ARTICLE IN PRESS

JID: NEUCOM [m5G;July 10, 2017;21:22]

Neurocomputing 000 (2017) 1–9

Contents lists available at ScienceDirect

Neurocomputing

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

Extreme learning machine based transfer learning algorithms: A survey

Syed Moshfeq Salaken*, Abbas Khosravi, Thanh Nguyen, Saeid Nahavandi

Institute for Intelligent Systems Research and Innovation, Deakin University, 75 Pigdons Road, Waurn Ponds, Victoria 3216, Australia

ARTICLE INFO

Article history:
Received 6 November 2016
Revised 12 June 2017
Accepted 13 June 2017
Available online xxx

Communicated by Dr. G.-B. Huang

Keywords:
Transfer learning
Extreme learning machine

ABSTRACT

Extreme learning machine (ELM) has been increasingly popular in the field of transfer learning (TL) due to its simplicity, training speed and ease of use in online sequential learning process. This paper critically examines transfer learning algorithms formulated with ELM technique and provides state of the art knowledge to expedite the learning process ELM based TL algorithms. As this article discusses available ELM based TL algorithm in detail, it provides a holistic overview of current literature, serves as a starting point for new researchers in ELM based TL algorithms and facilitates identification of future research direction in concise manner.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Machine learning (ML) has become instrumental in solving difficult problems in the domain of forecasting, classification and clustering. However, conventional machine learning algorithms assume that the underlying distribution of training samples and testing samples will be the same. Unfortunately, this assumption is often violated in practice and therefore, conventional ML algorithms fail to serve their purpose in this context [1,2]. The solution to this problem is particularly difficult as collecting enough labeled samples from domain of interest is often time consuming and very expensive. The sub-field of machine learning that deals with this distribution shift issue is commonly termed as transfer learning (TL). TL algorithms attempts to solve this learning problem by training with seemingly unrelated samples from a different, but related, domain.

Extreme learning machine (ELM) has recently started gaining momentum in the field of transfer learning. Speed of learning and the simplicity of implementation of ELM algorithm has been instrumental in the popularity of ELM. The ease of using online sequential learning techniques in ELM has played a key part as well. This article reviews all articles on transfer learning using extreme learning machine available on Google scholar, Scopus and ScienceDirect at the time of writing. The objective of this article is to provide concise overview of techniques used by published works that combine ELM and transfer learning. It also comments on the strengths and limitations of each of the proposed algorithms while also recommending future aspects of research in transfer learning with ELM. Table 1 presents overview of techniques, strengths and limitations of articles discussed. The main contributions of this paper are:

- Describing the algorithm of extreme learning machine, kernel techniques and online learning algorithm of extreme learning machine in simple words so that it serves the purpose of easy consumption for researchers in other fields of transfer learning.
- Summarizing all available articles in the field of transfer learning with ELM using same set of notations (while the original articles used different notations) so that the merit of each algorithm can be understood easily.
- It also provides a set of recommendations for future researchers interested in TL using ELM to establish a common framework for comparison and benchmarking.
- The algorithms are discussed in a way that allows for identification of the gaps in the literature and therefore makes it simple to find future research directions in the field of TL using ELM.

The remainder of this paper is organized as follows: Section 2 discusses the concept of extreme learning machine (ELM), kernel techniques and online sequential learning in the context of ELM, Section 3 discusses existing ELM based TL algorithms in detail, Section 5 provides some recommendation for future research in ELM based TL algorithms and finally Section 6 concludes the paper.

* Corresponding author.
E-mail addresses: syed.salaken@gmail.com, ssalaken@deakin.edu.au, syed.salaken@research.deakin.edu.au (S.M. Salaken).

http://dx.doi.org/10.1016/j.neucom.2017.06.037
0925-2312/© 2017 Elsevier B.V. All rights reserved.

Please cite this article as: S.M. Salaken et al., Extreme learning machine based transfer learning algorithms: A survey, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.06.037

Table 1
Overview of different algorithms.

		Articles
Techniques used	Online sequential learning	[3–7]
	Kernel function	[5,8,9]
	Regularization on target domain samples	[8,10,11]
	Transfer samples	[12]
	Spare representation	[13]
	Ensemble	[3]
	Weighting	[14]
	Stacking	[9]
Strengths	Feature adaptiveness	[4]
	Attempts to avoid negative transfer	[14,15]
	Potential to become general framework	[4,10,11]
	Multiple source domain	[7]
Limitations	Small drift assumption (e.g. small $\beta_S - \beta_T$)	[8,12,14]
	Prone to overfitting	[14]

2. Preliminaries

For the sake of readability, let us define some notations. Throughout the whole article, following notations will apply:

- β: output weights of ELM
- **H**: hidden layer output matrix (i.e. resultant matrix of activation function on input features and bias)
- w: input weights
- b: bias
- **I**: identity matrix
- **K**: kernel matrix
- C, C_1, C_2 etc.: regularization constants
- **T**: label matrix
- ξ: prediction error ($\xi = \mathbf{H}\beta - \mathbf{T}$)
- **X**: input matrix
- S in subscript: source domain
- T in subscript: target domain
- O: output of ELM (i.e. $O = \mathbf{H}\beta$)

2.1. Transfer learning

Transfer learning is a sub-field of machine learning that deals with the distribution shift of data among domains. Traditionally, machine learning algorithms assume that both training and testing data follow similar distributions. This is a strong assumption and therefore, conventional machine learning algorithms performs poorly when this condition is not met. Unfortunately, there are many cases when data collection in the task of interest is difficult, expensive and sometime impossible [1]. Transfer learning is the specific branch of machine learning that aims to tackle this issue by learning on relevant data. Historically, it has been called by different names including domain adaptation, covariate shift, learning to learn etc. We encourage the reader to go through the excellent survey [1] for an in-depth understanding of different aspects and algorithms in the domain of transfer learning. For the sake of completeness, the concept of source and target domains is described in the sequel.

In the terminology of transfer learning, the domain of interest where learning is required is called target domain. Target domain usually has no or little amount of labeled samples. The amount of available samples are not enough to train any learning algorithm. However, it is assumed that there is at least one related domain with enough samples which can be used for training purpose and extract relevant knowledge. These domains, from where relevant data is collected for training, is called source domain(s). Proper identification of source domain is crucial as using drastically different training samples may adversely effect the learning process. When this happens, it is called negative transfer. It is very crucial for the TL algorithms to put mechanisms in place to avoid negative transfer. However, very few ELM based algorithms actively designed this mechanism in their proposed learning process.

