Adaptive Segmentation and Sequence Learning of Human Activities from Skeleton Data

David Ada Adama, Ahmad Lotfi*, Robert Ranson

School of Science and Technology, Nottingham Trent University, Clifton Lane, Nottingham, NG11 8NS, United Kingdom

Abstract

Discovering underlying patterns for predicting future actions from spatio-temporal human activity information is a fundamental component of research related to the development of expert systems in human activity recognition and assistive robotics. Current research focuses on classification or learning representations of activities for various applications. However, not much attention is given to the pattern discovery of activities which have a major role in the prediction of unseen actions. This paper proposes a novel Adaptive Segmentation and Sequence Learning (ASSL) framework which aims at segmenting unlabelled observations of human activities from extracted 3D joint information. Learning from these obtained segments provides information about the underlying patterns of activity sequences needed in predicting subsequent actions. In the proposed method, the temporal accumulated motion energy of body parts in an activity is utilised in the segmentation process to obtain key actions from unlabelled activity sequences since body parts show changes in acceleration and deceleration during an activity. Based on the segments obtained, the temporal sequence of transitions across activity segments are learned by employing a Long Short-Term Memory Recurrent Neural Network. This ASSL technique has been evaluated using both an experimental human activity dataset and a public activity dataset, and achieved a better performance when compared with other techniques including an Auto-regressive Integrated Moving Average, Support Vector Regression and Gaussian Mixture Regression Models in learning to predict patterns of activity sequences.

*Corresponding author

Email addresses: david.adama2015@my.ntu.ac.uk (David Ada Adama), ahmad.lotfi@ntu.ac.uk (Ahmad Lotfi), robert.ranson@ntu.ac.uk (Robert Ranson)

Keywords: Human Activity Recognition, Sequence Learning, Long Short-Term Memory, Activity Segmentation, Mean-shift clustering, Key Pose.

1. Introduction

The advances in technology have seen more research on the development of expert systems related to human activities and their applications in everyday life. Learning the sequences of human activities is one aspect that is daunting in many of such applications. Due to the variability in the human nature of conducting activities, it is often not possible to attain a generalised model for identifying sequences used for activity predictions. This is due to understanding the underlying patterns of activities which in many cases are not explicit.

A popular area of the application of expert systems for human activity sequence learning is human-robot interaction. For example, assistive robots require abilities to learn human activities in order to function autonomously. Such activities usually require the coordination of different joints in the body to accomplish activities such as "pick and place" of an object activities. Robots equipped with preset instructions (or models) to carry out predefined functions limits them to only certain tasks as they do not possess the intelligence required to evolve their knowledge into executing functions which may differ from the preset knowledge. Also, such models become obsolete as new tasks are encountered since they are not able to adapt to dynamic situations which are inherent in most practical applications. This is primarily due to variations in activity sequences, thus the need to investigate the varying patterns of human activities. To offer a solution to such cases, it is imperative to break down these activities into constituent elements and extract relevant information used in simplifying the process of recognising various human activity patterns. Fig. 1 illustrates the underlying concept of how human activity patterns can be inferred and learned from processing extracted visual 3D information.

There are two main categories of learning algorithms suitable for human activity learning: *Batch learning* and *Sequence learning*. Classical batch learning algorithms predict output for new data when a complete training set of data is used. In this case, the new data samples are presented simultaneously and when desired. However, a complete training dataset is often not available in advance for most practical applications. In applications such as human activity prediction (Adama et al., 2018), healthcare monitoring (Anderez et al., 2020) and

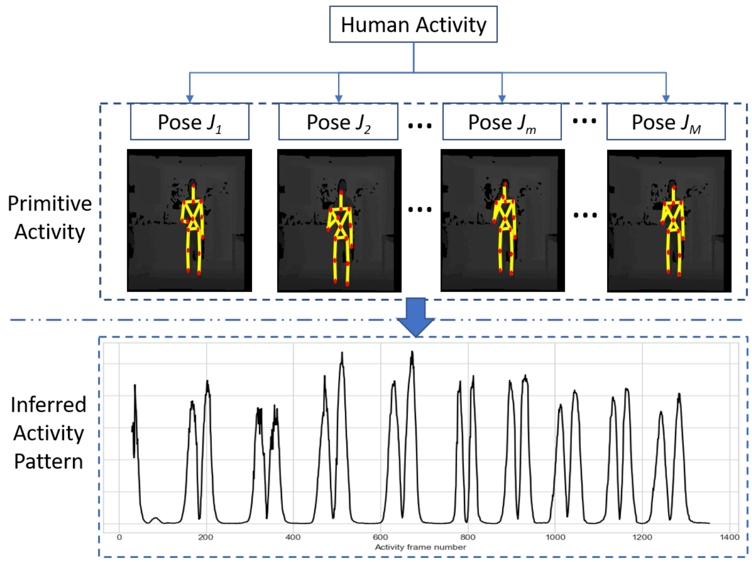


Fig. 1. An illustration of learning underlying patterns of simple primitive human activity sequences from 3D temporal information.

industrial functions (Suresh et al., 2010) in which temporal changes within a task are being observed, the classical batch learning algorithms are rather infeasible for learning. Sequence learning is executed in a series of occurrences of samples within a given training dataset. Samples are used in the algorithm one after another and discarded after learning. This implies that the computational time and memory required for learning is reduced, and the learning process can accommodate temporal changes associated with tasks (Suresh et al., 2010). In most cases of humans executing tasks, the path of actions may vary, however, each path contains approximately a similar order of true segments. To effectively learn such sequences of tasks, there are two key challenges which are often encountered. The *segmentation* of tasks wherein given the observed task path, the start and end positions of constituent actions through the path are identified and *sequential learning* of essential underlying actions (Lioutikov et al., 2017). The task segmentation is critical in sequence learning for modelling and interpreting tasks information as it facilitates the adaptation of learning sequences in unseen situations (Krishnan et al., 2017).

The main contributions of the work presented in this paper are summarised as follows:

- The paper proposes a novel adaptive segmentation and sequence learning (ASSL) approach for human activity pattern discovery from unlabelled sequences of observed activities.

- Exploiting the temporal accumulated motion energy of human actions through activity sequences for extracting key actions points during activities.

- Applying the ASSL approach to different human activity datasets. Besides, the ASSL approach is compared with other well-known sequence learning approaches and the results are presented.

The remainder of this paper is organised as follows: Section 2 discusses works related to this paper. Section 3 describes the research methodology explaining an overview of the proposed framework. In Section 4, the method proposed in this work for unsupervised human activity segmentation is presented and Section 5 follows with a description of the sequence learning method used in learning the activity segments constructed. Section 6 describes the application of the proposed model to human activity datasets and the results obtained. In Section 7, the performance of the proposed ASSL is compared with other sequence learning approaches and conclusions of the work are drawn in Section 8.

2. Related Work

There is a growing interest in research related to learning human activity sequences. This section presents a review of relevant works in two categories: the segmentation of human activities for detecting constituent actions, and activity modelling through sequential learning/prediction.

2.1. Action Detection and Segmentation

Recently, human activity recognition has received much attention with a lot of research undertaken for its applications in different areas (Adama et al., 2018; Lara and Labrador, 2013; Presti and Cascia, 2016). Most of the proposed activity recognition models (Presti and Cascia, 2016) can attain impressive performances in their respective areas of application. The majority focus on supervised approaches to activity recognition in which there is a sufficient amount of labelled data available to build training models. However, in real-world situations where obtaining labels for activities is a rather daunting task, supervised methods for activity recognition may not be feasible (Adama et al., 2019). On the other hand, unsupervised learning methods, like clustering (Comaniciu and Meer, 2002) are best suited for such applications.

An aspect of activity recognition which tends to be a challenge for many systems is detecting underlying/constituents actions in activities. This information is important in determining the structure of activities which is important when considering trends or sequences in such activities (Li and Fu, 2014). Therefore, segmentation is performed on data to obtain partitions which represent certain characteristics in activities. This is a vital step in investigating activity sequences. Existing approaches to segmentation of human activity differ in terms of the following categories (Aminikhanghahi and Cook, 2017; Aminikhanghahi and Cook, 2019); the activity types that are modelled, the sensing technology used to acquire information and the computational intelligence methods used in the segmentation process.

With a focus on human activity recognition from 3D human skeleton joints information, i.e. the joint positions or angles, different methods have been proposed for detecting actions in an activity. The authors in (Li and Fu, 2014) proposed a method for detecting atomic actions which they call *actionlets* using motion velocity. The method combined the Harris corner detector and Lucas Kanade (LK) optical flow to get velocity magnitudes. In our previous work (Adama et al., 2019), a key frame extraction method using the combined motion energy of all body joints in an activity has been proposed. Other works using the kinetic energy poses to determine key poses in activities are found in (Nunes et al., 2017; Shan and Akella, 2014). These methods then apply different machine learning algorithms for classification of actions obtained for activity recognition.

