# Deep learning for automatic stereotypical motor movement detection using wearable sensors in autism spectrum disorders

Nastaran Mohammadian Rad [a,b,c,*], Seyed Mostafa Kia [d,e], Calogero Zarbo [a], Twan van Laarhoven [b], Giuseppe Jurman [a], Paola Venuti [f], Elena Marchiori [b], Cesare Furlanello [a]

[a] *Fondazione Bruno Kessler, Trento, Italy*
[b] *Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands*
[c] *Department of Information Engineering and Computer Science, University of Trento, Trento, Italy*
[d] *Donders Centre for Cognitive Neuroimaging, Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, The Netherlands*
[e] *Department of Cognitive Neuroscience, Radboud University Medical Centre, Nijmegen, The Netherlands*
[f] *Department of Psychology and Cognitive Science, University of Trento, Trento, Italy*

## ARTICLE INFO

## ABSTRACT

Autism Spectrum Disorders are associated with atypical movements, of which stereotypical motor movements (SMMs) interfere with learning and social interaction. The automatic SMM detection using inertial measurement units (IMU) remains complex due to the strong intra and inter-subject variability, especially when handcrafted features are extracted from the signal. We propose a new application of the deep learning to facilitate automatic SMM detection using multi-axis IMUs. We use a convolutional neural network (CNN) to learn a discriminative feature space from raw data. We show how the CNN can be used for parameter transfer learning to enhance the detection rate on longitudinal data. We also combine the long short-term memory (LSTM) with CNN to model the temporal patterns in a sequence of multi-axis signals. Further, we employ ensemble learning to combine multiple LSTM learners into a more robust SMM detector. Our results show that: (1) feature learning outperforms handcrafted features; (2) parameter transfer learning is beneficial in longitudinal settings; (3) using LSTM to learn the temporal dynamic of signals enhances the detection rate especially for skewed training data; (4) an ensemble of LSTMs provides more accurate and stable detectors. These findings provide a significant step toward accurate SMM detection in real-time scenarios.

## 1. Introduction

Autism spectrum Disorder (ASD) is a complex and heterogeneous neuro-developmental disorder. Specific symptoms in ASD are difficulties in social interactions, repetitive or restricted behaviors, and verbal/nonverbal communication difficulties. Prevalence of ASD is reported to be 1 in 88 individuals [1]. While the majority of studies have mainly focused on social and communication problems of ASD children, the repetitive and restricted behaviors associated with ASD individuals are also objects of interest because of their effect on the learning performance and socialization; and also as an indicator of distress [2–4]. Stereotypical Motor Movements (SMMs) are the major group of atypical repetitive behaviors in children with ASD. SMMs occur without evoking stimuli and include hand flapping, body rocking, and mouthing [5–7]. SMMs

have a specific negative effect on the quality of life of ASD children: they can affect negatively on the performance of children during learning new skills and while using the learned skills [8]. Furthermore, since these type of movements are socially abnormal, they cause difficulties in interaction with pairs in the school or other social settings [9]. In some cases, the severity of SMMs can lead to a meltdown event and even can cause self-damaging behaviors [10].

There are three traditional approaches for measuring the SMMs [11]: (1) paper-and-pencil rating, (2) direct behavioral observation, and (3) video-based methods. Paper-and-pencil rating is an interview-based approach which suffers from the subjectivity in rating. Furthermore, it cannot accurately detect the intensity, amount, and duration of SMMs [12]. In the direct behavioral observation approach, therapists can directly observe and record the sequences of SMMs. This method is not also a reliable approach due to the several reasons [11,13]: first, in high speed movements, it is hard for therapists to accurately observe and document all SMM

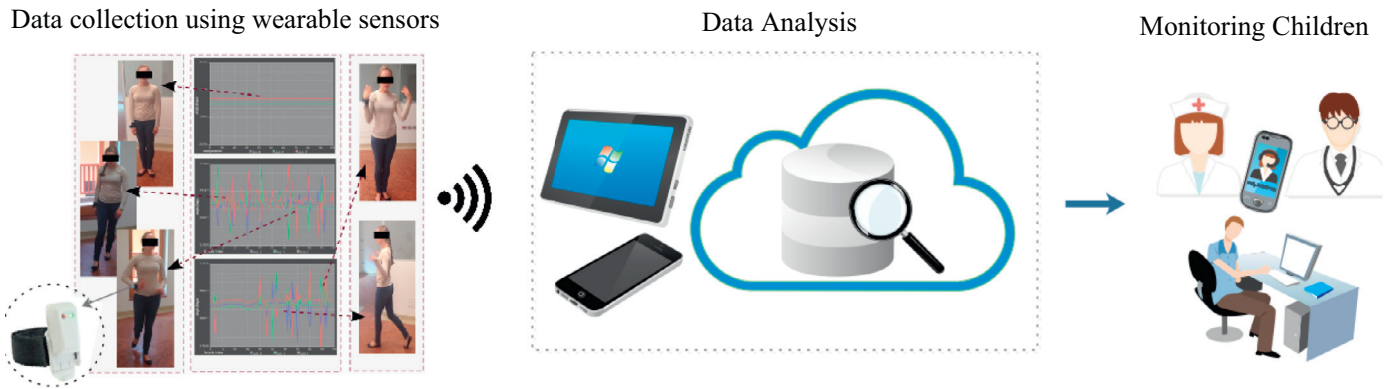Data collection using wearable sensors  Data Analysis  Monitoring Children



**Fig. 1.** A real-time automatic SMM detection system. Inertial Measurement Units (IMUs) can be used for data collection. The collected data can be analyzed locally or remotely to detect SMMs. In case of detecting abnormal movements, an alert is sent to a therapist, caregiver, or parents.

sequences instantaneously. Second, determining the start and end time of the SMM sequences is difficult. Third, it is impossible for therapists to concurrently record all environmental conditions and SMMs. Video-based approaches are based on video capturing, off-line coding, and analysis of SMMs. Since multiple coding sessions of the captured videos are feasible, this observational method is much more accurate than two previous approaches, but it is time-consuming and unpractical as a clinical tool [14].

Considering the high prevalence rate of autism in children [15] and the limitations of existing methods for measuring SMMs, it is essential to develop time efficient and accurate methods for automatic SMM detection. Developing a real-time SMM detection and quantification system would be advantageous for ASD researchers, caregivers, families, and therapists. Such a system would provide a powerful tool to evaluate the adaptation of subjects with ASD to diverse life contexts within an ecologic approach. In particular, it helps to mitigate the meltdown behaviors that are anticipated by the increase in atypical behaviors. Any automatic quantification of atypical movements would indeed help caregivers and teachers to defuse the mechanism leading to stereotyped behaviors by involving children in specific activities or social interactions. Such involvement decreases the frequency of SMMs and gradually alleviates their duration and severity [16,17]. A real-time implementation of SMM detection system (see Fig. 1) would help therapists to evaluate the efficacy of behavioral interventions.

Inertial measurement units (IMUs) provide effective tools for measuring the frequency, intensity, and duration of physical activities over a time period via embedded accelerometer, gyroscope, and magnetometer sensors. Due to the small size and possibility of embedding in the mobile phones, IMUs have been adopted as common and useful sensors for wearable devices to measure the physical activities in either constrained and free-living environments [18–20]. In recent years, human activity recognition using IMU sensors has been widely studied. Most of these studies tried to extract time and frequency domain features such as mean, standard deviation, skewness, and FFT peaks from raw signals to feed them to a classifier for activity identification [21,22]. According to the achieved results in human activity recognition systems, applying pattern recognition on the collected data by IMU sensors can reliably and accurately detect physical activities which are an evidence for the possibility of applying such techniques to automatically detect SMMs in ASD children [23–36]. Despite meaningful amount of research in this direction, few challenges for automatic SMM detection using wearable sensors still remain unsolved especially in real-time applications.

One of the important challenges for accurate SMM detection is to extract a set of effective and robust features from the IMU sig-

nal. As in many other signal processing applications, SMM detection is commonly based on extracting handcrafted features from the IMU signals. So far, a wide variety of feature extraction methods have been used in the literature. Generally two main types of features are extracted from the accelerometer signal [37]: (i) time domain features, (ii) frequency domain features. For time domain features, some statistical features such as mean, standard deviation, zero-crossing, energy, and correlation are extracted from the overlapping windows of the signal. In the case of frequency features the discrete Fourier transform is used to estimate the power of different frequency bands. In addition, the Stockwell transform [38] is proposed by [33] for feature extraction from 3-axis accelerometer signals in order to provide better time-frequency resolution for non-stationary signals. In spite of their popularity, manual feature extraction and selection suffer from subjectivity and time inefficiency [39] that restrict the performance and also the application of SMM detection systems in real-time scenarios.

Another challenge toward developing a real-time SMM detection system is personalization due to the intra and inter-subject variability [33,35]. This challenge, despite its crucial importance, has been undervalued [33]. Intra-subject variability is mainly due to the high variability in the intensity, duration, frequency, and topography of SMMs in each individual with ASD. Inter-subject differences are defined by the same variability across different individuals. Existence of these two types of variability within and across ASD persons motivates the necessity of developing an adaptive SMM detection algorithm that is capable to adjust to new patterns of behaviors. Feature learning and transfer learning [40] can be considered as candidate solutions to attack these challenges. To this end, here we present an extended version of our previous efforts in [41,42] with four main contributions: (1) robust feature learning from multi-sensor IMU signals; (2) enhancing the adaptability of SMM detection system to new data via parameter transfer learning; (3) improving the detection rate by incorporating the temporal dynamics of signals in the feature learning process; and (4) using principles of the ensemble learning to enhance the detection rate.

To achieve our first goal, we propose a new application of the deep learning paradigm in order to directly *learn* discriminating features for detecting SMM patterns. In particular, we use a convolutional neural network (CNN) [43] to bypass the commonly used feature extraction procedure. The idea of the CNN is inspired by the visual sensory system of living creatures [44]. Following this idea, LeCun et al. [45] developed a deep CNN architecture to address a pattern recognition problem in computer vision. Having fewer connections and parameters due to the weight sharing property, CNNs are easier to train compared to other deep neural net-

works. Currently, CNN solutions are among the best-performing systems on pattern recognition systems specifically for the handwritten character [45] and object recognition [46]. Beyond audio and image recognition systems, CNNs are successfully applied on various types of signals. Mirowski et al. [47] applied CNN on EEG signals for seizure detection. In the domain of psychophysiology, for the first time Martinez et al. [39] proposed a model based on CNN to predict affective states of fun, excitement, anxiety, and relaxation. Their proposed model was tested on skin conductance and blood volume pulse signals. Recent studies show the advantageous of applying CNN on accelerometer signals for human activity recognition [48–50].