2.2. Extreme learning machine

Huang et al. [16] proposed a new version of neural network in 2004 called extreme learning machine (ELM). This new learning framework is essentially a single layer feed forward network where input weights are chosen randomly and output weights are determined analytically using a Moore–Penrose generalized inverse (p. 147 of [17]). The concept of extreme learning machines are counter-intuitive from the perspective of a traditional neural network in the sense that ELMs do not learn the inputs weights that connect the inputs to hidden layer. However, Huang et al. [16] showed the ELMs can reach the minimum training error [18–20] while also attaining lowest norm of weights [16], which implies that ELMs have better generalization performance [21]. The complete algorithm for extreme learning machine is described in Algorithm 1.

Algorithm 1 Extreme learning machine.

Input: Training set, $\mathcal{N} = \{(x_i, t_i)\} | x_i \epsilon \mathcal{R}^n, t_i \epsilon \mathcal{R}^m, i = 1, 2, \ldots, N$, activation function $g(x)$, number of neurons in hidden layer N_h
Output: Output weights, β

1: Choose random values for input weights w_i and bias b_i, $i = 1, 2, \ldots, N_h$
2: Calculate hidden layer output matrix, H, which is defined as:

$$H = \begin{bmatrix} g(w_1 x_1 + b_1) & \cdots & g(w_{N_h} x_1 + b_{N_h}) \\ g(w_1 x_2 + b_1) & \cdots & g(w_{N_h} x_2 + b_{N_h}) \\ \cdots & \cdots & \cdots \\ g(w_1 x_N + b_1) & \cdots & g(w_{N_h} x_N + b_{N_h}) \end{bmatrix}_{N \times N_h} \quad (1)$$

3: Calculate output matrix, β as

$$\beta = H^{\dagger} T \quad (2)$$

where, $H^{\dagger}$ is the Moore–Penrose generalized inverse of H that minimizes the L_2 norm of both $||H\beta - T||$ and $||\beta||$. T is the label matrix of the training set defined as

$$T = \begin{bmatrix} t_1^T \\ t_2^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (3)$$

where m is the dimension of labels for each training samples
4: **return** β

The activation function $g(x)$ is not limited to sigmoidal activation functions and tend to be very generalized. The readers are encouraged to go through [22] for a detailed analysis. The Moore–Penrose generalized inverse in Eq. (2) can be computed as follows:

$$\mathbf{H}^{\dagger} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \quad (4)$$

If regularization is applied in the ELM, the learning problem turns into a minimization problem which can be formulated as follows:

$$\min_{\beta} \mathcal{L} : \frac{1}{2} ||\beta||^2 + C.\frac{1}{2}.\sum_{i=1}^{N} ||\xi_i||^2 \quad (5)$$

subject to $\mathbf{H}(x_i)\beta = t_i - \xi_i, i = 1, 2, \ldots, N$. ξ denotes the prediction error of ELM.

ELM is very capable learner with many different configurations and architectures including, but not limited to, SLFN [23], local receptive fields [24], multi-layer [25,26] and auto-encoder [27]. They are universal approximators [22,28] and frequently used in regression [22,29], classification [22,30] and clustering [31,32] among many other applications. It is also shown that ELM unify the solution framework for regression, binary classification and multi-class classification [22]. However, for the purpose of brevity and conciseness, their descriptions are not included in this transfer learning focused review and readers are encouraged to go through the cited articles for better understanding. In the sequel, concepts relevant for transfer learning is discussed in the following sections.

2.3. Kernel extreme learning machine

Kernel extreme learning machines [22] are traditional ELMs with a kernel function. A kernel function is used to derive a kernel from original feature matrix. Common kernel functions include radial basis function, sinusoidal function, exponential functions etc. When a kernel matrix $\mathbf{K}$ is known, output weights β can be calculated as follows:

$$\beta = \left(\frac{\mathbf{I}}{C} + \mathbf{K}\right)^{-1}\mathbf{T} \tag{6}$$

where $\mathbf{I}$ is identity matrix, $\mathbf{C}$ is regularization coefficient and $\mathbf{T}$ is the label matrix of the training set.

2.4. Online sequential learning using extreme learning machine

Liang et al. [33] proposed online learning mechanism for extreme learning machine. The idea behind online sequential learning captures the fact that a base learner can be updated by using small amount of incoming data. This ensures that previously trained base learner is updating its knowledge about the new data by learning on small chunks at a time. More specifically, if H_0 and T_0 is the hidden layer output matrix and label matrix from first chunk of data (which is often the biggest amount available at the time of initial training), H_1 and T_1 is the new hidden layer output matrix and label matrix from new (and often smaller) chunk of data, then updated value of output matrix $\beta^{(1)}$ can be computed as follows:

$$\beta^{(1)} = \beta^{(0)} + \mathbf{K}_1^{-1}\mathbf{H}_1^T(\mathbf{T}_1 - \mathbf{H}_1\beta^{(0)}) \tag{7}$$

where $\mathbf{K}$ is the kernel matrix, $\mathbf{H}$ is the input layer hidden matrix and T is the label matrix. $\beta^{(0)}$ denotes the output matrix in the first training phase which is computed from $\mathbf{H}_0$ and $\mathbf{T}_0$

2.5. Reduced kernel extreme learning machine

Deng et al. [34] proposed reduced kernel extreme learning machine (RKELM). The idea behind RKELM is similar as kernel ELM (KELM). However, RKELM works only with randomly selected subset from input samples. The authors in [34] showed that RKELM can achieve similar level of accuracy as KELM with reduced computational cost. More specifically, if $\tilde{N}$ is the number of samples in randomly selected subset of training data, then $\tilde{N} \ll N$, where N is the number of samples in available data.

3. Existing works

3.1. Drift compensation in sensor measurement

Zhang and Zhang [10,11] proposed extreme learning machine based transfer learning framework in the context of sensor measurement drift compensation for E-Nose systems. They proposed two different ELM based transfer learning algorithms called DAELM-S and DAELM-T. The first algorithm learns an ELM classifier on source domain data and uses limited labeled target domain data in the regularization scheme. This is done by minimizing the following objective function:

$$\min_{\beta_S, \xi_S^i, \xi_T^i} \frac{1}{2}||\beta_S||^2 + C_S\frac{1}{2}\sum_{i=1}^{N_S}||\xi_S^i||^2 + C_T\frac{1}{2}\sum_{j=1}^{N_T}||\xi_T^i||^2 \tag{8}$$

where S and T denote the source and available target domain samples, respectively. ξ carries the same meaning as in Eq. (2).

