AUC Based Extreme Learning Machines for Supervised and Semi-supervised Imbalanced Classification

Guanjin Wang, *Member, IEEE,* Kok Wai Wong, *Senior Member, IEEE* and Jie Lu, *Fellow, IEEE*

***Abstract*—Extreme learning machines (ELM) has been theoretically and experimentally proved to achieve promising performance at a fast learning speed for supervised classification tasks. However, it does not perform well on imbalanced binary classification tasks and tends to get biased towards the majority class. Besides, since a large amount of training data with labels are not always available in the real world, there is an urgent demand to develop an efficient semi-supervised version of ELM for imbalanced binary classification tasks. In this paper, owing to the distinct insensitivity of Area Under the ROC curve (AUC) to both class skews and changes of class distributions, we focus the study on integrating AUC maximization into the ELM framework to tackle with imbalanced binary classification tasks well. By demystifying the AUC metric with the ELM framework, we develop a new AUC based ELM called AUC-ELM for imbalanced binary classification, which essentially is revealed to be equivalent to an ELM on another transformed data space. Accordingly, its semi-supervised version called SAUC-ELM is also developed. Both AUC-ELM and SAUC-ELM have the distinctive merits: (1) they share the advantage of ELM in both generalization capability and training efficiency, and further uniquely tailored for imbalanced binary classification tasks; (2) In contrast to the existing imbalanced variants of ELM such as Class-specific Cost Regulation ELM and Semi-supervised ELM, they have fewer parameters to tune, thereby reducing the computational cost for model selection. Experiments on a heap of datasets show that both AUC-ELM and SAUC-ELM outperform the other comparative methods in terms of both classification performance and training speed.**

***Index Terms*—Extreme learning machine, imbalance learning, semi-supervised learning, AUC optimization**

I. Introduction

In the last few decades, extensive studies on data classification techniques [1], [2], [3], [4], [5], [6], [7], [8] have been carried out. Among them, single layer feedforward networks have attracted our attentions due to their approximation capability. The most popular learning algorithm to train single layer feedforward networks is back-propagation method [9], which uses gradient descent update rule to optimize the weights in the network. However, such a method may encounter stopping criteria, learning rate, learning epochs, and local minima issues. Other techniques such as generic and evolutionary algorithms [3], [4] have also been used to provide the global optimal solution, and yet they are still imperfect due to high computational cost. Support vector machine (SVM) [5] is another well-known training algorithm for single layer feedforward networks using a maximum margin classifier derived from a framework of structural risk minimization. SVM essentially is to solve a quadratic programming problem, which is more convenient to handle. In contrast to these methods, Huang et al. proposed a new batch learning algorithm - extreme learning machine (ELM) [6] for single layer feedforward networks. It only needs to update the output weights between the hidden and output layers by solving a regularized least squares (or ridge regression) problem, while the rest of parameters, such as the input weights and bias between input and hidden layers can be randomly assigned. ELM is theoretically and experimentally proved to have a comparable or better generalization capability with a fast learning speed compared to other popular learning algorithms in most cases [7], [10], [11], [12], [13], [14], [15].

However, ELM itself is not explicitly designed to overcome a common challenge - class imbalance, in which there is a much larger number of samples belonging to one class compared to another within a dataset [16]. For example, cancer diagnosis is a public domain to face skewed datasets. It is common to have a large number of normal cases and very few cancer cases. Most traditional machine learning methods such as ELM are directly trained on the imbalanced datasets and tend to be overwhelmed by the majority class, i.e., it may achieve very high accuracy on the majority class by compromising accuracy on the minority class, thus leading to poor classification performance. In the literature, two main approaches are proposed to resolve this problem, i.e., the re-sampling approach and algorithmic modification approach [17].

The re-sampling approach is designed to balance the class distribution by either removing some majority class samples (undersampling) [18] or adding some minority class samples (oversampling) [19]. Its main advantage is that it is independent of classifier construction and can work readily with most classifiers. In undersampling, samples can be removed randomly or based on specific criteria, e.g., removing samples far away from boundaries between two classes. In oversampling, samples can be randomly duplicated, or samples which are close to the boundaries are selected to duplicate. However, the former method may encounter the problem of information loss while the latter may suffer from the overfitting issue. In the algorithmic modification approach, cost-sensitive learning [20] is frequently used to deal with imbalanced datasets, which assigns higher misclassification cost or higher weight

G. Wang and K. Wong are with the Discipline of Information technology, Mathematics & Statistics, College of Science, Health, Engineering and Education, Murdoch University, WA, Australia (e-mail: Guanjin.Wang@murdoch.edu.au, K.Wong@murdoch.edu.au).