To fulfill our second goal, we employ the parameter transfer learning by pre-initializing the parameters of the CNN [51]. We hypothesize that this capability can be used to transfer the prior knowledge regarding the distribution of parameters from one dataset to another dataset that are collected in a longitudinal study. If successful, our method can be employed in order to enhance the adaptability of SMM detection system to new unseen data, thus facilitates its applications in wild real-world scenarios.

By the definition, SMMs are repetitive behaviors, thus temporal patterns stored in the sequence of samples are expected to contain valuable information. Our third contribution relies on the fact that the proposed CNN architecture does not fully exploit the temporally structured information in the IMU signal sequences. This is a general issue in SMM detection, in which the segments of IMU signals are treated as statistically independent samples. Therefore the possible long-term dependencies stored in the longer temporal intervals of the signal are ignored in the detection process. Long short-term memory (LSTM) [52] as a type of recurrent neural networks (RNN) has been effectively used for learning long-term temporal dependencies in sequential data such as speech recognition [53], handwriting recognition [54], and natural language modeling [55]. Recently, the LSTM has also been successfully used for human activity recognition using wearable technologies as a classic sequence analysis problem [50,56,57]. Considering these studies in human activity recognition, it is expected that learning the temporal patterns stored in the consecutive samples of IMU data to provide higher accurate SMM detectors. Thus, here we propose a deep architecture, stacking an LSTM layer on top of the CNN architecture, in order to learn the dynamic feature space on the sequence of IMU data. We further show that combining multiple LSTM models using an ensemble learning technique can improve the stability of results. To the best of our knowledge, it is the first time that a recurrent architecture is used for SMM detection using wearable technologies.

The rest of this paper is organized as follows: in Section 2 we first briefly review the formal definitions and concepts about CNN, LSTM, parameter transfer learning, and the ensemble of the best base learners approach. Then using the presented definitions we introduce the proposed CNN and LSTM architectures[1] for SMM detection on IMU signals. The experimental materials and procedures are explained in this section. Section 3 compares our experimental results versus the state of the art solutions in SMM detection. Our results on a simulated dataset and two real datasets show that feature learning via the CNN outperforms handcrafted features in SMM classification. Furthermore, it is shown that the parameter transfer learning is beneficial in enhancing the SMM detection rate when moving to a new dataset. Finally our results illustrate that including the dynamics of recorded data in feature learning process improves the classification performance in SMM detection especially when an unbalanced training set is used in the training

phase. In Section 4 we discuss how the proposed deep architecture facilitates developing real-time SMM detection systems. We further discuss the main limitation of our method and state the possible future directions. Section 5 concludes this paper by summarizing our achievements.

## 2. Methods

### 2.1. Notation

Let $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_c \in \mathbb{R}^L$ be $c$ time-series of length $L$ that are recorded by a set of inertial measurement units (IMUs) (e.g., accelerometer, gyroscope, and magnetometer sensors) at the sampling frequency of $\nu$ Hz. Thus, $T = L/\nu$ represents the length of the signal in seconds. We refer to each $\mathbf{S}_i$ as a data channel. Now consider $\mathbf{X}_t \in \mathbb{R}^{c \times \nu}$ for $t \in \{1, 2, \ldots, T\}$ as a sample in the raw feature space that is constructed by concatenating time-series of $c$ data channels in a given time $t$. Let $y_t \in \{0, 1\}$ be the label associated to $\mathbf{X}_t$ where $y_t = 1$ corresponds to an SMM sample. In this text, we use boldface capital letters to represent matrices, boldface lowercase letters to represent vectors, and italic lowercase letters to represent scalars. We represent the matrix and element-wise multiplication between $\mathbf{A}$ and $\mathbf{B}$ matrices by $\mathbf{A} \cdot \mathbf{B}$ and $\mathbf{A} \odot \mathbf{B}$, respectively. Further, $[\mathbf{a}, \mathbf{b}]$ represents vector concatenation operation between $\mathbf{a}$ and $\mathbf{b}$ vectors.

### 2.2. Feature learning via convolutional neural network

The goal of an SMM detector is to predict the probability of being an SMM for a given sample $\mathbf{X}_t$, i.e., $P(y_t = 1 \mid \mathbf{X}_t)$. While the raw feature space ($\mathbf{X}_t$) is sensitive to intra and inter-subject differences, feature learning can provide the possibility to learn a new feature space that is robust over time and across different subjects. The aim of *feature learning* is to learn a linear or non-linear mapping function $\mathcal{F} : \mathbf{X}_t \mapsto \mathbf{x}_t'$, where $\mathbf{x}_t' \in \mathbb{R}^d$ is called the learned feature space. Then a classifier can be used in the learned feature space to estimate $P(y_t = 1 \mid \mathbf{x}_t')$.

Convolutional neural networks (CNNs) offer an effective infrastructure for feature learning. CNN benefits from invariant local receptive fields, shared weights, and spatio-temporal sub-sampling features to provide robustness over shifts and distortions in the input space [43]. A classic CNN has a hierarchical architecture that alternates convolutional and pooling layers in order to summarize large input spaces with spatio-temporal relations into a lower dimensional feature space. A 1D-convolutional layer receives the input signal $\mathbf{X}_t \in \mathbb{R}^{c \times \nu}$, convolves it with a set of $f$ filters with the length of $m$, $\mathbf{W} \in \mathbb{R}^{f \times c \times m}$, and produces a feature map $\mathbf{M}_t \in \mathbb{R}^{f \times \nu}$:

$$
\begin{aligned}
\mathbf{M}_t &= \mathbf{X}_t * \mathbf{W} \\
&= \begin{bmatrix}
\sum_{j=1}^{c} \sum_{i=1}^{m} w_{1,j,i} \times x_{j,1+i} & \cdots & \sum_{j=1}^{c} \sum_{i=1}^{m} w_{1,j,i} \times x_{j,\nu+i} \\
\sum_{j=1}^{c} \sum_{i=1}^{m} w_{2,j,i} \times x_{j,1+i} & \cdots & \sum_{j=1}^{c} \sum_{i=1}^{m} w_{2,j,i} \times x_{j,\nu+i} \\
\vdots & \vdots & \vdots \\
\sum_{j=1}^{c} \sum_{i=1}^{m} w_{f,j,i} \times x_{j,1+i} & \cdots & \sum_{j=1}^{c} \sum_{i=1}^{m} w_{f,j,i} \times x_{j,\nu+i}
\end{bmatrix}
\end{aligned}
\tag{1}
$$

where $*$ represents the convolution operator. The feature map is then fed to an activation function, generally the rectified linear unit (ReLU), to add non-linearity to the network and also to avoid the gradient vanishing problem [58], where:

$$
\mathbf{M}_t^+ = max(\mathbf{0}^{f \times \nu}, \mathbf{M}_t).
\tag{2}
$$

Here $max(.,.)$ represents the element-wise max operation. Finally, in order to reduce the sensitivity of the output to shifts and distortions, $\mathbf{M}_t^+$ is passed through a pooling layer which performs a local averaging or sub-sampling over a pooling window

---

[1] Here the architecture implies the customization of the structural parameters of the CNN such as the number of layers, the number and size of filters, etc.
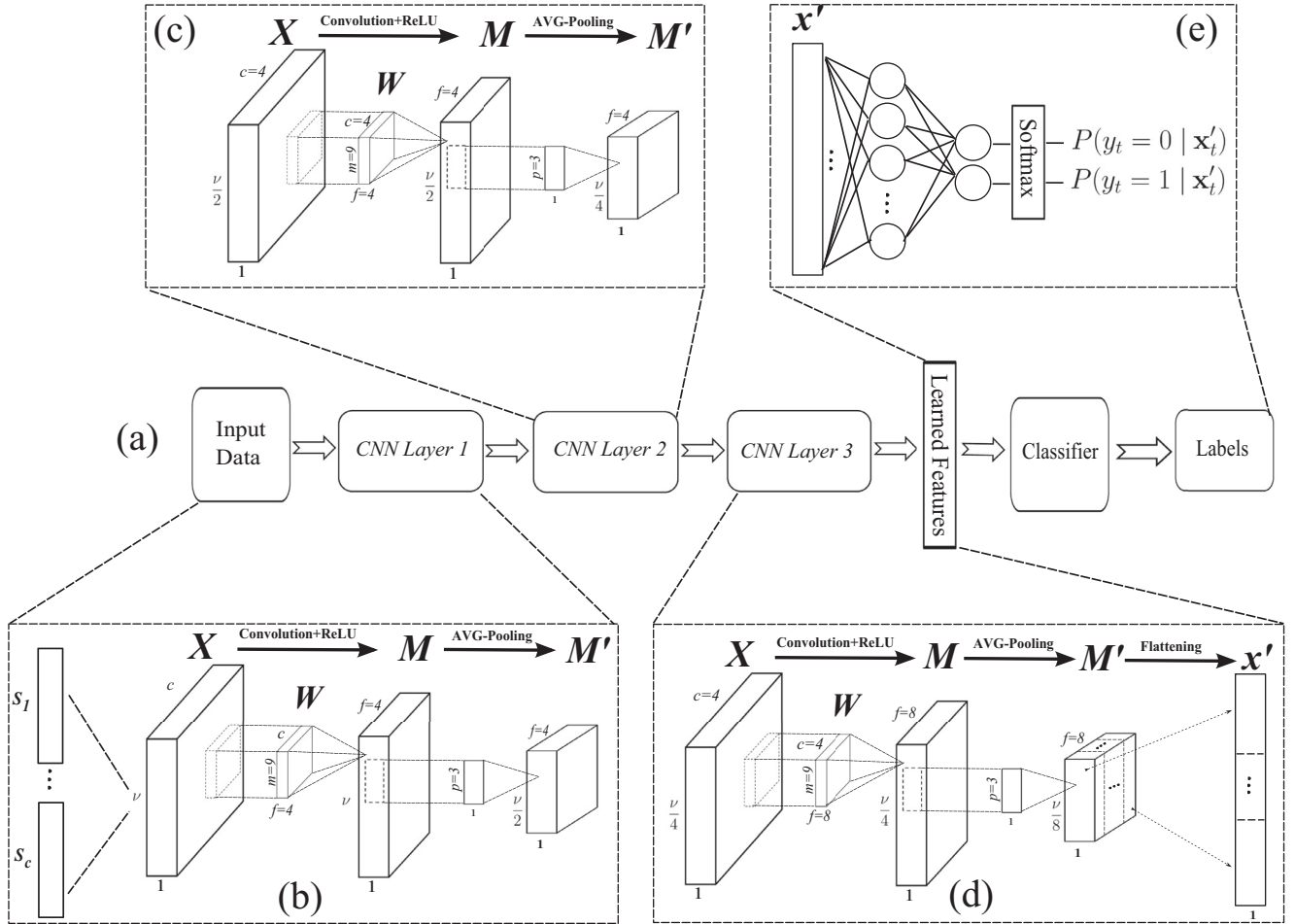
Fig. 2. (a) The proposed architecture for SMM detection in the static feature space using a three-layer CNN. (b) The first CNN layer. This layer receives the one-second long time-series of several IMU sensors at time $t$, i.e., $\mathbf{X}_t$, and transfer it to the first level reduced feature map $\mathbf{M}'_t$. (c) The second CNN layer that uses the first level reduced feature map $\mathbf{M}'_t$ as its input, and transfer it to the second-level reduced feature map. (d) The third CNN layer. The reduced feature map of this layer is reshaped to the learned feature vector $\mathbf{x}'_t$ using the flattening operation. (e) The learned feature vector is fed to a fully-connected followed by a softmax layer to classify the samples to SMM and no-SMM classes.

