Neural Networks 108 (2018) 466–478

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Soft + Hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection

Tharindu Fernando *, Simon Denman, Sridha Sridharan, Clinton Fookes

Image and Video Research Laboratory, SAIVT, Queensland University of Technology, Australia

ARTICLE INFO

Article history:
Received 17 February 2017
Received in revised form 31 August 2018
Accepted 5 September 2018
Available online 20 September 2018

Keywords:
Human trajectory prediction
Social navigation
Deep feature learning
Attention models

ABSTRACT

As humans we possess an intuitive ability for navigation which we master through years of practice; however existing approaches to model this trait for diverse tasks including monitoring pedestrian flow and detecting abnormal events have been limited by using a variety of hand-crafted features. Recent research in the area of deep-learning has demonstrated the power of learning features directly from the data; and related research in recurrent neural networks has shown exemplary results in sequence-to-sequence problems such as neural machine translation and neural image caption generation. Motivated by these approaches, we propose a novel method to predict the future motion of a pedestrian given a short history of their, and their neighbours, past behaviour. The novelty of the proposed method is the combined attention model which utilises both "soft attention" as well as "hard-wired" attention in order to map the trajectory information from the local neighbourhood to the future positions of the pedestrian of interest. We illustrate how a simple approximation of attention weights (i.e. hard-wired) can be merged together with soft attention weights in order to make our model applicable for challenging real world scenarios with hundreds of neighbours. The navigational capability of the proposed method is tested on two challenging publicly available surveillance databases where our model outperforms the current-state-of-the-art methods. Additionally, we illustrate how the proposed architecture can be directly applied for the task of abnormal event detection without handcrafting the features.

1. Introduction

Understanding and predicting crowd behaviour in complex real world scenarios has a vast number of applications, from designing intelligent security systems to deploying socially-aware robots. Despite significant interest from researchers in domains such as abnormal event detection, traffic flow estimation and behaviour prediction; accurately modelling and predicting crowd behaviour has remained a challenging problem due to its complex nature.

As humans we possess an intuitive ability for navigation which we master through years of practice; and as such these complex dynamics cannot be captured with only a handful of hand-crafted features. We believe that directly learning from the trajectories of pedestrians of interest (i.e. pedestrian whose trajectory we seek to predict) along with their neighbours holds the key to modelling the natural ability for navigation we possess. We propose an approach that identifies salient information in both the pedestrian of interest and neighbouring trajectories and fuses this to predict the future path. To simplify learning, we use hard-wired attention weights for neighbouring trajectories, which leads to 50% faster model training for a minimal decrease in accuracy.

The approach we present in this paper can be viewed as a data driven approach which learns the relationship between neighbouring trajectories in an unsupervised manner. Our approach is motivated by the recent success of deep learning approaches (Goroshin, Bruna, Tompson, Eigen, & LeCun, 2015; Lai, Bo, & Fox, 2014; Madry, Bo, Kragic, & Fox, 2014) in unsupervised feature learning for classification and regression tasks.

1.1. Problem definition

The problem we have addressed can be defined as follows: Assume that each frame in our dataset is first preprocessed such that we have obtained the spatial coordinates of each pedestrian at every time frame. Therefore the trajectory of the ith pedestrian for the time period of 1 to T_{obs} can be defined as,

$$\mathbf{x_i} = [x_1, y_1, \ldots, x_{T_{obs}}, y_{T_{obs}}]. \quad (1)$$

The task we are interested in is predicting the trajectory of the ith pedestrian for the period of T_{obs+1} to T_{pred}, having observed the trajectory of the ith pedestrian from time 1 to T_{obs} as well as the trajectories all the other pedestrians in the local neighbourhood

* Corresponding author.
E-mail address: t.warnakulasuriya@qut.edu.au (T. Fernando).

https://doi.org/10.1016/j.neunet.2018.09.002
0893-6080/

during that period. This can be considered a sequence to sequence prediction problem where the input sequence captures contextual information corresponding to the spatial location of the pedestrian of interest and their neighbours, and the output sequence contains the predicted future path of the pedestrian of interest.

1.2. Proposed solution

To solve this problem we propose a novel architecture as illustrated in Fig. 1. For encoding and decoding purposes we utilise Long–Short Term Memory networks (LSTM) due to their recent success in sequence to sequence prediction (Bahdanau, Cho, & Bengio, 2014; Xu et al., 2015; Yoo, Park, Lee, Paek, & Kweon, 2015). We demonstrate the social navigational capability of the proposed method on two challenging publicly available surveillance databases. We demonstrate that our approach is capable of learning the common patterns in human navigation behaviour, and achieves improved predictions for pedestrians paths over the current state-of-the-art methodologies. Furthermore, an application of the proposed method for abnormal human behaviour detection is shown in Section 5.

2. Related work

2.1. Trajectory clustering

When considering approaches for learning motion patterns through clustering, Giannotti, Nanni, Pinelli, and Pedreschi (2007) have proposed the concept of "trajectory patterns", which represents the descriptions of frequent behaviours in terms of space and time. They have analysed GPS traces of a fleet of 273 trucks comprising a total of 112,203 points. Deviating from discovering common trajectories, Lee, Han, and Whang (2007) proposed to discover common sub-trajectories using a partition-and-group framework. The framework partitions each trajectory into a set of line segments, and forms clusters by grouping similar line segments based on density. Morris and Trivedi (2009) evaluated different similarity measures and clustering methodologies to uncover their strengths and weaknesses for trajectory clustering. With reference to their findings, the clustering method had little effect on the quality of the results achieved; however selecting the appropriate distance measures with respect to the properties of the trajectories in the dataset had great influence on final performance.

In a different line of work, clustering algorithms have been utilised to predict the co-salient objects (Han, Cheng, Li, & Zhang, 2017) in a group of images (Yao, Han, Zhang, & Nie, 2017), and detect common salient events in a collection of videos (Zhang, Han, Jiang, Ye, & Chang, 2017). However these works significantly vary from the proposed model as they are working with dense pixel level features compared to the sparse trajectory features that we are considering.

2.2. Human behaviour prediction

When predicting human behaviour the most common motion models are social force models (Helbing & Molnár, 1995; Koppula & Saxena, 2013; Pellegrini, Ess, & Gool, 2010; Xu, Denman, Fookes, & Sridharan, 2012; Yamaguchi, Berg, Ortiz, & Berg, 2011) which generate attractive and repulsive forces between pedestrians. Several variants of such approaches exist. Alahi, Ramanathan, and Li (2014) represent it as a social affinity feature by learning the pedestrian trajectories with relative positions whereas Yi, Li, and Wang (2015) observed the behaviour of stationary crowd groups in order to understand crowd behaviour. With the aid of topic models the authors in Wang, Ma, Ng, and Grimson (2008) were able to learn motion patterns in crowd behaviour without tracking objects. This approach was extended to incorporate spatio-temporal dependencies in Emonet, Varadarajan, and Odobez (2011) and Hospedales, Gong, and Xiang (2009).

Deviating from the above approaches, a mixture model of dynamic pedestrian agents is presented by Zhou, Tang, and Wang (2015), who also consider the temporal ordering of the observations. Yet, this model ignores the interactions among agents, a key factor when predicting behaviour in real world scenarios.

The main drawback in all of the above methods is that they utilise hand-crafted features to model human behaviour and interactions. Hand-crafted features may only capture abstract level semantics of the environment and they are heavily dependent on the domain knowledge that we possess.

In Alahi et al. (2016) an unsupervised feature learning approach was proposed. The authors have generated multiple LSTMs for each pedestrian in the scene at that particular time frame. They have observed the position of all the pedestrians from time 1 to T_{obs} and predicted all of their positions for the period T_{obs+1} to T_{pred}.

They have pooled the hidden states of the immediately preceding time step for the neighbouring pedestrians when generating their positions in the current time step. A more detailed comparison of this model with our proposed model is presented in Section 3.3.

More recently, a number of deep learning based approaches (Bartoli, Lisanti, Ballan, & Del Bimbo, 2017; Varshneya & Srinivasaraghavan, 2017; Zou, Su, Song, & Zhu, 2018) have been proposed for analysing human behaviour, which all emphasise the importance of fully capturing context information when predicting future events. However, they cannot be directly used to solve the problem that we are focusing on: predicting a person of interest's future motion given their and their neighbours previous motion. For instance, in Varshneya and Srinivasaraghavan (2017) the authors utilise additional spatial information in the video frames to model the spatial context in the scene. In contrast we utilise only trajectory information. The work of Bartoli et al. (2017) uses hand labelled annotations regarding human–human and human–object interactions to aid the decision making process. Hand labelling such events is a tedious process and such annotations are typically not present in public databases.

