Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 63 (2020) 30–41

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

ELM-HTM guided bio-inspired unsupervised learning for anomalous trajectory classification

Arif Ahmed Sekh [a,*], Debi Prosad Dogra [b], Samarjit Kar [c], Partha Pratim Roy [d], Dilip K. Prasad [e]

[a] *Department of Physics and Technology, UiT The Arctic University of Norway, Tromsø 9019, Norway*
[b] *School of Electrical Science, Indian Institute of Technology Bhubaneswar, Bhubaneswar 751013, India*
[c] *Department of Mathematics, National Institute of Technology Durgapur, Durgapur 713209, India*
[d] *Department of Computer Science, Indian Institute of Technology, Roorkee, Uttarakhand 247667, India*
[e] *Department of Computer Science, UiT The Arctic University of Norway, Tromsø 9019, Norway*

Received 11 June 2019; received in revised form 10 November 2019; accepted 26 April 2020
Available online 23 May 2020

Abstract

Artificial intelligent systems often model the solutions of typical machine learning problems, inspired by biological processes, because of the biological system is faster and much adaptive than deep learning. The utility of bio-inspired learning methods lie in its ability to discover unknown patterns, and its less dependence on mathematical modeling or exhaustive training. In this paper, we propose a new bio-inspired learning model for a single-class classifier to detect abnormality in video object trajectories. The method uses a simple but dynamic extreme learning machine (ELM) and hierarchical temporal memory (HTM) together referred to as ELM-HTM in an unsupervised way to learn and classify time series patterns. The method has been tested on trajectory sequences in traffic surveillance to find abnormal behaviors such as high-speed, unusual stops, driving in wrong directions, loitering, etc. Experiments have also been performed with 3D air signatures captured using sensors and used for biometric authentication(forged/genuine). The results indicate a significant gain over training time and classification accuracy. The proposed method outperforms in predicting long-time patterns by observing small steps with an average accuracy gain of 15% as compared to the state-of-the-art HTM. The method has applications in detecting abnormal activities in videos by learning the movement patterns as well as in biometric authentication.

Keywords: Trajectory analysis; Anomaly detection; ELM; HTM; Bio-inspired learning

1. Introduction

Time series data is one of the important sources of information used in various pattern understanding tasks. Trajectories as a sequence of data (Ahmed, Dogra, Kar, & Roy, 2018b) have been used in various tasks including but not limited to visual surveillance (Yi, Li, & Wang, 2016), traffic monitoring (Ahmed, Dogra, Kar, & Roy, 2018a), 3D signature analysis (Behera, Dogra, & Roy, 2018), etc. Learning through observation is the primary learning process adopted by human brain (Deng et al., 2015; Hawkins & Blakeslee, 2007). Human brain uses cognitive learning in various visual event identification, such as abnormal traffic movement detection, sign language recognition or air-writing understanding. In this paper, we

* Corresponding author.
E-mail addresses: arif.ahmed.sekh@uit.no (A.A. Sekh), dpdogra@iitbbs.ac.in (D.P. Dogra), samarjit.kar@maths.nitdgp.ac.in (S. Kar), proy.fcs@iitr.ac.in (P.P. Roy).

https://doi.org/10.1016/j.cogsys.2020.04.003
1389-0417/© 2020 The Author(s). Published by Elsevier B.V.

demonstrate the usability of learning from unlabeled data applicable to trajectory anomaly detection. We have introduced a hierarchical and feedback-based learning algorithm inspired from learning of human brain. The proposed method uses hierarchical temporal memory (HTM) (Edwards et al., 2017; Fan, Sharad, Sengupta, & Roy, 2016) to learn the normality model from unlabeled data. Next, the model has been used to learn a single class classifier using extreme learning machine (ELM) to find abnormalities in time series. The method has been tested on two applications, (i) finding surveillance abnormalities from moving objects trajectories (ii) air signatures acquired for biometric authentication, where the low-level movement patterns are complex. Fig. 1 depicts the overall framework of the proposed method. The framework consists of 4 components. (1) A set of unlabeled trajectories are extracted and used for training, (2) Trajectories are encoded using SDR unit, (3) An HTM module and (4) An ELM module are combined using feedback to classify and estimate normality score.

1.1. Motivation and contributions

Since the emergence of artificial intelligence, researchers are trying to link it with bio-inspired systems for solving various computer vision and machine learning problems. Despite striking similarities between artificial intelligence and biological brain, deep understanding of the human visual system applied in pattern understanding is still far from the perfection. The main success of bio-inspired learning methods is the ability of discovering unknown patterns (Cui, Ahmad, & Hawkins, 2017). State-of-the-art neural networks (NN)-based learning architectures rely on mathematical modeling and expensive training. Such systems often demand an entirely new set of training data when newer patterns are discovered.

In this paper, we have made the following contributions: (i) We have proposed a new bio-inspired online-learning model for a single-class classifier to detect abnormality in time series data. (ii) The proposed method fuses two state-of-the-art bio-inspired learning methods, namely ELM and HTM using feedbacks, where HTM learns the low-level pattern similarity and ELM learns the high-level features. (iii) It has been tested on video object trajectories to find abnormal patterns. The method has also been applied on 3D air signatures used in biometric applications.

Rest of the paper is organized as follows. In Section 2, we have discussed the proposed ELM-HTM method for classifying normality of trajectory, including overview of the HTM and ELM methods and ELM-HTM fusion technique. In Section 3, we present the results using traffic junction videos, and 3D air signature trajectories. Finally, in Section 4, we conclude our paper by highlighting some key future extensions of the present work.

1.2. Related work and background

Learning, predicting, and classifying complex temporal pattern is challenging due to several reasons such as complex structure (Lee et al., 2017), large amount low-level pattern variations (Cui, Surpur, Ahmad, & Hawkins, 2016), dynamic in nature (Alahi et al., 2016), expensive training dependent (Donahue et al., 2015), etc. Firstly, the real-world sequence data often have changing statistics and required online learning capabilities to deal with the changes of patterns in the continuous time domain. Secondly, sequence learning needs an automatic prediction algorithm to deal with accurate prediction. Thirdly, sequence data are often mixed with noise. Lastly, most of the machine learning algorithms typically tuned to a set of task-specific hyperparameters. However, good sequence learning algorithms demand small number of