with size of $p$ elements and calculates the reduced feature map $\mathbf{M}'_t \in \mathbb{R}^{f \times \frac{\nu}{u}}$. In fact, a pooling layer reduces the resolution of a feature map by factor of $\frac{1}{u}$ where $u$ is the stride (or step) size. Max-pooling and average-pooling are two commonly used pooling functions which compute the maximum or average value among the values in a pooling window, respectively. In average-pooling for $m'_{i,j} \in \mathbf{M}'_t$, $i \in \{1, \ldots, f\}$, and $j \in \{1, \ldots, \frac{\nu}{u}\}$, we have:

$$m'_{i,j} = \frac{1}{p} \sum_{k=1}^{p} m_{i,(j-1) \times u + k}. \tag{3}$$

Alternatively, in the max-pooling each element of the reduced feature map is the maximum value in a corresponding pooling window:

$$m'_{i,j} = max(m_{i,(j-1) \times u + 1}, m_{i,(j-1) \times u + 2}, \ldots, m_{i,(j-1) \times u + p}). \tag{4}$$

The reduced feature map $\mathbf{M}'_t$ can be used as the input to the next convolutional layer, i.e., $\mathbf{X}_t$ of the next layer. In general, the reduced feature map computed by stacking several convolution, ReLU, and pooling layers is flattened as a vector before the classification step. The flattening step is performed by collapsing the rows of $\mathbf{M}'_t$ in the form of a vector. The resulting vector is called the learned feature space $\mathbf{x}'_t$ that represents a new representation of the original feature space. This new representation is typically fed to a fully connected neural network followed by a softmax layer for the classification purposes.

In this paper, and for the purpose of SMM detection on the multi-sensor IMU data, we propose to use a three-layer CNN to transform the time-series of multiple sensors to a new feature space. The proposed architecture is shown in Fig. 2. Three convolutional layers are designed to have 4, 4, and 8 filters with the length of 9 samples (i.e., 0.1 s), respectively. The length of the pooling window and pooling stride are fixed to 3 ($p = 3$) and 2 ($u = 2$), respectively. The pooling stride of 2 reduces the length of feature maps by the factor of 0.5 after each pooling layer. The output of the third convolutional layer after flattening provides the learned feature vector. Then, the learned feature vector is fed to a two-layer fully-connected network with 8 and 2 neurons that are connected to a softmax layer. A dropout [59] rate of 0.5 is used in the fully connected layers to avoid the overfitting problem. Since only the information in $\mathbf{X}_t$ is used to compute $\mathbf{x}'_t$ and then predict $y_t$, we refer to the learned feature space via this CNN architecture as the static feature space.

### 2.3. Parameter transfer learning via network pre-initialization

The quality and characteristics of recorded IMU signals varies not only from subject to subject but also from time to time in a single subject. Therefore it is important that the SMM detector system be able to adapt to new streams of signals in longitudinal scenarios. In this paper, we explore the possibility of parameter trans-

fer learning via network pre-initialization in order to transfer the learned patterns to the newly seen data in a different time span. In this direction we first formalize the background theoretical concepts.

In the statistical learning theory, the goal is to learn a task $\mathcal{T}$ in a certain domain $\mathcal{D}$. A *domain* $\mathcal{D} = \{\mathcal{X}, \rho_{\mathcal{X}}\}$ is defined as a possible conjunction between an input space $\mathcal{X}$ and a marginal probability distribution $\rho_{\mathcal{X}}$. For example in the SMM detection context, the recorded IMU signal for different subjects can be considered as different domains as the marginal probability distribution $\rho_{\mathcal{X}}$ is different from one subject to another. Similarly different domains can be defined by time in longitudinal data collection scenarios. Given a domain $\mathcal{D}$, a *task* $\mathcal{T} = \{\mathcal{Y}, \Phi\}$ is defined as a predictive function $\Phi$ from $\mathcal{D}$ to the output space $\mathcal{Y}$. For example in this study $\Phi$ is the SMM detector, and $\mathcal{Y}$ represents the categorical output space of SMM and no-SMM samples. Assume $\mathcal{D}_S$, $\mathcal{D}_T$, $\mathcal{T}_S$, and $\mathcal{T}_T$ to represent the source domain, target domain, source task, and target task, respectively. *Transfer learning* aims to benefit from the knowledge in the source domain and task in order to improve the predictive power in the target domain when $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ [40].

In this study, we are interested in the application of *parameter transfer learning* via pre-initializing the parameters of a deep neural network, as a well-established technique in the deep learning community, to improve the SMM prediction performance across different subjects and time intervals. To this end, we define the source domain $\mathcal{D}_S$ as the IMU signal which is collected on several subjects at the time span $T_1$. Similarly the target domain $\mathcal{D}_T$ is defined as the IMU signal which is collected on several subjects at the time span $T_2$. Assume $\Phi_S$ be the learned predictive function, i.e., the CNN classifier, in the source domain. We use the learned parameters in $\Phi_S$ to pre-initialize the parameters of the predictive function in the target domain $\Phi_T$. In simpler words, instead of random pre-initialization, we initialize the parameters of CNN classifier in the target domain with the learned CNN parameters in the source domain. We hypothesize that such a knowledge transfer via learned parameters improves the prediction performance in the longitudinal studies where the data are collected at different time intervals.

### 2.4. SMM detection in dynamic feature space using LSTM

In SMM detection using static feature space (see Section 2.2) only the data in $\mathbf{X}_t$ is used to predict $y_t$. Thus it is implicitly assumed that the sequence of samples over time are independent and identically distributed (i.i.d). But in reality, this assumption is not valid as the samples in consecutive time steps are highly dependent. Therefore, it is expected that accounting for this dependency would improve the performance of the SMM detector. Following this hypothesis, we propose to use a long short-term memory (LSTM) layer to model the temporal dependency between the consecutive time steps of the recorded signal.

Let $\mathbf{x}'_t$ be a set of static features that are extracted or learned from samples in the raw feature space (i.e., from $\mathbf{X}_t$). Here we assume the CNN architecture explained in Section 2.2 is used to compute $\mathbf{x}'_t$. Then, let $\mathbf{c}_t \in \mathbb{R}^q$ and $\mathbf{h}_t \in \mathbb{R}^q$ to represent the cell state and output of an LSTM unit at time step $t$, respectively, where $q$ is the number of neurons in the LSTM unit. We will refer to $\mathbf{h}_t$ as the *dynamic* feature space. The LSTM unit receives $\mathbf{x}'_t$, $\mathbf{h}_{t-1}$, and $\mathbf{c}_{t-1}$ as its inputs, and computes $\mathbf{c}_t$ and $\mathbf{h}_t$ as follows:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{5}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot (1 - e^{-2 \times \mathbf{c}_t}) \odot (1 + e^{-2 \times \mathbf{c}_t})^{-1}. \tag{6}$$

Here $\mathbf{f}_t \in \mathbb{R}^q$ is called the forget gate vector and its elements are real numbers between 0 and 1 that decide how much information
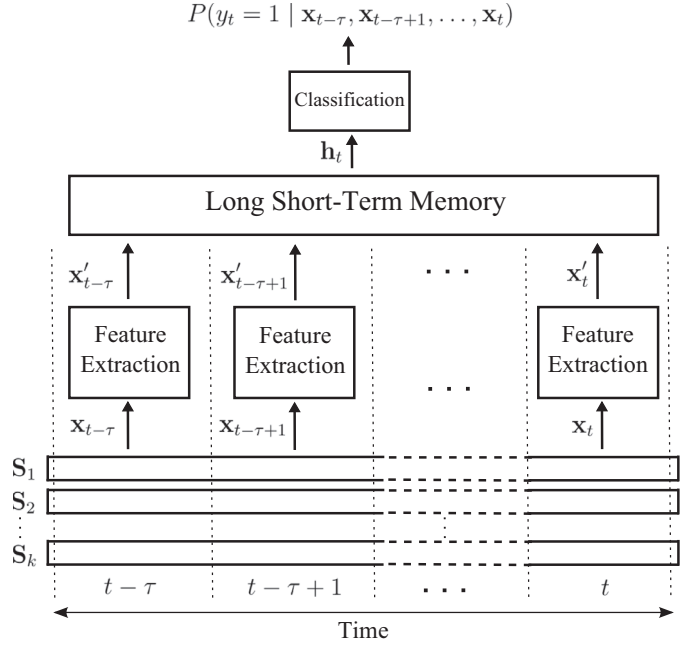


**Fig. 3.** The proposed architecture for SMM detection in the dynamic feature space using long short-term memory. Each feature extraction block contains a trained three-layer CNN architecture (see Fig. 2).

to be passed from $\mathbf{c}_{t-1}$ to $\mathbf{c}_t$. During the learning phase, the forget gate learns the forget weight matrix $\mathbf{W}_f$ and the forget bias vector $\mathbf{b_f}$. $\mathbf{f}_t$ is computed by

$$\mathbf{f}_t = (1 + e^{-(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}'_t] + \mathbf{b}_f)})^{-1}. \tag{7}$$