Furthermore in Zou et al. (2018) the authors utilise very short trajectories with total length of 17 points, where they observe 9 points of the trajectory and predict the next 8 points. Their generative adversarial model requires a larger database to train hence they have utilised a dataset which has more than 40,000 key point tracklets, but with short trajectory segments. In contrast we are interested in modelling much longer trajectories (total of 40 frames) and we believe this task to be much more challenging, and have more practical applications in surveillance, automation and abnormal event detection, compared to modelling shorter trajectories.

2.3. Attention models

Attention-based mechanisms are motivated by the notion that, instead of decoding based on the encoding of a single element or fixed-length part of the input sequence, one can attend a specific area (or important areas) of the whole input sequence to generate the next output. Importantly, we let the model learn what to attend to based on the input sequence and what it has produced so far.

In Bahdanau et al. (2014) have shown that attention-based RNN models are useful for aligning input and output word sequences for neural machine translation. This was followed by the works by Xu et al. (2015) and Yao et al. (2015) for image and video captioning respectively. According to Sharma, Kiros, and Salakhutdinov (2015), attention based models can be broadly categorised into soft attention and hard attention models, based on the method

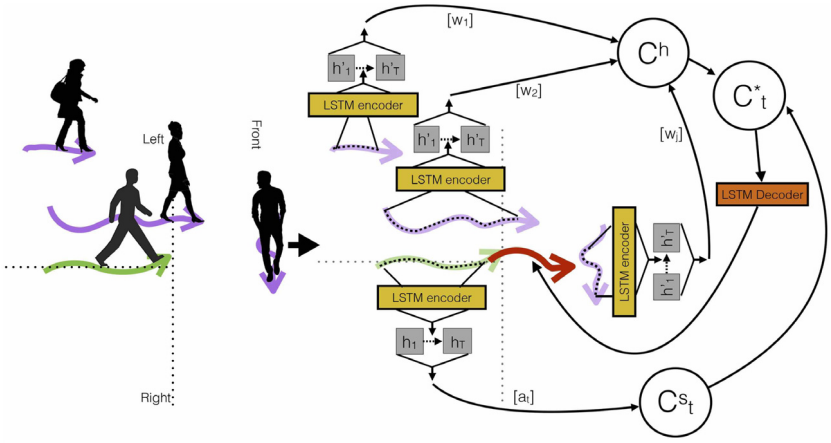


Fig. 1. A sample surveillance scene (on the left): The trajectory of the pedestrian of interest is shown in green, and has two neighbours (shown in purple) to the left, one in front and none on right. Neighbourhood encoding scheme (on the right): Trajectory information is encoded with LSTM encoders. A soft attention context vector C_t^s is used to embed the trajectory information from the pedestrian of interest, and a hardwired attention context vector C_t^h is used for neighbouring trajectories. In order to generate C_t^s we use a soft attention function denoted a_t in the above figure, and the hardwired weights are denoted by w. The merged context vector is then used to predict the future trajectory for the pedestrian of interest (shown in red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

that it uses to learn the attention weights. *Soft attention* models (Bahdanau et al., 2014; Sharma et al., 2015; Xu et al., 2015; Yao et al., 2015) can be viewed as "supervised" guiding mechanisms which learn the alignment between input and output sequences through backpropagation. *Hard attention* (Mnih, Heess, Graves, et al., 2014; Williams, 1992) is used by Reinforcement Learning to predict an approximate location to focus on. With reference to Sharma et al. (2015), learning hard attention models can become computationally expensive as it requires sampling.

Still, soft aligning multiple feature sequences is computationally inefficient as we need to calculate an attention value for each combination of input and output elements. This is feasible in cases such as neural machine translation where we have a 50-word input sequence and generate a 50-word output sequence, but prohibitively expensive in a surveillance setting when a target has hundreds of neighbours, and we have to learn the attention weight values for all possible value combinations for each of the neighbouring trajectories. We tackle this problem via the merging of soft attention and hardwired attention in our framework.

3. Proposed approach

3.1. LSTM encoder–decoder framework

In contrast to past attention models (Bahdanau et al., 2014; Xu et al., 2015) which align a single input sequence with an output sequence, we need to consider multiple feature sequences in the form of trajectory information from the pedestrian of interest and their neighbours when predicting the output sequence. Aligning all features together is not optimal as they have different degrees of influence (i.e. a person walking directly next to the target has greater influence than a person several metres away). Soft attention models are deterministic models which are trained using back-propagation. Therefore, aligning each input trajectory sequence separately via a separate soft attention model is computationally expensive.

We show that we can overcome this problem with a set of hardwired weights which we calculate based on the distance between each neighbour and the pedestrian of interest. When considering navigation, as distance is the key factor which determines the neighbour's influence, it acts as a good generalisation.

The proposed LSTM Encoder–Decoder framework is shown in Fig. 2. Due to its computational complexities, soft attention (denoted a_t in Fig. 2) is used only when embedding the trajectory information from the pedestrian of interest. We show that by approximating the required attention for the neighbours through hardwired attention weights (w_j), which we calculate based on the distance between the neighbouring pedestrian and the pedestrian of interest, we can generate a good approximation of their influence. The methodology of generating soft and hardwired attentions is outlined in the following subsections.

3.1.1. LSTM Encoder

In a general Encoder–Decoder framework, an encoder receives an input sequence $\mathbf{x}$ from which it generates an encoded sequence h. In the context of this paper, the input sequence for the pedestrian i is given in Eq. (1) and the encoded sequence is given by,

$$h_i = [h_1, \ldots, h_T]. \tag{2}$$

The encoding function is an LSTM, which can be denoted by,

$$h_t = \text{LSTM}(\mathbf{x}_t, h_{t-1}). \tag{3}$$

With the aid of above equation we encode the trajectory information from the pedestrian of interest as well as each trajectory in the local neighbourhood.

3.1.2. LSTM Decoder

Before considering how the combined context vector C_t^* is formulated, the concept of time dependent context vector can be illustrated as follows. For a general case, let s_{t-1} be the decoder hidden state at time $t-1$, $\mathbf{y}_{t-1}$ be the decoder output at time $t-1$, C_t be the context vector at time t and f be the decoding function. The decoder output at time t is given by,

$$\mathbf{y}_t = f(s_{t-1}, \mathbf{y}_{t-1}, C_t), \tag{4}$$

as defined by Bahdanau et al. (2014) such that distinct context vectors are given for each time instant. The context vector depends on the encoded input sequence $h = [h_1, \ldots, h_t]$.

In the proposed approach, the given trajectory (i.e. $\mathbf{x} = [x_1, y_1, \ldots, x_{T_{obs}}, y_{T_{obs}}]$) for the pedestrian of interest is encoded and used to generate a soft attention context vector, C_t^s. With the aid of distinct context vectors we are able to focus different degrees of attention towards different parts of the input sequence, when

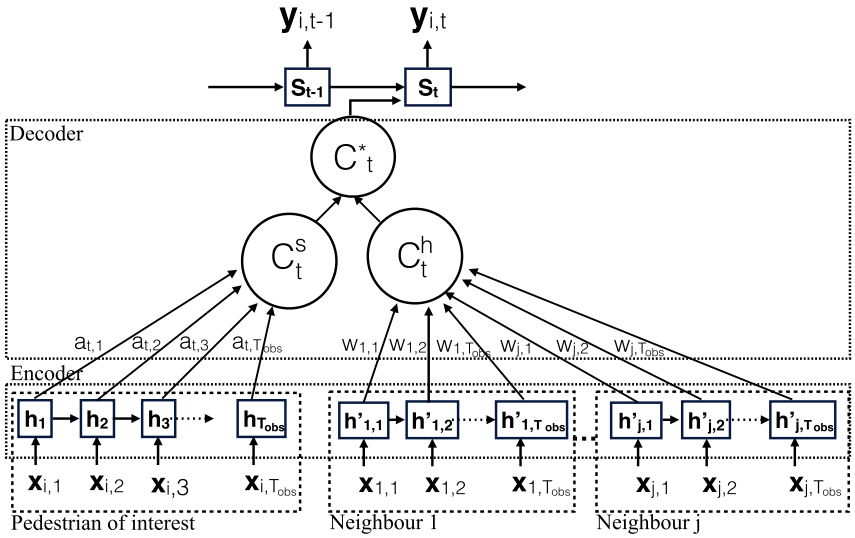


Fig. 2. The proposed *Soft + Hardwired* Attention model. We utilise the trajectory information from both the pedestrian of interest and the neighbouring trajectories. We embed the trajectory information from the pedestrian of interest with the soft attention context vector C_t^s, while neighbouring trajectories are embedded with the aid of a hardwired attention context vector C_t^h. In order to generate C_t^s we use a soft attention function denoted a_t in the above figure, and the hardwired weights are denoted by w. Then the merged context vector, C_t^*, is used to predict the future state $\mathbf{y}_{i(t)}$.

predicting the output sequence. The soft attention context vector C_t^s can be computed as a weighted sum of hidden states,

$$C_t^s = \sum_{j=1}^{T_{obs}} \alpha_{tj} h_j. \tag{5}$$

In Bahdanau et al. (2014), the authors have shown that the weight α_{tj} can be computed by,

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T} exp(e_{tk})}, \tag{6}$$

$$e_{tj} = a(s_{t-1}, h_j), \tag{7}$$

and the function a is a feed forward neural network for joint training with other components of the system.