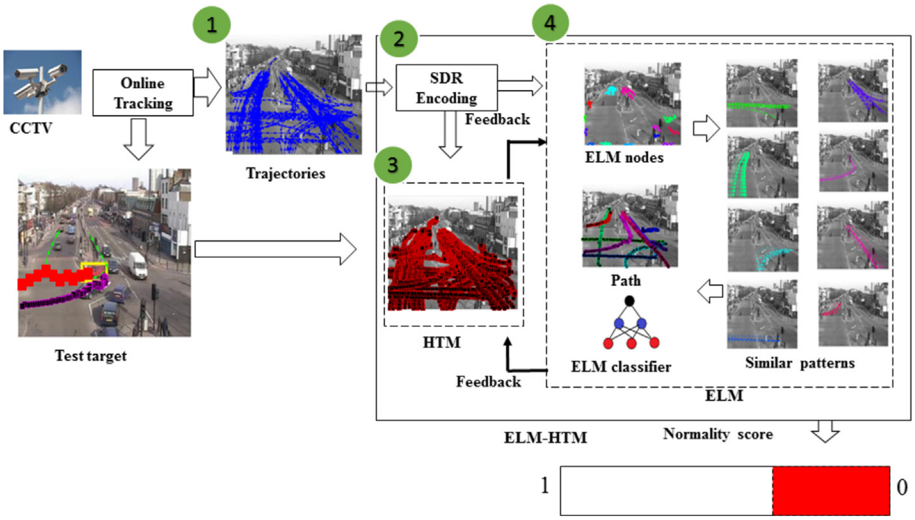


Fig. 1. Flow and key points of the proposed method. Steps are marked in green circle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

hyperparameters or sometimes no hyperparameter to be tuned for a wide range of applications. A number of neural networks-based learning architectures have been proposed to deal with the sequence learning problems (Längkvist, Karlsson, & Loutfi, 2014). Time delay neural network (TDNN) (Meng, Bianchi-Berthouze, Deng, Cheng, & Cosmas, 2016) is an input delay-based neural network. Long short term neural network (LSTM) (Alahi et al., 2016; Sutskever, Vinyals, & Le, 2014) is used in many applications to learn and predict abnormality based on recurrent neural network (RNN). Unsupervised methods using unlabeled data relay on probability of events and clustering methods. Rodríguez-Serrano Singh (2012) have proposed a probability-based hidden Markov model, where each state is weighted using the probabilistic weight and a lower probability represents higher abnormality. Campo, Baydoun, Marcenaro, Cavallaro, and Regazzoni (2018) have proposed a self organizing map to construct different cluster of patterns in an unsupervised way. Xu, Zhou, Lin, and Zha (2015) have proposed shrinkage-based unsupervised clustering method. The low frequent clusters are considered to be abnormal. Such learning methods can be used in sequence learning applications. Recently a bio inspired learning method that uses cognitive learning referred to as HTM, has been proposed by Cui et al. (2017). The method uses similar pyramidal cell structures found in *neocortex* layers and it has applied in various pattern anomaly detections (Cui, Ahmad, & Hawkins, 2016; Wu, Zeng, & Yan, 2018). HTM found to be a good solution in low-level prediction and classification tasks, especially when the data are unlabeled (Ahmad, Lavin, Purdy, & Agha, 2017), it is observed that the method is sensitive to the local patterns. Similar tasks have been solved using extreme learning machine (ELM) approaches (Huang, Zhou, Ding, & Zhang, 2012; PPark & Kimark & Kim, 2017), where the pattern is represented using high-level concept such as nodes. The primary advantage of ELM is its simple architecture (a single hidden layer model). It requires less data and consumes less time to train as compared to conventional deep learning architectures (LeCun, Bengio, & Hinton, 2015). The advantages of the HTM (Hawkins & George, 2016) method is the similarity of the method with human brain model, which is fast and adaptive. HTM focused on the local patterns and suitable for anomaly detection. On the other side the ELM can be used for classifying patterns represented by the high-level features called hidden nodes.

Preliminary of ELM and HTM Theory: Extreme learning machines (ELM) or online sequential extreme learning machines (OS-ELM) (Tang, Deng, & Huang, 2016) are trained using a single-hidden layer flashforward network. It has been reported that universal approximation and classification capabilities of ELM provides good generalization in various real world problems (G.-B. Huang & Chen, 2008; G.B. Huang, Chen, & Siew, 2006). ELM uses three-layered architecture: input, hidden, and output layers. The bias and input weights are randomly generated and fixed during the entire learning process. A typical single hidden layer-based ELM model with L number of hidden nodes consists of the output weights (β), and $G(a,b,x)$ as a sigmoid function for each node. The method minimizes the cost function given in (1), where H is the hidden layer output matrix and $\mathbb{T}$ is the training matrix. The main drawback of an ELM is the random weight assignment during learning process. To overcome the limitation, we have used restricted Boltzmann machine (RBM) (Pacheco, Krohling, & da Silva, 2018) to extract the statistical weights of the nodes by probability distributions. Fig. 2(a) shows a typical ELM network.

$$min\|H\beta - \mathbb{T}\|^2 \tag{1}$$

Hierarchical temporal memory (HTM) (Cui et al., 2017; Edwards et al., 2017) is considered as one of the highly popular neuroscience inspired machine learning method. Its primary advantages are (i) it can be trained by unlabeled data (ii) it can efficiently discover spatial and temporal patterns (iii) it is online and can be trained in real-time and (iv) it has higher noise tolerance. The structure of a typical HTM-based systems is presented in Fig. 2(b).

HTM networks are fed with the sequences represented by sparse distributed representations (SDR). The method is similar to neural functionality of human brain. Each activity/pattern is represented by sparse collection of active cells. For example, a pattern of size 15 can be 000100001110000, where 1 represents active and 0 is inactive. Typical, HTM models learn spatial patterns as well as the transition between pattern in temporal domain. The SDR coefficients are learned online. The trained neuron set is represented by a matrix known as mini-column. A typical SDR has been used with HTM spatial pooling (SP) (Cui et al., 2017) to reduce the size of a pattern representation to produce high-level patterns. A typical HTM network is represented by active/inactive binary matrix. A pattern similarity is measured from the similarity in SDR representation of the patterns. It is measured using the overlap bit of the SDR. For example, Fig. 3 presents two sequences of size 25 with an overlap in 5 bits. The overlap is calculated using the dot (.) product. The sequences reconsidered similar if the overlap bit position is less than the minimum overlap bit (θ). Hence the method is sensitive to θ for identifying the similar patterns from unlabeled data.

HTM can learn such pattern similarity from online streaming of data and can deal with temporal patterns. The main drawback of HTM learning is, the system is highly sensitive to the overlap parameter (θ). A higher or lower value may affect the classification accuracy. To deal with this problem, we have taken θ from the high-level learning using ELM. Initially, we group similar patterns using ELM and extract θ by taking the maximum overlap bit. More about HTM learning process can be found in (Hawkins & Ahmad, 2016; Wu et al., 2018).