J. Lu is with the Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia (e-mail: Jie.Lu@uts.edu.au).

to the minority class samples. For example, for specific cancer detection, the misclassification cost of a cancer case is set to be higher than the normal case, so that the bias of the majority class is shifted towards the more significant minority class. By utilizing this strategy, some variants of ELM such as Weighted ELM [21], boosting weighted ELM [22], ensemble weighted ELM [23] and CCR-ELM [24] have been proposed to handle class imbalance problems. However, for most applications, we lack the prior information of the cost distribution and thus, how to determine the cost matrix remains a challenge. On the other hand, most commonly used classifiers are built by optimizing an objective function that is related to accuracy or error rate. These performance metrics could be misleading for imbalanced classification. To overcome this issue, the Area Under the ROC curve (AUC) [25] is alternatively used to evaluate the models due to its robustness against class skews. An early study [26] pointed out that the large variances of AUC for imbalanced datasets suggest that there may be significantly different AUC values for two classifiers sharing similar accuracy. Such a finding posed an encouraging sign for comprehensive studies on straight-forward AUC optimization. In other words, when AUC maximization is the target, a classifier that is designed to optimize AUC directly can have a significant advantage over class imbalance problems [27], [28]. In the past decades, AUC maximizing versions of various learning algorithms have been developed which in fact lead to higher AUC values supported by empirical evidences [27], [29], [30], [31], [32]. However, these studies focus on supervised learning, where learning algorithms are trained using labeled samples only. In practice, it is very challenging to collect a large number of samples with labels, and the intensive manual labeling work is often required. Rather, it is easy and cheap to collect unlabeled samples. To overcome the shortage of supervised learning, semi-supervised learning [33] for improving the classification performance that makes use of both labeled and unlabeled data for training has been recently developed. The relevant studies on semi-supervised ELM can be found in [34], [35], [36], [37], [38], [39].

As a result, it is necessary to bridge the advantage of ELM algorithm and AUC optimization to achieve both supervised and semi-supervised imbalanced learning. In this study, we propose a supervised ELM based method AUC-ELM by combining AUC optimization with the ELM framework, and then further extend it into its semi-supervised version SAUC-ELM. We expect that the proposed algorithms will be effective to incorporate labeled and unlabeled samples into the training process and provide high generalization performance on imbalanced datasets. The contributions of this work can be summarized as follows.

1) AUC-ELM is proposed to mainly solve class imbalance problems by integrating AUC metric optimization into the ELM framework. Based on the output expression of ELM, we demystify the AUC metric into its geometrical interpretation from a new perspective. AUC-ELM is theoretically derived to be equivalent to the corresponding ELM on the transformed input space, and hence inherit the traditional ELM's excellent generalization capability and training efficiency.
2) SAUC-ELM is a semi-supervised version of AUC-ELM. To best of our knowledge, by demystifying the proposed semi-supervised AUC metric, this is the first attempt to solve semi-supervised imbalanced learning problems by simply incorporating AUC optimization into the semi-supervised ELM framework.
3) Different from the existing ELM based imbalanced learning methods such as CCR-ELM and SS-ELM, AUC-ELM and SAUC-ELM have fewer parameters to tune, thereby reducing the workload for the model selection procedure.
4) Experimental results on benchmark imbalanced binary datasets demonstrate that both AUC-ELM and SAUC-ELM achieve promising classification performance and fast training speed, in contrast to the comparative methods.

This paper is organized as follows. Section II outlines the related work on ELM for balanced, imbalanced, and semi-supervised binary classification. The proposed methods AUC-ELM and its semi-supervised version SAUC-ELM are presented in Section III and Section IV, respectively. Experimental results are reported and analyzed in Section V. Section VI ends this paper with a conclusion and future work.

II. Related work

A vast amount of experimental evidences have revealed that ELM is effective for both classification and regression tasks. Here, we only review ELM for binary classification tasks to keep our study focus. Suppose we have a training set with N samples for a supervised binary classification problem, i.e., $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, $\mathbf{Y} = \{y_i\}_{i=1}^N$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$. ELM aims at learning a discriminant rule based on the training data.

Generally speaking, ELM contains the input layer, hidden layer and output layer. In particular, we can randomly assign hidden neurons using any nonlinear piecewise continuous functions, such as the Gaussian function

$$g(\boldsymbol{x}; \boldsymbol{\alpha}, b) = \exp(-\frac{||\boldsymbol{x} - \boldsymbol{\alpha}||^2}{b^2})$$

and Sigmoid function

$$g(\boldsymbol{x}; \boldsymbol{w}, b) = \frac{1}{1 + \exp(-(\boldsymbol{w}^T \boldsymbol{x} + b))}$$

where $(\boldsymbol{\alpha}, b)$ or $(\boldsymbol{w}, b)$ are the parameters of the corresponding mapping functions, respectively, and $||\cdot||$ represents the Euclidean norm.

According to ELM's theory in [7], all the parameters in the mapping functions of the hidden layer can be randomly generated in terms of any continuous probability distribution e.g., the uniform or Gaussian distribution on $(-1, 1)$. In other words, different from the classical feedforward neural networks and SVM, the training of ELM only deals with how to tune the output weights between the hidden layer and output layer for binary classification tasks, which actually means that the training of ELM is equivalent to solving a regularized least squares regression problem. Therefore, training ELM has been

theoretically and empirically proved to be more efficient than training SVM or learning with back-propagation.