As an extension to the decoder model proposed in Bahdanau et al. (2014), we have added a set of hardwired weights which we use to generate *hardwired attention* (C_t^h). Utilising the hardwired attention model we combine the encoded hidden states of the neighbouring trajectories in the local neighbourhood.

Hardwired attention weights are designed to incorporate the notion of distance between the pedestrian of interest and his or her neighbours into the trajectory prediction model. The closer a neighbouring pedestrian, the higher their associated weight, because that pedestrian has a greater influence on the trajectory that we are trying to predict.

The simplest representation scheme can be given by,

$$w_{(n,j)} = \frac{1}{\text{dist}(n,j)}, \tag{8}$$

where $\text{dist}(n,j)$ is the distance between the nth neighbour and the pedestrian of interest at the jth time instance, and $w_{(n,j)}$ is the generated hardwired attention weight. This idea can be extended to generate the context vector for the hardwired attention model.

Let there be N neighbouring trajectories in the local neighbourhood and $h'_{(n,j)}$ be the encoded hidden state of the nth neighbour at the jth time instance, then the context vector for the hardwired attention model is defined as,

$$C_t^h = \sum_{n=1}^{N} \sum_{j=1}^{T_{obs}} w_{(n,j)} h'_{(n,j)}. \tag{9}$$

We then employ a simple concatenation layer to combine the information from individual attentions. Hence the combined context vector can be denoted as,

$$C_t^* = \tanh(W_c[C_t^s; C_t^h]), \tag{10}$$

where W_c is referred to as the set of weights for concatenation. We learn this weight value also through back-propagation.

The final prediction can now be computed as,

$$\mathbf{y}_t = \text{LSTM}(s_{t-1}, \mathbf{y}_{t-1}, C_t^*), \tag{11}$$

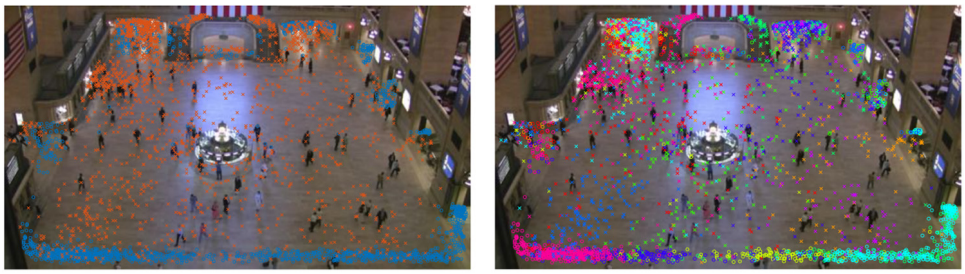
where the decoding function f in Eq. (4) is replaced with a LSTM decoder as we are employing LSTMs for encoding and decoding purposes.

3.2. Model learning

The given input trajectories in the training set are clustered based on source and sink positions and we run an outlier detection algorithm for each cluster considering the entire trajectory. For clustering we used DBSCAN (Ester, peter Kriegel, Sander, & Xu, 1996) as it enables us to cluster the data on the fly without specifying the number of clusters. As we do not possess the ground truth clusters, hyper parameters[1] of the DBSCAN algorithm were chosen experimentally. We analysed the clustering results visually and tuned the parameters until a sensible separation was obtained. Fig. 3(a) shows the source (in blue circles) and sink positions (in orange x) for the training set of trajectories in the GC dataset. It is clearly evident that the there exist multiple entry and exit points in the dataset. In Fig. 3(b) we show the clustering results where we have clustered the trajectories based on the source and sink positions. Source and sink positions are colour coded based on the respective cluster identity. It can be seen that the clustering algorithm has separated the trajectories based on the spatial dissimilarities they exhibit.

After clustering we learn a separate trajectory prediction model for each generated cluster. When modelling the local neighbourhood of the pedestrians of interest, we have encoded the trajectories of those closest 10 neighbours in each direction, namely

[1] epsilon = 0.50, minPts = 100.



(a) Unclustered points (b) Clustered points

Fig. 3. Source (as circles) and sink positions (as crosses) for the training set of trajectories in the GC dataset (Yi et al., 2015). (a) Unclustered points showing high variability in source and sink positions. (b) clustered trajectories based on source and sink, which generate a sensible grouping. Source and sink positions are colour coded based on the respective cluster identity.

front, left and right. If there exist more than 10 neighbours in any direction, we have taken the first (closest) 9 trajectories and the mean trajectory of the rest of the neighbours. If a trajectory has less than 10 neighbours, we create dummy trajectories such that we have 10 neighbours, and set the weight of these dummy neighbours to 0.

Note that the previous clustering process is performed only on the training set. When testing the model, we are concerned with predicting the pedestrian trajectory given the first T_{obs} locations. To select the appropriate prediction model to use, the mean trajectory for each cluster for the period of 1 to T_{obs} is obtained using,

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{n'} I_{j,i}\mathbf{x}_{i,t}}{\sum_{i=1}^{n'} I_{j,i}}, \quad \text{for t} \in \{1, \ldots, T_{obs}\} \text{ and j} \in \{1, \ldots, c\} \tag{12}$$

where there are n' trajectories in the training set, c number of clusters and $I_{j,i}$ is an indicator function which evaluates as,

$$I_{j,i} = \begin{cases} 1 & \text{if } \mathbf{x}_i \in j\text{th cluster.} \\ 0 & o.w \end{cases} \tag{13}$$

In the testing phase, the given trajectories are assigned to the closest cluster centre while considering those mean trajectories as the cluster centroids. This process is illustrated in Fig. 4.

3.3. Comparison to the Social-LSTM model of Alahi et al. (2016)

In this section we draw comparisons between the current state-of-the-art technique and the proposed approach. In Alahi et al. (2016), for each neighbouring pedestrian, the hidden state at time $t-1$ is extracted out and fed as an input to the prediction model of the pedestrian of interest. Let there be N neighbours in the local neighbourhood and $h'_{(n,t-1)}$ be the hidden state of the nth neighbour at the time instance $t-1$. Then the process can be written as,

$$H'_t = \sum_{n=1}^{N} h'_{(n,t-1)}, \tag{14}$$

and the hidden state of the pedestrian of interest at the tth time instance is given by,

$$h_t = \text{LSTM}(h_{t-1}, \mathbf{x}_{t-1}, H'_t), \tag{15}$$

where h_{t-1} refers to the hidden state and $\mathbf{x}_{t-1}$ refers to the position of the pedestrian of interest at the $t-1$ time instance. The authors are passing H'_t and $\mathbf{x}_{t-1}$ through embedding functions before feeding it to the LSTM model, but in order to draw direct comparisons we are using the above notation. In Alahi et al. (2016), the hidden state of the pedestrian of interest at the tth time instance depends only on his or her previous hidden state, the position in the previous time instance and the pooled hidden state of the immediately preceding time step for the neighbouring pedestrians (see Eq. (15)). Comparing to our model, we are considering the entire set of hidden states for the pedestrian of interest as well as the neighbouring pedestrians when predicting the tth output element (see Eqs. (5)–(11)).

As humans we tend to vary our intentions time to time. For an example consider the problem of navigating in a train station. A person may start walking towards the desired platform and then realise that he has not got a ticket and then make a sudden change and move towards the ticket counter. When applying the LSTM model proposed by Alahi et al. (2016) to such real world scenarios, by observing the immediate preceding hidden state one can generate reactive behaviour to avoid collisions but when doing long term path planning, even though the LSTM is capable of handling long term relationships, the prediction process may go almost "blindly" towards the end of the sequence (Jia, Gavves, Fernando, & Tuytelaars, 2015) as we are neglecting vital information about pedestrian's behaviour under varying contexts. In contrast, the proposed combined attention model considers the entire sequence of hidden states for both the pedestrian of interest and his or her neighbours and then we utilise time dependent weights which enables us to vary their influence in a timely manner.

Additionally, we observed that even in unstructured scenes such as train stations, airport terminals and shopping malls where multiple source and sink positions are present, still there exists dominant motion patterns describing the navigation preference of the pedestrians. For instance taking the same train station example, although the main problem that we are trying to solve here is to navigating while avoiding collisions, humans demonstrate different preferences in doing so. One pedestrian may be there to meet passengers and hence is wandering in a free area, while another pedestrian may aim to get in or out of the train station as quickly as possible. Therefore one single LSTM model is not sufficient to capture such ambiguities in navigational patterns. We observed that such distinct preferences in navigation generate unique trajectory patterns which can be easily segmented via the proposed clustering process. Therefore in contrast to Alahi et al. (2016), we are learning a different trajectory prediction model for each trajectory cluster.