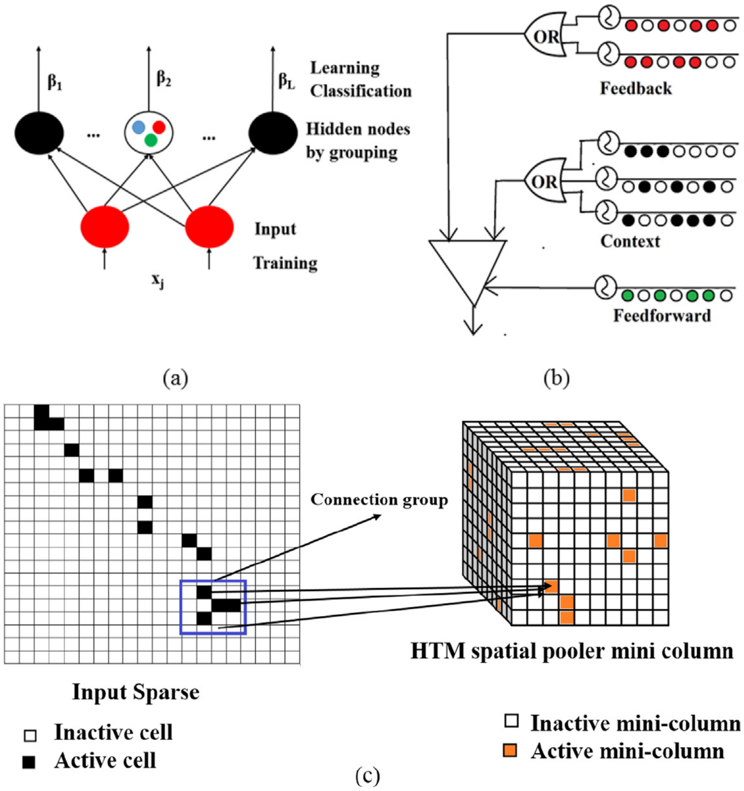


Fig. 2. (a) Typical structure of HTM network. HTM is bio-inspired method consists of local context, feedback and flashforward. The method is similar structure and decision making with human neuron (b) A typical ELM is a single-layered neural network. ELM uses a single hidden layer for learning (c) HTM spatial pooling (SP) layer converts the input pattern to a spatio-temporal minicolumns, the activated cells column are represented by the filled color. This mechanism represents the input patterns into a spatio-temporal patterns with reduced data (pooling). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

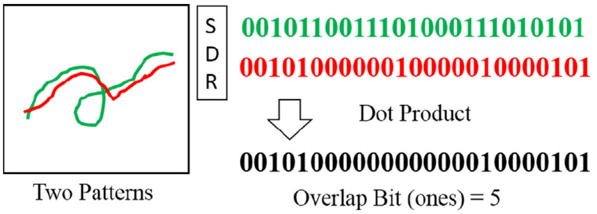


Fig. 3. Example of two patterns and overlap bit.

2. Proposed methodology

In this section, we present an unsupervised learning method that is based on a single layer extreme learning machine and hierarchical temporal memory (ELM-HTM). The method has been used to model a single class classifier. It can learn normality characteristics from unlabeled data and produce normality scores for the test data.

2.1. Trajectory representation and encoding

A trajectory is defined by the spatio-temporal positions of targets say car, pedestrian, fingertip, etc. A trajectory can be formally defined using (2), where $p_i(x_i, y_i, t_i)$ represents the instantaneous position of an object at time t_i in 2D. In 3D (e.g. when it represents the fingertip positions during air signature (Behera et al., 2018)), it can additionally hold the depth information, thus making it a four tuple, $p_i(x_i, y_i, z_i, t_i)$. Trajectories can be obtained by tracking targets using multi-object tracking in case of video applications and sensors can be used to track finger movements during air signatures. Though the low-level information of trajectory have already been used in various machine learning algorithms, however, due to unavailability of labelled data is a real challenge for the research community. Therefore, designing high-level features to represent motion patterns that can be used for classification, has been taken as a research challenge. In the next section, we describe how sparse distributed representation (SDR) can be successfully used to extract meaningful features from the trajectory. These features are then used to classify trajectories using the proposed ELM-HTM guided bio-inspired unsupervised single class classifier to understand abnormalities.

$$T = \{p_1, p_2, p_3, \ldots, p_n\} \tag{2}$$

2.2. Learning with unlabeled data

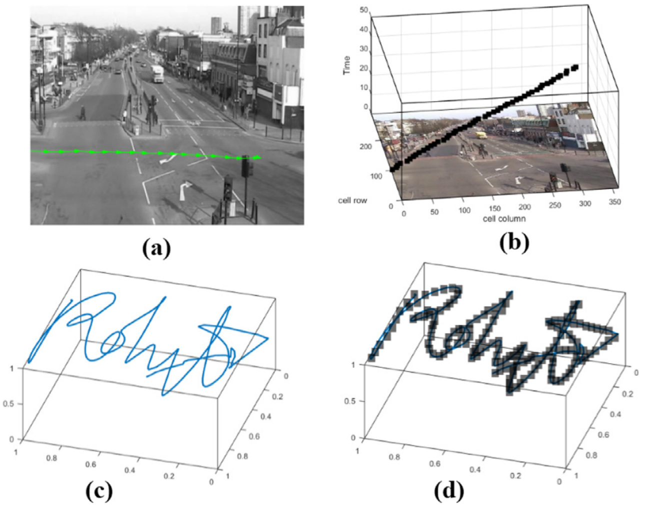
Applications such as computer vision aided traffic surveillance, GPS-guided object tracking or, sensor-guided air writing demand scalable solution that can learn the dynamic nature of the patterns present in the past observations. However, the challenges in designing an acceptable learning method can be broadly categorized into (i) unavailability of sufficient labeled trajectories (ii) the method should be online (iii) the learning method should deal with the dynamic nature of the pattern considering temporal sequence of events (iv) the method should learn using small amount of training data and (v) the training time should be minimum. To design such a scalable system, we have designed a new framework ELM-HTM by fusion.

2.3. ELM-HTM guided trajectory classifier

In this section, we have discussed the proposed ELM-HTM learning algorithm. First, the trajectories are represented using SDR. Next, a HTM spatial pooler (SP) is applied to reduced the complexity of the trajectory. In each temporal position, a single cell is activated with respect to t and the trajectory is represented by a set of active cells. The binary matrix is generated by replacing active cells by $1's$ and inactive cells by $0's$. Fig. 4 shows typical moving object trajectories used in visual surveillance 3D air signature trajectory analysis with the help of SDR.