The second algorithm aims to learn a classifier on the very limited labeled target domain data while the unlabeled data from target domain are also exploited by getting a predicted output set by another classifier trained on source domain. The exploitation of unlabeled target domain samples occurs in the form of regularization while another terms controls overfitting in learning process. The regularization in the context of DAELM-T compares the output of source domain trained ELM classifier with that of another ELM classifier trained on limited target domain data. This is an interesting idea as it attempts to utilize the full knowledge of limited labeled data from target domain while also extracting regulated knowledge from unlabeled data in the target domain by passing them to a well tuned classifier from source domain. In other words, it tries to minimize the error between classifiers trained on source and limited labeled target domain data. The idea in this algorithm is implemented by minimizing the following cost function:

$$\min_{\beta_T} \frac{1}{2}||\beta||^2 + C_T\frac{1}{2}\sum_{i=1}^{N_T}||\xi_T^i||^2 + C_{Tu}\frac{1}{2}\sum_{j=1}^{N_{Tu}}||\mathcal{E}||^2 \tag{9}$$

where, $\mathcal{E} = H_{Tu}\beta_S - H_{Tu}\beta_T$. $H_{Tu}\beta_s$ is the prediction on unlabeled target domain samples by an ELM classifier trained with the source domain data.

The authors of [10,11] have demonstrated the effectiveness of their algorithm on long term sensor drift data from UCI repository. Both of the proposed algorithm showed better performance even though the DAELM-T requires more labeled samples to perform better.

Yan and Zhang [12] proposed another ELM based transfer learning framework for electronic nose (E-Nose) systems. They proposed transfer sample based coupled task learning (TCTL) framework which can use conventional machine learning algorithms to improve the performance in target domain. The proposed TCTL algorithm uses a small number of labeled sample from target domain (i.e. samples from slave device with drift) to learn a classifier. The essential contribution of TCTL algorithm is a cost function that consists of a loss term over source domain training samples, a term that attempts to reduce the difference of source and target domain in a projected space and a term that attempts to minimize the difference between projection vectors for source and target domain. The key point to note here is the use of so called "transfer samples" from both source and target domain. The paper does not describe how the "transfer samples" from source domains are selected. However, without same number of "transfer samples" from these domain, computation of the second term in the objective function would not be possible. The cost function of [12] can be described as follows:

$$\min_{\beta_S, \beta_T} \frac{1}{2}||\xi_S||^2 + C_1||\mathcal{E}_{sim}||^2 + C_2||\beta_S - \beta_T||^2 \tag{10}$$

where ξ represents the training error over source domain, C_1, C_2 represents coefficient constants, $\mathcal{E}_{sim} = (X_{S_{TS}}\beta_S - X_{T_{TS}}\beta_T)$ represents similarity of transfer samples projected by respective domain. $X_{S_{TS}}$ and $X_{T_{TS}}$ represents the transfer samples from source and target domains, respectively. β_S and β_T represents output weights of ELM trained on source domain samples and available target domain samples, respectively.

In addition to this, it is essential to have same number of features/variables in the source and target domain in [12]. The third term in the cost function enforces similar projection vectors for source and target domain under the assumption that deviation in the E-Nose system is small. Therefore, performance of the proposed algorithm under drastic changes between domains is subject to investigation. The authors of [12] demonstrated the performance of their algorithm over 10 datasets and compares with a number of relevant algorithms. Even though the proposed TCTL algorithm was outperformed by DAELM [8] and ensemble classifiers in most cases, it performs reasonably well in those cases. In the remaining cases, TCTL algorithm outperform all other algorithms.

3.2. Data classification

Liet al. [8] uses an extreme learning machine (ELM) to tackle the transfer learning (TL) problem and proposed a TL-ELM algorithm. They first train an ELM classifier on source domain training set and then optimize the classifier on available target domain dataset. More specifically, in the dual formation of the problem, they first aim to find two different penalty parameters using source domain knowledge. Afterwards, they optimize the remaining parameters of derived Lagrangian using already computed penalty parameters using available target domain knowledge. The cost function to achieve this can be described as follows:

$$\min_{\beta_T,\beta_S} \frac{1}{2}||\beta_T||^2 + C_T\sum_{i=1}^{N}\xi_T^i + \frac{1}{2}C_1||\beta_T-\beta_S||^2 \tag{11}$$

In addition to this TL-ELM algorithm, the authors of [8] proposed another algorithm called TL-DAKELM. This TL-DAKELM aims to utilize kernel ELM along with existing domain adaptation methods to solve transfer learning problem. In this approach, the authors uses default Gaussian kernel, find the bandwidth and penalty parameter and then solve for the remaining tuning parameters. However, unlike the first algorithm TL-ELM, the article does not mention whether the source domain dataset is used to find the first set of parameters. It is important to note that both of the proposed algorithms require labeled target domain data. The paper uses a number of UCI datasets including Banana, Titanic, Waveform, Flare Solar, 20 newsgroups as well as ORL and Yale face dataset to compare the performance of their algorithm to others. While they briefly mentioned the source and target domain dataset selection process for 20 newsgroups dataset, it is completely absent for remaining datasets. It would have been informative for the reader to have a hint on the selection process, specially for the seemingly unrelated dataset like Banana, Titanic etc. Moreover, even though their algorithm shows some improvement for most of the dataset, the lack of repeated testing on the paper prohibits the reader from achieving any insight on the stability of the performance of the proposed algorithm.

Liet al. also proposed another variant of ELM based transfer learning algorithm in [13] which focuses on sparse representation optimization through graph Laplacian computation. Essentially, the authors attempt to learn a free sparse representation from unlabeled data, which is followed by a dictionary learning phase. Afterwards, a TL-ELM classifier is trained. The essence of their proposed framework is achieved by embedding the source domain labels, target domain labels and predictions from classifiers trained on source domain data in a common latent space within a regularization framework. Similar to their previous work in [8], seemingly unrelated datasets from UCI dataset is used in the performance comparison section. However, it is not discussed why some unrelated source domain is able to improve the performance in target domain classification task.