Assume there is n_k hidden neurons in the hidden layers, which are randomly assigned to project the data from the input space onto a n_k-dimensional feature space. We denote by $\boldsymbol{h}(\boldsymbol{x}_i) \in \mathbb{R}^{1\times n_h}$ the output vector of the hidden layer with respect to $\boldsymbol{x}_i$, and $\boldsymbol{\beta} \in \mathbb{R}^{n_h\times 1}$ the output weight vector that connect the hidden layer with the output layer. As a result, ELM has its output as follows:

$$f(\boldsymbol{x}_i) = \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta}, \quad i = 1, \cdots, N. \tag{1}$$

in which $\boldsymbol{\beta}$ is obtained by minimizing the sum of the training errors. That is to say, ELM attempts to solve the following formulation

$$\begin{aligned} \min_{\boldsymbol{\beta}\in\mathbb{R}^{n_h\times 1}} \quad & \frac{1}{2}||\boldsymbol{\beta}||^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t.} \quad & \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta} = y_i - \xi_i, \quad i = 1, \cdots, N \end{aligned} \tag{2}$$

where the first term is a regularization term to prevent overfitting issues. ξ_i is the training error regarding the i-th sample and C is a penalty parameter.

By substituting the constraints into the objective function, we obtain the following equivalent unconstrained optimization problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{n_h\times 1}} L_{ELM} = \frac{1}{2}||\boldsymbol{\beta}||^2 + \frac{C}{2}||\boldsymbol{Y} - \mathbf{H}\boldsymbol{\beta}||^2 \tag{3}$$

where $\mathbf{H} = [\boldsymbol{h}(\boldsymbol{x}_1)^T, \cdots, \boldsymbol{h}(\boldsymbol{x}_N)^T]^T \in \mathbb{R}^{N\times n_h}$ and $\boldsymbol{Y} = [y_1, y_2, \cdots, y_N]^T$.

The above problem is known as the ridge regression or regularized least squares regression, and its solution is determined in [6], [7], [10], which can be represented as below:

$$\boldsymbol{\beta}^* = \big(\mathbf{H}^T\mathbf{H} + \frac{\mathbf{I}_{n_h}}{C}\big)^{-1}\mathbf{H}^T\boldsymbol{Y}, \quad N > n_h \tag{4}$$

or equivalently

$$\boldsymbol{\beta}^* = \mathbf{H}^T\big(\mathbf{H}\mathbf{H}^T + \frac{\mathbf{I}_N}{C}\big)^{-1}\boldsymbol{Y}, \quad N < n_h \tag{5}$$

Ever since ELM was proposed in [6], various studies have substantially contributed to ELM's theories, variants and applications. A survey about ELM can be found in [10]. At present, the ELM techniques have been developed into deep ELMs for massive data [40], [41], [42], on-line learning [43], [44] and so on. Since our study focuses on imbalanced and semi-supervised learning of ELM, we briefly review them as follows.

Firstly, class imbalance is a common challenge in practice such as cancer detection [45], computer vision [46] and fraud detection [47], where one class has a much larger number of training samples than the other class. The problem associated with class imbalance learning is that traditional learning models tend to be partial to the majority class and the overall performance is deteriorated. To adapt traditional ELM to handle the class imbalance problem effectively, some variants of ELM like Weighted ELM (WELM), Boosting weighted ELM [22], Regularized Weighted Circular Complex valued ELM [48], Ensemble weighted ELM [23], Class-specific Cost Regulation ELM (CCR-ELM) [24] and Class-specific ELM [49] were proposed. When these ELM's variants run on training samples, their performances heavily depend on the weights. However, how to find better weights is still a challenging problem. What is more, the tuning of too many weights may be time consuming and even impracticable. For example, CCR-ELM [24] uses class-specific cost regularization to combat imbalanced data, which has to tune two regularization parameters by grid search on (2^{-24}, 2^{-23},$\cdots$,2^{24},2^{25}) to find optimal values. That would be 2500 different combinations of regularization parameters to run on the training dataset which is computationally intensive.

Secondly, most existing studies on ELM are primarily used for supervised learning. In reality, however, it is much more difficult to acquire sufficient labeled samples than the unlabeled ones. In such a dilemma, the performance of supervised learning algorithms may sacrifice as useful information hidden in the unlabeled data are not used. To overcome the disadvantage of supervised ELMs, ELM is introduced to the framework of semi-supervised learning to expand its applicability further. The manifold regularization (MR) [50] is widely used in the area of semi-supervised learning, which attempts to extract the geometric information from both labeled and unlabeled data and make the smoothness of classifiers along the intrinsic manifold by adding a regularization term. Following MR framework or deep learning, SS-ELM [35] and other variants [34], [36], [37], [38], [39], [51], [52] about semi-supervised learning of ELM were well developed. However, none of these methods targeted imbalanced data. Moreover, how to determine the graph Laplacian matrix for MR gained from both labeled and unlabeled data in advance is not a trival work in practice.

III. Supervised AUC-ELM for imbalanced binary classification

In this section, a supervised AUC-ELM is proposed for imbalanced binary classification. AUC is a better evaluation metric than accuracy for class imbalance learning due to its invariance to the class ratios and therefore it is frequent to use. Intuitively, $AUC(s)$ reflects the probability in the sense of the mathematical expectation that the scoring of the majority sample $\boldsymbol{x}^+$ is greater than that of the minority sample $\boldsymbol{x}^-$, if $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ both are randomly sampled from the majority and minority classes. According to [25], [53], $AUC(s)$ can be expressed as

$$AUC(s) = \mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x}^-)) \tag{6}$$

where $\mathcal{D}^+$ and $\mathcal{D}^-$ denote the distributions of the majority and minority classes, respectively. s is a scoring function, for example, $s(\boldsymbol{x}) = h(\boldsymbol{x})\boldsymbol{\beta}$. $\mathbb{E}$ is the corresponding mathematical expectation. $\mathbf{1}(\cdot)$ is the indicator function which returns value 1 if the condition is satisfied; 0 otherwise.