The proposed method fuses ELM and HTM model together, where HTM has been used to learn low-level noisy patterns (bits) and ELM has been used to learn high-level features (region-to-region patterns). The ELM module is a single hidden-layered architecture as described earlier. Fig. 5 depicts the flow of the proposed method.

The main designing challenges of such system is that the normal patterns in a surveillance are not fixed. Therefore, the normality can vary time to time. To begin with, we first convert the trajectories into SDR and passes them through the ELM-HTM model. HTM is used to predict sequence by observing small number of steps. A given input sequence x_t is converted in SDR as $a(x_t)$. HTM predicts the sequence as $\sigma(x_{t-1})$. The predicted sequence is highly dependent on the overlap and match ratio (θ). We have calculated θ using the feedbacks received from ELM module. In ELM module, θ is calculated from the average match and overlap within the group of similar patterns it belongs. The prediction error is represent in (3), where the error (E_t) is the scaler normalization of $a(x_t)$. The model changes the underlying statistics automatically by online learning. E_t is inversely proportional to the count of the common bit patterns. It becomes 0 when the prediction is correct. In case of traffic monitoring, an abnormal situation can be treated as an undiscovered pattern of movement such as loitering or illegal u-turns in highway traffic.



(a) (b)

(c) (d)

Fig. 4. (a) A typical object trajectory extracted by visual tracking. (b) SDR representation of the trajectories, black cell represents 1 and others are 0. (c) 3D air signature trajectory extracted by tracking fingertip (d) SDR representation of the signature, black cell represents 1 and others are 0.

$$E_t = 1 - \frac{\sigma(x_{t-1}).a(x_t)}{|a(x_t)|} \tag{3}$$

HTM can identify the potential outliers and normal patterns. A range of similarly looking abnormal patterns may be present in a normal class. Now the question is: *How much normal these patterns are?* (Albusac, Vallejo, Castro-Schez, Glez-Morcillo, & Jiménez, 2014; Mabrouk & Zagrouba, 2017). We assume that object trajectories in surveillance videos typically demonstrate region-to-region movements of the objects. A region-to-region path can be considered as high-level information that can be used to understand normality of the trajectory. The normality concept can be used to find abnormalities in normal patterns such as infrequent U-turn, over-speeding vehicles, vehicles moving in wrong directions, unusual stops, loitering, etc. Lower the normality, higher the chance of abnormality. Once the outliers are extracted‘ using HTM, we have applied ELM to discover normality index of a test pattern. However, deciding the number of hidden layers in ELM network is challenging. The number of hidden layers should be chosen based on the variation of patterns present in the data. A higher number of nodes for a simple scenario with a small variation of patterns may overfit the model. A smaller number of nodes for a complex dataset may not be sufficient. The method is described hereafter.

The process is initiated by representing each trajectory by their origin and terminal cells as described in (4).

$$T = \{C_{start}, C_{end}\} \tag{4}$$

These cells are then incrementally grouped using density-based clustering algorithm known as DBSCAN (Ester, Kriegel, Sander, & Xu, 1996). It is an unsupervised clustering algorithm regulated by the maximum distance from the neighbourhood (ϵ). Each group of cells/regions is then represented as a neuron segment. The ELM-HTM model is then dynamically constructed and it is modified from the online feedback. Number of segments detected after density based clustering provide us the clue to decide the

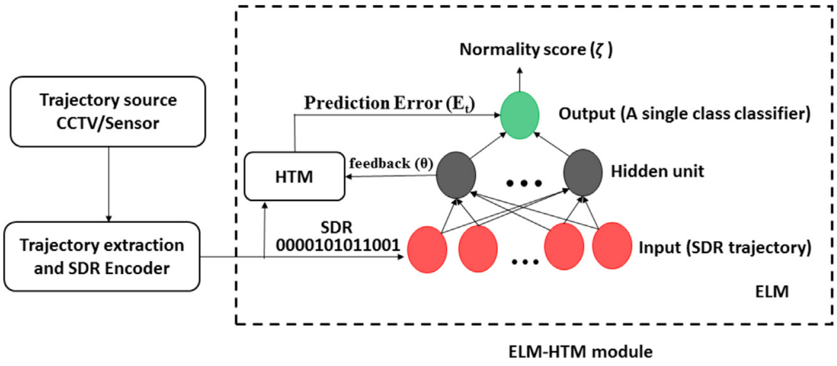


Fig. 5. The working nature of the ELM-HTM method. First, the trajectories are extracted by object/fingertip tracking and converted in SDR. The low-level patterns have been learned using HTM and the probabilistic score of region-to-region movement patterns are learned using ELM. HTM uses the feedback received from ELM (θ) to calculate prediction error (E_i). The HTM-ELM classifier fuse these score to learn normality and classify abnormalities.

number of hidden layers. Since the nodes of the input layer of the ELM are fully connected with the hidden layer, it is therefore a meaningful guess to use number of segments as the number of hidden layers. This ensures that any trajectory represented using SDR can ideally be checked against all possible region-to-region movements. Next, region-to-region movements of the objects are expressed as paths using the activation cells of the SDR. The hidden layers in the ELM architecture encapsulate individual probability as well as inter-regions transition probabilities. We have used a global averaging method to extract average path from the training samples and a distance score of normality during the classification.

Restricted Boltzmann machine (RBM) (Pacheco et al., 2018) has been used to generate the weights of the hidden layers. It is realistic to assume that infrequent paths are lesser probable to be normal. In Algorithm 1, we present the method to obtain various parameters discussed earlier to learn normality using the proposed ELM-HTM framework.