2.2. Sequential Modelling of Activities

The study of sequence learning algorithms are reported (Suresh et al., 2010; Cui et al., 2016; Wen and Wang, 2017; Zhu et al., 2018). Sequence learning algorithms are used for the analysis of patterns generated through a series of observed information for recognition or classification of activities (Zhu et al., 2018). Machine learning researchers have studied sequence learning over so many decades. This led to the development of statistical models such as Hidden Markov Models (HMM) (Fine et al., 1998; Rabiner and Juang, 1986) and Autoregressive Integrated Moving Average (ARIMA) (Durbin and Koopman, 2012) which were introduced for time series and temporal pattern recognition problems (Cui et al., 2016). Recurrent Neural Networks (RNNs) have since evolved to solve sequence prediction problems due to their recurrent lateral structure. Long-Short Term Memory (LSTM), a type of RNN, have a unique ability to selectively pass information across time and are able to model significantly long-term dependencies due to the gating mechanism they possess (Hochreiter and Schmidhuber, 1997). LSTMs also can deal with the vanishing gradient problem. This has seen impressive performances in a variety of real-world applications.

Concerning human activities, attempts to model human activity sequences have been studied by various researchers (Wen and Wang, 2017; Medina-Quero et al., 2018) using different temporal models for human activities recognition. HMM is used over predefined motion features of 3D joint positions to learn the dynamics of human actions (Lv and Nevatia, 2006). Conditional Random Field (CRF) is another generative model employed in modelling human actions. The CRF is used in (Han et al., 2010) to estimate motion patterns that correspond to manifold subspace of 3D joint position features for human action recognition. Similar approaches employing generative models to model activities are also proposed in (Shan and Akella, 2014; Ofli et al., 2014). The 3D joint positions obtained through skeleton tracking tend to be noisy. Therefore, when the change between actions is small, without the accurate selection of features, recognising precise action states becomes difficult. This tends to undermine the performance of generative models. Such models require an adequate amount of data for training as they are prone to over-fitting. Dynamic Time Warping (DTW) (Choi and Kim, 2018) is another solution used in modelling actions by defining the distance between two temporal sequences of activity actions. The learning can then be achieved through nearest-neighbour classification. However, the performance of DTW is dependent on a good measure of the samples similarity. It could also suffer from temporal misalignment when handling periodic actions which could lead to degrading its performance (Li and Prakash, 2011). Reyes-Ortiz et al. (2016) have proposed a Transition-Aware Human Activity Recognition (TAHAR) system for the real-time classification of physical human activities. The system combined the probabilistic output of consecutive activity predictions of a Support Vector Machine (SVM) with a heuristic filtering approach to address issues regarding the occurrence of transitions between activities and unknown activities to the proposed learning algorithm. From their results, the system was able to situations with and without activities transition information. Similar works for sequential learning of human activities employing LSTM RNN are seen in the works by Liu et al. (2016) and Li et al. (2017).

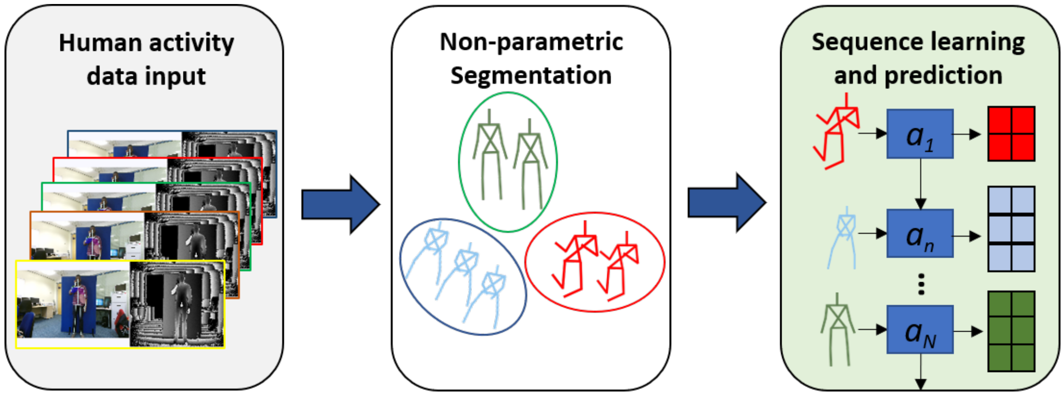


Fig. 2. Overview of the proposed approach to the Adaptive Segmentation and Sequence Learning (ASSL) of human activity.

These works demonstrate the effectiveness of segmentation and sequence modelling in exploring the underlying patterns in sequential data. This paper extends the approach of detecting key actions proposed in (Adama et al., 2019) for the segmentation of human activities by proposing an ASSL approach. Following from the identification of key actions, the non-parametric segmentation of 3D skeletal data of human activities obtained. This is then used in an LSTM model for the prediction of activity actions. In the following section, the problem statement is described and key definitions used in this work are presented.

3. Methodology

To address the challenges of segmentation and sequence learning of human activities, a novel framework for Adaptive Segmentation and Sequence Learning (ASSL) is proposed using visual information of activities. An overview of the ASSL framework is depicted in Fig. 2. There are three distinct steps in the proposed ASSL framework as described below:

1. Initially, key actions from observed human activity information are obtained. Human activities contain a large number of actions for which only the key aspects are relevant. By exploiting the temporal accumulated motion energy of each action through the sequence, the key actions can be extracted from the points of change in acceleration and deceleration of activity motion.

2. While segments of activities can be inferred from manual annotations, this creates a burden in *supervised* situations where high-dimensional data would require large

amounts of annotations to obtain feasible segments which can be learned. A non-parametric technique for feature space analysis is applied for *unsupervised* segmentation of relevant activity actions.

3. From the segments obtained, a Recurrent Neural Network (RNN) method for sequence learning called Long Short-Term Memory (LSTM) is used to learn activity sequences.

This work will benefit expert systems applications which require learning the underlying sequences in human actions through activities.

3.1. Definitions

Given a set of observed human activities $A = \{a_1, a_2, \ldots, a_n, \ldots, a_N\}$ performed by actors. The observations are obtained using an RGB-depth sensor. Each demonstration of an activity a_n within the observed activities set is a discrete time sequence of activity poses. An activity pose J as represented by;

$$J = [j_1, j_2, \ldots, j_m, \ldots, j_M], \quad \text{for } J \in \mathbb{R}^{3\times M}, \tag{1}$$

is a feature space which represents 3D human skeleton joints with coordinates. M represents the total number of joints in J with each joint, j_m, with coordinates x_m, y_m, z_m corresponding to horizontal, vertical and depth positions respectively.

Definition 1. Key action, $\overline{J}$ is defined as the important atomic level action performed during an activity. Key actions extracted from an activity represent a subset of poses $\overline{J} \subset a_n$, for $n = 1, 2, ..., N$, which occurs in varying time instants of an executed activity.

Definition 2. Activity segmentation is defined by a function C in which each key action, $\overline{J}_b$, $b = 1, 2, ..., B$, of an activity a_n is assigned a value, Q_z, $z = 1, 2, ..., Z$, corresponding to a unique activity segment represented as:

$$C : a_n \longmapsto (\overline{J}_b)_{1,2,\ldots,B}, \quad \text{for } \overline{J}_b \in Q_z \tag{2}$$

where b is the index of the key action through the activity sequence and B is the number of key actions contained in a_n. Each segment derived comprises similar activity key actions through a temporal sequence.

Definition 3. Activity action sequence, S, is defined as the temporal ordering of all B key actions obtained from activity a_n. A repetition of similar key actions may be observed in the sequence at points where a_n contains actions which are repeated at different temporal instances. A representation of this definition is presented as:

$$S = (\overline{J}_b)_{b=1}^{B} \tag{3}$$

3.2. Assumptions

For the research presented in this paper, certain assumptions are made. They are:

- The observed sequence of a human activity comprises of unlabelled atomic actions which this work aims to identify through a process of adaptive segmentation.

- The number of key poses $\overline{J}_B$ that make up an activity is not given. This is drawn from the fact that each activity can be segmented into key poses which make up for the relevant aspects that define the activity. However, this number is not pre-defined from activity observations in the proposed model.

3.3. Problem Statement 1

Given an observed sequence of human activity obtained using an RGB-depth sensor, the first phase is the segmentation of an unlabelled sequence into meaningful representations of similar actionlets. The segments obtained represent a collection of similar actions which may (or may not) fulfil temporal order relationship constraints.

The task of segmentation from an unlabelled activity sequence is addressed in this work using an adaptive approach to segmentation. The following steps are proposed for use in obtaining the function C for the segmentation of an activity.

Detection of key actions (or poses): Key actions of an activity are relevant in the process of learning an activity sequence. This is mainly because an activity can be executed in different forms whilst certain key aspects through the observation of an activity can uniquely identify it. As mentioned in the Introduction section, the motion energy feature of actions through an activity can be used in obtaining these key actions. The key actions are therefore identified by applying a filtering method of moving average crossovers of the motion energy. The description of how this is implemented is presented in the next Section.

Non-parametric feature space clustering: The key actions obtained from the filtering process of the motion energy feature are clustered using a Mean-Shift feature space analysis method. This method performs the clustering in terms of similarity of the motion energy of key actions.

3.4. Problem Statement 2

To learn the sequence S of transition of actions from one activity segment to another, it is important to note that an activity is not executed in only one possible sequence. An activity can be executed with different temporal orders of constituent actions. This results in a challenge of learning a generalised sequence for an activity.

The sequence of actions from one segment to another occur in intervals. The LSTM-RNN algorithm, which is predominantly used in predicting time series, is applied in learning the sequence of distinct actions within the activity segments. This method is used as the algorithm is able to capture infinitely long sequences and predict succeeding occurrences based on the memory gates.

The architecture of the ASSL approach for human activities from 3D skeleton information as proposed in this paper is depicted in Fig. 3. This comprises three stages of activity data input from an RGB-Depth sensor, segmentation of human activity and sequential learning and prediction. Details of these stages are provided in the proceeding sections.