4. Experiments

We present the experimental results on two publicly available human trajectory datasets: New York Grand Central (GC) (Yi et al., 2015) and Edinburgh Informatics Forum (EIF) database (Majecka, 2009). The Grand Central dataset consist of around 12,600 trajectories whereas the Edinburgh Informatics Forum database contains around 90,000 trajectories. We have conducted 2 experiments. For the first experiment on the Grand Central dataset, after

Fig. 4. Model selection process during testing illustrated on the GC dataset (Yi et al., 2015). We select the appropriate cluster for the input trajectory (shown as a dashed purple line) by considering the spatial distance between it and the respective cluster centroids which are created by taking the mean trajectory for each cluster for the period of 1 to T_{obs}.

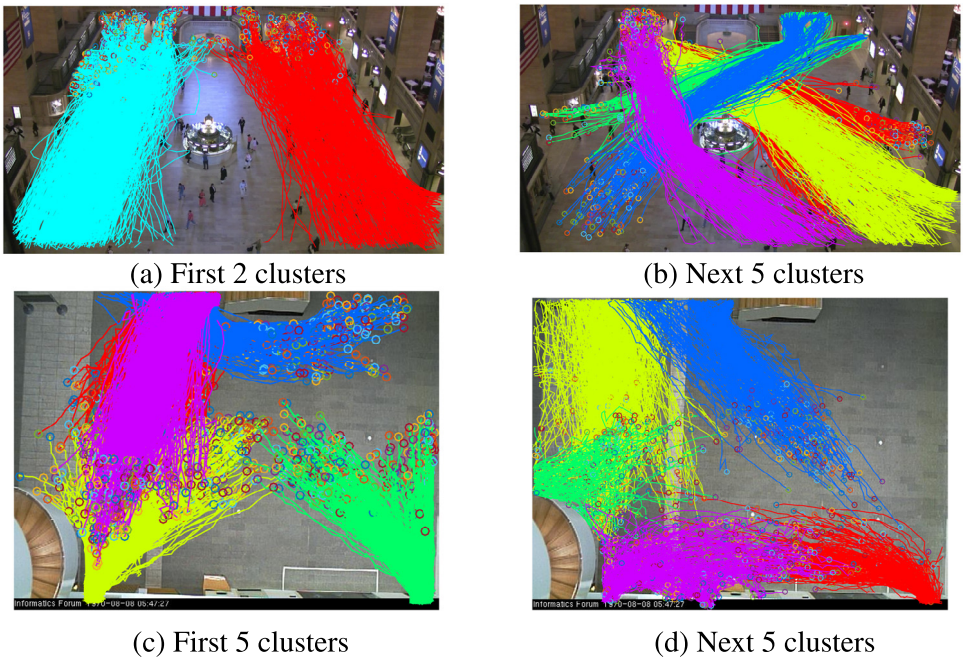


Fig. 5. Clustering results for Grand Central (a, b) (Yi et al., 2015) and Edinburgh Informatics Forum (c, d) (Majecka, 2009) Datasets.

filtering out short and fragmented trajectories,[2] we are left with 8000 trajectories, and train our model on 5000 trajectories and evaluate the prediction accuracy on 3000 trajectories. In the next experiment we considered 7 days worth trajectories from Edinburgh Informatics Forum database, trained our model on 10,000 trajectories and tested on 6000 trajectories (tracks from data files Jun 17 & 18, July 14 & 20, Aug 27, Sep 01 and Oct 09). These data files are specifically chosen as they possess more denser, hence more challenging, crowd behaviour compared to others. We could not utilise the entire 92,000 trajectories provided in the database as we needed to retain a comparatively similar sizes in between the two datasets.

As there are no standard training/testing splits for the above 2 datasets we separated the trajectories randomly. We trained the proposed model as well as all the baselines from scratch on the same splits, hence we can directly compare their evaluation results.

Prior to learning trajectory models, we employ clustering to separate the different modes of human motion. This allows us to learn separate models for different behaviours, such as one model for a pedestrian who is buying tickets and another for those who are directly entering or leaving the train station. We believe that these different motion patterns generate unique pedestrian behavioural styles and that are well captured through separate models. This can be achieved via clustering trajectories based on entire trajectory, but this will produce very large number of clusters or large number of outliers due to the wide variation in the different modes of human motion. As a result, in each cluster, we would have very few examples to train our prediction models on. Therefore as a solution to the above stated problem we cluster the trajectories based only on the enter/exit zones. As illustrated in Fig. 5 this approach works reasonably well at separating different modes of human motion.

[2] We consider short trajectories to be those with length less than the time period that we are considering (40 frames) whereas fragmented trajectories are trajectories which have discontinuities between 2 consecutive frames due to noise in the tracking process.

Even with clustering based on entry and exit points, the way that a person moves through the environment and how they are influenced by neighbours will vary considerably. For instance consider the clusters represented in green and blue in Fig. 5(b). The way that an intersecting trajectory travelling straight up in the scene, from bottom left towards up left, will affect green and blue clusters differently because of their different exit zones. It is general human nature to try to avoid collisions while keeping the expected heading direction. Therefore entry/exit zones based clustering is sufficient to capture this.

4.1. Quantitative results

Similar to Alahi et al. (2016) we report prediction accuracy with the following 3 error metrics. Let n be the number of trajectories in the testing set, $\mathbf{x}_{i,t}^{pred}$ be the predicted position for the trajectory i at tth time instance, and $\mathbf{x}_{i,t}^{obs}$ be the respective observed positions then,

1. *Average displacement error (ADE):*

$$ADE = \frac{\sum_{i=1}^{n}\sum_{t=T_{obs}+1}^{T_{pred}}(\mathbf{x}_{i,t}^{pred} - \mathbf{x}_{i,t}^{obs})^2}{n(T_{pred} - (T_{obs}+1))}. \tag{16}$$

2. *Final displacement error (FDE) :*

$$FDE = \frac{\sum_{i=1}^{n}\sqrt{(\mathbf{x}_{i,T_{pred}}^{pred} - \mathbf{x}_{i,T_{pred}}^{obs})^2}}{n}. \tag{17}$$

3. *Average non-linear displacement error (n-ADE):* The average displacement error for the non-linear regions of the trajectory,

$$n-ADE = \frac{\sum_{i=1}^{n}\sum_{t=T_{obs}+1}^{T_{pred}} I(\mathbf{x}_{i,t}^{pred})(\mathbf{x}_{i,t}^{pred} - \mathbf{x}_{i,t}^{obs})^2}{\sum_{i=1}^{n}\sum_{t=T_{obs}+1}^{T_{pred}} I(\mathbf{x}_{i,t}^{pred})}, \tag{18}$$

where,

$$I(\mathbf{x}_{i,t}^{pred}) = \begin{cases} 1 & \text{if } \frac{d^2 y_{i,t}}{dx_{i,t}^2} \neq 0. \\ 0 & o.w \end{cases} \tag{19}$$

In all experiments we have observed the trajectory (and its neighbours) for 20 frames and predicted the trajectory for the next 20 frames. Compared to Alahi et al. (2016), which has considered sequences of 20 frames total length, we are considering more lengthy sequences (with a total of 40 frames) as in Baccouche, Mamalet, Wolf, Garcia, and Baskurt (2011) the authors have shown that LSTM models tend to generate more accurate results with lengthy sequences.

In the experimental results, shown in Table 1, we compare our prediction model with the state-of-the-art. As the baseline models we implemented the Social Force (SF) model from Yamaguchi et al. (2011) and the Social LSTM (S-LSTM) model given in Alahi et al. (2016). For the S-LSTM model a local neighbourhood of size 32px was considered and the hyper-parameters were set according to Alahi et al. (2016). In order to make direct comparisons with Alahi et al. (2016), the hidden state dimensions of encoders and decoders of all OUR models were set to be 300 hidden units.

For the SF model, preferred speed, destination, and social grouping factors are used to model the agent behaviour. When predicting the destination, a linear support vector machine was trained with the ground truth destination areas detected in Section 3.2.

In order to evaluate the strength of the proposed model, we compare this combined attention model (OUR_{cmb}) with 4 variations on our proposed approach: (1) OUR_{pi}, which ignore the neighbouring trajectories and considers only the soft attention component derived from the trajectory of the person of interest when making predictions; (2) OUR_{sc} which omits the clustering stage such that only a single model (using combined attention weights) is learnt; (3) (OUR_{sft}) in which when calculating the influence of each trajectory, for the both pedestrian of interest as well as neighbouring trajectories, only soft attention is used; and (4) (OUR_{hw}) which utilises hard wired weights for both the pedestrian of interest as well as neighbouring trajectories influence. When calculating hardwired weights of the trajectory of pedestrian of interest we used the inverse weighted distance from last known point of the trajectory (i.e. $x_{T_{obs}}, y_{T_{obs}}$).