Algorithm 1. ELM-HTM learning

```
Require:
  Training data $\{x_i, t_i\}$ with $N$ samples
  Maximum threshold for DBSCAN $(\epsilon), minPts$
Ensure: $\{x_i, t_i\}$ are unlabeled
  Learn HTM module
  Number of hidden node of the ELM $(\kappa)$ = Number
    of cluster obtained $DBSCAN(\{x_i, t_i\}, \kappa, minPts)$
  weight of the $i^{th}$ node $(\beta_i) = p(\kappa|\{x_i, t_i\})$, where
    $p(\kappa|\{x_i, t_i\})$ is calculated using (Pacheco et al., 2018)
  Extract average paths $(\{\eta_i\})$ from $\kappa_i$ to
    $\kappa_j = DBA(\{x_i, t_i\}), \{x_i, t_i\} \in \kappa_i, \kappa_j$
  Calculate $\theta_i = \frac{1}{n}\sum_{i=1}^{n} match(x_i), x_i \in \kappa_i$
  return $\kappa, \beta, \{\eta_i\}, \{\theta_i\}$
```

1-Class Classifier: The topmost layer of the ELM architecture is a softmax layer and it is used to identify the normality index of a given pattern or trajectory represented in SDR encoding. During learning, the layer estimates the average paths (η) and stores as path model. We have used DTW Barycenter Averaging (DBA) (Petitjean, Ketterlin, & Gançarski, 2011) to obtain the average path that is needed in the final stage of classification. DBA is a global averaging method that iteratively performs the refining and minimization operations of the distance using dynamic time warping (DTW). The output of the layer is a fuzzy variable (0 to 1), where 1 represents absolutely normal and 0 represents possibly abnormal conditions. This layer combines (i) the output from the hidden layers of the ELM to understand the normality as region-to-region pattern (ii) the HTM prediction error (E_t) to understand low-level pattern similarity and (iii) pattern distance/path deviation (ϕ) from the path model to take the final decision. ϕ is calculated by taking minimum of each Hausdorff distance from the average path. Fig. 6 depicts the concept of average path and deviation.

The normality distance (ζ) is extracted by the classification algorithm defined in Algorithm 2, where H_d is the Hausdorff distance and E_i is the prediction error feedback received from HTM module. Higher the distance, lower the chance of normality. The score is normalized between 0 and 1 using the distribution of ζ and E_i during learning. Fig. 7 shows an example of the learning results in 10-min QMUL dataset video and the constructed ELM. Two potential regions (blue and red) represent two hidden nodes in the ELM, where, (a) 5 min training video from QMUL dataset video, 37 targets are tracked and a set of trajectories ($\{T\}$) are extracted. (b) SDR encoded, where the trajectories are represented by active or 1 and inactive or 0. The black boxes represent active cells (1) obtained during training, (c) representation of the patterns by initial and final cells, (d) region segmentation using DBSCAN clustering, where each color represents a different region in the scene, and number of such regions (κ) is the number of class extracted by DBSCAN, and (e) constructed ELM of the scene. Here, the number of hidden nodes is equals to κ. We have found two such nodes (red and blue) in this case. (f) DBA-based average path (η) repressed using SDR, black box represents 1.

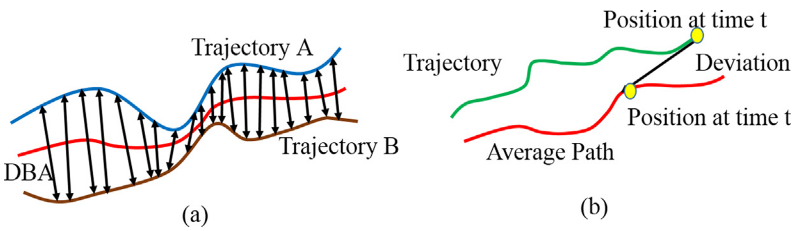


Fig. 6. The concept of average path and path deviation used in classification. The black arrows represent the positions at same time intervals.

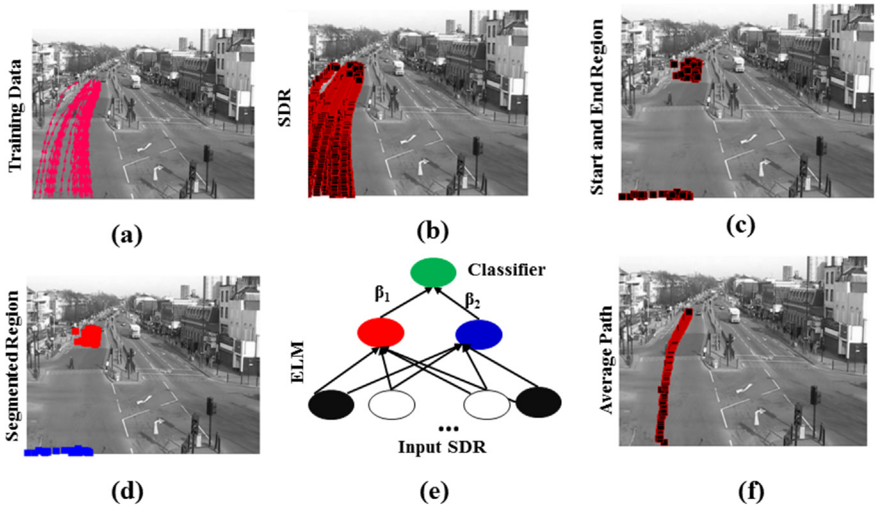


Fig. 7. The figure demonstrates the ELM learning method.

Algorithm 2. ELM-HTM classifier

```
Require:
  Test trajectory $T_i = \{x_i, t_i\}$ with $X$ samples
Ensure: $\{x_i, t_i\}$ may complete or incomplete
  $E_t$ = prediction error from HTM
  $\alpha$ = ELM layer output score
  if $E_t$ is outlier OR $\alpha$ is outlier then
      Activate alarm as abnormal
  else
      $T_p$ is complete trajectory of $T_i$ predicted by HTM
      $\phi = \min(H_d(T_p, \eta_i)), \phi$ is normality distance from path
      $\zeta = E_t \odot \alpha \odot \phi, \zeta$ is the final normality score, $\odot$ is normalized average operator
      if $\zeta < \Delta$, where $\Delta$ is expert defined normality threshold then
          Activate alarm as abnormal
      else
          Display normality score $\zeta$
      end if
  end if
```

3. Experiments and results

To present the effectiveness of the method, we have used two types of trajectories. We have applied the classifier to find abnormalities in surveillance videos recorded at traffic junction/roadway crossing using static camera. Also, the method has been applied on finger trajectories obtained during 3D air signatures for biometric authentication. We have used a 50 user 3D air signature dataset (Behera et al., 2018). In the context of visual surveillance, two videos datasets, namely QMUL (Loy, Hospedales, Xiang, & Gong, 2012) (30 min) is a traffic activity. The video contains 786 number of trajectories of targets where 21 targets were marked as abnormal. A long duration video (10 h) is recorded. The video contains 12009 targets among 42 are abnormal. High speed, loitering, illegal u-turn, driving in wrong direction, and unusual stops were marked as abnormal. An air signature dataset was prepared using leap motion sensor by tracking of fingers. The dataset contains valid air signatures and forgery signatures of various users. A genuine signature is normal and forged signature is assumed to be abnormal.