4. Activity Segmentation

Segmentation of human activity is relevant in the analysis of trends in transitions from one activity state to another. This section describes the process of activity segmentation using the extracted human activity information.

4.1. Key Action Point Detection with Motion Energy

Human activity consists of movement sequences generated by different body parts. It is worth noting that not all aspects of an activity movement sequence are necessary to define an activity. Certain aspects of the sequence can be executed in different forms and still result in a similar activity. To simplify an activity to the relevant action points that constitute the

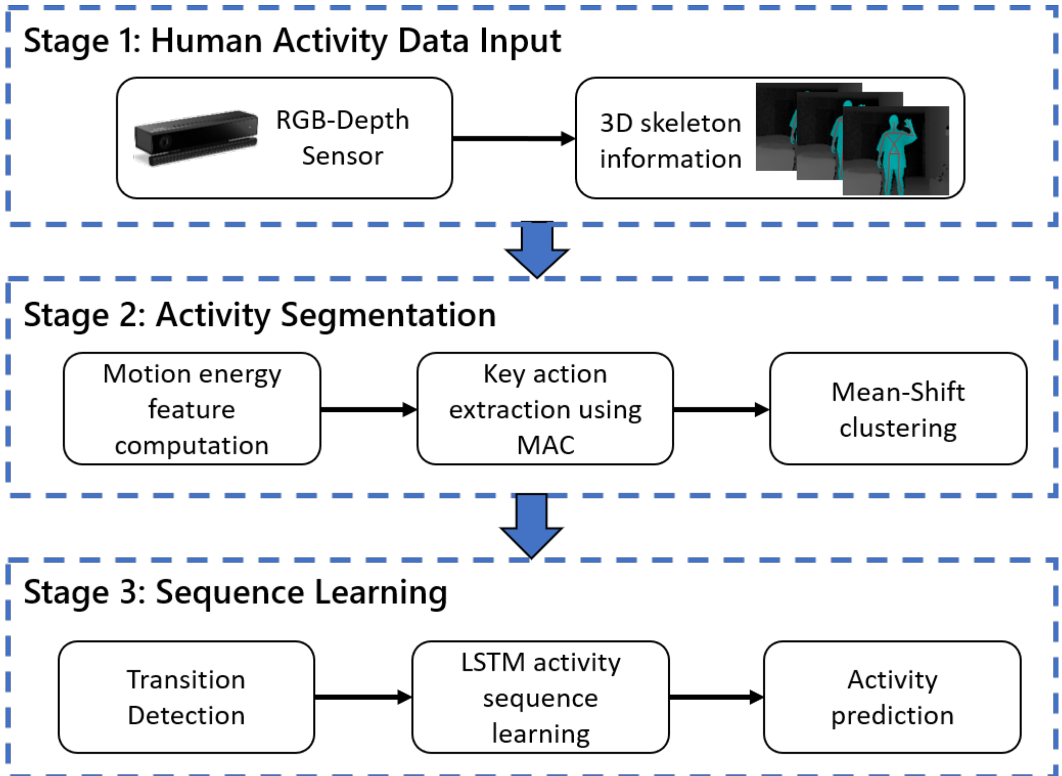


Fig. 3. Architecture of the proposed ASSL approach for human activities from 3D skeleton information which comprises activity input, segmentation and sequence learning stages respectively.

sequence, key poses are selected. This is achieved by leveraging the motion energy obtained from activity sequences.

4.1.1. Extraction of Motion Energy

The motion energy of activity poses as first proposed by (Shan and Akella, 2014) is based on the fact that joints show changes in acceleration and deceleration through an activity. This information is significant when considering the identification of the key action points of activities. Following from the representation of an activity pose given in Equation 1, the motion energy E_l for each pose is computed as the sum of motion energies for each joint in the pose;

$$E_l(J) = \sum_{m=1}^{M} E_l(j_m) \tag{4}$$

where j_m is a joint in the pose. It is assumed that the mass of all joints to be equally one unit due to the fact that it is impossible to obtain the actual mass of a joint from the information obtained using RGB-Depth sensors. Computing the joint velocities using the temporal change ΔT in the position d of joints during an activity, the motion energy can be expressed as:

$$E_l(J) = \frac{1}{2}\sum_{m=1}^{M}(v_{j_m})^2 \tag{5}$$

where, v_{j_m} represents the velocity of joint j_m and is expressed as $v_{j_m} = \frac{d_m^c - d_m^p}{\Delta T}$, d_m^c is the current joint position and d_m^p is the previous joint position. By substituting v_{j_m} in Equation 5, the motion energy of each joint is computed using the following equation:

$$E_l(J) = \frac{1}{2}\sum_{m=1}^{M}\left(\frac{d_m^c - d_m^p}{\Delta T}\right)^2 \tag{6}$$

4.1.2. Moving Average Crossover of Motion Energy

The Moving Average (MA) is a filtering technique often applied to get overall trends in data. This technique is used to highlight long-term cycles in time series data by smoothing out short-term variations (Droke, 2001). It works by creating series of averages of different time windows from a dataset over a given distribution.

Most of the works employing motion energy for identifying key action points of activities set threshold values of energy from a random exploration of selected points in order to extract the relevant points of interest in an activity (Nunes et al., 2017; Shan and Akella, 2014; Zhu et al., 2015). The energy thresholds are selected by repeated experiments of different threshold values and the observations below the threshold value are selected as key poses. The MA of the extracted motion energy of poses are used in filtering the motion energy signal extracted from an activity sequence.

A different approach is proposed to use crossovers of two Simple Moving Averages (SMAs) of the extracted motion energy in identifying the relevant key poses of an activity. The SMA is an un-weighted mean of a set of data points. This is taken from equal sets of data to ensure variations in the mean and data points are aligned and not shifted in time. Given the motion energy obtained in Equation 4, the SMA for the motion energy signal of an activity can be computed using the following expression:

$$SMA = \frac{\sum_{r=0}^{\alpha-1} E_l(J)_{t-r}}{\alpha} \tag{7}$$

where α is the value of the period selected for MA and $t - r$ is the position of the selected observation within α. This is expressed in a simplified form as follows;

$$SMA_{E_l} = \frac{E_l(J)_t + E_l(J)_{t-1} + ... + E_l(J)_{t-(\alpha-1)}}{\alpha} \tag{8}$$

Two moving averages are selected in this work - a short-term average (fast moving average) α_f and a long-term moving average (slow moving average) α_s. The MA crossovers are obtained from points where the SMAs for both α_f and α_s intersect. These points indicate significant changes in motion energy of activity poses and are used as reference points for their corresponding key actions in an activity sequence as presented in the following equation.

$$\overline{J_b} = SMA_{\alpha_s} \cap SMA_{\alpha_f} \tag{9}$$

Following the acquisition of the key action points, activity segments are obtained by application of a non-parametric feature space analysis technique - In this case, mean-shift clustering for associating key actions to clusters of similar actions.

4.2. Non-Parametric Clustering for Segmentation

Prior to learning the sequence of actions in an activity for prediction, it is necessary to know the segments that make up an activity. This information is not easily determined by mere observation of the key actions obtained from exploration of the motion energy feature. Also, the number of segments is defined for an activity as these can vary depending on the sequence observed. Therefore, the use of a non-parametric method of clustering key actions is proposed to determine the number of segments in an activity sequence and assign the obtained key actions to their respective segments before learning can be achieved. A mean-shift clustering approach is adopted here (Comaniciu and Meer, 2002). The mean-shift approach builds upon the concept of Kernel Density Estimation (KDE) (Parzen, 1962) which estimates the hidden distribution for a dataset by placing a kernel on each point contained in the dataset. The description of the mode of application for the proposed segmentation of human activity is provided below.

Given B key action points, $\overline{J_b}$ $b = 1, ..., B$, on a 2-dimensional space computed for an activity. As described in Section 4.1, these points correspond to the motion energies of key

Algorithm 1 Segmentation of human activity from joints coordinate skeleton information.

Input:

Instances of 3D skeleton joints coordinate of human activities $A = \{a_1, a_2, ..., a_N\}$, in which each observation of activity a is a pose $J = [j_1, j_2, ..., j_M]$;

Activity time window t;

Moving average periods α_s, α_f;

Output:

Activity segments obtained as a function C for assigning each key action to a segment;

Procedure:

```
for a = 1 to N do
    Find the velocity of each observation J within t;
    Compute the motion energy for J: E_l(J) = sum_{m=1}^{M} E_l(j_m);
    Compute the simple moving average of the motion energy with the periods α_s, α_f:
  SMA = (sum_{r=0}^{α-1} E_l(J)_{t-r}) / α;
    Key action points, J̄_b = SMA_{α_s} ∩ SMA_{α_f};
end for
Assign J̄_b to a cluster Q_z which is determined by a non-parametric mean-shift clustering
  technique;
return Q_Z = C(J̄_b).
```

action positions. The kernel density estimate for the key action points with kernel K with a bandwidth parameter h is:

$$f(\overline{J}) = \frac{1}{Bh^2} \sum_{b=1}^{B} K\left(\frac{\overline{J} - \overline{J_b}}{h}\right) \tag{10}$$

with K satisfying the following two conditions:

1. $\int K(\overline{J}) d\overline{J} = 1, \quad$ and

2. $K(\overline{J}) = K(|\overline{J}|) \quad$ for all values of $\overline{J}$.