The proposed model outperforms the SF model and S-LSTM model on both datasets. The error reduction is more evident for the GC dataset where there are multiple source and sink positions, different crowd motion patterns are present and motion paths are heavily crowded. Comparing the results of $\mathbf{OUR_{sc}}$ (proposed approach without clustering) against the S-LSTM model we can see that regardless of the clustering process the proposed combined attention architecture is capable of improving the trajectory prediction. For all the measured error metrics $\mathbf{OUR_{sc}}$ has outperformed S-LSTM and (OUR_{pi}) demonstrating that it is important to preserve historical data for both the pedestrian of interest as well as the neighbours. Secondly when comparing $\mathbf{OUR_{sft}}$ and $\mathbf{OUR_{cmb}}$ it is evident that hard wired weights act as a good approximation of the neighbours influence without inducing a heavy computational cost (see Section 5.1 for details). The OUR_{hw} model has poor performance as it fails to approximate the importance of the historical behaviour. This is largely because when predicting the future trajectory, different portions of the historic trajectories carry various information that could be utilised for inferring in different situations. This information is context dependent and cannot be hardcoded. For instance the giving way behaviour of the pedestrian in Fig. 8(e)–(h), which is observed in first few frames is highly relevant when predicting behaviour of that pedestrian near large crowd groups. However it is of little importance in other contexts. Hence the OUR_{hw} model fails to generate accurate predictions.

When comparing results of $\mathbf{OUR_{cmb}}$ against $\mathbf{OUR_{sc}}$ it is evident that the clustering process has partitioned the trajectories based on these different semantics and via utilising separate models for each cluster, we are able to generate more accurate predictions. The combined attention model is capable of learning how the neighbours influence the current trajectory and how this impact varies under different neighbourhood locations.

Furthermore, we would like to point out that, because of the separate model learning process we were able to predict the final destination positions with more precision compared to baseline models where they do not consider the environmental configurations of the unstructured scene. While the proposed approach does not explicitly model the environment, a certain amount of environmental information is inherently encoded in the entry and exit locations, which the proposed approach is able to leverage.

4.2. Qualitative results

In Figs. 6 and 7 we show prediction results of the S-LSTM model, SF model and our combined attention model (OUR_{cmb}) on the GC and EIF datasets respectively. When observing Fig. 6 it should be noted that our model generates better predictions in heavily crowded areas as well as sparsely crowded areas in Fig. 7. As we are learning a separate model for each cluster, the prediction models are able to learn different patterns of influences from neighbouring pedestrians. For instance in the 1st and 3rd column of Fig. 6 we demonstrate how the model adapts in order to avoid collisions. In the last row of Fig. 6 we show some failure cases. The reason for such deviations from the ground truth was mostly due to sudden changes in destination. Even though these trajectories do not match the ground truth, the proposed method still generates

Table 1
Quantitative results. In all the methods forecast trajectories are of length 20 frames. The first 2 rows represent the Average displacement error, rows 3 to 4 are for Final displacement error and the final 2 rows are for Average non-linear displacement error.

Metirc	Dataset	SF	S-LSTM	OUR_{sc}	OUR_{pi}	OUR_{hw}	OUR_{sft}	OUR_{cmb}
ADE	GC	3.364	1.990	1.878	2.041	2.717	**1.092**	1.096
	EIF	3.124	1.524	1.392	1.685	1.430	**0.901**	0.986
FDE	GC	5.808	4.519	4.317	5.277	4.512	**2.873**	3.011
	EIF	3.909	2.510	2.345	3.089	2.932	**1.054**	1.311
n-ADE	GC	3.983	1.781	1.701	2.304	3.121	**0.976**	0.985
	EIF	3.394	2.398	2.098	2.415	2.997	**0.901**	**0.901**

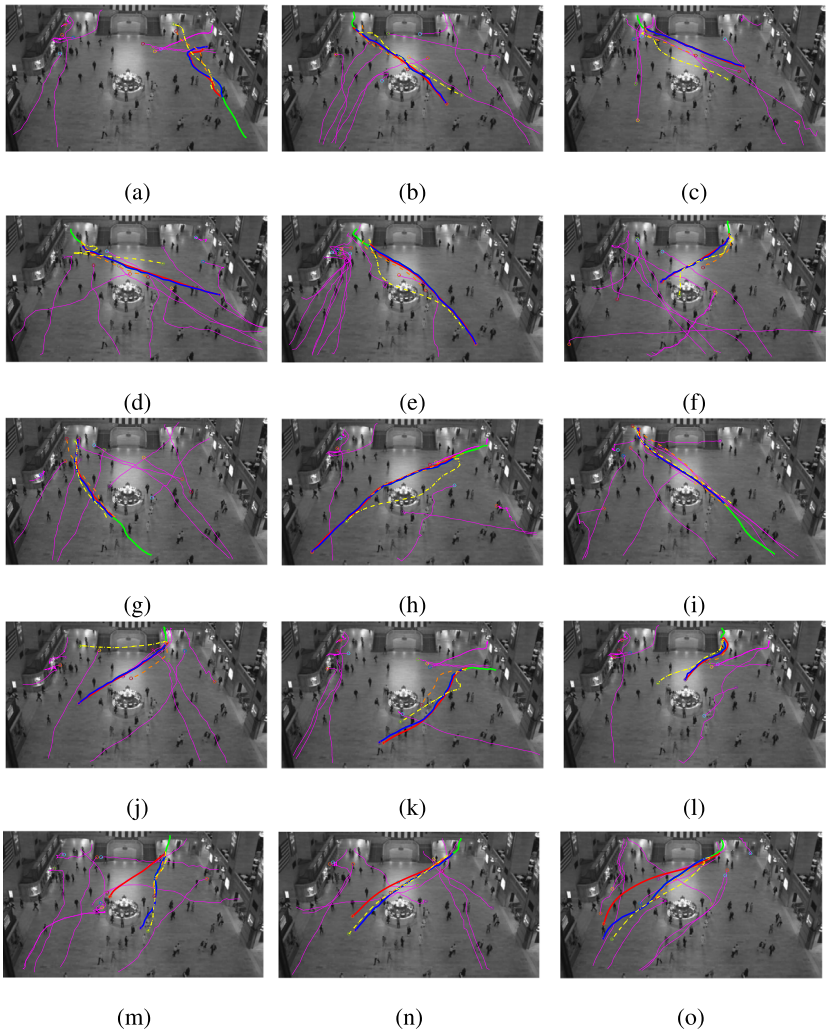


Fig. 6. Qualitative results for the GC dataset (Yi et al., 2015): Given (in green), Ground Truth (in Blue), Neighbouring (in purple) and Predicted trajectories from $\mathbf{OUR_{cmb}}$ model (in red), from **S-LSTM** model (in yellow), from **SF** model (in orange). (a) to (l) accurate predictions and (m) to (o) some erroneous predictions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

plausible trajectories. For instance, in Fig. 6(m) and (o) the model moves side ways to avoid collision with the neighbours in the left and right directions.

Three example scenarios that illustrate the advantage of attending to all the hidden states within that particular context are shown in Fig. 8. The first row shows an example where the pedestrian of interest exhibits 2 modes of motion (walking and running) within the same trajectory. In the second row we have an example where the previous context of the pedestrian of interest is useful in the final prediction. Third row shows an example where pedestrian exhibits a group motion. The first three columns show the progression of the motion from the start of the trajectory (first column) to the point directly before the prediction is made (third column). The given trajectory of the person of interest is shown in green and the neighbouring trajectories are shown in purple. In order to preserve visibility without occlusions we have plotted only the closest 2 neighbours in each direction. Finally in the fourth column we have presented the respective predictions from $\mathbf{OUR_{cmb}}$ model (in red), from **S-LSTM** model (in yellow), from $\mathbf{OUR_{sc}}$ model (in cyan). The ground truth observations are shown in blue.