In practice, because $\mathbf{1}(\cdot)$ is not continuous, we can approximately maximize Eq. (6) by replacing the indicator function $\mathbf{1}(\cdot)$ using a convex and continuous surrogate function. In this study, we prefer $\Big(1 - \big(s(\boldsymbol{x}^+) - s(\boldsymbol{x}^-)\big)\Big)^2$ as the surrogate function. In particular, when $s(\boldsymbol{x}) = h(\boldsymbol{x})\boldsymbol{\beta}$ in ELM, we

may view $\Big(1-\big(h(\boldsymbol{x}^+)-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)$ as a random variable z. According to the well-known mathematical formula: $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}(z^2)=\big(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}(z)\big)^2+\mathbb{E}_{z\sim\mathcal{D}}\Big(\big(z-\mathbb{E}_{z\sim\mathcal{D}}(z)\big)^2\Big)$, we have

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]=\Big(1-\big(h(\boldsymbol{x}^+)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2\\&+\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)-\Big(1-\big(h(\boldsymbol{x}^+)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2\Big]\\&=\Big(1-\big(h(\boldsymbol{x}^+)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2+\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(\big(h(\boldsymbol{x}^+)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2\Big]\end{aligned}\tag{7}$$

Similarly, we have

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]\\&=\big(1-\big(\boldsymbol{c}^+-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\big)^2+\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\Big[\Big(\big(\boldsymbol{c}^+-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]\end{aligned}\tag{8}$$

where $\boldsymbol{c}^+=\mathbb{E}_{\boldsymbol{x}^+\sim\boldsymbol{D}^+}\big(h(\boldsymbol{x}^+)\big)$, $\boldsymbol{c}^-=\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\big(h(\boldsymbol{x}^-)\big)$. Therefore we have

$$\begin{aligned}AUC(s)&=\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(s(\boldsymbol{x}^+)-s(\boldsymbol{x}^-)\big)^2\Big]\\&=\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]\\&=\frac{1}{2}\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\Big\{\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]\Big\}\\&+\frac{1}{2}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big\{\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]\Big\}\\&=\frac{1}{2}\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2\\&+\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(h(\boldsymbol{x}^-)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2\Big]\Big]\\&+\frac{1}{2}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(\boldsymbol{c}^+-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\\&+\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\Big[\Big(\big(\boldsymbol{c}^+-h(\boldsymbol{x}^+)\big)\boldsymbol{\beta}\Big)^2\Big]\Big]\end{aligned}\tag{9}$$

where $\boldsymbol{c}^+$ and $\boldsymbol{c}^-$ remain the same expressions in Eq. (8). As $\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\big[\big((h(\boldsymbol{x}^-)-\boldsymbol{c}^-)\boldsymbol{\beta}\big)^2\big]=\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\big[\big((h(\boldsymbol{x}^-)-\boldsymbol{c}^-)\boldsymbol{\beta}\big)^2\big]$, $\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\big[\Big(\big(\boldsymbol{c}^+-h(\boldsymbol{x}^+)\big)\boldsymbol{\beta}\Big)^2\big]=\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\big[\big((\boldsymbol{c}^+-h(\boldsymbol{x}^+))\boldsymbol{\beta}\big)^2\big]$, we have

$$\begin{aligned}AUC(s)=&\frac{1}{2}\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\Big[\Big(\big(\boldsymbol{c}^+-h(\boldsymbol{x}^+)\big)\boldsymbol{\beta}\Big)^2\Big]+\frac{1}{2}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(\big(h(\boldsymbol{x}^-)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2\Big]\\&+\frac{1}{2}\mathbb{E}_{\boldsymbol{x}+\sim\mathcal{D}^+}\Big[\Big(\big(1-\big(h(\boldsymbol{x}^+)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2\Big]+\frac{1}{2}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(\boldsymbol{c}^+-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]\end{aligned}\tag{10}$$

According to Eq. (6), $AUC(s)$ involves pair-wise samples from two classes, so it is quadratic in the number of training samples. However, according to Eq. (10), after $\boldsymbol{c}^+$ and $\boldsymbol{c}^-$ are computed in advance, $AUC(s)$ becomes linearly dependent on the number of training samples, which will greatly reduce the computational burden of the AUC(s) optimization.

What is more, $AUC(s)$ in Eq. (10) has clear geometrical interpretation. Its first two terms represent the variances of the outputs of ELM for two classes, respectively. Its third and fourth terms represent the surrogate losses of the outputs of ELM for each class with respect to the mean output of ELM for the other class. Therefore, after demystifying AUC(s) in this way, the rationale of AUC(s) becomes very intuitive and interpretable.