Using a tangent hyperbolic function, $\tilde{\mathbf{c}}_t \in \mathbb{R}^q$ provides new candidate values between $-1$ and 1 for $\mathbf{c}_t$ by learning $\mathbf{W}_c$ and $\mathbf{b_c}$:

$$\tilde{\mathbf{c}}_t = (1 - e^{-2 \times (\mathbf{W}_c \cdot [y_{t-1}, \mathbf{x}'_t] + \mathbf{b}_c)}) \odot (1 + e^{-2 \times (\mathbf{W}_c \cdot [y_{t-1}, \mathbf{x}'_t] + \mathbf{b}_c)})^{-1}, \tag{8}$$

where $\mathbf{i}_t \in \mathbb{R}^q$ is the input gate vector with elements between 0 and 1. These values determine the level of new information in $\tilde{\mathbf{c}}_t$ to be transferred to the cell state $\mathbf{c}_t$. $\mathbf{i}_t$ is computed based on $\mathbf{W_i}$ and $\mathbf{b_i}$ as follows:

$$\mathbf{i}_t = (1 + e^{-(\mathbf{W_i} \cdot [y_{t-1}, \mathbf{x}'_t] + \mathbf{b_i})})^{-1}. \tag{9}$$

Finally, $\mathbf{o}_t \in \mathbb{R}^q$ is the output gate vector that filters the cell state $\mathbf{c}_t$ to generate the output of the LSTM unit $\mathbf{h}_t$:

$$\mathbf{o}_t = (1 + e^{-(\mathbf{W_o} \cdot [y_{t-1}, \mathbf{x}'_t] + \mathbf{b_o})})^{-1}. \tag{10}$$

In this paper, a fully-connected layer with dropout of 0.2 is used to transfer the output of the LSTM layer at time $t$, i.e., $\mathbf{h}_t$, to the input of the softmax layer $\mathbf{z}_t = [z_t^{(0)}, z_t^{(1)}]^T \in \mathbb{R}^2$:

$$P(y_t = 1 \mid \mathbf{x}_{t-\tau}, \mathbf{x}_{t-\tau+1}, \ldots, \mathbf{x}_t) = \frac{e^{z_t^{(1)}}}{e^{z_t^{(0)}} + e^{z_t^{(1)}}}, \tag{11}$$

where $\tau$ represents the number of previous time steps that are used as the input to the LSTM layer. Fig. 3 presents a schematic overview of the proposed architecture.

### 2.5. Ensemble of the best base learners

Due to the random initialization and using stochastic optimization algorithms on random mini-batches in training deep learning models, retraining the same model on the same training set results in heterogeneous approximations of the target classifier. This heterogeneity is the direct result of reaching different local optimums in optimizing a complex non-convex error surface. One possible approach to overcome this problem is *ensemble learning* (EL) [60].

**Fig. 4.** (a) The configuration of the EXLs3 sensor on the right hand. (b),(c) The simulated data are collected during daily work activities (e.g., writing, typing, etc.). (d),(e) The subjects are asked to intermittently perform hand flapping upon receiving a start/stop cue from the instructor.

The main idea behind EL is to combine the knowledge learned by individual classifiers in order to achieve a more superior and stable performance. It is shown that in general an ensemble of classifiers works better than every single classifier due to the statistical, computational, and representational reasons [61]. Considering the success of deep learning ensembles in pattern recognition and signal processing applications [62–65], in this study we are interested in applying classifier selection voting approach [66] to combine an ensemble of the best base learners.

Let $(\mathbf{X}_{tr}, \mathbf{y}_{tr})$ and $(\mathbf{X}_{ts}, \mathbf{y}_{ts})$ to be the corresponding sample/target pairs in the training and test sets, respectively. Then assume $\mathcal{C} = \{c_1, c_2, \ldots c_l\}$ be a set of $l$ base learners trained on the training set. Our goal is to first find a set of $b$ best classifiers $\mathcal{C}^* \subseteq \mathcal{C}$ based on a performance measure $\alpha$ on the training set, and then to combine their prediction on the test set using majority voting in the prediction phase. Algorithm 1 summarizes this approach.

---

**Algorithm 1** The training and test procedures in the majority voting on a set of $b$ best models.

1: **procedure** TRAINING($\mathcal{C}, \mathbf{X}_{tr}, \mathbf{y}_{tr}$)
2:     **for all** $c_i \in \mathcal{C}$ **do**
3:         Predict $\hat{\mathbf{y}}$ using $c_i$ on $\mathbf{X}_{tr}$.
4:         Evaluate $\hat{\mathbf{y}}$ and store the performance in $\alpha_i$.
    **end**
5:     **for** $i \leftarrow 1, l$ **do**
6:         Store the best $i$ classifiers in $\mathcal{C}_i^*$.
7:         Predict $\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_i$ using classifiers in $\mathcal{C}_i^*$ on $\mathbf{X}_{tr}$.
8:         Compute majority voting $\tilde{\mathbf{y}}$ on predictions in $\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_i$.
9:         Evaluate $\tilde{\mathbf{y}}$ and store the performance in $\alpha_i$.
    **end**
10:     Find the best value for $b$ by $b = \text{argmax}_i(\alpha_i)$.
11:     return $\mathcal{C}_b^*$.

12:

13: **procedure** TEST($\mathcal{C}_b^*, \mathbf{X}_{ts}, \mathbf{y}_{ts}$)
14:     Predict $\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_b$ using classifiers in $\mathcal{C}_b^*$ on $\mathbf{X}_{ts}$.
15:     Compute majority voting $\tilde{\mathbf{y}}$ on predictions in $\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_b$.
16:     Evaluate $\tilde{\mathbf{y}}$ and store the performance in $\alpha$.
17:     return $\alpha$.

---

### 2.6. Experimental materials

We assess the performance of the proposed methods on both simulated and real data. In the following, we describe the datasets and the procedures that are used for data preparation.

#### 2.6.1. Simulated data

In a simulation setting, 5 healthy subjects (3 females and 2 males) are asked to emulate stereotypical movements in a con-

**Table 1**
Number of samples in SMM and no-SMM classes in three datasets.

| Data | Subjects | No-SMM | SMM | All | SMM/All |
|------|----------|--------|-----|-----|---------|
| Simulated data | Sub1 | 13,875 | 4075 | 17,950 | 0.23 |
| | Sub2 | 11,686 | 6224 | 17,910 | 0.35 |
| | Sub3 | 13,694 | 4246 | 17,940 | 0.24 |
| | Sub4 | 12,428 | 5532 | 17,960 | 0.31 |
| | Sub5 | 13,583 | 4367 | 17,950 | 0.24 |
| | Total | 65,266 | 24,444 | 89,710 | 0.27 |
| Real data1 | Sub1 | 21,292 | 5663 | 26,955 | 0.21 |
| | Sub2 | 12,763 | 4372 | 17,135 | 0.26 |
| | Sub3 | 31,780 | 2855 | 34,635 | 0.08 |
| | Sub4 | 10,571 | 10,243 | 20,814 | 0.49 |
| | Sub5 | 17,782 | 6173 | 23,955 | 0.26 |
| | Sub6 | 12,207 | 17,725 | 29,932 | 0.59 |
| | Total | 106,395 | 47,031 | 153,426 | 0.31 |
| Real data2 | Sub1 | 18,729 | 11,656 | 30,385 | 0.38 |
| | Sub2 | 22,611 | 4804 | 27,415 | 0.18 |
| | Sub3 | 40,557 | 268 | 40,825 | 0.01 |
| | Sub4 | 38,796 | 8176 | 46,972 | 0.17 |
| | Sub5 | 22,896 | 6728 | 29,624 | 0.23 |
| | Sub6 | 2375 | 11,178 | 13,553 | 0.82 |
| | Total | 145,964 | 42,810 | 188,774 | 0.23 |

trolled environment. Each participant wore an EXLs3 sensor[2], a miniaturized electronic device with the function of real-time IMU, fixed on the right wrist using a wristband (see Fig. 4(a)). EXLs3 sensor records three-axis accelerometer, gyroscope, and magnetometer data (it has 9 data channels in total). The sensor was set to transmit three-axis $\pm 16\,g$ acceleration and $\pm 2000\,dps$ angular velocity at the $100\,Hz$ sampling rate. The participants were instructed to perform their normal working activities such as sitting, writing, and typing; while intermittently performing hand flapping upon receiving a start/stop cue from the instructor (see Fig. 4(b)–(e)). The total period of SMMs is organized somehow to keep the distribution of two classes comparable with real datasets where 27% of samples are in the SMM class (see Table 1 and Section 2.6.2). The total duration of each experiment was 30 min organized in three 10 min sessions. Real-time coding is undertaken during sessions to annotate the starting and ending time of movements. The captured data were band-pass filtered with a cut-off frequency of $0.1\,Hz$ to remove the DC components. Then the signal was segmented to 1 s long (i.e., 100 time-points) using a sliding window. The sliding window was moved along the time dimension with 10 time-steps resulting in 0.9 overlaps between consecutive windows[3].

---

### 2.6.2. Real data

We use the data presented in [33] wherein the accelerometer data were collected for 6 subjects with autism in a longitudinal study[4]. The data were collected in the laboratory and classroom environments while the subjects wore three 3-axis wireless accelerometers and engaged in body rocking, hand flapping, or simultaneous body rocking and hand flapping. The accelerometer sensors were attached to the left and right wrists, and on the torso. Offline annotation based on a recorded video is used to annotate the data by an expert. Two separate collections are available: the first collection, here we call it *Real Data1*, was recorded by MITes sensors at 60 Hz sampling frequency [31]. The second collection *Real Data2*, was recorded three years after the first recording on the same subjects using Wockets sensors with the sampling frequency of 90 Hz. The sampling rate of two recordings is equalized by re-sampling the signal in *Real Data1* to 90 Hz using linear interpolation. The cut-off high pass filter at 0.1 Hz is applied in order to remove the DC components of the signal. Similar to [33], the signal is segmented to 1-second long overlapped intervals using a sliding window. The amount of overlap is set to 10 time-points resulting in 0.87 overlap between consecutive windows. Table 1 summarizes the number of samples in no-SMM and SMM classes for each subject. The difference in the number of samples in SMM and no-SMM classes shows unbalanced nature of the real data, where in Real Data1 and Real Data2 datasets 31% and 23% of samples are in the SMM class, respectively.

### 2.7. Experimental setups and evaluation

To investigate the effect of static and dynamic feature learning and parameter transfer learning on the performance of SMM detection, we conducted four experiments. Keras library [67] is used in our implementations[5].