The first condition is required to ensure normalisation of the density estimate while the second condition relates to the symmetry of the data space containing all key action points of an activity. By applying a Gaussian symmetric kernel function for $K(\overline{J})$, the gradient of the density estimator in Equation 10 takes the form:

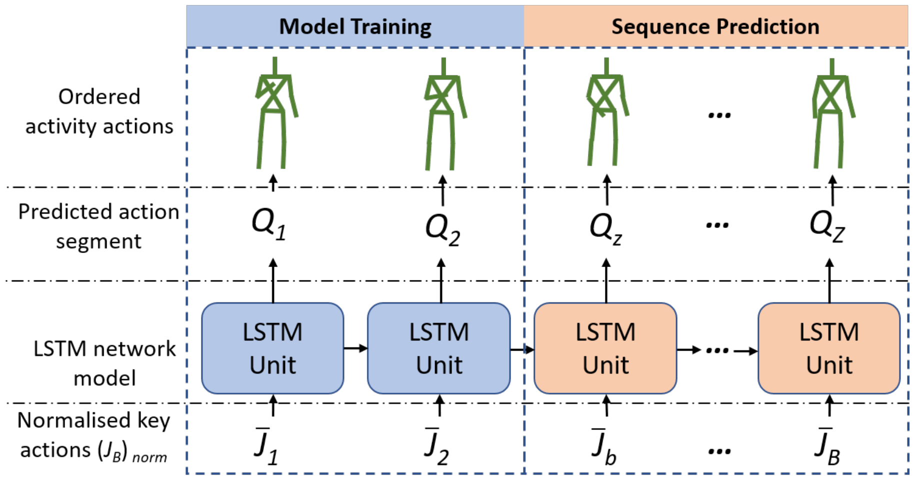


Fig. 4. LSTM structure for sequential learning and prediction of key action segments of human activity.

$$\nabla f(\overline{J}) = \frac{2}{Bh^4}\left(\sum_{b=1}^{B} g\left(\left|\frac{\overline{J}-\overline{J_b}}{h}\right|\right)\right)\vec{X}(\overline{J}) \tag{11}$$

where $\vec{X}(\overline{J})$ is the mean-shift vector pointing in the direction of increasing density and is represented as:

$$\vec{X}(\overline{J}) = \left(\frac{\sum_{b=1}^{B}\overline{J_b}g\left(\left|\frac{\overline{J}-\overline{J_b}}{h}\right|\right)}{\sum_{b=1}^{B} g\left(\left|\frac{\overline{J}-\overline{J_b}}{h}\right|\right)} - \overline{J}\right) \tag{12}$$

and $g(|\overline{J}|)$ is the derivative of the Gaussian kernel.

With the KDE computed, the mean-shift procedure is carried out by successive;

- Computation of the mean-shift vector $\vec{X}(\overline{J_b})$ at the location of each key action point $\overline{J_b}$,
- Translation of each action point $\overline{J_b} \rightarrow \overline{J_b} + \vec{X}(\overline{J_b})$,
- Repeat until convergence, that is, where the gradient density function is zero.

Afterwards, the key action points identified at the same points are segmented as belonging to the same cluster Q_z. For further details of convergence, readers are referred to (Comaniciu and Meer, 2002). Algorithm 1 list the procedure for activity segmentation proposed in this paper.

5. Sequence Learning and Prediction Model

The sequence learning stage involves the learning of activity sequences from the segmented key actions. An LSTM network (Hochreiter and Schmidhuber, 1997) is used to learn the long-term contextual dependencies between key actions of an activity. The segmented key actions are used as input to the network for learning the dependencies between the action segments. This is further extended to predicting sequential actions of activities. Fig. 4 illustrates the structure of an LSTM network as applied in this work. The LSTM comprises of the following components: input gate i_t, forget gate f_t, a cell with a self-recurrent connection and output gate o_t. The key action segments obtained for an activity are normalised for standardisation of the values, thus resulting in $Q_{norm} = \{\overline{J}_{1Q_1}, ..., \overline{J}_{BQz}\}_{norm}$. By taking Q_{norm} as input to the network, the network is updated every t timestep by iterating through all instances of the normalised key actions using the following equations;

$$i_t = \sigma(W^i(\overline{J}_{bQ_z}(t)) + U^i H_{t-1} + V^i) \tag{13}$$

$$f_t = \sigma(W^f(\overline{J}_{bQ_z}(t)) + U^f H_{t-1} + V^f) \tag{14}$$

$$o_t = \sigma(W^o(\overline{J}_{bQ_z}(t)) + U^o H_{t-1} + V^o) \tag{15}$$

$$g_t = tanH(W^g(\overline{J}_{bQ_z}(t)) + U^g H_{t-1} + V^g) \tag{16}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{17}$$

$$H_t = o_t \odot tanH(c_t) \tag{18}$$

where, $\sigma(\cdot)$ and $tanH(\cdot)$ are the sigmoid and hyperbolic functions respectively. W, U, V are parameters of the LSTM model. The operation $\odot$ denotes the element-wise multiplication of two vectors. The use of LSTM is due to its ability to map input activity sequences by recursively transforming current inputs Q_{norm} with the output hidden vector of previous steps H_{t-1}. Also, the vanish gradient problem inherent with RNN's is overcomed by the memory cell c_t which is computed, allowing the error derivatives to flow in a different path.

6. Application of Adaptive Segmentation and Sequence Learning Framework to 3D Skeleton Data of Daily Human Activity

This section reports the experimental procedure and results of applications of the proposed ASSL framework on 3D skeleton human activity datasets. To illustrate the application of the proposed work of ASSL of human activity sequences, the model proposed was applied to selected human activities. The proposed model is adaptive to different activities and thus gives it the ability to deal with complexities in activities.

To understand the methodology and its ability to solve the problems identified in the earlier Sections 3.3 and 3.4, the following hypotheses are proposed and evaluated.

Hypothesis 1. Where an unlabelled sequence of activity data is available, the segmentation technique proposed can be used to identify unique segments of an activity used for label assignments of actions in the sequence.

Hypothesis 2. Activity segments identified can be used to learn sequences for prediction with a reliable performance.

To address these hypotheses, two activities are selected from two real world human activity datasets; Dataset 1 - An experimental human activity dataset collected for this work and Dataset 2 - A benchmark public dataset, Cornell Activity Dataset (CAD-60) (Sung et al., 2011).

6.1. Experimental Design and Datasets

The motivation for the proposed ASSL framework is to address the issue of unlabelled sequences of human activities, in such cases where there is no knowledge *a priori* of constituent actions and their order, whilst there is the need to develop a system for identifying the patterns of activities. The experimental design and datasets used in evaluating the proposed framework are presented in this section.

6.1.1. Dataset 1 - Experimental Human Activity Dataset

The dataset generated to evaluate the proposed system in this work consists an activity which involves a person picking up an object placed on a surface. A Microsoft Kinect version 2 RGB-Depth sensor is used to acquire the 3D joint coordinate information of person. This information is obtained at 30 fps. This activity is chosen due to the proposed

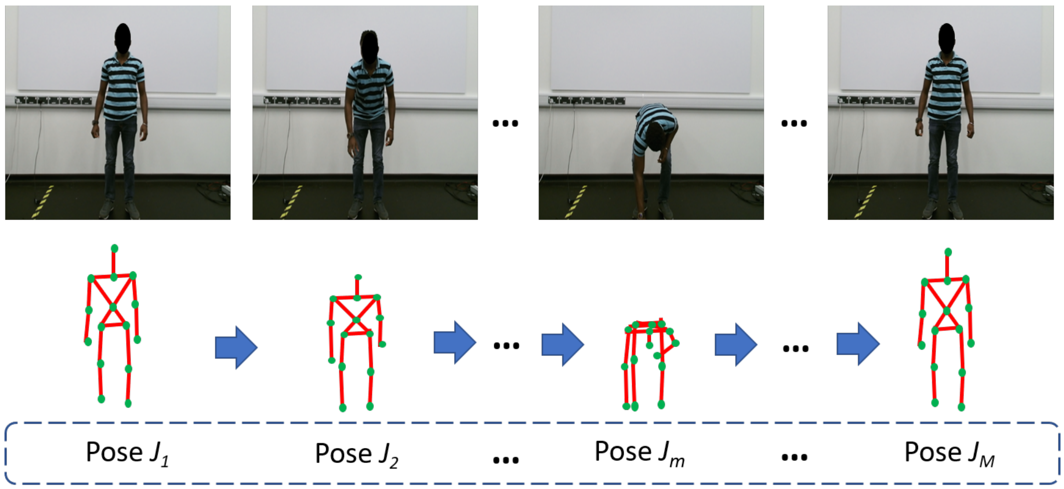


Fig. 5. Sample frames of *pick up object* activity obtained from the experimental activity dataset using an RGB-Depth sensor.

work being focused on enhancing the ability of assistive robots learning activity sequences for independent prediction of actions. Fig. 5 shows sample frames of the selected activity carried out by a person.

To obtain adequate amount of data to evaluate the ASSL framework, the activity is performed by three people. Each person is required to pick up an object from a flat surface repeatedly eight to ten times while the joint positions are recorded throughout the sequence. Table 1 shows the number of frames acquired from each person while carrying out the activity.

6.1.2. Dataset 2 - Cornell Activity Dataset (CAD-60)

The CAD-60 dataset (Sung et al., 2011) is based on human activity data obtained using an RGB-Depth sensor. The dataset comprises three modes of human activities data,

Table 1: Experimental dataset acquired from three actors for an activity - pick up object from a flat surface.