When considering the example shown in Fig. 8(a)–(d) the pedestrian of interest exhibits a running behaviour when entering the scene (Fig. 8(a)). Then at half way through her trajectory she shifts her behaviour to walking (Fig. 8(b)). Therefore at the time of the prediction (Fig. 8(c)), the hidden states of the baseline model will be dominated by the walking behaviour as it is the

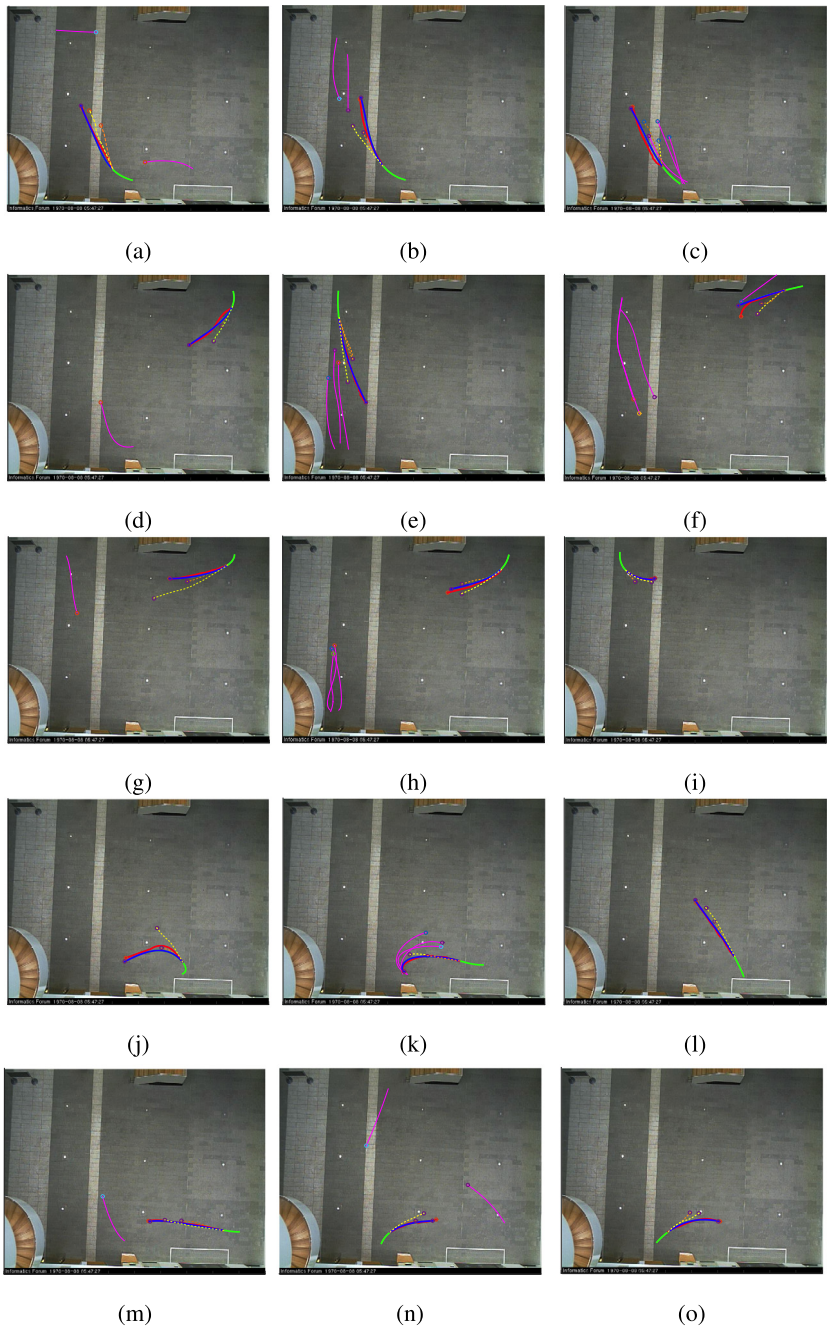


Fig. 7. Qualitative results for the EIF dataset (Majecka, 2009): Given (in green), Ground Truth (in Blue), Neighbouring (in purple) and Predicted trajectories from **OUR$_{cmb}$** model (in red), from **S-LSTM** model (in yellow), from **SF** model (in orange). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

most recent behaviour of this particular pedestrian. Therefore the predictions generated by the **S-LSTM** model are erroneous. But as we are attending to all the previous hidden states before predicting the future sequence, the multi model nature of that pedestrian's motion has been captured by the **OUR$_{sc}$** and **OUR$_{cmb}$** models.

Another example scenario where the proposed model out performs its baseline is shown in Fig. 8(e)–(h). When deciding upon whether to give way or to cut through the pedestrian group, models **OUR$_{sc}$** and **OUR$_{cmb}$** outperform the **S-LSTM** model. While attending to the historical data in 8(f) we can see the behaviour of the pedestrian of interest under a similar context. Therefore the proposed models can anticipate that the preferred behaviour of the pedestrian of interest under such context is giving way to the others. But in the **S-LSTM** model as it is attending only to the immediate preceding hidden states and those long range dependencies are not captured.

In the example shown in Fig. 8(i)–(l) we illustrate the importance of considering the entire history of the neighbours. The way that the neighbours affect to a pedestrian during group motion vastly differs to the impact group motion has on a pedestrian walking alone. For example pedestrians moving as part of a group tend to walk at a similar velocity, keeping small distances between themselves, stopping or turing together; whereas pedestrians walking alone try to keep a safe distance between themselves and their neighbours to avoid collisions. These notions can be quickly captured while observing the neighbourhood history. Therefore the predictions generated by both **OUR$_{sc}$** and **OUR$_{cmb}$**

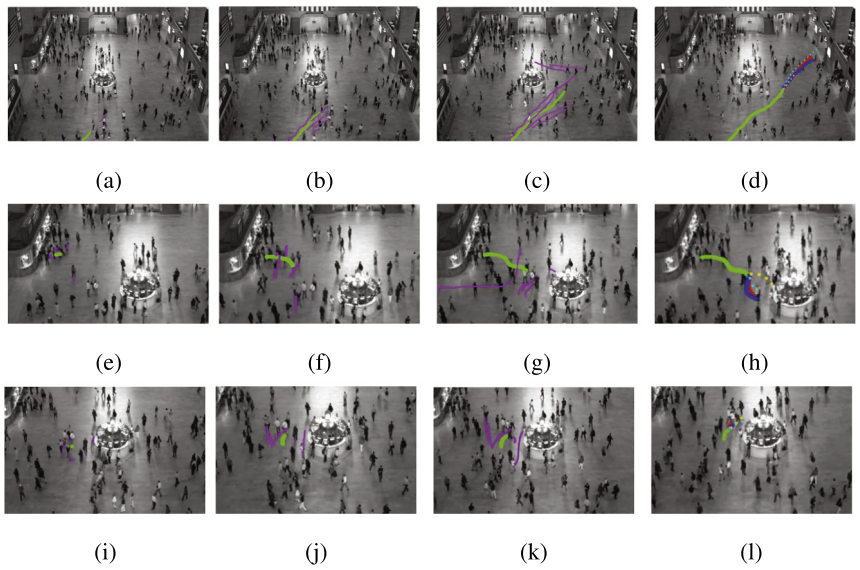


Fig. 8. Example scenarios from the GC dataset (Yi et al., 2015): Columns (left to right): first observation of the trajectory; half way through; last observation prior to prediction; prediction from the respective models. 1st row shows an example where the pedestrian exhibits 2 modes of motion (walking and running) within the same trajectory. 2nd row shows an example where the previous context of the pedestrian is useful in prediction. 3rd row shows an example where the pedestrian is moving as part of a group. Colours: Given trajectory (in green), Ground Truth (in Blue), Neighbouring (in purple) and Predicted trajectories from **OUR$_{cmb}$** model (in red), from **S-LSTM** model (in yellow), from **OUR$_{sc}$** model (in cyan). In order to preserve visibility without occlusions we have plotted only the closest 2 neighbours in each direction. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

models outperform the baseline **S-LSTM** model which only considered the immediate preceding hidden state of the neighbours when generating the predictions.

When comparing the predictions from **OUR$_{sc}$** model against **OUR$_{cmb}$** model it is evident that the more spatial context specific predictions are generated by **OUR$_{cmb}$** as it has been specifically trained on the examples from that particular spatial region. Therefore it anticipates the motion of a particular pedestrian more accurately. Still in all the example scenarios **OUR$_{sc}$** is shown to be capable of generating acceptable predictions compared to the baseline model, showing that the proposed combined attention mechanism is capable of generating more accurate and realistic trajectories than the current state-of-the-art.

5. Discussion

5.1. Hardware and implementation details

We evaluated the average time taken to complete a single epoch for the GC training set as well as the time taken to predict 1000 trajectories on the GC testing set, for the proposed method, its variants (i.e. only soft attention, etc.), and the baseline models. The evaluations along with the number of trainable parameters of each model are presented in Table 2. The implementation of the proposed method is completed using Keras (Chollet, 2017) with Theano (Bergstra et al., 2010) backend. The proposed module does not require any special hardware such as GPUs to run. The models are trained on a single core of an Intel Xenon E5-2680 v3 2.50 GHz CPU.