### 2.7.1. Experiment 1: Static feature learning

The main aim of this experiment is to compare the effectiveness of feature learning using a deep neural network versus raw feature space and handcrafted features in an across-subject SMM detection setting. To evaluate the effect of both feature extraction and feature learning on the SMM classification performance, first, without any feature extraction the signals in raw feature space are used as the input to a support vector machine (SVM) classifier. In this case, all data channels of each sample $\mathbf{X}_t$ are collapsed into a feature vector (with length of $900 = 9 \times 100$ in simulated data and $810 = 9 \times 90$ in real data case). Second, to evaluate the detection performance using handcrafted features we extracted all features mentioned in [33] including time, frequency, and Stockwell transform features, then, we replicated the across-subject SMM detection experiment in [33]. In this setting we used exactly the same implementation provided by the authors[6] in the feature extraction and classification steps. Third, a CNN architecture (see Section 2.2) is used to learn a middle representation of the multi-sensor signal. In this experiment, all effective parameters of CNN (weights and biases) are initialized by drawing small random numbers from the normal distribution. The stochastic gradient descent with momentum (the momentum is fixed to 0.9) is used for training the network. All these steps are performed only on the training data to ensure unbiased error estimation. Due to the random initialization

of weights and employing stochastic gradient descent algorithm for optimization, results can be different from one training run to another. Therefore, we repeated the whole procedure of learning and classification 10 times and the mean and standard variation over runs are reported. It is important to emphasize that, similar to [33], in all three parts of this experiment the number of samples in minority class is used to randomly draw a balanced training set.

### 2.7.2. Experiment 2: parameter transfer learning

As discussed before, deep neural networks provide the capability of parameter transfer learning via network pre-initialization. We applied this experiment only on two real datasets in order to investigate the possibility of transferring learned knowledge from one dataset to another in a longitudinal data collection setting. This experiment is similar to Experiment 1, except for the network initialization step. Instead of random initialization, here we firstly train the CNN on one balanced real dataset, e.g., Real Data1, and then we use the learned parameters for pre-initializing the parameters of CNN before training on another balanced real dataset, e.g., Real Data2. Similar to previous experiment, we repeated the whole experiment 10 times to evaluate the standard deviation of the classification performance.

### 2.7.3. Experiment 3: training on the unbalanced training set

As explained, in Experiment 1 and 2 we balanced the training set based on the number of samples in minority class. Even though balancing the training set improves the quality of the trained model but in fact it suffers from some deficits: (1) by balancing the training set we impose a wrong prior assumption on the original distribution of data. As shown in Table 1 in real datasets around 0.3 of samples belong to SMM class, when by balancing the dataset we assume it is 0.5; (2) by balancing the training set we cannot employ the full richness of the data as we need to remove significant amount of samples from the training set; (3) in some practical scenarios, such as real-time adaptation or classification on the sequence of streamed data, balancing the training set is impractical. Considering these limitations, in this experiment in order to evaluate the effect of balancing on the performance of CNN model we evaluate the performance of the proposed CNN architecture in predicting SMMs when unbalanced training sets are used in the training phase.

### 2.7.4. Experiment 4: dynamic feature learning

In this experiment, we are interested in answering three main questions: (1) what are the advantages of learning temporal representation of IMU signals for reliable SMM detection? (2) how long is the most informative time interval in IMU signals for detecting abnormal movements? 3) what is the optimal configuration for the LSTM unit? To answer these questions, we applied the proposed LSTM architecture in Section 2.4 on the three benchmark datasets with different values for $\tau$ and $q$, i.e., time steps and neuron number, respectively. We set $\tau = \{1, 3, 5, 10, 15, 25, 50\}$ and $q = \{5, 10, 20, 30, 40, 50\}$. The LSTM unit is trained on the extracted features by the CNN using the RMSProp [68] optimizer. The learned dynamic features via LSTM ($\mathbf{h}_t$) are classified to target classes using a softmax classifier. It is worthwhile to emphasize that, in this setting, since the order of samples in the training set matters, balancing the training set is impossible, thus we use the original unbalanced data.

### 2.7.5. Experiment 5: ensemble of LSTMs

To explore the possible advantage of combining multiple classifiers, we used the procedure explained in Section 2.5 in order to combine a set of $b$ best base learners. In this experiment, we used the LSTM models in the Experiment 4 as base learners for the classification of unbalanced data. For all datasets, we used the same

---

configurations for the LSTM models by fixing $\tau = 25$ and $q = 40$. We set $l = 10$ and used the F1-score for the performance metric $\alpha$ in Algorithm 1. The experiment is repeated 10 times to evaluate the standard deviation over the mean prediction performance.

### 2.7.6. Evaluation

In all experiments the leave-one-subject-out scheme is used for model evaluation in order to measure the robustness of the trained model against inter-subject variability. Due to the unbalanced class distributions in the test set, we computed the F1-score to evaluate the classification performance:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad ,$$

where true positive (TP), false positive (FP), and false negative (FN) rates are used as follows to compute the precision and recall:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}.$$

## 3. Results

### 3.1. Feature learning outperforms handcrafted features

The classification performances summarized in Table 2 compare the quality of feature learning via CNN with raw and handcrafted feature spaces on three datasets. In all three datasets, the classification performance of SMM detection on the handcrafted and learned features is higher than the classification performance on the raw feature space. This observation demonstrates the importance of feature extraction/learning for detecting SMMs. Furthermore, the comparison between the results achieved by handcrafted and learned features illustrates the efficacy of feature learning over the manual feature extraction in SMM prediction. The learned feature space reaches on average 0.10 and 0.20 higher F1-score than the handcrafted features in case of simulated data and

real data1, respectively, while in case of real data2 its performance declines by 0.05. These results support the overall efficacy of feature learning versus handcrafted features in extracting robust features for across-subject SMM detection. These conclusions are even further confirmed in Fig. 5(a)–(c), where the t-distributed Stochastic Neighbor Embedding (t-SNE) [69] technique is employed to visualize the different feature spaces in a 2-dimensional space. We used the average of Fisher's separability score [70] across two t-SNE dimensions to quantify the separability of samples in two classes for different feature spaces. Fig. 5(a) shows 2D t-SNE distribution of SMM and no-SMM samples in the raw feature space, where there is a high overlap between the samples of two classes. This high overlap is also well-reflected in the low Fisher's separability score in raw feature space (0.02). Fig. 5(b) depicts the distribution of samples of two classes in handcrafted feature space. The samples in two classes are barely separable and the Fisher's separability score is 0.03. Fig. 5(c) displays the 2D t-SNE space for the learned features via the CNN architecture. In this case the separability score is improved significantly to 0.10.

### 3.2. Parameter transfer learning is beneficial in longitudinal studies

As mentioned in Section 2.7.2, the aim of our second experiment was to investigate the possibility of transferring learned knowledge from one dataset to another using parameter transfer learning. Our results in Table 2 shows that transferring knowledge from one dataset to another in a longitudinal study, by pre-initializing the parameters of CNN model improves the average classification performance of the SMM detectors by 0.04 and 0.21 in Real Data1 and Real Data2 datasets, respectively.

### 3.3. Training on unbalanced data decreases the performance

The results in Table 2 illustrate the negative effect of using unbalanced training set in training CNN architecture in randomly initialized (feature learning) and pre-initialized (transfer learning) scenarios. The performance of SMM detection in feature learning

**Table 2**

Results for SMM detection using raw, handcrafted, static, dynamic feature spaces, and ensemble learning in three benchmarked datasets. The results show that feature learning generally outperforms raw and handcrafted feature spaces. In addition, the parameter transfer learning has a positive effect on the performance of the CNN classifier. Furthermore, training the CNN classifier on unbalanced training sets causes the performance drop in feature learning and transfer learning scenarios. Using the LSTM network to extract dynamic features from the signal alleviates this problem to some degrees. Ensemble of LSTMs shows more stable performance compared to single LSTM classifiers.