Based on Eq. (10), we may take the following empirical formulation to train the classification model $s(\boldsymbol{x})=h(\boldsymbol{x})\boldsymbol{\beta}$ from the **:**

$$\begin{aligned}\overline{AUC}(s)=&\frac{1}{N^+}\sum_{i=1}^{N^+}\Big(\big(\boldsymbol{c}^+-h(\boldsymbol{x}_i^+)\big)\boldsymbol{\beta}\Big)^2+\frac{1}{N^-}\sum_{j=1}^{N^-}\Big(\big(h(\boldsymbol{x}_j^-)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2\\&+\frac{1}{2}\sum_{i=1}^{N^+}\Big(\big(1-(h(\boldsymbol{x}_i^+)-\boldsymbol{c}^-\big)\boldsymbol{\beta}\Big)^2+\frac{1}{2}\sum_{j=1}^{N^-}\Big(\big(1-(\boldsymbol{c}^+-h(\boldsymbol{x}_j^-)\big)\boldsymbol{\beta}\Big)^2\end{aligned}\tag{11}$$

We can readily verify that $\mathbb{E}\big(\overline{AUC}(s)\big)=AUC(s)$.

Likewise in [27], [28], [29], [30], [31], [32] where $AUC(s)$ is directly adopted in their objective functions, we apply the above $\overline{AUC}(s)$ to formulate the following objective function to design a ELM based classifier for imbalanced learning

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{n_h\times 1}}\quad\frac{1}{2}\boldsymbol{\beta}^2+\gamma\overline{AUC}(s)\tag{12}$$

where γ is the given trade-off parameter between the first and second terms in Eq. (12). After substituting Eq. (11) into Eq. (12), we have

$$\begin{aligned}\min_{\boldsymbol{\beta}\in\mathbb{R}^{n_h\times 1}}J_{AUC-ELM}=&\boldsymbol{\beta}^T\Big(\frac{1}{2}\mathbf{I}+\frac{\gamma}{2N^+}\sum_{i=1}^{N^+}\big(\boldsymbol{c}^+-h(\boldsymbol{x}_i^+)\big)^T\big(\boldsymbol{c}^+-h(\boldsymbol{x}^+)\big)\\&+\frac{\gamma}{2N^-}\sum_{j=1}^{N^-}\big(h(\boldsymbol{x}_j^-)-\boldsymbol{c}^-\big)^T\big(h(\boldsymbol{x}_j^-)-\boldsymbol{c}^-\big)\Big)\boldsymbol{\beta}\\&+\frac{\gamma}{2N^+}\sum_{i=1}^{N^+}\Big(\big(1-\big(h(\boldsymbol{x}_i^+)-\boldsymbol{c}^-\big)\Big)\boldsymbol{\beta}\Big)^2\\&+\frac{\gamma}{2N^-}\sum_{j=1}^{N^-}\Big(1-\big(\boldsymbol{c}^+-h(\boldsymbol{x}_j^-)\big)\boldsymbol{\beta}\Big)^2\end{aligned}\tag{13}$$

Obviously, the first term implies more generalization capability than the classical ELM, while the second and third terms represent two loss functions which are only dependent on N^+ and N^-, respectively. From this perspective, Eq. (12) or equivalently Eq. (13) is a special ELM. We call it AUC-ELM.

After denoting $\mathbf{A}=\mathbf{I}+\frac{\gamma}{N^-}\sum_{j=1}^{N^-}\big(h(\boldsymbol{x}_j^-)-\boldsymbol{c}^-\big)^T\big(h(\boldsymbol{x}_j^-)-\boldsymbol{c}^-\big)+\frac{\gamma}{N^+}\sum_{i=1}^{N^+}\big(\boldsymbol{c}^+-h(\boldsymbol{x}_i^+)\big)^T\big(\boldsymbol{c}^+-h(\boldsymbol{x}_i^+)\big)$ and using the following mathematical transformation

$$(h^{'}(\boldsymbol{x}_i),y_i^{'})=\begin{cases}\Big(\sqrt{\frac{1}{2p}}\big(h(\boldsymbol{x}_i)-\boldsymbol{c}^-\big),\sqrt{\frac{1}{2p}}\Big), & y_i=+1\\ \Big(\sqrt{\frac{1}{2(1-p)}}\big(h(\boldsymbol{x}_i)-\boldsymbol{c}^+\big),-\sqrt{\frac{1}{2(1-p)}}\Big), & y_i=-1\end{cases}\tag{14}$$

where $p=\frac{N^+}{N^++N^-}=\frac{N^+}{N}$, Eq. (12) becomes

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{n_h\times 1}}J_{AUC-ELM}=\quad\frac{1}{2}\boldsymbol{\beta}^T\mathbf{A}\boldsymbol{\beta}+\frac{\gamma}{2N}\sum_{i=1}^{N}(y_i^{'}-h^{'}(\boldsymbol{x}_i)\boldsymbol{\beta})^2\tag{15}$$

Referring to the classical ELM in Eq. (3), we readily have its solution

$$\boldsymbol{\beta}=\begin{cases}(\frac{N}{\gamma}\mathbf{A}+\mathbf{H}^{'T}\mathbf{H}^{'})^{-1}\mathbf{H}^{'T}\boldsymbol{Y}^{'}, & \text{if}\quad N>n_h\\ \mathbf{H}^{'T}(\mathbf{H}^{'T}\mathbf{H}^{'}+\frac{N}{\gamma}\mathbf{A})^{-1}\boldsymbol{Y}^{'}, & \text{if}\quad n_h<N\end{cases}\tag{16}$$

where $\mathbf{H}^{'}=[h^{'}(\boldsymbol{x}_1)^T,h^{'}(\boldsymbol{x}_2)^T,\cdots,h^{'}(\boldsymbol{x}_N)^T]^T\in\mathbb{R}^{N\times n_h}$, $\boldsymbol{Y}^{'}=[y_1^{'},y_2^{'},\cdots,y_N^{'}]^T$, $N=N^++N^-$. Please note, $\mathbf{H}^{'}$, $\mathbf{A}$ can be computed in advance.