When analysing the results in Table 2 we observe a vast difference in training times between OUR$_{sft}$ and rest of the models. OUR$_{sft}$ requires the network to jointly back propagate through all the neighbours and learn the significance of each input trajectory. This is to be expected as OUR$_{sft}$ contains over twice the number of parameters as OUR$_{cmb}$. The OUR$_{hw}$ model is more light weight in training as the only learning components are the encoding and decoding LSTMs. The proposed OUR$_{cmb}$ model is slightly more time consuming compared to S-LSTM, OUR$_{pi}$ and OUR$_{hw}$ as it considers the entire history of the pedestrian of interest as well as the neighbours, however it shows a substantial reduction in training time compared to training everything with soft attention (i.e. OUR$_{sft}$). The evaluated test times show slightly variable test times for OUR$_{sft}$, OUR$_{hw}$, OUR$_{pi}$ and OUR$_{cmb}$ models. This is because, even though they have similar neural components, in the process of evaluating we pass the hidden state activations through an expectation calculation process whereas the hardwired weights are calculated using a distance function.

Table 2
Evaluation of runtimes (in seconds) with total number of trainable parameters: Train — Average time taken to complete a single training epoch. Test — Average time taken to predict 1000 trajectories.

Method	Elapsed time		Parameters
	Train	Test	
S-LSTM	73.8	11.01	253,557
OUR$_{pi}$	82.8	12.38	268,259
OUR$_{hw}$	70.2	13.11	278,210
OUR$_{sft}$	123.6	13.18	762,310
OUR$_{cmb}$	84.6	13.15	318,451

5.2. Effect of the neighbouring trajectories

We trained the *Our$_{cmb}$* model on the GC dataset while considering a varying number of neighbours to predict the trajectory of the pedestrian of interest. The change in ADE against the number of pedestrians is presented in Fig. 9. When selecting the neighbours we consider the distance between a specific trajectory and the pedestrian of interest and select the closest neighbours, as outlined in Section 3.2. The plot illustrates the importance of capturing context information from the neighbouring trajectories. The ADE is very high when we are not selecting any of the neighbours and gradually decreases as we incorporate information from neighbours who are close by. The error converges once we incorporate around 8 neighbours, and then starts to increase again as we keep adding more distant neighbours. We believe this is because this information added beyond 12–15 neighbours is redundant and affects the learning capacity of the model as each LSTM unit has a limited hidden state dimension which should store this encoded

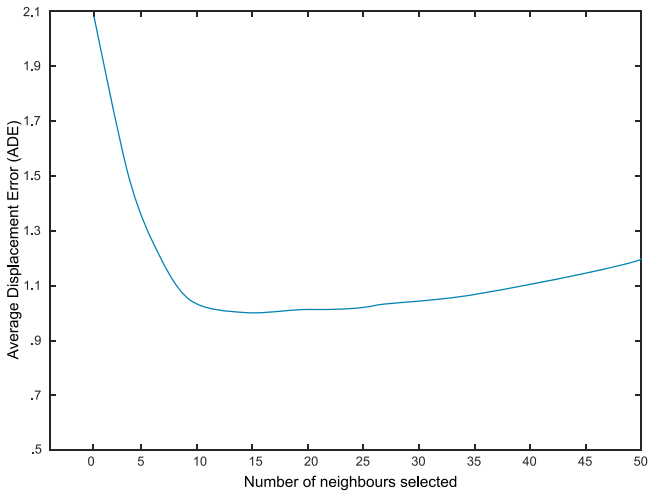


Fig. 9. Change in ADE against the number of neighbours selected.

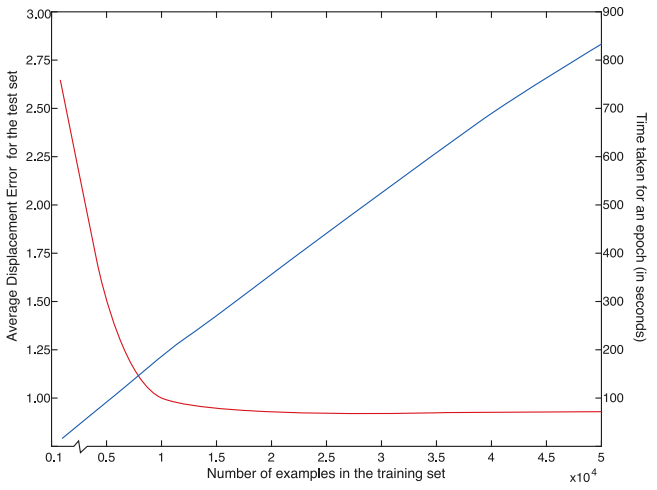


Fig. 10. Change in ADE and training time for an epoch, against training set size. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

information. This further supports our intuition that the neighbouring trajectories play a significant role when predicting the future trajectory of a particular pedestrian, and a good approximation of their influence is captured via the distance between them.

5.3. Effectiveness of soft + hardwired

When comparing Tables 1 and 2 the OUR_{sft} method is significantly more time consuming however there is only a slight improvement in accuracy. In contrast without such tedious back propagation needing to consider all the neighbouring trajectories in the neighbourhood, the proposed soft + hardwired method (i.e. OUR_{cmb}) has been able to generate accurate predictions with significantly greater time efficiency. The OUR_{hw} model is more time efficient however it leads to inaccurate predictions as the model looses the contextual information that can aid the future decision making due to this approximation. Similarly, the higher time efficiency of models OUR_{pi}, S-LSTM and SF are coupled with higher prediction error rates. The proposed soft + hardwired coupling mechanism, which is only slightly more time consuming, is able to generate predictions with much higher precision.

Furthermore when observing the qualitative results presented in Figs. 6 and 7 it is evident that the number of neighbours fluctuates. However the pedestrian models only accept fixed number of neighbours. Hence if we train the entire model with soft attention it is computationally inefficient as the model has to understand the presence of a dummy trajectory and learn through back propagation to set its weight to zero. Hardwired weights provide a more efficient, yet effective way to solve this issue via estimating the effect of neighbours through a simple distance measure.

5.4. Impact of the training set size

In order to analyse the robustness of the proposed model against different training set sizes we analyse the distribution of the Average Displacement Error (ADE) and training time for an epoch, for different training set sizes for the EIF dataset. For this evaluation we used the same testing set used in Section 4.1, and it is not used for training.

In Fig. 10 we visualise ADE against different training set sizes (in red) and the elapsed time per epoch (in blue). As could be expected, the training time increases gradually as more samples are added to the corpus. However the model accuracy converges around 10,000 training examples and we do not observe substantial improvement, irrespective of the introduction of additional examples. This highlights the light weight capacity of the proposed model and its ability to anticipate the repetitive nature of the human motion.

6. Abnormal behaviour detection

The proposed framework can be directly applied for detecting abnormal pedestrian behaviour. A naive approach would be to predict the trajectory for the period of T_{obs+1} to T_{pred} while observing the same trajectory over this time period and measuring the deviation between the observed and the predicted trajectories. If the deviation is greater than a threshold, then an abnormality can be said to have occurred. However due to the adaptive nature of deep neural networks, abnormal behaviours such as: (i) sudden turns and changes in walking directions; and (ii) trajectories with abnormal velocities; may not be classified as abnormal events.

We observe that the hidden states of the LSTM encoder decoder framework hold vital information which is used to model the walking behaviour of the pedestrian of interest. Hence, if his or her behaviour is abnormal then the hidden state values for that pedestrian should be distinct from those of a normal pedestrian.

With that intuition we randomly selected 500 trajectories from the Grand Central dataset and predicted the trajectories for those pedestrians. The trajectories were hand labelled for abnormal behaviour, considering sudden turns and changes in walking direction and abnormal velocities as the set of abnormal behaviours. The dataset consists of 445 normal trajectories and 55 abnormal trajectories. Then we extracted the encoded hidden states ($h_{(t)} = [h_{(1)}, \ldots, h_{(T_{obs})}]$) for the given trajectory for that pedestrian and the hidden states used for decoding ($s_{(t)} = [s_{(T_{obs+1})}, \ldots, s_{(T_{pred})}]$). The resultant hidden states are passed through DBSCAN to detect outliers. With the proposed approach we detected 441 trajectories as being normal and 59 trajectories as abnormal. The resultant detections are given in Table 3.

Analysing the classification results, we see that false alarms are mainly due to behaviours that are erroneously detected as abnormal being uncommon in the database. Cases such as people changing direction to buy tickets, and passengers wandering in the free area are detected as abnormal due to the fact that they are not significantly present in the subset of trajectories selected for this task.