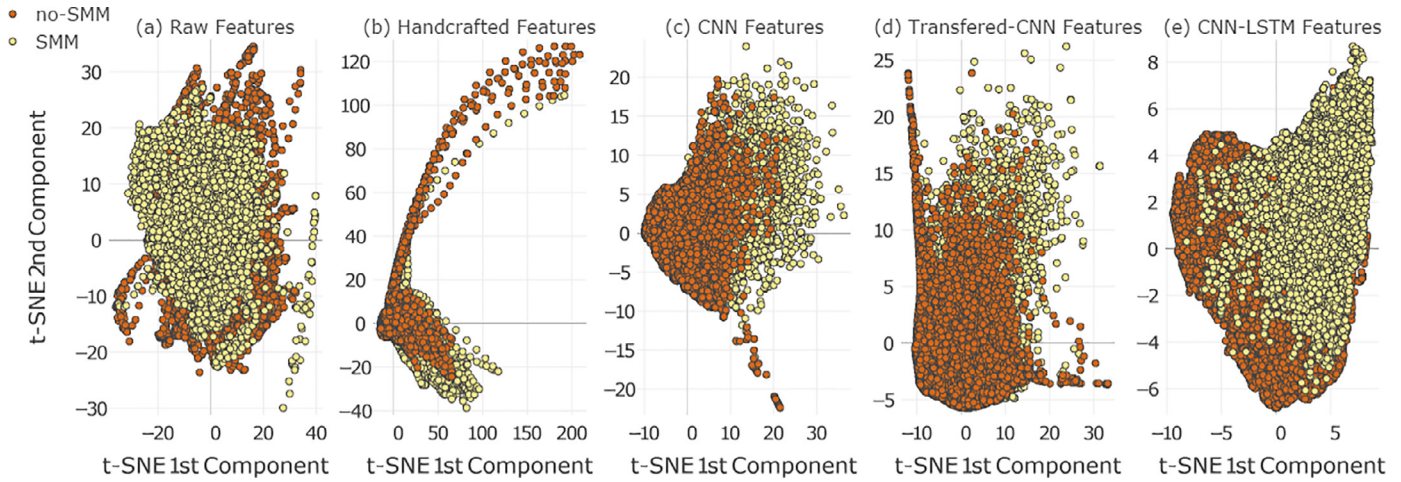| Data | Sub | Balanced training sets | | | | Unbalanced training sets | | | | |
|------|-----|-----------------|----------------------|------------------|----------------------|-----------------------------|----------------------|-------------------------------|-----------------------------|-------------------|
| | | Raw features | Handcrafted features | Feature learning | Transfer learning | Feature learning (1 s) | Transfer learning | Feature learning (2.5 s) | Dynamic feature learning | Ensemble learning |
| Simulated | 1 | 0.29 | 0.71 | 0.78 ± 0.05 | - | 0.73 ± 0.13 | - | 0.95 ± 0.01 | 0.95 ± 0.01 | 0.95 ± 0.00 |
| | 2 | 0.84 | 0.86 | 0.86 ± 0.03 | - | 0.78 ± 0.09 | - | 0.86 ± 0.13 | 0.95 ± 0.01 | 0.96 ± 0.01 |
| | 3 | 0.55 | 0.76 | 0.80 ± 0.01 | - | 0.75 ± 0.13 | - | 0.95 ± 0.03 | 0.97 ± 0.01 | 0.97 ± 0.00 |
| | 4 | 0.76 | 0.48 | 0.85 ± 0.03 | - | 0.73 ± 0.12 | - | 0.97 ± 0.01 | 0.96 ± 0.02 | 0.97 ± 0.01 |
| | 5 | 0.38 | 0.77 | 0.79 ± 0.01 | - | 0.80 ± 0.04 | - | 0.91 ± 0.01 | 0.90 ± 0.02 | 0.91 ± 0.00 |
| | Mean | 0.56 | 0.72 | 0.82 ± 0.03 | - | 0.76 ± 0.11 | - | 0.93 ± 0.06 | 0.95 ± 0.01 | 0.95 ± 0.01 |
| Real Data1 | 1 | 0.44 | 0.74 | 0.74 ± 0.02 | 0.71 ± 0.02 | 0.70 ± 0.02 | 0.71 ± 0.03 | 0.73 ± 0.04 | 0.77 ± 0.03 | 0.80 ± 0.00 |
| | 2 | 0.32 | 0.37 | 0.75 ± 0.02 | 0.73 ± 0.01 | 0.63 ± 0.03 | 0.63 ± 0.04 | 0.68 ± 0.04 | 0.71 ± 0.03 | 0.74 ± 0.00 |
| | 3 | 0.22 | 0.50 | 0.68 ± 0.04 | 0.70 ± 0.03 | 0.57 ± 0.08 | 0.59 ± 0.06 | 0.56 ± 0.13 | 0.68 ± 0.05 | 0.72 ± 0.01 |
| | 4 | 0.44 | 0.73 | 0.92 ± 0.01 | 0.92 ± 0.00 | 0.88 ± 0.01 | 0.88 ± 0.01 | 0.93 ± 0.00 | 0.91 ± 0.01 | 0.93 ± 0.00 |
| | 5 | 0.56 | 0.44 | 0.51 ± 0.04 | 0.68 ± 0.05 | 0.51 ± 0.08 | 0.58 ± 0.07 | 0.51 ± 0.04 | 0.52 ± 0.04 | 0.51 ± 0.01 |
| | 6 | 0.56 | 0.46 | 0.90 ± 0.01 | 0.94 ± 0.01 | 0.79 ± 0.07 | 0.81 ± 0.09 | 0.86 ± 0.12 | 0.90 ± 0.02 | 0.91 ± 0.00 |
| | Mean | 0.42 | 0.54 | 0.74 ± 0.03 | 0.78 ± 0.03 | 0.68 ± 0.06 | 0.70 ± 0.06 | 0.71 ± 0.08 | 0.75 ± 0.03 | 0.77 ± 0.01 |
| Real Data2 | 1 | 0.47 | 0.43 | 0.61 ± 0.11 | 0.68 ± 0.05 | 0.33 ± 0.14 | 0.36 ± 0.08 | 0.47 ± 0.15 | 0.53 ± 0.09 | 0.59 ± 0.03 |
| | 2 | 0.23 | 0.26 | 0.20 ± 0.04 | 0.22 ± 0.04 | 0.11 ± 0.03 | 0.16 ± 0.04 | 0.13 ± 0.05 | 0.26 ± 0.06 | 0.29 ± 0.02 |
| | 3 | 0.01 | 0.03 | 0.02 ± 0.01 | 0.02 ± 0.01 | 0.02 ± 0.01 | 0.02 ± 0.01 | 0.02 ± 0.01 | 0.02 ± 0.02 | 0.02 ± 0.02 |
| | 4 | 0.32 | 0.86 | 0.72 ± 0.03 | 0.77 ± 0.02 | 0.71 ± 0.14 | 0.83 ± 0.03 | 0.90 ± 0.02 | 0.76 ± 0.09 | 0.87 ± 0.02 |
| | 5 | 0.38 | 0.73 | 0.21 ± 0.09 | 0.75 ± 0.09 | 0.14 ± 0.09 | 0.09 ± 0.02 | 0.23 ± 0.17 | 0.35 ± 0.15 | 0.43 ± 0.08 |
| | 6 | 0.50 | 0.07 | 0.36 ± 0.13 | 0.91 ± 0.05 | 0.62 ± 0.08 | 0.70 ± 0.16 | 0.68 ± 0.21 | 0.96 ± 0.01 | 0.98 ± 0.00 |
| | Mean | 0.32 | 0.40 | 0.35 ± 0.08 | 0.56 ± 0.05 | 0.32 ± 0.1 | 0.36 ± 0.08 | 0.40 ± 0.13 | 0.48 ± 0.08 | 0.53 ± 0.04 |

**Fig. 5.** The distribution of SMM and no-SMM samples in the 2-dimensional t-SNE space for (a) raw feature space, (b) handcrafted features, (c) static feature space learned by CNN, (d) static feature space learned by pre-initialized CNN, and (e) dynamic feature space learned by LSTM. Feature learning increases the separability of samples in two classes compared to raw and handcrafted features.
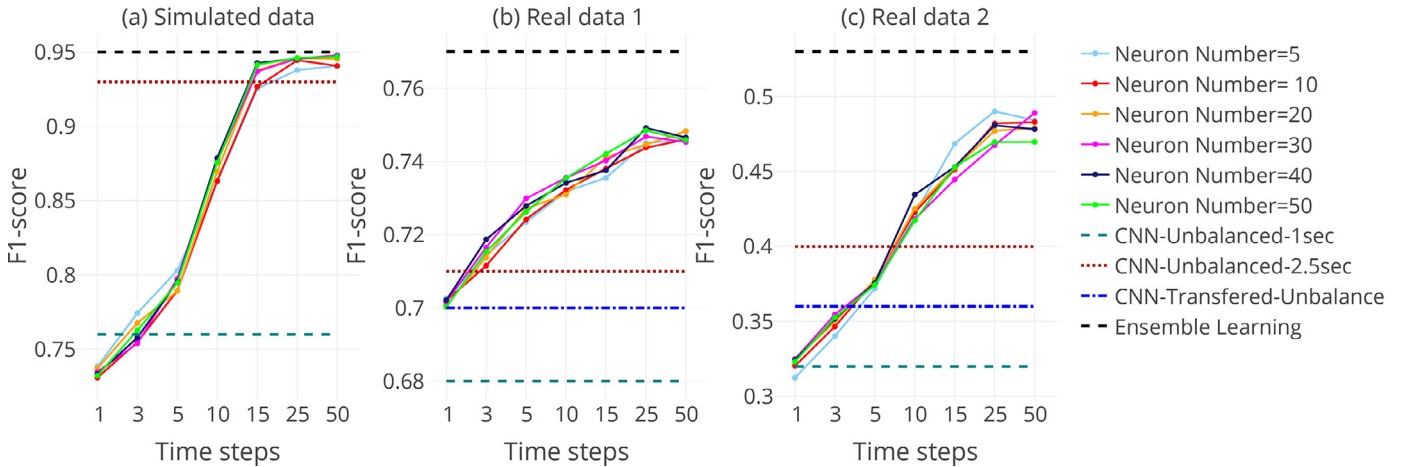


**Fig. 6.** Comparison between the classification performances of CNN and LSTM for different time-steps ($\tau$) and number of neurons $q$, on three datasets and when an unbalanced training set is used for training the networks. The results show the superiority of the dynamic feature space over the static feature space. While the number of neurons in the LSTM unit has little effect on the performance, using around 2.5 s long interval is the best choice for extracting effective dynamic features from the IMU signals. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

scenario drops by 0.06 and 0.03 in Real Data1 and Real Data2 datasets, respectively. This performance drop is even more pronounced in the transfer learning scenario where we observe 0.08 and 0.20 performance drop in the corresponding datasets.

### 3.4. Dynamic feature learning outperforms static feature learning

Fig. 6 compares the averaged SMM classification performance over subjects in the static feature space via the CNN (the green dashed line for plain feature learning and the blue dotted-dashed line for transfer learning) with the dynamic feature space via the LSTM, the latter with different values for $\tau$ (x-axis) and $q$ (line colors). Here in all settings, an unbalanced training set is used in the training phase. The results on three datasets illustrate that learning the temporal representation of signals with an LSTM unit, consistently across datasets, improves the classification performance compared to the static feature learning via the CNN. The classification performance improves by increasing $\tau$, and it reaches its highest performance around $\tau = 25$. Considering the consistency of the best $\tau$ value for different subjects and different datasets, it can be concluded that using around 25 time-steps, i.e., around 2.5 s long interval, for extracting dynamic features is the best choice for SMM

detection purposes. On the other hand, the results show the negligible effect of the number of LSTM neurons ($q$) on the detection performance, thus, a value around 10 can be considered a reasonable choice.

To further benchmark the advantage of dynamic feature learning via LSTM, we used the CNN architecture for the SMM detection on the best length for the time intervals, i.e., on 2.5 s time intervals. The results on the three datasets are summarized in Table 2 and Fig. 6 (the dotted red line). The results confirm the superiority of dynamic feature learning compared to static feature learning despite using longer time intervals for learning static features.