Remark 1. AUC-ELM is a special ELM on the data space transformed by using Eq. (14). Compared to two trade-off parameters in CCR-ELM, Obviously, the regularization parameter γ has an important impact on the performance of AUC-ELM. In order to optimize γ, we can use cross-validation

with grid search, or leave-one-out cross valuation with the predicted residual sum of squares (PRESS) static [15]. Also, WELM [21] or CCR-ELM [24] is primarily motivated by the accuracy metric, while AUC-ELM is motivated by the AUC metric, which is more stable and robust for imbalanced binary classification tasks. Therefore, as the first attempt, AUC-ELM exhibits the great potential to handle imbalanced binary classification tasks.

Based on the above discussions, the implementation of AUC-ELM is summarized in Algorithm 1.

Algorithm 1 AUC-ELM

Input Given N labeled samples $\{\boldsymbol{x}_i, y_i\}$, $i = 1, 2, \cdots, N$ $\quad (N = N^+ + N^-)$
Output the mapping function of AUC-ELM, i.e., $h(\boldsymbol{x})\boldsymbol{\beta}$
Step 1 Initialize an ELM network of n_h hidden neurons by randomly assigning input weights and biases, and calculate the output matrix $\mathbf{H}^{'}$ of the hidden neurons.
Step 2 Compute the matrics $\mathbf{A}$ and the vector $\boldsymbol{Y}'$ according to their above definitions.
Step 3 Choose the trade-off parameter γ
Step 4
If $n_h < N$
Then compute the output weight vector $\boldsymbol{\beta}$ using the first formula in Eq. (16)
Else Compute $\boldsymbol{\beta}$ using the second formula in Eq. (16)
Step 5 Return the mapping function $h(\boldsymbol{x})\boldsymbol{\beta}$

Remark 2. Comparing Eq. (15) with Eq. (3), we can easily find that once the matrices $\mathbf{A}$, $\mathbf{H}^{'}$, and the vector $\boldsymbol{Y}^{'}$ are fixed in advance, the computational complexity of solving Eq. (15) keeps the same as that of solving Eq. (3). According to their respective definitions given as above, the calculation of the matrix $\mathbf{A}$ requires $O(N^+N^+ + N^-N^-) \approx O((N^+)^2)$ computational burdens (since N^- is generally much less than N^+). The calculation of both $\mathbf{H}^{'}$ and $\boldsymbol{Y}^{'}$ obviously requires $O(N^+ + N^-)$ computational burdens (see Eq. (14)). Therefore, in contrast to the classical ELM, AUC-ELM has an extra computational complexity, i.e., $O((N^+)^2 + N^+ + N^-) \approx O((N^+)^2)$.

IV. Semi-supervised AUC-ELM: SAUC-ELM

A. On SAUC and $\overline{SAUC}$

When on-hand training data contain unlabeled samples, we can not directly apply the AUC or $\overline{SAUC}$ metric to guide the semi-supervised learning for imbalanced binary classification. In order to sufficiently leverage all the useful information from the available training data with and without labels, here we view all the available training data as a whole, and then define a new AUC metric specific for semi-supervised learning to express the sum of two AUC values between the labeled samples and the whole dataset. The merit of such a new definition exists in that while AUC for semi-supervised training data can not be directly estimated, this new AUC metric can help us do it by using the following Theorem 1 to reveal the relationship between them.

Definition 1: AUC metric $SAUC$ for both labeled and unlabeled binary data is defined as

$$\begin{aligned} SAUC(s) =& \mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\mathbf{1}\big(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x})\big) \\ &+\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}\big(s(\boldsymbol{x}) \geq s(\boldsymbol{x}^-)\big) \end{aligned} \tag{17}$$

where $\overline{\mathcal{D}}$ denotes the distribution of all the labeled and unlabeled samples in the training data.

Theorem 1: For a binary classification problem, given an arbitrary scoring function s, there exists only a constant difference between its $AUC(s)$ value and $SAUC(s)$ value, i.e.,

$$SAUC(s) = AUC(s) + \frac{1}{2} \tag{18}$$

Proof: According to the formulation in Eq. (2), we observe

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\mathbf{1}\big(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x})\big) \\ =&\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}^-\cup\overline{\mathcal{D}}^+}\mathbf{1}\big(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x})\big) \\ =&(1-\lambda)\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}\big(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x}^-)\big) \\ &+\lambda\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^{'}\sim\overline{\mathcal{D}}^+}\mathbf{1}\big(s(\boldsymbol{x}^+) \geq (\boldsymbol{x}^{'})\big) \end{aligned} \tag{19}$$

where λ is the fixed yet unknown percentage of the majority samples to the minority samples. Please note, $\mathcal{D}^+$ and $\overline{\mathcal{D}}^+$ actually represent the sample distribution of majority samples. Because the probability that a randomly selected majority sample is ranked higher than another randomly selected majority sample from the same distribution always keeps a constant, i.e., $\frac{1}{2}$. Therefore, we have