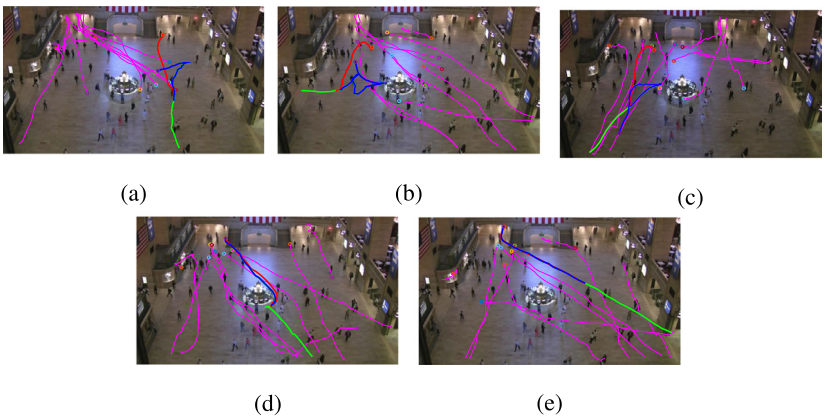


Fig. 11. Abnormal event detections on the GC dataset (Yi et al., 2015): (a)–(c) abnormal behaviour detected due to sudden change of moving direction. Abnormal behaviour due to sudden circular turn (d) and abnormal velocity in (e).

Table 3
Abnormal event detection with the proposed algorithm: This approach has detected 47 out of 55 ground truth abnormal events.

		Ground truths		Total
		Abnormal	Normal	
Predicted	Abnormal	**47**	12	**59**
	Normal	8	**433**	**441**
	Total	**55**	**445**	**500**

Table 4
Abnormal event detection with naive approach: This approach has detected only 29 out of 55 ground truth abnormal events.

		Ground truths		Total
		Abnormal	Normal	
Predicted	Abnormal	**29**	24	**53**
	Normal	26	**421**	**447**
	Total	**55**	**445**	**500**

We compare this approach to the naive approach given above. It is evident that some abnormal trajectories are misclassified as normal behaviour due to its lack of deviation from the observed trajectory. See Table 4.

Some examples of detected abnormal events are shown in Fig. 11. The first row shows the abnormal behaviour detected due to sudden change of moving direction. Even though, in the examples shown (d) and (e), there is not a significant deviation between the predicted path and the observed path, our abnormal event detection approach has accurately classified the event due to the sudden circular turn in the trajectory in (d) and abnormal velocity in (e).

7. Conclusion

In this paper we have proposed a novel neural attention based framework to model pedestrian flow in a surveillance setting. We extend the classical encoder–decoder framework in sequence to sequence modelling to incorporate both soft attention as well as hard-wired attention. This has a major positive impact when handling longer trajectories in heavily cluttered neighbourhoods. The hand-crafted hard-wired attention weights approximate the neighbour's influence and make the application of attention models pursuable for real world scenarios with large number of neighbours. We tested our proposed model in two challenging publicly available surveillance datasets and demonstrated state-of-the-art performance. Our new neural attention framework exhibited a stronger ability to accurately predict pedestrian motion, even in the presence of multiple source and sink positions and with high crowd densities observed. Furthermore, we have shown how the proposed approach can support abnormal event detection through hidden state clustering. This approach is able to accurately detect events in challenging situations, without handcrafting the features. Apart from direct applications such as abnormal behaviour detection, improving passenger flow in transport environments, this framework can be extended to any application domain where modelling multiple co-occurring trajectories is necessary. Some potential areas include modelling aircraft movements, ship trajectories and vehicle traffic.

Acknowledgment

This research was supported by the Australian Research Council's Linkage Project LP140100282 "Improving Productivity and Efficiency of Australian Airports". The authors also thank QUT High Performance Computing (HPC) for providing the computational resources for this research.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*.
Alahi, A., Ramanathan, V., & Li, F. F. (2014). Socially-aware large-scale crowd forecasting. In *CVPR* (pp. 2211–2218).
Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011). Sequential deep learning for human action recognition. In *HBU* (pp. 29–39). . http://dx.doi.org/10.1007/978-3-642-25446-8_4.
Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
Bartoli, F., Lisanti, G., Ballan, L., & Del Bimbo, A. (2017). Context-aware trajectory prediction. arXiv preprint arXiv:1705.02503.
Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., et al. (2010). Theano: A cpu and gpu math compiler in python. In *Proceedings of 9th python in science conference* (pp. 1–7).
Chollet, F. (2017). Keras. URL http://keras.io,2017.
Emonet, R., Varadarajan, J., & Odobez, J. M. (2011). Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model. In *CVPR* (pp. 3233–3240).
Ester, M., peter Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise (pp. 226–231).
Giannotti, F., Nanni, M., Pinelli, F., & Pedreschi, D. (2007). Trajectory pattern mining. In *ACM SIGKDD* (pp. 330–339). . http://dx.doi.org/10.1145/1281192.1281230.
Goroshin, R., Bruna, J., Tompson, J., Eigen, D., & LeCun, Y. (2015). Unsupervised feature learning from temporal data. arXiv preprint arXiv:1504.02518.
Han, J., Cheng, G., Li, Z., & Zhang, D. (2017). A unified metric learning-based framework for co-saliency detection. *IEEE Transactions on Circuits and Systems for Video Technology*.
Helbing, D., & Molnár, P. (1995). Social force model for pedestrian dynamics. *Physical Review E*, *51*, 4282–4286. http://dx.doi.org/10.1103/PhysRevE.51.4282.
Hospedales, T. M., Gong, S., & Xiang, T. (2009). A markov clustering topic model for mining behaviour in video. In *ICCV* (pp. 1165–1172).

Jia, X., Gavves, E., Fernando, B., & Tuytelaars, T. (2015). Guiding the long-short term memory model for image caption generation. In *Proceedings of the IEEE international conference on computer vision* (pp. 2407–2415).

Koppula, H. S., & Saxena, A. (2013). Anticipating human activities using object affordances for reactive robotic response. In *IROS* (p. 2071).

Lai, K., Bo, L., & Fox, D. (2014). Unsupervised feature learning for 3d scene labeling. In *ICRA* (pp. 3050–3057). . http://dx.doi.org/10.1109/ICRA.2014.6907298.

Lee, J. G., Han, J., & Whang, K. Y. (2007). Trajectory clustering: A partition-and-group framework. In *ACM SIGMOD* (pp. 593–604). . http://dx.doi.org/10.1145/1247480.1247546.

Madry, M., Bo, L., Kragic, D., & Fox, D. (2014). ST-HMP: unsupervised spatio-temporal feature learning for tactile data. In *ICRA* (pp. 2262–2269). . http://dx.doi.org/10.1109/ICRA.2014.6907172.

Majecka, B. (2009). *Statistical models of pedestrian behaviour in the forum* (MSc Dissertation), School of Informatics, University of Edinburgh.

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in neural information processing systems* (pp. 2204–2212).

Morris, B., & Trivedi, M. M. (2009). Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *CVPR* (pp. 312–319).

Pellegrini, S., Ess, A., & Gool, L. J. V. (2010). Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV* (pp. 452–465).

Sharma, S., Kiros, R., & Salakhutdinov, R. (2015). Action recognition using visual attention. arXiv preprint arXiv:1511.04119.

Varshneya, D., & Srinivasaraghavan, G. (2017). Human trajectory prediction using spatially aware deep attention models. arXiv preprint arXiv:1705.09436.

Wang, X., Ma, K. T., Ng, G. W., & Grimson, W. E. L. (2008). Trajectory analysis and semantic region modeling using a nonparametric bayesian model. In *CVPR* (pp. 1–8).

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*, 229–256.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML, Vol. 14* (pp. 77–81).

Xu, J., Denman, S., Fookes, C. B., & Sridharan, S. (2012). Unusual scene detection using distributed behaviour model and sparse representation. *AVSS*, 48–53.

Yamaguchi, K., Berg, A. C., Ortiz, L. E., & Berg, T. L. (2011). Who are you with and where are you going? In *CVPR* (pp. 1345–1352).

Yao, X., Han, J., Zhang, D., & Nie, F. (2017). Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering. *IEEE Transactions on Image Processing*, *26*(7), 3196–3209.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., et al. (2015). Describing videos by exploiting temporal structure. In *ICCV* (pp. 4507–4515).

Yi, S., Li, H., & Wang, X. (2015). Understanding pedestrian behaviors from stationary crowd groups. In *CVPR* (pp. 3488–3496).

Yoo, D., Park, S., Lee, J. Y., Paek, A. S., & Kweon, I. S. (2015). Attentionnet: Aggregating weak directions for accurate object detection. In *ICCV* (pp. 2659–2667).

Zhang, D., Han, J., Jiang, L., Ye, S., & Chang, X. (2017). Revealing event saliency in unconstrained video collection. *IEEE Transactions on Image Processing*, *26*(4), 1746–1758.

Zhou, B., Tang, X., & Wang, X. (2015). Learning collective crowd behaviors with dynamic pedestrian-agents. *Journal of Computer Vision*, *111*, 50–68.

Zou, H., Su, H., Song, S., & Zhu, J. (2018). Understanding human behaviors in crowds by imitating the decision-making process. arXiv preprint arXiv:1801.08391.