The effect of learning the temporal representation on the separability of SMM and no-SMM samples is shown in Fig. 5(e). The higher Fisher's separability score (0.17) in the dynamic feature space compared to static feature spaces can be considered as the basis for the higher classification performance of the proposed architecture, demonstrating the importance of learning dynamic features using an LSTM based architecture. Furthermore, employing the dynamic-feature representation computed by the LSTM, improves the detection rates when unbalanced training sets are used (see Table 2). This observation is consistent across three datasets.
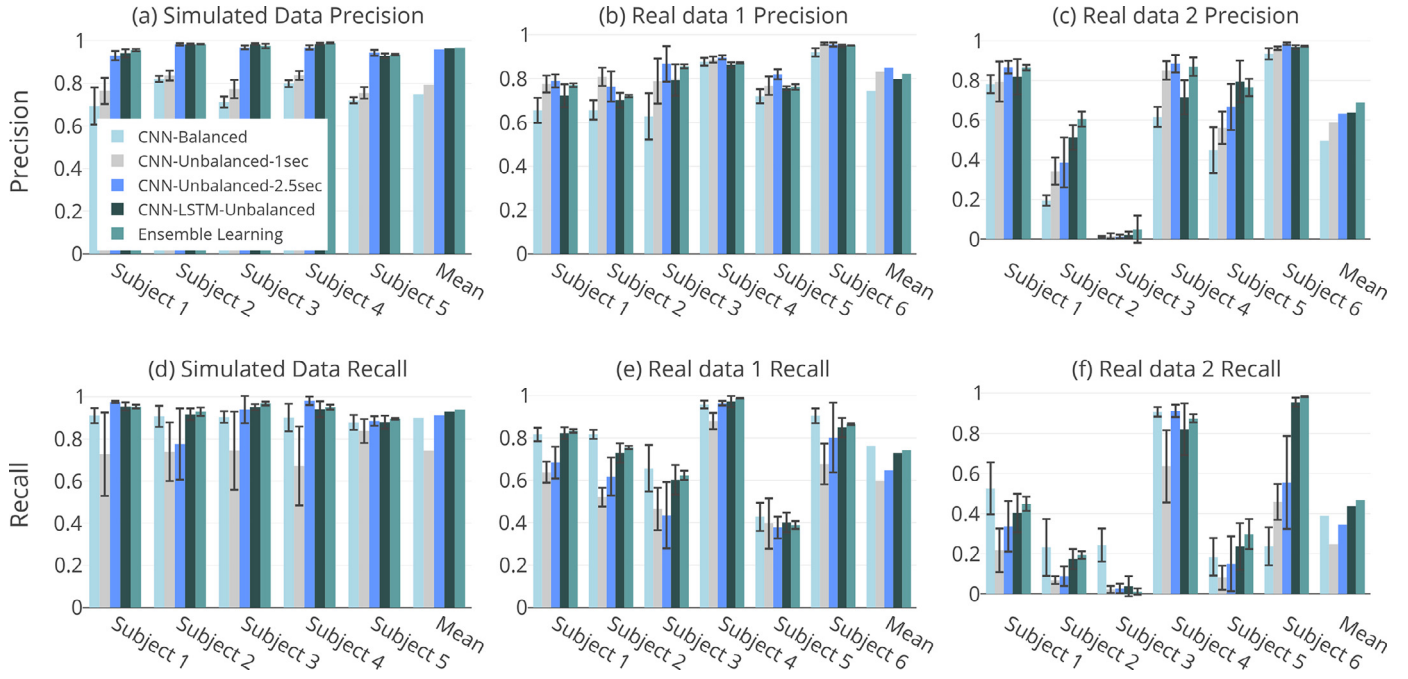
**Fig. 7.** A comparison between precision and recall rates of classifiers trained on balanced/unbalanced training sets for static/dynamic feature representations. Using dynamic feature space provides an SMM detection system with higher sensitivity and specificity.

Fig. 7 further explores the superiority of the dynamic feature representation when the training set is unbalanced. In the static feature space case, balancing the training set and enforcing the wrong prior class distribution into the classification task, despite higher recall rate, affects negatively the precision of the classifier. In other words, the classifiers have higher false alarm rate, which could be problematic in real-world applications. This deficit is recovered in the case of dynamic feature representation where the classifier presents higher precision rate and comparable recall with respect to static features. In fact, the LSTM-based architecture by enforcing the true prior distribution of data into the training process and, at the same time using all the recorded samples, provides an SMM detection system with higher sensitivity and specificity.

*3.5. Ensemble of LSTMs stabilizes the performance*

The last column of Table 2 summarizes the results of the ensemble approach. The results show slight boost in the mean performance compared to single LSTM classifiers, especially on real data2 (see dashed black line in Fig. 6). Fig. 7 shows that both precision and recall contribute equally to this improvement in F1-scores. In addition to the higher performance, the main advantage of EL is demonstrated by the low variability of results. This reduction in the variability is well-reflected in the reduced standard deviation around the mean performance in real datasets (0.02 and 0.04 reduction in real data1 and real data2 datasets, respectively). In other words an ensemble of LSTMs provides more reliable SMM detector in comparison to every single LSTM classifier.

**4. Discussion**

*4.1. Toward real-Time automatic SMM detection*

In this study, we proposed an original application of deep learning for SMM detection in ASD children using wearable technologies. To this end, we used a three-layer CNN architecture for learning a more discriminative and robust feature space in an across-subject SMM detection scenario. Our experimental results, on the

simulated and real data, support the superiority of learning middle representation of IMU signal over traditional feature extraction methods in automatic SMM detection. Further, we showed that the parameter transfer learning via network pre-initialization provides the infrastructure for effective knowledge transfer from one dataset to another in a longitudinal setting. We also presented an application of LSTM in SMM detection context for extracting dynamic features on the sequence of IMU data. In a comparison with the static feature space learned via CNN architecture, we illustrated the advantage of employing the temporal information in improving the separability of SMM and no-SMM samples. We experimentally showed that using around 2.5 s long interval for extracting dynamic features is the best choice for SMM detection purposes. Further, we showed using around 10 neurons in the LSTM unit is a reasonable choice in order to extract the dynamics of samples over time. As a side-advantage of learning a dynamic feature space, we experimentally demonstrated the higher performance of our method when, in a real world setting, the distribution of samples in SMM and no-SMM classes is highly skewed. We showed, while the skewness of samples negatively affects the performance of the SMM detector in the static feature space, exploiting the temporal patterns of multi-sensor IMU signals recovers its performance. This advantage facilitates training high-performing models by exploiting whole data sequences in real-time SMM detection scenarios. Our effort, for the first time in the SMM detection context, demonstrated the superiority of recurrent structures in extracting discriminative temporal patterns from IMU signals. As the final contribution, considering the important role of ensemble learning for classifying the stream data in non-stationary environments [65,66], we employed ensemble of the best base learners technique to improve the reliability of the SMM detector. In summary, our results show that feature learning, transfer learning, ensemble learning, and learning temporal structures from multi-modal IMU signals improve the performance of SMM detection systems especially in more realistic scenarios.

Developing real-time mobile applications for detecting the abnormal movements such as SMMs can be considered as a final goal in the context of automatic SMM detection using wearable sensors.

At the moment there are numerous challenges in real-time human activity recognition using wearable sensors, namely [19,71,72]: (1) designing effective feature extraction and inference methods; (2) recognition on realistic data; (3) the adaptability of system to new users. Addressing these issues demands a huge investment in research toward finding robust and effective features that can be extracted in a reasonable time from the stream of IMU signal. Our proposal to learn a middle representation of the signal, that is robust to the signal variability of a single subject data over time and also to the across-subject variability, can be considered as an effective solution in this direction. In addition, the parameter transfer learning capability besides the possibility of incremental training of the proposed deep architecture facilitates the online adaptation of an automatic SMM detector in real-time scenarios. This finding overlooks the subject specific [73], and monolithic [35,74] activity recognition systems opening new frontiers toward adaptable activity recognition systems which are more appropriate for real-time usages. At the end, the high detection performance on unbalanced training sets achieved in the dynamic feature space facilitates the application of our method on the realistic data when the incoming data samples are highly skewed.

### 4.2. Limitation and future work

Even though the deep architecture introduced in this study provides a significant step toward a more accurate automatic SMM detection system in real-time scenarios, but it suffers from a considerable limitation: the proposed fully supervised scheme for training the SMM detection model is problematic for its online adaptation. This problem comes from the fact that in real applications the system has no access to the labels of incoming samples during usage by a new user. Therefore, the adaptation to new unseen data should be performed only based on the input unlabeled data. This limitation motivates future researches on the online adaptation of the system in an unsupervised manner. One possible solution in this direction is transductive transfer learning [40] where the basic assumption is that no labeled data in the target domain are available. Therefore adopting a transductive transfer learning strategy in the adaptation phase can be considered as a possible future direction to extend this work.

### 5. Conclusions

In this study we addressed the problem of automatic SMM detection in the framework of deep learning. We used a 3-layer CNN architecture to learn a robust feature space from multi-sensor/modal IMU signals. We illustrated how the proposed architecture can be employed for parameter transfer learning in order to enhance the adaptability of SMM detection system to new data. Further, we showed incorporating the temporal dynamics of the signal in the feature learning process, by combining the CNN architecture with an LSTM unit, improves the SMM detection rate in real-world scenarios especially in case of unbalanced data. We further illustrated the advantage of ensemble learning to provide more stable and reliable SMM detectors. Our results demonstrate high potentials of deep learning paradigm to address the crucial challenges toward real-time SMM detection systems using wearable technologies.

### Acknowledgments

## References

[1] J. Baio, Prevalence of Autism Spectrum Disorders: Autism and Developmental Disabilities Monitoring Network, 14 Sites, United States, 2008. Morbidity and Mortality Weekly Report. Surveillance Summaries. Volume 61, Number 3., Centers for Disease Control and Prevention, 2012.

[2] J. Hernandez, I. Riobo, A. Rozga, G.D. Abowd, R.W. Picard, Using electrodermal activity to recognize ease of engagement in children during social interactions, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014, pp. 307–317.

[3] T. Chaspari, C.-C. Lee, S. Narayanan, Interplay between verbal response latency and physiology of children with autism during ECA interactions, 8, 2012, p. 8.

[4] E. Hedman, L. Miller, S. Schoen, D. Nielsen, M. Goodwin, R. Picard, Measuring autonomic arousal during therapy, in: Proc. of Design and Emotion, 2012, pp. 11–14.

[5] G. Berkson, R. Davenport, Stereotyped movements of mental defectives: I. initial survey., Am. J. Mental Deficiency (1962).

[6] M.H. Lewis, A.A. Baumeister, Stereotyped mannerisms in mentally retarded persons: animal models and theoretical analyses, Int. Rev. Res. Mental Retard. 11 (1982) 123–1261.

[7] R.M. Ridley, H.F. Baker, Stereotypy in monkeys and humans, Psychol. Med. 12 (1982) 61–72.

[8] J.W. Varni, O.I. Lovaas, R.L. Koegel, N.L. Everett, An analysis of observational learning in autistic and normal children, J. Abnormal Child Psychol. 7 (1979) 31–43.

[9] R. Jones, D. Wint, N. Ellis, The social effects of stereotyped behaviour, J. Intellectual Disability Res. 34 (1990) 261–268.

[10] C.H. Kennedy, Evolution of stereotypy into self-injury. (2002).

[11] R.L. Sprague, K.M. Newell, Stereotyped movements: Brain and behavior relationships., Am. Psychol. Associat., 1996.