Xu and Wang [3] investigates ELM for data stream classification. Since the data stream classification task is sequential, learning speed is crucial and hence, ELM is a suitable candidate for this task due to the low training time according to the authors. The researchers in [3] proposed IDS-ELM algorithm which is able to detect and address concept drift in data stream classification. IDS-ELM algorithm exploits the fact that output weights of an ELM classifier can be adjusted for a new block of incoming data by using current activation matrix from new samples and the old output weights as well as old activation matrix from previously trained ELM classifier (see Eq. (6) for details). This concept is very similar to online sequential ELM (OS-ELM) (see Section 2.4). However, IDS-ELM determines a drastic change of concept in the incoming data streams by observing the accuracy of incrementally updated classifiers. When the deviation of accuracy between successive updates is greater than Hoeffding Bound, the IDS-ELM algorithm decides that a drastic change has occurred and therefore, it trains a new batch of ELM classifiers with randomly selected activation function from six different choices (i.e. sigmoidal, sinusoidal, multiquadratic, Gaussian, Hardlim and triangular basis). This is a simple implementation of ensemble classification. It determines the optimal number of hidden neurons based on the strategy described below and continue until the accuracy of weighted ensemble classifiers is lower than Hoeffding Bound. Accuracy of the ensemble of classifiers is determined based on weighted voting, where weights are inverse of (1 − accuracy).

Xu and Wang [3] also proposed an algorithm to determine the optimal number of neurons in the hidden layer of ELM. This algorithm follows the underlying principle of binary search and search for optimal number of hidden neurons in the range of [0, m], where m is the smaller number between the number features and number of samples in the input data. The optimal number of hidden neurons is determined based on the lowest error from a number of ELM classifiers trained with hidden neurons taken from this range following a binary search like method where the search is initiated with 0.5*m number of neurons in the hidden layer. This IDS-ELM algorithm then compares the accuracy of this initial ELM with two different ELMs taken from the right and left of its search domain each i.e. ELM trained with number of hidden neurons that is the middle number of [1, 0.5m) and (0.5m, m]. Based on the accuracy comparison, the proposed algorithm determines the direction of further search. This process continues until the difference of accuracy falls below a certain threshold. The authors of [3] demonstrated the success of their algorithm in several synthetic and UCI datasets. The result is mixed and no algorithm in comparison clearly wins most of the tests. In many cases, conventional neural network using back-propagation algorithm outperforms the proposed IDS-ELM algorithm. The authors recognize this fact and shows the advantage of IDS-ELM in terms of computation speed. However, this is trivial as ELM classifiers does not utilize iterative learning to tune the input weights. Although their results show that IDS-ELM is more stable in terms of mean accuracy, it does not imply the superiority of IDS-ELM in any way.

3.3. Wi-Fi localization

Jiang et al. [4] proposed a feature adaptive ELM framework in the context of Wi-Fi localization problem. Since the nature of Wi-Fi network requires continuous addition and removal of access points, models trained on outdated data will not be able to work once any kind of change is applied. Therefore, their feature adaptive model falls into transfer learning category.

The proposed framework in [4] has the ability to update the old model using small amount of data. This approach falls into the domain of online learning and hence, the proposed model is called termed as Feature Adaptive Online Sequential Extreme Learning

Machine (FA-OSELM) algorithm. In essence, the contributions of this algorithm include a input-weight transfer matrix and a input-weight supplement vector to adjust the changing dimension of feature matrices among domains in conjunction with online sequential learning. The proposed method in not limited only to ELM and has the potential to become a general framework in transfer learning if the base learner has incremental learning ability.

The FA-OSELM framework starts by training an ELM classifier on the initial data and then addresses the changing number of features in new batch of data (resulting from a Wi-Fi access point network change) by introducing input-weight transfer matrix and input-weight supplement vector to generate new feature matrix from old feature matrix and new samples. This mechanism either stretches or shrinks the feature matrix depending on the dimension of new samples from changed network while also ensuring that the required changes are reflected in the new feature matrix. Essentially, input-weight transfer matrix handles the dimensionality change of the old feature matrix with respect to the new samples and input-weight supplement vector addresses the adjustment of value in the new feature matrix. Once the old feature matrix is adjusted with respect to the new data, an incremental learning is done to update the parameters of the already trained classifier (see Section 2.4 for details of this process). Therefore, this model does not need to retain old data for a complete retraining once a new batch of data is available due to a Wi-Fi network update. The authors of [4] analytically derived the incremental parameter update formula to implement the sequential online learning ability. Although this FA-OSELM framework is specifically designed for Wi-Fi localization problem, the authors has also demonstrated the success of their framework in image segmentation datasets from UCI, making this algorithm a good general transfer learning framework.

3.4. Remote sensing

Zhou et al. [14] proposed an weighted ELM based transfer learning classification (WELMTC) framework in the context of remote sensing monitoring. The premise of the work lies in the fact that many factors including different atmospheric condition, land cover and phenological changes often drastically changes the elements of images and machine learning models trained on old images does not work well anymore. Therefore, transfer learning has the potential to prevent the wastage of historical data and may yield better performance with a small sample of new images.

The proposed framework in [14] starts by training an ELM classifier on historical data, which is considered as source domain data in their work, and then finding the new output connection weights by minimizing a cost function over new sample images (limited target domain data) and an error term with respect to old connection weights. The cost function can be described as follows:

$$\min_{\beta_T}(1-C)||\mathbf{T}_T-\mathbf{H}_T\beta_T||\Phi_T+C||\beta_T-\beta_S|| \tag{12}$$

where β_S is learned from weighted ELM (weights are placed in a diagonal matrix with each element is greater than zero). Φ_T is the weight matrix for available target domain samples. For weighted ELM, β_S can be calculated according to the following equation:

$$\beta_S=(\mathbf{H}_S^T\Phi_S\mathbf{H}_S)^{-1}\mathbf{H}_S^T\Phi_S\mathbf{T}_S \tag{13}$$

where Φ_S is the source domain weight matrix.

The authors have analytically shown that the old connection weights from the historical ELM classifier (e.g. the classifier trained on source domain data) contains the required historical information to perform the transfer of knowledge in the context of remote sensing images. More specifically, the proposed cost function minimizes the prediction error between the prediction of an ELM classifier trained on available target domain samples and the target domain ground truth while also minimizing the difference between historical output weights and current output weights in a regularized manner. The authors of [14] have adopted an aggressive approach where they keep learning classifier parameter until all images (i.e. both source domain and available target domain images) are classified correctly. This is a potential pitfall of overfitting and needs to be further investigated. However, unlike other research works mentioned above, the framework in [14] considers the effect of negative transfer and implements measure to prevent negative transfer.