Activity	**Number of frames**			**Total**
	Person 1	**Person 2**	**Person 3**	
Pick up object	1804	1663	1355	4822

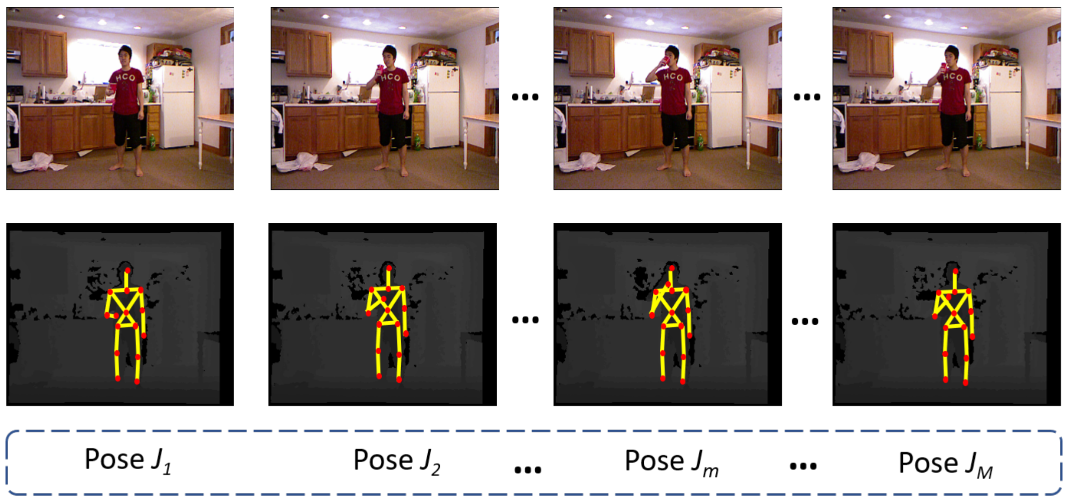


Fig. 6. Sample frames of *drinking water* activity obtained using an RGB-Depth sensor contained in the CAD-60 dataset (Sung et al., 2011). The sample shows RGB images and the corresponding depth image with the tracked skeleton overlaid.

RGB images, Depth images and 3D skeleton joint coordinates observed from a person performing an activity. The skeleton joint data consists of joint coordinates information of 15 joints. The dataset is recorded at a frame rate of 15fps using a Microsoft Kinect sensor and includes recordings for 12 human activities namely; Rinsing mouth, brushing teeth, wearing contact lens, talking on the phone, drinking water, opening pill container, cooking (chopping), cooking (stirring), talking on couch, relaxing on couch, writing on whiteboard, working on computer and a sequence of random plus stationary activities. The data is collected from four participants with each performing each activity.

Most applications of this dataset are based on activity classification and therefore involve the use of all activities within the dataset. However, to demonstrate the work proposed in this paper, a single activity from the dataset is selected and used in our evaluations. The activity chosen is the *drinking water* activity as there are more motions involved in the activity when compared to the remainder activities available in the dataset. This creates a scenario with varying motion patterns to test the robustness of the framework. Sample frames of varying actions occurring throughout the activity sequence are shown in Fig. 6. The samples show a person drinking water with the tracked skeleton joints overlaid on the depth images. The activity is performed repeatedly $2-3$ times.

6.2. Experimental Human Activity Dataset Results and Evaluation

To evaluate the performance of the proposed framework on the experimental dataset, it is implemented in stages, starting with the segmentation process - the computation of motion energy, detection of key action points and the non-parametric clustering for key action segmentation. This is then followed by the sequence learning and prediction of the obtained key actions.

6.2.1. Key Action Identification using Motion Energy

Applying the approach to identifying key action points of an activity, the motion energy is computed for 3D joint positions data obtained from each person. A window size, w_s, of one second is used which corresponds to 30 frames of activity to compute the motion energy. Fig. 7a shows the motion energy obtained from person 1 of the experimental dataset. The figure shows the changes in the cumulative motion energy which is a result of continuous acceleration and deceleration of body joints through the activity sequence.

In the proposed framework, the key actions are identified at points of minimum and maximum motion energies. Applying the simple moving average technique, after multiple experiments with different values of SMA_{α_s} and SMA_{α_f}, 30 and 15 frames are selected for both moving averages respectively. Fig. 7b depicts the key action points identified from the motion energy computed in Fig. 7a. The green plot shows the SMA_{α_s} while the red plot shows the SMA_{α_f}. The crossover points of both moving averages are identified by the blue dots in Fig. 7b. These points represent the key actions $\overline{J_B}$ in the activity sequence from the data. Similarly, the key actions are obtained for all participants in the experimental dataset.

6.2.2. Non-parametric Clustering of Experimental Dataset

Due to the varying nature of the activities performed from one individual to another, there are variations in motion energy values from person to person. To tackle this difficulty, the motion energy of the key actions identified for each participant's activity are normalised for standardisation across all participants. Fig. 8 shows the representation of normalised motion energies of identified key actions for all persons in the dataset. A total of 202 key action frames are identified from all three participants which shows a significant reduction when compared to the total number of frames 4822 as shown in Table 1. This emphasises the

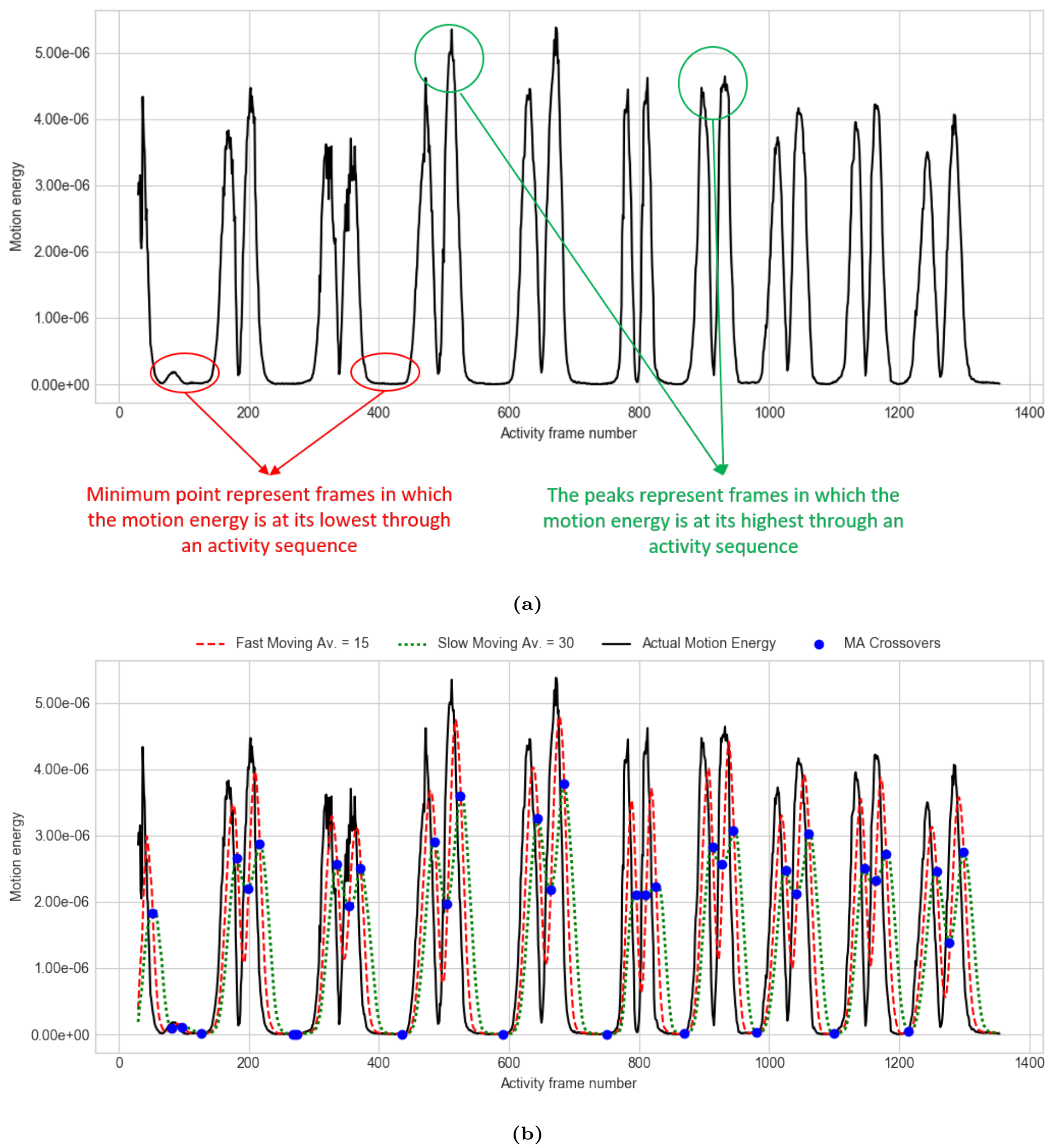


Fig. 7. Key action identification for *pick up object* activity from person 1 in the experimental dataset. (a) Motion energy plot for person 1 from the experimental dataset. The energy is computed using a 1 second window = 30 frames. (b) Motion energy plot with identified crossover points of two moving averages which represent the identified key action points of the activity. $SMA_{\alpha_f} = 15$ and $SMA_{\alpha_s} = 30$.

need for the segmentation process to reduce the computational complexities when learning the activity sequence.