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\mathbf{1}(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x})) \\ =&(1-\lambda)\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}\big(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x}^-)\big) + \frac{\gamma}{2} \end{aligned} \tag{20}$$

Similarly, we easily have

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}\big(s(\boldsymbol{x}) \geq s(\boldsymbol{x}^-)\big) \\ =&\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}^-\cup\overline{\mathcal{D}}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}\big(s(\boldsymbol{x}) \geq s(\boldsymbol{x}^-)\big) \\ =&\lambda\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}\big(s(\boldsymbol{x}) \geq s(\boldsymbol{x}^-)\big) \\ &+(1-\lambda)\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}^-}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}\big(s(\boldsymbol{x}) \geq s(\boldsymbol{x}^-)\big) \\ =&\lambda\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}\big(s(\boldsymbol{x}) \geq s(\boldsymbol{x}^-)\big) + \frac{1-\lambda}{2} \end{aligned} \tag{21}$$

With Eq. (20) and Eq. (21), we immediately have

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\mathbf{1}(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x})) + \mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}(s(\boldsymbol{x}) \geq s(\boldsymbol{x}^-)) \\ =&\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\mathbf{1}(s(\boldsymbol{x}^+) \geq s(\boldsymbol{x}^-)) + \frac{1}{2} \end{aligned} \tag{22}$$

i.e., $SAUC(s) = AUC(s) + \frac{1}{2}$, which means Theorem 1 holds true.

Theorem 1 is very important, since it reveals that when $SAUC(s)$ achieves the maximum, AUC will also achieve its maximum. This theoretical result provides the solid foundation for this work here. That is to say, when using the surrogate function $\Big(1-\big(s(\boldsymbol{x}^+)-s(\boldsymbol{x}^-)\big)\Big)^2$, we can maximize SAUC(s) approximately by minimizing the following formulation:

$$\begin{aligned} SAUC(s) =&\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\Big[\Big(1-\big(s(\boldsymbol{x}^+)-s(\boldsymbol{x})\big)\Big)^2\Big] \\ &+\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(s(\boldsymbol{x})-s(\boldsymbol{x}^-)\big)\Big)^2\Big] \end{aligned} \tag{23}$$

When $s(\boldsymbol{x}) = h(\boldsymbol{x})\boldsymbol{\beta}$ in ELM, similar to the mathematical derivations in the last section, we have

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-h(\boldsymbol{x})\big)\boldsymbol{\beta}\Big)^2\Big] \\ =&\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\Big[\Big(1-\big(h(\boldsymbol{x}^+)-\boldsymbol{c}\big)\boldsymbol{\beta}\Big)^2\Big] + \mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\Big[\Big(\big(h(\boldsymbol{x})-\boldsymbol{c}\big)\boldsymbol{\beta}\Big)^2\Big] \end{aligned} \tag{24}$$

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}^-}\mathbb{E}_{\boldsymbol{x}^-\sim\overline{\mathcal{D}}}\Big[\Big(1-\big(h(\boldsymbol{x})-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]\\ =&\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\Big[\Big(1-\big(\boldsymbol{c}-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\Big]+\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\Big[\big(h(\boldsymbol{x})-\boldsymbol{c}\big)\boldsymbol{\beta}\Big)^2\Big]\end{aligned} \tag{25}$$

where $\boldsymbol{c}=\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}(h(\boldsymbol{x}))$.

Therefore

$$\begin{aligned}SAUC(s)=&2\mathbb{E}_{\boldsymbol{x}\sim\overline{\mathcal{D}}}\big[\big(h(\boldsymbol{x})-\boldsymbol{c}\big)\boldsymbol{\beta}\big)^2\big]+\mathbb{E}_{\boldsymbol{x}^+\sim\mathcal{D}^+}\big[\big(1-\big(h(\boldsymbol{x}^+)-\boldsymbol{c}\big)\boldsymbol{\beta}\big)^2\big]\\&+\mathbb{E}_{\boldsymbol{x}^-\sim\mathcal{D}^-}\big[\big(1-\big(\boldsymbol{c}-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\big)^2\big]\end{aligned} \tag{26}$$

In essence, optimizing SAUC(s) asks all the outputs of ELM for labeled and unlabeled samples fall around the mean output of ELM as close as possible (see the first term in Eq. (26)), while the second and third terms in Eq. (26) represent the surrogate losses of the outputs of ELM for each class with respect to the mean output of ELM along different directions. Therefore, SUAC(s) has clear geometrical interpretation. Referring to $\overline{AUC}$ for AUC, we can develop the empirical formulations $\overline{SAUC}$ for SAUC, i.e.,

$$\begin{aligned}\overline{SAUC}(s)=&\frac{2}{N}\sum_{i=1}^{N}\boldsymbol{\beta}^T(h(\boldsymbol{x}_i)-\boldsymbol{c})(h(\boldsymbol{x}_i)-\boldsymbol{c})^T\boldsymbol{\beta}\\&+\frac{1}{N^+}\sum_{i=1}^{N^+}\big(1-(h(\boldsymbol{x}_i^+)-\boldsymbol{c})\boldsymbol{\beta}\big)^2+\frac{1}{N^-}\sum_{i=1}^{N^-}\big(1-(\boldsymbol{c}-h(\boldsymbol{x}_i^-))\boldsymbol{\beta}\big)^2\end{aligned} \tag{27}$$

where $N=N^++N^-+N_u$, N_u denotes the total number of all the unlabeled samples. We can easily verify that $\mathbb{E}(\overline{SAUC})=SAUC$.