The proposed framework adopts an AdaBoost (more specifically, TrAdaBoost for TL problems) [35] like strategy where weight of misclassified target domain sample is increased to magnify its significance. On the other hand, if a historical image (i.e. source domain image) is misclassified by the new ELM classifier, it is considered less relevant and corresponding weight is decreased. Since this a greedy strategy to filter relevant samples, the proposed classification framework can potentially discard all the historical information and may turn into a simple ELM classifier trained only on available target domain images. The authors of this work has successfully demonstrated the superiority of their framework on two different remote sensing dataset (namely, SPOT-5 from Dalian, China and TM remote sensing images from Panjin, China) over online sequential ELM, classic ELM, SVM and Bayesian ARTMAP. It would be interesting to see the comparison of their classifier with respect to other transfer learning algorithms like TrAdaBoost [35], specially because the proposed algorithm in [14] seems to highly influenced by TrAdaBoost algorithm.

3.5. Activity recognition

Deng et al. [5] applied transfer learning in the field of cross-person activity recognition using ELM. They proposed an algorithm called Transfer learning Reduced Kernel Extreme Learning Machine (TransRKELM) which utilized the concept from reduced kernel extreme learning machine (see Section 2.5 for details on RKELM) and online sequential extreme learning machine (see Section 2.4 for details on online learning for ELM). The reduced kernel technique in ELM effectively reduces the sample size by randomly selecting a smaller subset of available sample. This helps to reduce computational cost at the cost of accuracy. It becomes particularly important when learning needs to be done on embedded devices.

The researcher in [5] used approximately 10% of available samples to build the reduced kernel extreme learning machine (RKELM). These 10% samples are considered as source samples and the trained RKELM serves as the base learner in the proposed algorithm. Output weights in RKELM can be computed according to the following equation:

$$\beta=\left(\frac{\mathbf{I}}{C}+\tilde{\mathbf{K}}^T\mathbf{K}\right)\tilde{\mathbf{K}}\mathbf{T} \tag{14}$$

where $\tilde{\mathbf{K}}$ is the reduced kernel of input features (i.e. kernel matrix for reduced source samples).

Subsequently, in the online knowledge transfer stage, the proposed algorithm collects new samples from users during recognition phase. It is essentially the initial usage period of new user. The recognition (i.e. test) samples with high confidence classification confidence are retained as new training samples. Once the number of retained samples crosses a predefined threshold, they are used to update the parameters of base ELM using sequential update method (see Eq. (6) for update formula). A potential pitfall to this new sample acquisition strategy is getting wrong label for these new training samples. This may results into negative transfer and deteriorates the performance of base learner. The authors of [5] demonstrated

Please cite this article as: S.M. Salaken et al., Extreme learning machine based transfer learning algorithms: A survey, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.06.037

the performance of their algorithm on UCI dataset which is available at (http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones). The demonstration shows that the proposed (TransRKELM) is clearly outperformed by Huang's KELM and SVM. It is important to note that Huang's KELM and SVM did not use any new samples to update their parameters during the test phase. It would be interesting to see a comparison of the base learning before and after the updates. Even though proposed algorithm trains much faster than traditional ELM or SVM, this advantage is clearly outweighed by the significant decrease in accuracy.

Chen et al. [6] investigated an ELM based transfer learning problem in the context of activity recognition using mobile device. The proposed model follows two stage approach i.e. offline training of a base learner and online parameter update using limited amount of data, which is similar to many other ELM based transfer learning frameworks described above. However, the proposed algorithm in [5] requires a number of preprocessing steps before training an ELM base learner. More specifically, the proposed algorithm process the accelerometer signals from different axes into a single magnitude series to get synthesized acceleration, extract some statistical and frequency domain features and finally applies principal component analysis to get robust features in a reduced dimensional space. These steps are applied to both initial data (source domain samples) and incremental update data (target domain samples). Finally, an ELM base learner (e.g. a classifier) is trained with source domain data in first stage. In the second stage, the algorithm attempts to update the parameter of the model by learning from new samples. These new samples are collected in the same way as described in [5] and hence suffers from the same potential problem of updating parameters using wrongly labeled data. Specifically, the proposed algorithm applies the base learner on new data, possibly from a new person using a new mobile device, classify them using base learner and identify the incoming samples with high confidence in classification. The confidence is calculated according the following equation:

$$O_{T_i} = O_{T_i} - \min(O_T) \tag{15}$$

$$\text{confidence}_i = \frac{\max(O_{T_i})}{\sum(O_{T_i})} \quad i = 1, 2, \dots, N \tag{16}$$

Afterwords, these high confidence samples are used as target domain samples and parameters of base ELM classifier is updated using these samples. This online parameter update idea is the same as online sequential ELM training. The researchers in [6] used their own data, which is collected by their own device built using a XSens MTx sensor containing triaxial accelerometer, a triaxial gyroscope and a compass, to demonstrate the performance of proposed algorithm. If made publicly available, this data can serve as a benchmark dataset for activity recognition model research. Interestingly, the proposed algorithm does not improve performance of the classifier in the same location after adaptation (i.e. incremental parameter update of base learner using target domain samples). However, it outperforms the base learner when applied to a new location. This interesting result implies that negative transfer is happening in the adaptation phase (i.e. online update) for the same location and requires further investigation.

3.6. Deep representation learning

Even though deep learning or representation learning is different from transfer learning by definition, they are closely intertwined. As proven time and again, deep learning can capture the structure of the data. This is inherently important for transfer learning as those high level structures can contribute to transfer of knowledge between domains. For this reason, this article discusses the work done by Yu et al. [9] on deep representation learning using extreme learning machine. In this work, the authors of [9] proposed a series of stacked ELM to learn higher level view of inputs. Random shift and linear kernels have been used as stacking element. More specifically, if $\mathbf{X}$ is the input matrix and $\mathbf{T}$ is the label matrix, then the following event of actions take place:

- a weight matrix, $\mathbf{W}$ is generated randomly whose elements are chosen from a normal distribution
- hidden layer output matrix $\mathbf{H}$ is determined as follows: $\mathbf{H} = \mathbf{WX}$. Conventional ELM determines $\mathbf{H}$ using activation function at this step.
- output layer weights are calculated using $\tilde{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$
- output of ELM is calculated as $\mathbf{O} = \mathbf{H}\tilde{\beta}$
- with another randomly generated matrix, called projection matrix $\mathbf{Q}$, calculate $P = \sigma(\mathbf{X} + \alpha.\mathbf{Q}.\mathbf{O})$
- set $\mathbf{X} = \mathbf{P}$
- the above process is continued for the next ELM in the stacked structure
- once all ELM in the stacked structure process the above workflow, the output prediction function is computed as $\mathrm{f}(\mathbf{X}) = \mathbf{X}.\mathbf{W}.\tilde{\beta}$

Here, the kernel function $\sigma(.)$ can be any kernel function e.g. sigmoidal function, radial basis function etc. α is a shift control parameter with respect to the original input data.