3.1. Results using video data

We present the results of classification by varying several factors such as clustering distance threshold, training data size, and number of steps. First experiment demonstrated how training time and accuracy vary over the training size. The experiment has been conducted 10 times for each training size with different set of data and the average results have been reported. Accuracy has been measured in terms of successful classification of identifying abnormal trajectories (object). In the second experiments, we have demonstrated the target movement (trajectory) prediction. We have used 80% of data as training and 20% as testing. In each case, we have predicted user movements until the targets disappear through scene boundary. The experiments have been conducted by varying the number of frames. This experiment also considered a 10-fold cross validation.

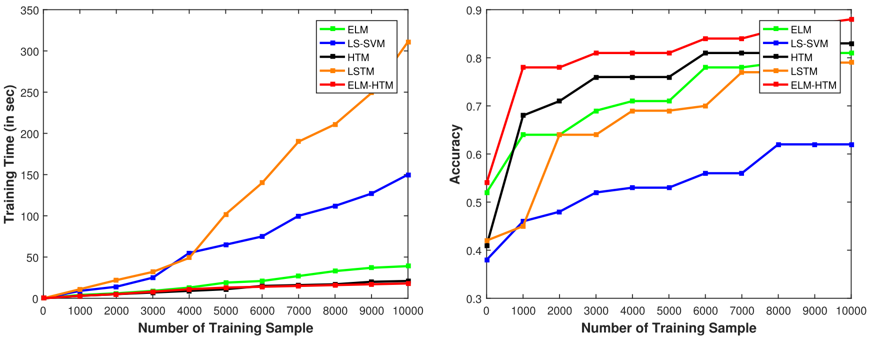
Effects on Training Sample Size: First, we present the training time against the number of training sequences. Fig. 8(a) shows training time verses number of training samples obtained in our recorded residential traffic video dataset. Fig. 8(b) presents the accuracy in such training samples. It has been observed that the training process consumes significantly lesser amount of time even if the sample number increases. For example, with a set of approximately 12 k trajectories, the training took a few seconds on a desktop PC without GPU (Intel core i3, 2.6 Ghz, 8 GB RAM), which is highly encouraging. This is essential for typical real-time learning applications. ELM-HTM method consume similar time as compared to HTM. It is due to the simplistic architecture of the ELM. However, it may be observed that accuracy of the 1-class classifier does not vary significantly even if the training size increases manyfold. Typical sequence classifiers such as LSTM takes more time and cannot achieve accuracy at per with the proposed ELM-HTM framework. Prediction capability has been considered as an important metrics in time series data analysis. We present results of two experiments to demonstrate the prediction capability of the proposed method. First, we have calculated the classification accuracy compared to number of steps observed. We have considered the situation after 5 h of learning in our recorded video.

In Fig. 9(a) we present the result. It has been found that the proposed method outperform when the observed steps are low. Though, the method perform similar to HTM when higher number of steps are observed. We have also perform another experiment after 5 h of learning to understand how much future steps can be predicted accurately. Fig. 9(b) shows the results of prediction accuracy with respect to number of frames need to predict. In this experiment we found a significant improvement of long-future prediction compared to the state-of-the-art methods.

It is also been observed that when the training data increases with time, a ELM-HTM model dynamically adopts the situation by reconstructing the ELM structure. A typical ELM with fixed number of hidden nodes is not effective. It may increase false negatives and affect in the final classification accuracy. Fig. 10 shows a typical ELM network constructed after learning the normality index for varying duration applied on the residential traffic video. Fig. 11(a) shows a comparative analysis of accuracy over time by varying number of nodes in our dataset. It is observed that the hypothesis of taking same number of hidden nodes according to the number of cluster seems to be valid. It has been due to the working principle of ELM, ELM demands less number of nodes when we have small variation in the data. We have measured the variation of data by clustering the regions.

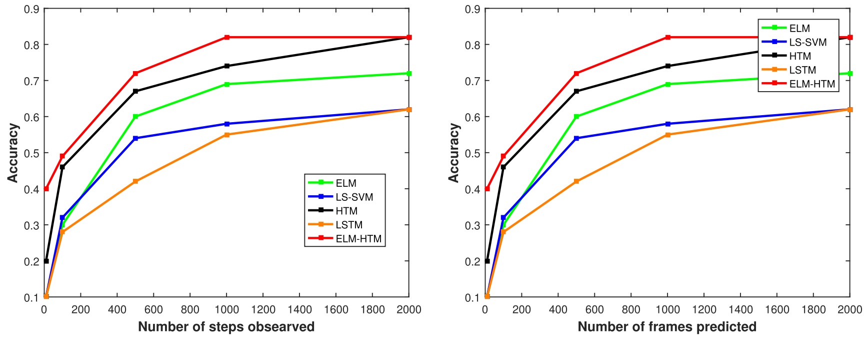
Effect of the DBSCAN Parameter: Though, the proposed method is unsupervised and targeted to a least user iteration, it is also depends on some parameter such as the clustering radios in DBSCAN (ϵ). If ϵ is increased, the number of hidden layer also increased in ELM. Fig. 13(b) presents the result of the classification accuracy after learning 20-min QMUL dataset videos. It is observed that, when the value of ϵ in between $20 - 30$, we have achieved maximum accuracy. In our setting, we have used 20 as the standard setting for all the cases.

Effect of the Normality Threshold (Δ): The normality threshold is somehow sensitive and it depends on diversity of patterns present in the data. Very low or very high threshold can impact the accuracy of the system. If Δ is significantly low, the system is less restricted, i.e., only high deviating patterns are considered as abnormal. A high Δ leads to a highly restricted environment where a small deviation of pattern can be treated as abnormal. Fig. 11(b) presents the accuracy, precision, recall, and F1 scores varying threshold. It has been observed that a value between 0.1 to 0.3 can be reasonably good for this dataset. When the method is applied to a signature verification, we have



(a) (b)

Fig. 8. (a) Result of the training time versus accuracy in our recorded video. It is observed that the method consumes significantly less amount of time due to the signal-based bio-inspired nature of the algorithm. (b) Results of number of training data and accuracy in our recorded video. It is observed that the accuracy for detecting abnormalities increased significantly compared to HTM due the high-level feedback from the ELM layer in final decision.



(a) (b)

Fig. 9. (a) Accuracy after 5 h of learning in our dataset video. Proposed ELM-HTM method gained 20% accuracy in early prediction (with a less number of observed steps) and also gain a little accuracy after observing large number of steps. The result is expected as we have used high-level feedback from the ELM for prediction. For the same reason the method outperform for predicting long-term future movement. (b) Shows the capability of higher-order prediction in our dataset after 5 h of learning. It is reflected that the proposed method also perform better for long-term prediction compared to the state-of-the-art methods.