The normalised values are then clustered using the non-parametric method described

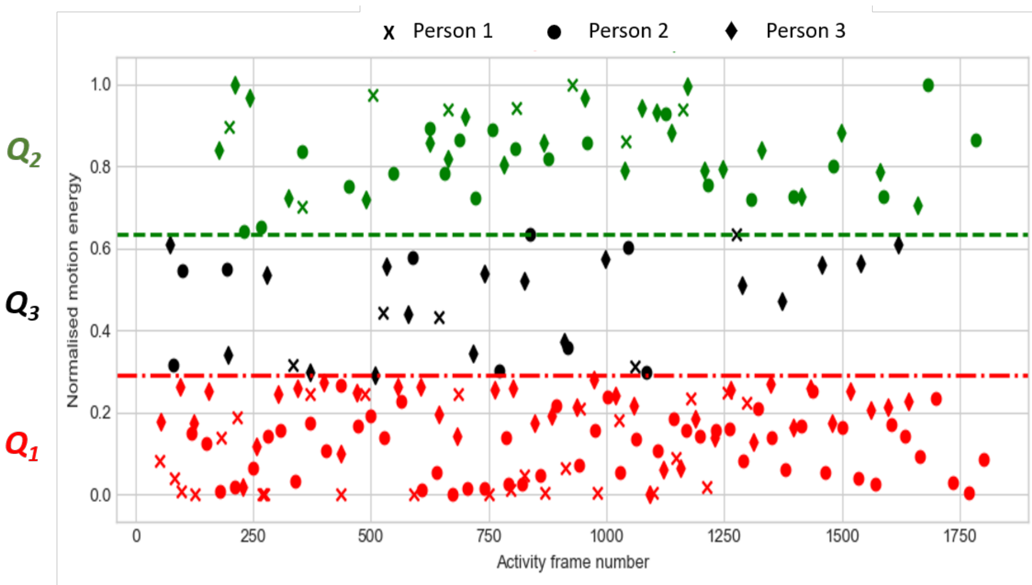


Fig. 8. Normalised motion energy with action segment identification of key actions for all participants in the experimental human activity dataset corresponding to the *pick up object activity*.

earlier. The results obtained from clustering is also represented in Fig. 8. It can be observed that for the selected activity three segments corresponding to Q_1, Q_2 and Q_3, are identified and the boundaries of the segments as obtained from the results are represented by the horizontal line plots (green and orange) shown on the figure. Fig. 9 shows the distribution of the number of key action points identified in each activity segment for all participants.

6.2.3. Sequence Learning of Experimental Human Activity Dataset

The sequence learning model is grounded on the results obtained from the activity segmentation process. To investigate the performance, the outputs from the segmentation process are fed as input to the learning model and a comparison is made between the results obtained and the actual activity sequence observed. This comparison is done in terms of the Mean Absolute Error (MAE), Mean Absolute Scaled Error (MASE) and Root-Mean-Square Error (RMSE) for the predictions made. The performance of the sequence learning model in this work depends on a proper segmentation of the unlabelled activity sequences observed.

The performance of the sequence learning framework is evaluated on the experimental dataset. Considering the dataset consists of 3 participants, a leave-one-out cross validation

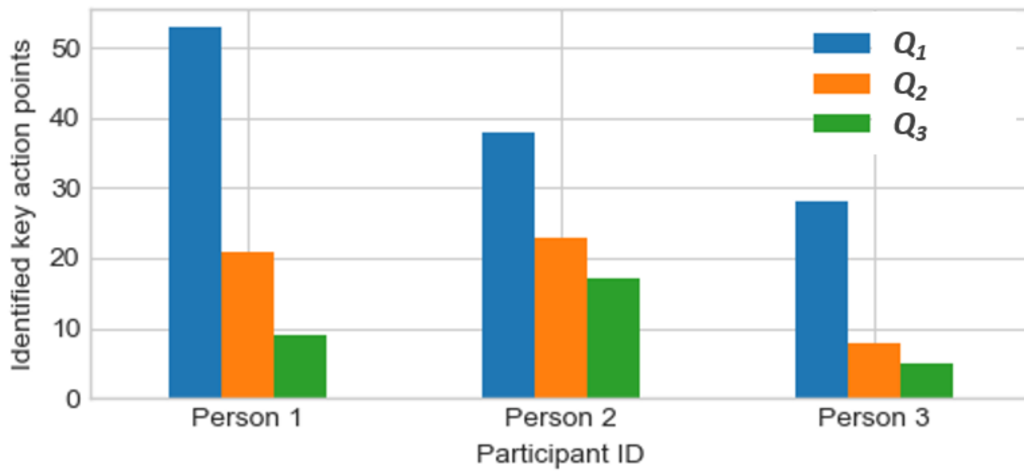


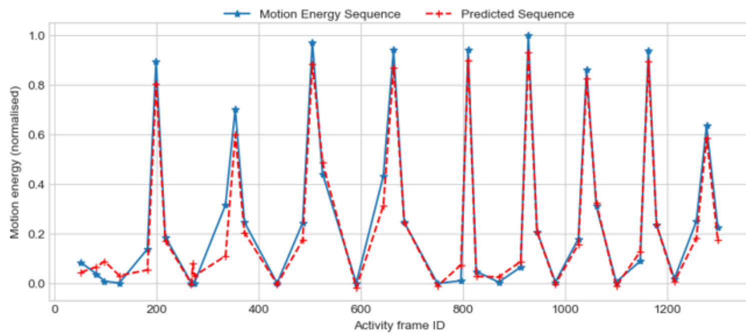
Fig. 9. Activity segmentation distribution for participants in the experimental human activity dataset.

approach is used in experiments to learn sequences of key action occurrences for an activity. Two participants are used in training the model and the remainder is left out for testing. This is done through consecutive iterations with each participant used in testing the model.

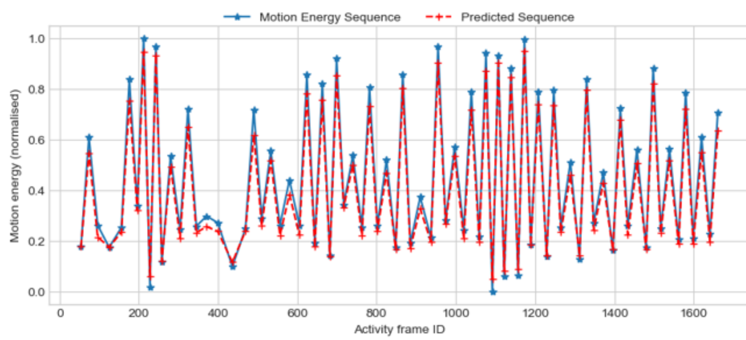
Fig. 10 shows the result of the sequence learning model on the prediction of the activity sequence contained in the experimental dataset. Table 2 shows the result when the experimental dataset is applied to the proposed ASSL model. The results produced RMSE values of 0.055, 0.049 and 0.050 respectively for all three participants in the dataset when each was tested using the leave-one-out cross validation. The lower the RMSE value the better the result in predicting the sequence. The variation in the structure of the sequence between the remainder two person's data used when training the model and the structure of the person 1 used in testing the model produced a higher RMSE value (0.055) in comparison with the RMSE value obtained for the other two. This can be attributed to the nature of the activity sequence for person 1, i.e. the speed of the activity.

6.3. CAD-60 Dataset Results and Evaluation

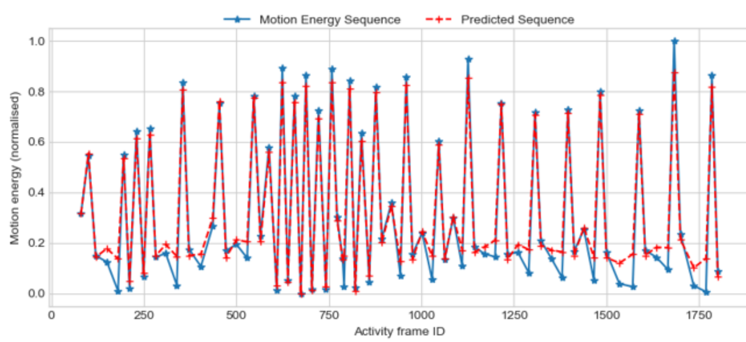
The segmentation process applied to the CAD-60 dataset using the same values of simple moving averages as in the case of the experimental activity dataset to identify key actions which are segmented resulted in a similar number of activity segments. The distribution of key actions identified in each segment is given Fig. 11. This shows a similar ratio in the distribution of key actions identified for all actors except for the case of *actor 1*. This



(a)



(b)



(c)

Fig. 10. Performance of sequence learning model on the prediction of experimental dataset activity sequence. (a) Person 1. (b) Person 2. (c) Person 3.

infers that for the activity - *drinking water* - performed by all actors, there are three atomic actions that define the activity. The order in which the actions occur define the activity sequence. It is important to note that the segments identified in the experiments with the CAD-60 experiment are not the same as those of the experimental activity dataset.