It is important to realize that, unlike conventional deep learning which utilizes unsupervised learning in each layer (by setting the input as the target/output), the proposed DrELM algorithm is using the actual labels in each layer as the output. It would be interesting to see the what the stacked ELM network learns in terms of memory (e.g. deep neural networks learns strokes and structures if applied on image) for image processing. Interested readers are encouraged to go through a comprehensive review of different deep neural network architectures with corresponding applications in [36].

3.7. Q learning

Pan et al. [15] proposed a transfer learning set-up to solve Q learning problem. Q learning is a commonly used algorithm in reinforcement learning and normally used to determine behavior selection policies. The authors in [15] proposed a three stage transfer learning framework to estimate the Q values where only a limited amount of data is available. They utilized multiple related data as source domain to enhance the learning. The stages in the proposed algorithms are task space transfer mechanism, sample space transfer mechanism and ELM based Q learning. The first stage determines similarity between tasks based on state similarity and task similarity among source domains and target domain. Task similarity is computed by calculating likelihood ratio of samples between source and target domains. In the second stage, the algorithm select some source domain samples as additional training samples in the target domain based on the probability of belonging to target domain. The proposed algorithm include task similarities while calculating probability of belongingness. This step is put in place to essentially avoid negative transfer. It ensures that no irrelevant samples are being incorporated in the training set. Once this sample transfer stage is complete, ELM is trained based on the already available target domain data and transferred samples. The whole process is continues until a fixed number of iteration is completed. The authors of [15] showed the effectiveness of their algorithm through a simulated case study where a boat tries to cross a river with string current force. The most successful algorithm will be able to choose policy with least time steps and distance measurement. The algorithm proposed by the researchers has two variants i.e. SST-ELMQ and MST-ELMQ. These are essentially two variants of same algorithm but one works with only one source domain while the other works with multiple source domains. The MST-ELMQ algorithm outperforms other algorithms in terms of time steps but a single layer back-propagation based neural network outperform both SST-ELMQ and MST-ELMQ in terms of distance measurement. Thus, it would be interesting to see the success of the proposed algorithm on a few other datasets.

3.8. Transportation mode recognition

Chen et al. [7] proposed an ELM based transfer learning algorithm in the context of transportation mode recognition problem. Essentially, the proposed algorithm, TransELM, is an implementation of online sequential ELM where the incremental samples are selected from the source domain though the use of multi-class trustable intervals of mean and median in source domain. More specifically, the algorithm computes the probability of a particular source sample to belong to a particular class. Afterwards, for each class in the source domain, mean and standard deviation of these probabilities are calculated and stored. At this stage, previously trained ELM (trained on source domain samples) is applied on available target domain samples. Similar to the source domain, probability of the target domain samples belonging to each class is computed and a correspondence to source domain class is decided based on the mean and median of those probabilities. It effectively implies that a particular sample can belong to different classes in source and target domain. However, the TransELM algorithm will merge the samples from source domain to that of target domain based on this correspondence to synthetically generate new samples in the target domain. If there is no strong match for a source domain sample to a particular target domain samples, it will be discarded as noise. This procedure is essentially a sample selection method based on relevancy. Once this step is completed, an online sequential ELM is trained with these new samples to update the parameters of ELM.

The authors have demonstrated the performance of their algorithm on a dataset consisting of six different transportation modes i.e. bicycling, driving, taking light-rail, staying still, taking bus and walking. This dataset is collected using Nokia N82 model which provided both accelerometer and GPS data. These data are collected from volunteers without any restriction to place the mobile in any particular portion of body. The proposed algorithm significantly outperforms traditional ELM in two cases out of six cases. In those 4 cases, the difference is not significant. However, the comparison was done only with traditional ELM which never saw those incremental samples. Therefore, it would be interesting to see comparison with other transfer learning algorithm with the same incremental samples.

3.9. Knowledge transfer in computer vision

Zhang and Zhang [37] proposed a TL algorithm that handles domain adaptation problem in computer vision domain. They proposed an algorithm called ELM based domain adaptation framework called EDA which is also able to generalize to multiple view scenario (i.e. when images of an object from different angles are taken into account). EDA formulates the learning problem by adding three components into the minimization setup of cost function. First components addresses the learning error when trained only with source domain data. The seconds term addresses the learning error of a so called category transformation matrix. The idea behind this category transformation matrix is to minimize the target domain prediction error by minimizing the difference of weighted truth and predicted labels. Interestingly, the authors in [37] multiplied the truth values instead of the predicted values. They also enforced a condition to keep this transformation matrix close to unity. Essentially, this second term performs a label adaption functionality in the knowledge transfer process. The third term attempts to minimize the difference of two base classifiers trained on source data. One of these classifiers must be an ELM classifier whereas the other one can be any traditional non-ELM learner (e.g. SVM, ANN, KNN etc.).

The researchers compared the proposed algorithm on 4 different datasets, namely YouTube and Consumer videos, 3DA Office dataset, 4DA Extended Office dataset and Bing-Caltech dataset. They have shown that proposed algorithms considerably outperforms other baseline methods on all dataset considered. Their experimental framework is satisfactory in the sense that they report mean prediction accuracy and standard errors from multiple runs. However, their experiment lacks the presence of statistical significance test.

In another work, Zhang et al. [38] proposed a reconstruction themed algorithm called Latent Spare Domain Transfer (LSDT) to tackle knowledge adaptation problem in computer vision set-up. The proposed LSDT algorithm learns a latent space between source and target domain by utilizing a few samples from source domain. The number of source domain samples are limited in order to avoid overfitting issues. In this process, a spare reconstruction matrix is learned simultaneously along with a latent space projection matrix. Afterwards, combined source and target domain data is used in a spare subspace clustering method, proposed in [38], to reconstruct target domain. Finally, a learner is used to learn the input–output mapping. The authors in [38] demonstrated the success of the proposed algorithm on multiple benchmark datasets and compared their performance with several baseline algorithms. They have shown that, in most cases, the proposed LSDT algorithm outperforms other baseline methods.