[12] D.A. Pyles, M.M. Riordan, J.S. Bailey, The stereotypy analysis: an instrument for examining environmental variables associated with differential rates of stereotypic behavior, Res. Development. Disabilities 18 (1997) 11–38.

[13] N.C. Gardenier, R. MacDonald, G. Green, Comparison of direct observational methods for measuring stereotypic behavior in children with autism spectrum disorders, Res. Development. Disabilities 25 (2004) 99–118.

[14] J.L. Matson, M. Nebel-Schwalm, Assessing challenging behaviors in children with autism spectrum disorders: a review, Res. Development. Disabilities 28 (2007) 567–579.

[15] S. Goldman, C. Wang, M.W. Salgado, P.E. Greene, M. Kim, I. Rapin, Motor stereotypies in children with autism and other developmental disorders, Development. Med. Child Neurol. 51 (2009) 30–38.

[16] S. Lee, S.L. Odom, R. Loftin, Social engagement with peers and stereotypic behavior of children with autism, J. Positive Behav. Intervent. 9 (2007) 67–79.

[17] R.L. Loftin, S.L. Odom, J.F. Lantz, Social interaction and repetitive motor behaviors, J. Autism and Development. Disorders 38 (2008) 1124–1135.

[18] M.J. Mathie, A.C. Coster, N.H. Lovell, B.G. Celler, Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement, Physiol. Measur. 25 (2004) R1.

[19] O.D. Lara, M.A. Labrador, A survey on human activity recognition using wearable sensors., IEEE Commun. Surveys Tutorials 15 (2013) 1192–1209.

[20] J.L. Helbostad, B. Vereijken, C. Becker, C. Todd, K. Taraldsen, M. Pijnappels, K. Aminian, S. Mellone, Mobile health applications to promote active and healthy ageing, Sensors 17 (2017).

[21] J. Baek, G. Lee, W. Park, B.-J. Yun, Accelerometer signal processing for user activity detection, in: Knowledge-Based Intelligent Information and Engineering Systems, Springer, 2004, pp. 610–617.

[22] M. Gjoreski, H. Gjoreski, M. Lutrek, M. Gams, How accurately can your wrist device recognize daily activities and detect falls? Sensors 16 (2016).

[23] T. Westeyn, K. Vadas, X. Bian, T. Starner, G.D. Abowd, Recognizing mimicked autistic self-stimulatory behaviors using HMMs, in: Ninth IEEE International Symposium on Wearable Computers (ISWC'05), IEEE, 2005, pp. 164–167.

[24] C.-H. Min, A.H. Tewfik, Y. Kim, R. Menard, Optimal sensor location for body sensor network to detect self-stimulatory behaviors of children with autism spectrum disorder, in: Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE, IEEE, 2009, pp. 3489–3492.

[25] C.-H. Min, A.H. Tewfik, Automatic characterization and detection of behavioral patterns using linear predictive coding of accelerometer sensor data, in: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, IEEE, 2010, pp. 220–223.

[26] H. Min Cheol, A.H. Tewfik, Novel pattern detection in children with autism spectrum disorder using iterative subspace identification, in: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2010, pp. 2266–2269.

[27] N. Gonçalves, J.L. Rodrigues, S. Costa, F. Soares, Automatic detection of stereotyped hand flapping movements: two different approaches, in: RO-MAN, 2012 IEEE, IEEE, 2012, pp. 392–397.

[28] N. Goncalves, J.L. Rodrigues, S. Costa, F. Soares, Automatic detection of stereotypical motor movements, Procedia Eng. 47 (2012) 590–593.

[29] J.L. Rodrigues, N. Gonçalves, S. Costa, F. Soares, Stereotyped movement recognition in children with ASD, Sensors Actuators A 202 (2013) 162–169.

[30] F. Albinali, M.S. Goodwin, S. Intille, Detecting stereotypical motor movements in the classroom using accelerometry and pattern recognition algorithms, Pervasive Mobile Comput. 8 (2012) 103–114.

[31] F. Albinali, M.S. Goodwin, S.S. Intille, Recognizing stereotypical motor movements in the laboratory and classroom: a case study with children on the autism spectrum, in: Proceedings of the 11th international conference on Ubiquitous computing, ACM, 2009, pp. 71–80.

[32] M.S. Goodwin, S.S. Intille, F. Albinali, W.F. Velicer, Automated detection of stereotypical motor movements, J. Autism Development. Disorders 41 (2011) 770–782.

[33] M.S. Goodwin, M. Haghighi, Q. Tang, M. Akcakaya, D. Erdogmus, S. Intille, Moving towards a real-time system for automatically recognizing stereotypical motor movements in individuals on the autism spectrum using wireless accelerometry, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014, pp. 861–872.

[34] T. Plötz, N.Y. Hammerla, A. Rozga, A. Reavis, N. Call, G.D. Abowd, Automatic assessment of problem behavior in individuals with developmental disabilities, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 391–400.

[35] U. Großekathöfer, N.V. Manyakov, V. Mihajlovi, G. Pandina, A. Skalkin, S. Ness, A. Bangerter, M.S. Goodwin, Automated detection of stereotypical motor movements in autism spectrum disorder using recurrence quantification analysis, Frontiers Neuroinformatics 11 (2017) 9.

[36] A.M. Amiri, N. Peltier, C. Goldberg, Y. Sun, A. Nathan, S.V. Hiremath, K. Mankodiya, Wearsense: detecting autism stereotypic behaviors through smartwatches, Healthcare 5 (2017).

[37] N.F. Ince, C.-H. Min, A. Tewfik, D. Vanderpool, Detection of early morning daily activities with static home and wearable wireless sensors, EURASIP J. Adv. Signal Process. 2008 (2008) 31.

[38] R.G. Stockwell, L. Mansinha, R. Lowe, Localization of the complex spectrum: the s transform, Signal Process. IEEE Trans. 44 (1996) 998–1001.

[39] H.P. Martinez, Y. Bengio, G.N. Yannakakis, Learning deep physiological models of affect, Computat. Intell. Mag. IEEE 8 (2013) 20–33.

[40] S.J. Pan, Q. Yang, A survey on transfer learning, Knowl. Data Eng. IEEE Trans. 22 (2010) 1345–1359.

[41] N. Mohammadian Rad, C. Furlanello, Applying deep learning to stereotypical motor movement detection in autism spectrum disorders, in: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016, pp. 1235–1242, doi:10.1109/ICDMW.2016.0178.

[42] N.M. Rad, S.M. Kia, C. Zarbo, G. Jurman, P. Venuti, C. Furlanello, Stereotypical motor movement detection in dynamic feature space, in: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016, pp. 487–494, doi:10.1109/ICDMW.2016.0076.

[43] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, The Handbook of Brain Theory and Neural Networks, 3361, 1995.

[44] K. Fukushima, Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybernetics 36 (1980) 193–202.

[45] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 86, 1998, pp. 2278–2324.

[46] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[47] P.W. Mirowski, Y. LeCun, D. Madhavan, R. Kuzniecky, Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG, in: Machine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on, IEEE, 2008, pp. 244–249.

[48] J.B. Yang, M.N. Nguyen, P.P. San, X.L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, in: Proceedings of the 24th International Conference on Artificial Intelligence, AAAI Press, 2015, pp. 3995–4001.

[49] M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, in: Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on, IEEE, 2014, pp. 197–205.

[50] T. Zebin, P.J. Scully, K.B. Ozanyan, Human activity recognition with inertial sensors using a deep learning approach, in: 2016 IEEE SENSORS, 2016, pp. 1–3, doi:10.1109/ICSENS.2016.7808590.

[51] F.J.O. Morales, D. Roggen, Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations, in: Proceedings of the 2016 ACM International Symposium on Wearable Computers, ACM, 2016, pp. 92–99.

[52] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural comput. 9 (1997) 1735–1780.

[53] F. Eyben, M. Wöllmer, B. Schuller, A. Graves, From speech to letters-using a novel neural network architecture for grapheme based asr, in: Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on, IEEE, 2009, pp. 376–380.

[54] P. Doetsch, M. Kozielski, H. Ney, Fast and robust training of recurrent neural networks for offline handwriting recognition, in: Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on, IEEE, 2014, pp. 279–284.

[55] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, arXiv:1409.2329 (2014).

[56] F.J. Ordóñez, D. Roggen, Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, Sensors 16 (2016) 115.

[57] N.Y. Hammerla, S. Halloran, T. Ploetz, Deep, convolutional, and recurrent models for human activity recognition using wearables, arXiv:1604.08880 (2016).

[58] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. http://www.deeplearningbook.org

[59] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting., J. Mach. Learn. Res. 15 (2014) 1929–1958.

[60] T.G. Dietterich, Ensemble learning, in: The Handbook of Brain Theory and Neural Networks, 2, MIT Press: Cambridge, MA, 2002, pp. 110–125.

[61] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, first, Wiley-Interscience, 2004. http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471210781

[62] L. Deng, J. Platt, Ensemble deep learning for speech recognition(2014).

[63] L.-p. Jin, J. Dong, Ensemble deep learning for biomedical time series classification, Computat. Intell. Neurosci. 2016 (2016).

[64] Y. Guan, T. Ploetz, Ensembles of deep lstm learners for activity recognition using wearables, arXiv:1703.09370 (2017).

[65] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woniak, Ensemble learning for data stream analysis: a survey, Inf. Fusion 37 (2017) 132–156.

[66] H.M. Gomes, J.P. Barddal, F. Enembreck, A. Bifet, A survey on ensemble learning for data stream classification, ACM Comput. Surv. 50 (2017) 23:1–23:36.

[67] F. Chollet, Keras, 2015. https://github.com/fchollet/keras.

[68] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude, COURSERA: Neural Netw. Mach. Learn. 4 (2012).

[69] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.

[70] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, 2012.

[71] M.A. Labrador, O.D.L. Yejas, Human Activity Recognition: Using Wearable Sensors and Smartphones, CRC Press, 2013.

[72] J.W. Lockhart, G.M. Weiss, Limitations with activity recognition methodology & data sets, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM, 2014, pp. 747–756.

[73] M. Berchtold, M. Budde, H.R. Schmidtke, M. Beigl, An extensible modular recognition concept that makes activity recognition practical, in: Annual Conference on Artificial Intelligence, Springer, 2010, pp. 400–409.

[74] O.D. Lara, A.J. Pérez, M.A. Labrador, J.D. Posada, Centinela: a human activity recognition system based on acceleration and vital sign data, Pervasive and Mobile Comput. 8 (2012) 717–729.