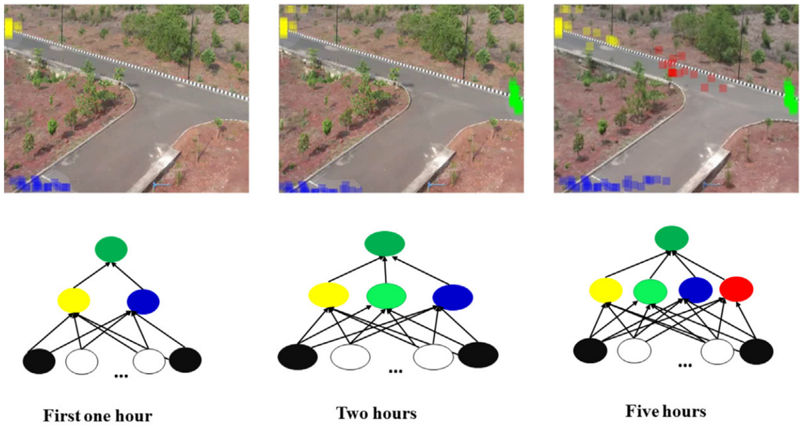
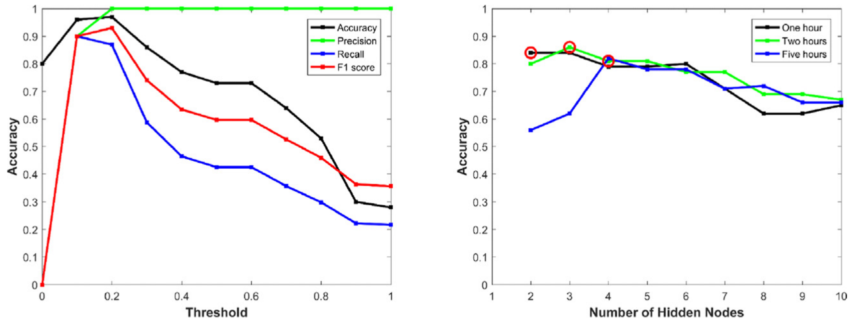


Fig. 10. Dynamic nature of the ELM-HTM learning. It is observed that the system is able to adopts the changes of patters during online learning. There are only two regions (blue and yellow) found during one hour, two other regions (green and red) have been discovered by long term learning (2 and 5 h). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) (b)

Fig. 11. (a) Accuracy, precision, recall, and F1 scores of abnormality detection varying threshold in our video dataset. (b) Accuracy of the ELM varying number of nodes. Region clustering suggests that the minimum number of required hidden nodes are 2, 3, and 4 for one, two, and five hours of videos, respectively. It has also been observed that the model performs the best in most of the cases with the suggested number of nodes (red markers). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

observed that a threshold of 0.8 is good to reduce false positive rate.

Case study: We now present a case study on a public junction video dataset. Fig. 12(a) presents the paths in the scene. The normality setting (Δ) has been fixed at 0.2, i.e., normality scores>0.2 are considered to be normal. Fig. 12(b) presents a scenario when a car taking an illegal U-turn. The proposed method extracts

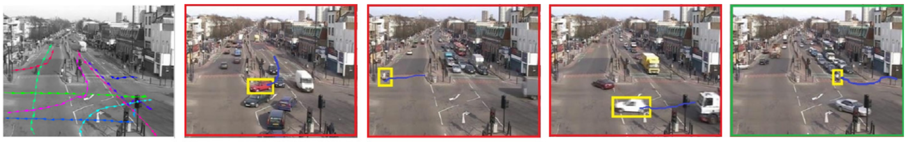
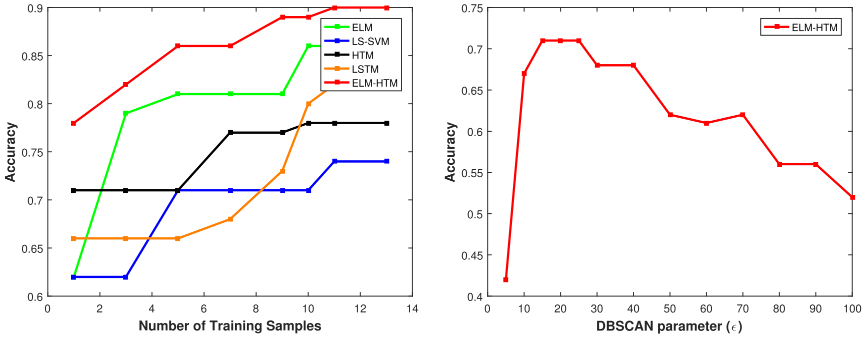

(a) (b) 0.05 (c) 0.09 (d) 0.15 (e) 0.18

Fig. 12. Some examples of abnormal activities in QMUL video dataset.



(a) (b)

Fig. 13. (a) Classification accuracy in 3D air signature data. Result indicates that the proposed method achieved 75% accuracy by observing one sample. The accuracy achieved maximum 90% accuracy observing 12 number of samples (b) Effect of the clustering parameter (ϵ) in ELM learning. 20 min of QMUL (Loy et al., 2012) junction video is used as training and rest 10 min have been used for testing by changing ϵ. It is observed that at $\epsilon = 20$ the method achieved maximum 75% accuracy.

$E_t = 0.01, \alpha = 0.02$, and $\phi = 0.12$ after normalization at some point of time and detects the pattern as abnormal with score = 0.05. In (c), an abnormal pedestrian movement is depicted (visiting infrequent zones) (d) a high speeding car (e) a failure scenario when a pedestrian is identified as abnormal as the pattern is observed for the first time, it becomes normal with score = 0.18 after observing similar patterns multiple times.

3.2. Results using 3D finger trajectory

To demonstrate the learning capability of a single pattern, we have tested the algorithm to verify 3D air signature (Behera et al., 2018) using the trajectory data. First, the classifier is trained using a single user signature. The normality threshold is set to 0.8, i.e. a signature above 0.8 normality considered as authenticate. We have demonstrated abnormal trajectory classification (we have used forged signatures) by varying the number of training samples. In each case, we have experimented 10 times and recorded the average accuracy. We have presented the accuracy of the classifier by randomly selecting training and testing data. Fig. 13(a) shows the accuracy. It is observed that the method achieved 80% accuracy by using only one training sample.

Comparative Analysis: We have compared the results with the state-of-the-art HTM[1] with fixed θ and ELM (PPark & Kimark & Kim, 2017)[2] with fixed number of hidden nodes, LS-SVM (Chen & Lee, 2015), and LSTM (Sutskever et al., 2014).[3] The methods are sensitive to various parameters and the results are reported using the best possible values of the parameter to achieve highest accuracy. The values are estimated by experimenting different values of the parameters on the same dataset and parameter setting with the highest accuracy is considered as the standard setting.