4. Comparative performance of ELM based TL algorithms

In this section, results of different ELM based TL algorithms are presented on multiple datasets. ELM based TL algorithms are used in many different domains of engineering applications and therefore, it is logical for these papers not to demonstrate their success on similar benchmark datasets. Since not all of the discussed articles measured their performance on same datasets, it is difficult to compare the performance of these algorithms. In addition, most papers differ from each other in experimental details, making it impossible to compare performance of an individual ELM-TR algorithm against other ELM-TR algorithms. However, this does not undermine the individual success of each of these paper. To demonstrate this, the best performing baseline methods are compared to the best performing ELM based TL method from the discussed papers on multiple benchmark datasets. The comparison is shown in Table 2. The first two datasets are common benchmark datasets used in many machine learning tasks while the last three datasets are specifically used in transfer learning. Note from Table 2 that, in most cases, a variant of ELM based TL algorithm outperforms the baseline method.

5. Recommendations for future research

While there is an increasing interest to use ELM in the context of transfer learning, there is a lack of completeness in the published articles. The articles that are already published validates the effectiveness of ELM to solve transfer learning problem. However, these articles failed to compare their performance against existing transfer learning algorithms/frameworks. It is recommended that future research compares the performance of proposed algorithm with existing ELM based transfer learning algorithms as well as established transfer learning frameworks e.g. TrAdaBoost [35]. Another important aspect of the experiment is stability analysis of

Table 2
Comparison of ELM based TL algorithms on different datasets.

Dataset	Best non-TL learner		ELM base learner		Reference
	Name	Accuracy	Name	Accuracy	
Diabetes	DBN	0.7805	DrELM^r	0.7822	[9]
	K-Means	0.8901	TFSR-ELM	0.9423	[13]
	TrAdaBoost	0.8725	TL-ELM	0.9624	[8]
MNIST	K-Means	0.6588	TFSR-ELM	0.847	[13]
	GFK	0.826	NLSDT	0.874	[38]
	SAE	0.9675	DrELM^r	0.9538	[9]
E-Nose multi-device	ELM	0.7031	TCTL+SEMI	0.9375	[12]
	ML-comgfk	0.6728	DAELM-T(50)	0.9186	[10]
3DA (Amazon-800)	SHFA	0.566	EDA_{SVM}	0.623	[37]
3DA (Webcam-800)	SHFA	0.559	EDA_{SVM}	0.625	[37]
Yale	LS-DAKSVM	0.7589	TL-DAKELM	0.8962	[8]

proposed algorithm. Most of the existing ELM-TL literature failed to report results from repeated experiments. Repeated experiment is crucial for ELM based algorithm as the input connection weights are chosen randomly in ELM algorithm. Finally, there is a tendency of using seemingly unrelated datasets as source domain in existing ELM-TL literature. It is important to establish the relevance of source domain with target domain. Without this step, negative transfer is more likely to occur. A summary of the recommendations is given below:

- Compare the result of proposed algorithm with base ELM and ELM based transfer learning algorithms.
- Compare the performance of proposed algorithms with at least one non-ELM based transfer learning algorithm e.g. TrAdaBoost.
- Use Monte-Carlo simulation method for stability analysis, if theoretical analysis is not possible. Report the result of proposed algorithm using either a box-plot, violin plot or mean and standard deviation from multiple runs.
- Provide reasoning to select a certain dataset as source domain. Following this recommendation will help avoiding negative transfer.

6. Conclusion

This paper discusses current state of extreme learning based transfer learning algorithms. It provides detailed discussion of each of the algorithms in simple terms while using common terminology for all reviewed articles. This particular aspect of this paper facilitates easy understanding of ELM based TL algorithm in different fields of study. Therefore, this paper can be used to understand basic elements of ELM based TL algorithms and will be particularly useful for researchers who are new in this field. Hence, this paper may serve as the starting point to begin search for gaps in current literature. In addition, this paper recommends some crucial steps for experimentation in future research. These recommendations are designed to ensure that future research works in ELM based transfer learning algorithms are credible, stable and reproducible.

Acknowledgment

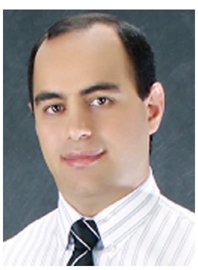
This research has been supported by Australian Government Research Training Program Scholarship.