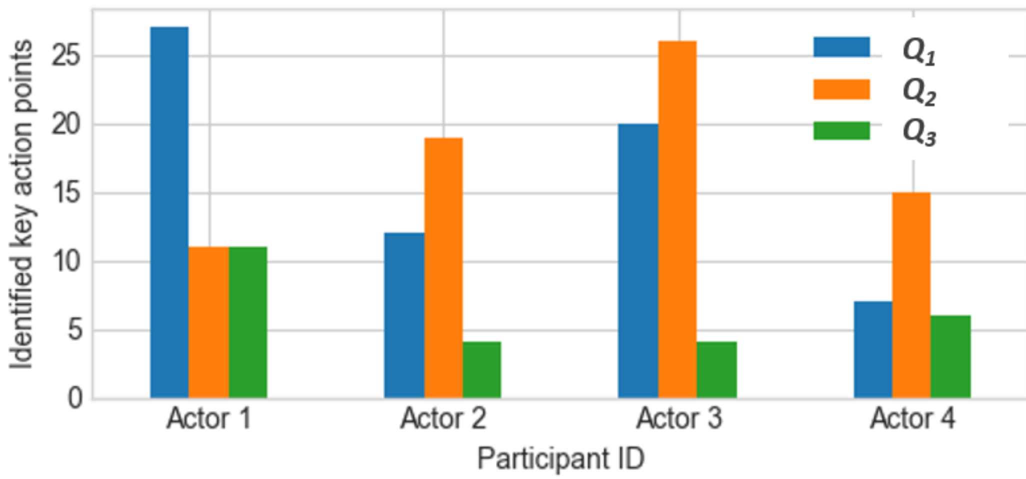


Fig. 11. Distribution of key action points in identified activity segments for all actors in the CAD-60 dataset.

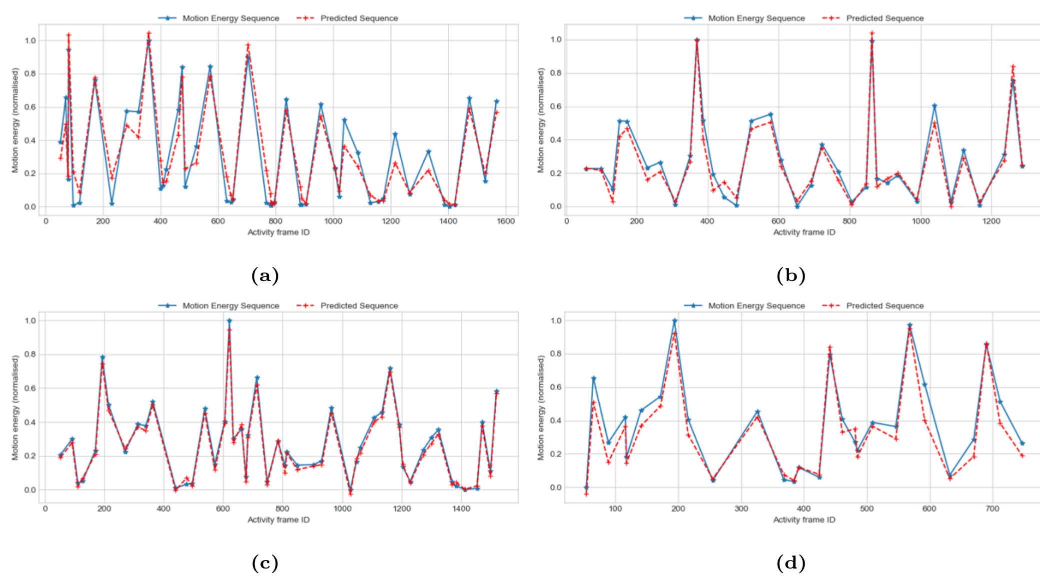


Fig. 12. Prediction performance of sequence learning model on the CAD-60 dataset. (a) Actor 1. (b) Actor 2. (c) Actor 3. (d) Actor 4.

Evaluating the performance of the sequence learning framework with the CAD-60 dataset is implemented in a similar method to the experimental dataset. A leave-one-out cross validation approach is also applied with each participant data used in testing while the remainder three are used in training the model. This is performed in consecutive iterations.

In Fig. 12, the prediction results for all actors are shown. The plots in the figure represent when each actors' activity data is left out from the training process and used to test the trained sequence learning model. Table 3 shows the prediction results obtained for the dataset with the ASSL. The RMSE values produced from predicting activity sequences for the data tested correspond to 0.092, 0.053, 0.025 and 0.078 for Actor's $1, 2, 3$ and 4 respectively. The low RMSE values show the model is able to learn with a high degree of reliability the activity sequence.

It should be noted that in the experiments a consideration was given to test the sequence learning model without the process of segmentation to extract key actions, that is, using the motion energy of all actions within the activity. This was done in the evaluation of the proposed model. Using all the actions, the clustering stage identified the actions as belonging to one cluster as opposed to the output of the clustering using the segmented key actions. From the visual observation of the activity, it is clear that this activity consists of more than one distinguishable action. Also, for both datasets used, the sequence learning model performed poorly in predicting the action sequences. This could be due to all actions identified as being the same.

7. Comparison with other Sequence Learning Models

This section presents a comparison of the proposed ASSL framework's performance with other statistical models widely used in learning sequences from time series data. The adaptive segmentation and sequence learning of 3D skeleton data of human activities framework primarily demonstrates that unlabelled actions and sequences of activities can be modelled for accurate prediction of unseen actions. This is beneficial for applications that require exploiting the underlying patterns to understand human tasks from visual observations while they are executed. This was demonstrated in the previous sections. To further emphasise the ability of the proposed framework to learn activity sequences, a comparison is made with other methods of sequence learning used in forecasting applications, an Autoregressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR) (Gascon-Moreno et al., 2012; Awad and Khanna, 2015) and Gaussian Mixture Regression (GMR). The basis for selecting the ARIMA model is because it comes from a well established area of computational intelligence. ARIMA models are also widely used in analysis of temporal pattern recognition and time series prediction. The SVR and

GMR models are techniques mostly applied in batch learning problems for forecasting purposes. These models are applied to both the experimental dataset and CAD-60 dataset described in Sections 6.1.1 and 6.1.2 respectively, with the same validation technique already described.

Autoregressive Moving Average (ARMA) models are amongst the most widely used statistical algorithms for modelling and predicting time series information (Smith et al., 2018). A generalisation of this model is the Autoregressive Integrated Moving Average (ARIMA) which is applied in situations where there is evidence of non-stationarity in data. In such cases, a differencing step, d, corresponding to the *Integrated* part of the model is applied to remove non-stationarity points (Ümit Çavus Büyüksahina and Ertekin, 2019). Afterwards, the ARMA model is applied on the stationary data. The implementation of ARIMA in this work follows the method described in (Ümit Çavus Büyüksahina and Ertekin, 2019). The Auto-Regressive, AR, component uses weighted linear combinations of previous values of the data sequence and performs a regression of the sequence against itself. Similarly, the Moving Average, MA, component attempts predicting a target using regression based

Table 2: Comparison of the proposed ASSL model's performance with ARIMA, SVR and GMR models on the experimental human activity dataset (the best results across all models in bold text).

Metric	**Method**	**Person 1** (error ± var.)	**Person 2** (error ± var.)	**Person 3** (error ± var.)
MAE	ASSL	**0.044 ± 0.005**	**0.025 ± 0.006**	**0.032 ± 0.004**
	ARIMA	0.228 ± 0.032	0.135 ± 0.036	0.132 ± 0.069
	SVR	0.057 ± 0.005	0.076 ± 0.006	0.072 ± 0.006
	GMR	0.345 ± 0.090	0.407 ± 0.090	0.309 ± 0.077
MASE	ASSL	0.152 ± 0.005	**0.122 ± 0.006**	**0.047 ± 0.004**
	ARIMA	0.586 ± 0.032	0.272 ± 0.036	0.291 ± 0.069
	SVR	**0.141 ± 0.005**	0.153 ± 0.006	0.244 ± 0.006
	GMR	0.849 ± 0.090	0.823 ± 0.090	1.046 ± 0.077
RMSE	ASSL	**0.055 ± 0.005**	**0.049 ± 0.006**	**0.050 ± 0.004**
	ARIMA	0.298 ± 0.032	0.198 ± 0.036	0.175 ± 0.069
	SVR	0.075 ± 0.005	0.088 ± 0.006	0.081 ± 0.006
	GMR	0.457 ± 0.090	0.506 ± 0.090	0.414 ± 0.077

Table 3: Comparison of the proposed ASSL model's performance with ARIMA, SVR and GMR models on the CAD-60 dataset (the best results are in bold text).

Metric	Method	**Actor 1** (error ± var.)	**Actor 2** (error ± var.)	**Actor 3** (error ± var.)	**Actor 4** (error ± var.)
MAE	ASSL	**0.072** ± **0.023**	**0.044** ± 0.015	**0.023** ± 0.012	**0.062** ± 0.018
	ARIMA	0.307 ± 0.190	0.202 ± 0.077	0.220 ± 0.109	0.255 ± 0.122
	SVR	0.123 ± 0.023	0.100 ± **0.014**	0.089 ± **0.010**	0.100 ± **0.017**
	GMR	0.302 ± 0.117	0.273 ± 0.062	0.239 ± 0.050	0.357 ± 0.093
MASE	ASSL	**0.281** ± **0.023**	**0.336** ± 0.015	**0.442** ± 0.012	**0.253** ± 0.018
	ARIMA	0.865 ± 0.190	0.690 ± 0.077	0.983 ± 0.109	0.802 ± 0.122
	SVR	0.312 ± 0.023	0.385 ± **0.014**	0.452 ± **0.010**	0.341 ± **0.017**
	GMR	0.765 ± 0.117	1.045 ± 0.062	1.208 ± 0.050	1.215 ± 0.093
RMSE	ASSL	**0.092** ± **0.023**	**0.053** ± 0.015	**0.025** ± 0.012	**0.078** ± 0.018
	ARIMA	0.339 ± 0.190	0.267 ± 0.077	0.264 ± 0.109	0.326 ± 0.122
	SVR	0.153 ± 0.023	0.119 ± **0.014**	0.105 ± **0.010**	0.130 ± **0.017**
	GMR	0.456 ± 0.117	0.370 ± 0.062	0.326 ± 0.050	0.469 ± 0.093

on past forecast errors. The parameters of the ARIMA model corresponding to coefficients of the orders of the model are d, p and q. p represents the number of time lags to consider. When $p = 0$, the mode is reduced to a MA model of q order. Similarly, if $q = 0$, the model becomes AR of p order. Details of the selection of the optimal parameters for the ARIMA model used are beyond the scope of this work. Readers are referred to (Ümit Çavus Büyüksahina and Ertekin, 2019) for more insight into ARIMA.