4. Concluding remarks and future direction

In this paper we have presented a new bio-inspired online-learning model for a single-class classifier to detect normality in time series data sequences. The method uses extremal learning machine (ELM) and hierarchical temporal memory (HTM) together called ELM-HTM in an unsupervised fashion to learn and classify time series patterns. The method has been tested on trajectory sequences in traffic surveillance to find abnormal behaviours and 3D air signatures that have been captured using sensors. The proposed method uses ELM feedback to HTM to refine the prediction and HTM feedback to the ELM classification layer to classify a pattern. The results indicate a significant gain over training time and classification accuracy. The method includes real-time learning and least user supervision. The method can be used in various time series data analysis where the normality is dynamic and vary time to time such as traffic flow analysis/movement pattern analysis by object tracking or GPS tracking and air writing

[1] https://github.com/numenta/htm.java.

[2] https://github.com/dclambert/Python-ELM.

[3] https://github.com/RobRomijnders/LSTM_tsc.

signature authenticating, etc. The future direction of the work may be extended to large volume of trajectories such as air-traffic, satellite movement, city traffic by GPS, crowd activity, etc.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Funding: The work has not been funded from anywhere.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors. Informed consent: Informed consent was obtained from all individual participants included in the study.

References

Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing, 262*, 134–147.

Ahmed, S. A., Dogra, D. P., Kar, S., & Roy, P. P. (2018a). Surveillance scene representation and trajectory abnormality detection using aggregation of multiple concepts. *Expert Systems with Applications*.

Ahmed, S. A., Dogra, D. P., Kar, S., & Roy, P. P. (2018b). Trajectory-based surveillance analysis: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*.

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 961–971).

Albusac, J., Vallejo, D., Castro-Schez, J. J., Glez-Morcillo, C., & Jiménez, L. (2014). Dynamic weighted aggregation for normality analysis in intelligent surveillance systems. *Expert Systems with Applications, 41*(4) , 2008–2022.

Behera, S. K., Dogra, D. P., & Roy, P. P. (2018). Fast recognition and verification of 3d air signatures using convex hulls. *Expert Systems with Applications*.

Campo, D., Baydoun, M., Marcenaro, L., Cavallaro, A., & Regazzoni, C. S. (2018). Unsupervised trajectory modeling based on discrete descriptors for classifying moving objects in video sequences. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 833–837). IEEE.

Chen, T.-T., & Lee, S.-J. (2015). A weighted ls-svm based learning system for time series forecasting. *Information Sciences, 299*, 99–116.

Cui, Y., Ahmad, S., & Hawkins, J. (2016). Continuous online sequence learning with an unsupervised neural network model. *Neural Computation, 28*(11), 2474–2504.

Cui, Y., Ahmad, S., & Hawkins, J. (2017). The htm spatial pooler: A neocortical algorithm for online sparse distributed coding. *Frontiers in Computational Neuroscience*, 11.

Cui, Y., Surpur, C., Ahmad, S., & Hawkins, J. (2016). A comparative study of htm and other neural network models for online sequence learning with streaming data. In *Neural Networks (IJCNN), 2016 International Joint Conference on* (pp. 1530–1538). IEEE.

Deng, L., Li, G., Deng, N., Wang, D., Zhang, Z., He, W., ... Shi, L. (2015). Complex learning in bio-plausible memristive networks. *Scientific Reports (Nature), 5*, 10684.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2625–2634).

Edwards, J. L., Saphir, W. C., Ahmad, S., George, D., Astier, F., & Marianetti, R. (2017). Hierarchical temporal memory (htm) system deployed as web service. US Patent 9,621,681.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd, 96*, 226–231.

Fan, D., Sharad, M., Sengupta, A., & Roy, K. (2016). Hierarchical temporal memory based on spin-neurons and resistive memory for energy-efficient brain-inspired computing. *IEEE Transactions on Neural Networks and Learning Systems, 27*(9), 1907–1919.

Hawkins, J., & Ahmad, S. (2016). Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in Neural Circuits, 10*, 23.

Hawkins, J., & Blakeslee, S. (2007). *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan.

Hawkins, J., & George, D. (2016). Methods, architecture, and apparatus for implementing machine intelligence and hierarchical memory systems. US Patent 9,530,091.

Huang, G.-B., & Chen, L. (2008). Enhanced random search based incremental extreme learning machine. *Neurocomputing, 71*(16–18), 3460–3468.

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

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42*(2), 513–529.

Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters, 42*, 11–24.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521* (7553), 436.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., & Chandraker, M. (2017). Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 336–345).

Loy, C. C., Hospedales, T. M., Xiang, T., & Gong, S. (2012). Stream-based joint exploration-exploitation active learning. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1560–1567). IEEE.

Mabrouk, A. B., & Zagrouba, E. (2017). Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications*.

Meng, H., Bianchi-Berthouze, N., Deng, Y., Cheng, J., & Cosmas, J. P. (2016). Time-delay neural network for continuous emotional dimension prediction from facial expression sequences. *IEEE Transactions on Cybernetics, 46*(4), 916–929.

Pacheco, A. G., Krohling, R. A., & da Silva, C. A. (2018). Restricted boltzmann machine to determine the input weights for extreme learning machines. *Expert Systems with Applications, 96*, 77–85.

Park, J.-M., & Kim, J.-H. (2017). Online recurrent extreme learning machine and its application to time-series prediction. In *Neural Networks (IJCNN), 2017 International Joint Conference on* (pp. 1983–1990). IEEE.

Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition, 44*(3), 678–693.

Rodríguez-Serrano, J. A., & Singh, S. (2012). Trajectory clustering in cctv traffic videos using probability product kernels with hidden markov models. *Pattern Analysis and Applications, 15*(4), 415–426.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Tang, J., Deng, C., & Huang, G.-B. (2016). Extreme learning machine for multilayer perceptron. *IEEE Transactions on Neural Networks and Learning Systems, 27*(4), 809–821.

Wu, J., Zeng, W., & Yan, F. (2018). Hierarchical temporal memory method for time-series-based anomaly detection. *Neurocomputing, 273*, 535–546.

Xu, H., Zhou, Y., Lin, W., & Zha, H. (2015). Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4328–4336).

Yi, S., Li, H., & Wang, X. (2016). Pedestrian behavior understanding and prediction with deep neural networks. In *European Conference on Computer Vision* (pp. 263–279). Springer.