References

[1] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Eng. 22 (10) (2010) 1345–1359.
[2] L. Zhang, D. Zhang, Evolutionary cost-sensitive extreme learning machine, IEEE Trans. Neural Netw. Learn. Syst. PP (99) (2016) 1–16, doi:10.1109/TNNLS.2016.2607757.
[3] S. Xu, J. Wang, A fast incremental extreme learning machine algorithm for data streams classification, Expert Syst. Appl. 65 (2016) 332–344.
[4] X. Jiang, J. Liu, Y. Chen, D. Liu, Y. Gu, Z. Chen, Feature adaptive online sequential extreme learning machine for lifelong indoor localization, Neural Comput. Appl. 27 (1) (2016) 215–225.
[5] W.-Y. Deng, Q.-H. Zheng, Z.-M. Wang, Cross-person activity recognition using reduced kernel extreme learning machine, Neural Netw. 53 (2014) 1–7.
[6] Y. Chen, Z. Zhao, S. Wang, Z. Chen, Extreme learning machine-based device displacement free activity recognition model, Soft Comput. 16 (9) (2012) 1617–1625.
[7] Z. Chen, S. Wang, Z. Shen, Y. Chen, Z. Zhao, Online sequential ELM based transfer learning for transportation mode recognition, in: Proceedings of the 2013 IEEE Conference on Cybernetics and Intelligent Systems (CIS), IEEE, 2013, pp. 78–83.
[8] X. Li, W. Mao, W. Jiang, Extreme learning machine based transfer learning for data classification, Neurocomputing 174 (2016) 203–210.
[9] W. Yu, F. Zhuang, Q. He, Z. Shi, Learning deep representations via extreme learning machines, Neurocomputing 149 (2015) 308–315.
[10] L. Zhang, D. Zhang, Domain adaptation extreme learning machines for drift compensation in E-Nose systems, IEEE Trans. Instrum. Meas. 64 (7) (2015a) 1790–1801.
[11] L. Zhang, D. Zhang, Domain adaptation transfer extreme learning machines, in: Proceedings of the ELM-2014, 1, Springer, 2015b, pp. 103–119.
[12] K. Yan, D. Zhang, Calibration transfer and drift compensation of E-Noses via coupled task learning, Sens. Actuators B Chem. 225 (2016) 288–297.
[13] X. Li, W. Mao, W. Jiang, Y. Yao, Extreme learning machine via free sparse transfer representation optimization, Memet. Comput. 8 (2) (2016) 85–95.
[14] Y. Zhou, J. Lian, M. Han, Remote sensing image transfer classification based on weighted extreme learning machine, IEEE Geosci. Remote Sens. Lett. 13 (10) (2016) 1405–1409.
[15] J. Pan, X. Wang, Y. Cheng, G. Cao, Multi-source transfer elm-based q learning, Neurocomputing 137 (2014) 57–64.
[16] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, 2, IEEE, 2004, pp. 985–990.
[17] D. Serre, Matrices: theory and applications, Vol. 216, Springer, New York, 2002. Print URL for the reference: https://link.springer.com/book/10.1007%2F978-1-4419-7683-3.
[18] G.-B. Huang, L. Chen, C.K. Siew, et al., Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[19] G.-B. Huang, Q.-Y. Zhu, C.K. Siew, Real-time learning capability of neural networks, IEEE Trans. Neural Netw. 17 (4) (2006) 863–878.
[20] G.-B. Huang, Learning capability and storage capacity of two-hidden-layer feedforward networks, IEEE Trans. Neural Netw. 14 (2) (2003) 274–281.
[21] P.L. Bartlett, For valid generalization, the size of the weights is more important than the size, Adv. Neural Inf. Process. Syst. (NIPS) 9 (1997) 134.
[22] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 42 (2) (2012) 513–529.
[23] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[24] G.-B. Huang, Z. Bai, L.L.C. Kasun, C.M. Vong, Local receptive fields based extreme learning machine, IEEE Comput. Intell. Mag. 10 (2) (2015) 18–29.
[25] G.-B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2) (2011) 107–122.
[26] J. Tang, C. Deng, G.-B. Huang, Extreme learning machine for multilayer perceptron, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2016) 809–821.
[27] L.L.C. Kasun, H. Zhou, G.-B. Huang, C.M. Vong, Representational learning with extreme learning machine for big data, IEEE Intell. Syst. 28 (6) (2013) 31–34.
[28] R. Zhang, Y. Lan, G.-b. Huang, Z.-B. Xu, Universal approximation of extreme learning machine with adaptive growth of hidden nodes, IEEE Trans. Neural Netw. Learn. Syst. 23 (2) (2012) 365–371.
[29] Y. Lan, Y.C. Soh, G.-B. Huang, Constructive hidden nodes selection of extreme learning machine for regression, Neurocomputing 73 (16) (2010) 3191–3199.
[30] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1) (2010) 155–163.

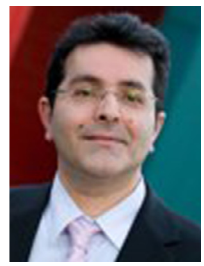
[31] C.K.L. Lekamalage, T. Liu, Y. Yang, Z. Lin, G.-B. Huang, Extreme learning machine for clustering, in: Proceedings of the ELM-2014, 28, Springer, 2015, pp. 435–444.
[32] G. Huang, S. Song, J.N. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405–2417.
[33] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006) 1411–1423.
[34] W. Deng, Q. Zheng, K. Zhang, Reduced kernel extreme learning machine, in: Proceedings of the Eighth International Conference on Computer Recognition Systems (CORES 2013), Springer, 2013, pp. 63–69.
[35] W. Dai, Q. Yang, G.-R. Xue, Y. Yu, Boosting for transfer learning, in: Proceedings of the Twenty-fourth International Conference on Machine Learning, ACM, 2007, pp. 193–200.
[36] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[37] L. Zhang, D. Zhang, Robust visual knowledge transfer via extreme learning machine-based domain adaptation, IEEE Trans. Image Process. 25 (10) (2016) 4959–4973.
[38] L. Zhang, W. Zuo, D. Zhang, LSDT: Latent sparse domain transfer learning for visual adaptation, IEEE Trans. Image Process. 25 (3) (2016) 1177–1191.

Syed Moshfeq Salaken has finished his B.Sc. in Electrical and Electronic Engineering in 2011 from Bangladesh University of Engineering and Technology, Bangladesh. He is currently pursuing a Ph.D. degree at Deakin University, Australia. His research interest include machine learning, fuzzy systems and data science.



Abbas Khosravi received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2002, the M.Sc. degree in electrical engineering from Amirkabir University of Technology, Tehran, Iran, in 2005, and the Ph.D. degree from Deakin University, Geelong, VIC, Australia, in 2010. Currently, he is a research fellow in the Institute of Intelligent Systems Research and Innovation (IISRI) at Deakin University. His primary research interests include development and application of artificial intelligence techniques for (meta)modeling, analysis, control, and optimization of operations within complex systems. He is a recipient of an Alfred Deakin Postdoctoral Research Fellowship in 2011.

Thanh Nguyen received the Ph.D. degree from Monash University, VIC, Australia, in 2013. Currently, he is a research fellow in the Institute of Intelligent Systems Research and Innovation (IISRI) at Deakin University. His primary research interests include bioinformatics, artificial intelligence and soft computing. He is a recipient of an Alfred Deakin Postdoctoral Research Fellowship in 2016.



Saeid Nahavandi received the B.Sc. (hons.), M.Sc., and Ph.D. degrees in automation and control from Durham University, Durham, U.K. He is the Alfred Deakin Professor, Chair of Engineering, and the director for the Institute of Intelligent Systems Research and Innovation (IISRI), Deakin University, Geelong, VIC, Australia. He has published over 350 peer reviewed papers in various international journals and conferences. He designed the worlds first 3-D interactive surface/motion controller. His research interests include modeling of complex systems, simulation-based optimization, robotics, haptics, and augmented reality. He is the Associate Editor of the IEEE SYSTEMS JOURNAL, an Editorial Consultant Board member for the International Journal of Advanced Robotic Systems, and an Editor (South Pacific Region) of the International Journal of Intelligent Automation and Soft Computing. He is a Fellow of Engineers Australia (FIEAust) and IET (FIET).

Please cite this article as: S.M. Salaken et al., Extreme learning machine based transfer learning algorithms: A survey, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.06.037