The SVR model as a supervised learning approach, has been applied as an effective tool in real-value function estimation and is characterised by the use of kernels. The model is trained by using a symmetrical loss function which penalises high and low misestimates equally. This model is used in the evaluation process to validate the proposed ASSL models performance. Implementations of the SVR and GMR models follow the methods in (Sung, 2004; Awad and Khanna, 2015).

7.1. Evaluation of the Results of ARIMA, SVR and GMR Prediction Models on the Experimental Dataset

The normalised key action points of the motion energy extracted from the experimental human activity are used as input to the ARIMA, SVR and GMR models as mentioned earlier. The results shown in Table 2 present the performance of all the models on the experimental dataset. As observed from the table, the proposed ASSL model had a better performance in terms of the MAE and RMSE than all the other models when observed across all the participants in the dataset. There is a significant difference in the MAE and RMSE performance obtained with the ASSL method outperforming all the other models. Next to the MAE and RMSE performance of the ASSL, the SVR model obtained comparable performance. However, the SVR model did slightly better than the ASSL model in terms of the MASE performance for person 1. As with most unsupervised learning structures, the ARIMA is able to predict data sequences with only the targeted data. It can also be noted from the results of Table 2 that the GMR model had the least performance across all the participants when compared with the other models. The GMR algorithm is known to be a fast learning model as it maximises only the likelihood. However, when it encounters many points, estimating the covariance matrices tends to be difficult. Therefore, the model diverges.

7.2. Evaluation of the Results of ARIMA, SVR and GMR models on CAD-60 Dataset

Table 3 shows a comparison of the results obtained for the performance of the ASSL framework with the ARIMA, SVR and GMR models on the CAD-60 dataset. Similar to the performance obtained with the experimental dataset, the proposed ASSL model outperformed all the other models with lower error values across all four actors. Similarly, the GMR was the worst-performing model on the CAD-60 dataset, except for the MASE for actor 1 where the ARIMA model had the highest error value.

The ARIMA model works as a regression model and therefore does not require labelled samples. However, the proposed approach is able to obtain labels through a non-parametric approach which is used in the later stage of sequence learning. This gives the ASSL method an edge over the ARIMA.

8. Conclusion and Future Work

In this paper, a novel adaptive technique for the segmentation and sequential learning of human activities is presented. The goal is to enable the discovery unknown activity patterns for prediction of future actions in an activity sequence, especially, for use in assistive robotics. Due to the dynamic nature of human behaviour, there are uncertainties associated with modelling actions performed in an activity. This work focused on proposing a model capable of adapting to variations that exist in actions through activity sequences. The use of 3D skeleton joint data obtained with RGB-Depth sensors makes it possible to acquire representations of actions for learning such activities.

The motion energy of skeleton joints is used as a feature in the segmentation process. This is due to changes in acceleration and deceleration observed in skeleton joints through a continuous sequence of activities. This feature is used in identifying key actions in an activity sequence from the moving average crossovers of the computed motion energy. This steps acts as filter stage as not all actions of an activity are relevant in predicting the activity sequence. We leverage the ability of LSTM model in learning activity sequences for predicting future actions of activities based on previous instances. The results show the performance of the LSTM sequence learning model is better than the unsupervised sequence learning approaches. Furthermore, learning sequences of activity from unlabelled activity structures are addressed. The segmentation approach used to identify labels from the structures made it possible to solve the unsupervised learning problem with a supervised technique of learning sequences.

Due to the challenges in this research area, the work presented in this paper has some limitations which will be addressed in future work. The work presented in this paper can be extended to include more subjects used in the experiments. This is needed due to the variation that exist from person to person performing an activity. This will add robustness to the sequence learning models. Furthermore, more research will be done on improving the performance of the sequence learning and prediction models in order to reduce the predictions errors in the results. Specifically, other variants of the LSTM RNN such as Bidirectional-LSTM will be tested.

References

D. A. Adama, A. Lotfi, C. Langensiepen, K. Lee, P. Trindade, Soft Computing 22 (2018) 7027–7039.

D. O. Anderez, A. Lotfi, A. Pourabdollah, Expert Systems with Applications 140 (2020).

S. Suresh, K. Dong, H. Kim, Neurocomputing 73 (2010) 3012 – 3019.

R. Lioutikov, G. Neumann, G. Maeda, J. Peters, The International Journal of Robotics Research 36 (2017) 879–894.

S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, K. Goldberg, The International Journal of Robotics Research 36 (2017) 1595–1618.

O. D. Lara, M. A. Labrador, IEEE Communications Surveys Tutorials 15 (2013) 1192–1209.

L. L. Presti, M. L. Cascia, Pattern Recognition 53 (2016) 130 – 147.

D. A. Adama, A. Lotfi, R. Ranson, P. Trindade, in: 2nd UK-RAS Conference on Embedded Intelligence (UK-RAS19, pp. 60–63.

D. Comaniciu, P. Meer, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 603–619.

K. Li, Y. Fu, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014) 1644–1657.

S. Aminikhanghahi, D. J. Cook, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 262–267.

S. Aminikhanghahi, D. J. Cook, Pervasive and Mobile Computing 53 (2019) 75 – 89.

D. A. Adama, A. Lotfi, C. Langensiepen, in: Advances in Computational Intelligence Systems, Springer International Publishing, 2019, pp. 303–311.

U. M. Nunes, D. R. Faria, P. Peixoto, Pattern Recognition Letters 99 (2017) 21 – 31.

J. Shan, S. Akella, in: 2014 IEEE International Workshop on Advanced Robotics and its Social Impacts, pp. 69–75.

Y. Cui, S. Ahmad, J. Hawkins, Neural Computation 28 (2016).

J. Wen, Z. Wang, Expert Systems with Applications 74 (2017) 19 – 28.

H. Zhu, H. Chen, R. Brown, Journal of Biomedical Informatics 84 (2018) 148 – 158.

S. Fine, Y. Singer, N. Tishby, Machine Learning 32 (1998) 41–62.

L. Rabiner, B. Juang, IEEE ASSP Magazine 3 (1986) 4–16.

J. Durbin, S. Koopman, Time series analysis by state space methods, Oxford University Press, New York, 2012.

S. Hochreiter, J. Schmidhuber, Neural Comput. 9 (1997) 1735–1780.

J. Medina-Quero, S. Zhang, C. Nugent, M. Espinilla, Expert Systems with Applications 114 (2018) 441 – 453.

F. Lv, R. Nevatia, in: Computer Vision – ECCV 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 359–372.

L. Han, X. Wu, W. Liang, G. Hou, Y. Jia, Image and Vision Computing 28 (2010) 836 – 849.

F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Journal of Visual Communication and Image Representation 25 (2014) 24 – 38.

H.-R. Choi, T. Kim, Mathematical Problems in Engineering 2018 (2018).

L. Li, B. A. Prakash, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11, pp. 185–192.

J.-L. Reyes-Ortiz, L. Oneto, A. SamÃ , X. Parra, D. Anguita, Neurocomputing 171 (2016) 754 – 767.

J. Liu, A. Shahroudy, D. Xu, G. Wang, in: Computer Vision – ECCV 2016, pp. 816–833.

K. Li, X. Zhao, J. Bian, M. Tan, in: 2017 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 1556–1561.

C. Droke, Moving Averages Simplified, Marketplace Books, 2001.

G. Zhu, L. Zhang, P. Shen, J. Song, L. Zhi, K. Yi, in: 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1209–1214.

E. Parzen, The Annuals of Mathematical Statistics 33 (1962) 1065–1076.

J. Sung, C. Ponce, B. Selman, A. Saxena, in: Proceedings of the 16th AAAI Conference on Plan, Activity, and Intent Recognition, AAAIWS'11-16, AAAI Press, 2011, pp. 47–55.

J. Gascon-Moreno, S. Salcedo-Sanz, E. Ortiz-Garcia, J. Acevedo-Rodriguez, J. A. Portilla-Figueras, Expert Systems with Applications 39 (2012) 8220 – 8227.

M. Awad, R. Khanna, Support Vector Regression, Apress, Berkeley, CA, 2015, pp. 67–80.

E. M. Smith, J. Smith, P. Legg, S. Francis, in: Advances in Computational Intelligence Systems, pp. 191–202.

Ümit Çavus Büyüksahina, S. Ertekin, Neurocomputing 361 (2019) 151 – 163.

H. G. Sung, Gaussian Mixture Regression and Classification, Ph.D. thesis, Rice University, 2004.