Based on Eq. (27), we may take the following empirical formulation as the objective function of the semi-supervised AUC-ELM.

$$\begin{aligned}&\min_{\boldsymbol{\beta}\in\mathbb{R}^{n_h\times 1}}J_{SAUC-ELM}\\=&\frac{1}{2}\boldsymbol{\beta}^2+\frac{\gamma}{2}\overline{SAUC}\\=&\frac{1}{2}\boldsymbol{\beta}^2+\frac{\gamma}{N}\sum_{i=1}^{N}\boldsymbol{\beta}^T\big(h(\boldsymbol{x}_i)-\boldsymbol{c}\big)^T\big(h(\boldsymbol{x}_i)-\boldsymbol{c}\big)\boldsymbol{\beta}\\&+\frac{\gamma}{2N^+}\sum_{i=1}^{N^+}\Big(1-\big(h(\boldsymbol{x}^+)\big)-\boldsymbol{c}\big)\boldsymbol{\beta}\Big)^2+\frac{\gamma}{2N^-}\sum_{i=1}^{N^-}\Big(1-\big(\boldsymbol{c}-h(\boldsymbol{x}^-)\big)\boldsymbol{\beta}\Big)^2\end{aligned} \tag{28}$$

After denoting $\mathbf{B}=\mathbf{I}+\frac{2\gamma}{N}\sum_{i=1}^{N}(h(\boldsymbol{x}_i)-\boldsymbol{c})^T(h(\boldsymbol{x}_i)-\boldsymbol{c})$ and using the following mathematical transformation:

$$(h^{'}(\boldsymbol{x}_i),y_i^{'})=\begin{cases}\Big(\sqrt{\frac{1}{2p}}\big(h(\boldsymbol{x}_i)-\boldsymbol{c}\big),\sqrt{\frac{1}{2p}}\Big), & y_i=+1\\ \Big(\sqrt{\frac{1}{2(1-p)}}\big(h(\boldsymbol{x}_i)-\boldsymbol{c}\big),-\sqrt{\frac{1}{2(1-p)}}\Big), & y_i=-1\end{cases} \tag{29}$$

where $p=\frac{N^+}{(N^++N^-)}=\frac{N^+}{N}$, we can equivalently express Eq. (28) as

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{n_h\times 1}}J_{SAUC-ELM}=\ \frac{1}{2}\boldsymbol{\beta}^T\mathbf{B}\boldsymbol{\beta}+\frac{\gamma}{2N}\sum_{i=1}^{N}\big(y_i^{'}-h^{'}(\boldsymbol{x}_i)\boldsymbol{\beta}\big)^2 \tag{30}$$

Referring to the classical ELM in Eq. (1), we can readily has its solution

$$\boldsymbol{\beta}=\begin{cases}(\frac{N}{\gamma}\mathbf{B}+\mathbf{H}^{'T}\mathbf{H}^{'})^{-1}\mathbf{H}^{'T}\boldsymbol{Y}^{'}, & \text{if}\quad N>n_k\\ \mathbf{H}^{'T}(\mathbf{H}^{'T}\mathbf{H}^{'}+\frac{N}{\gamma}\mathbf{B})^{-1}\boldsymbol{Y}^{'}, & \text{if}\quad n_k<N\end{cases} \tag{31}$$

where $\mathbf{H}^{'}$, $\boldsymbol{Y}^{'}$ are the same as in AUC-ELM. Based on the above discussions, the implementation of SAUC-ELM is summarized in Algorithm 2.

Remark 3. According to the definition of the matrix $\mathbf{B}$ in Eq. (30), its determination requires $O(N^2)=O((N^++N^-+N_u)^2)\approx O((N^++N_u)^2)$ computational burdens. After comparing Eq. (30) with Eq. (15), according to ***Remark 2***, we readily know that in contrast to the classical ELM, SAUC-ELM has an extra computational complexity, i.e., $O((N^++N_u)^2)$.

Algorithm 2 SAUC-ELM

Input Given N labeled samples $\{\boldsymbol{x}_i,y_i\}$, $i=1,2,\cdots,N$ $(N=N^++N^-)$, and unlabeled samples $\{\boldsymbol{x}_i\}$, $i=1,2,\cdots,N_u$
Output the mapping function of SAUC-ELM, i.e., $h(\boldsymbol{x})\boldsymbol{\beta}$
Step 1 Initialize an ELM network of n_h hidden neurons by randomly assigning input weights and bias, and calculate the output matrix $\mathbf{H}^{'}$ of the hidden neurons.
Step 2 Compute the matrix $\mathbf{B}$ and the vector $\boldsymbol{Y}^{'}$ according to their above definitions.
Step 3 Determine the trade-off parameter γ
Step 4
If $n_k<N$
Then compute the output weight vector $\boldsymbol{\beta}$ using the first formula in Eq. (31)
Else Compute $\boldsymbol{\beta}$ using the second formula in Eq. (31)
Step 5 Return the mapping function $h(\boldsymbol{x})\boldsymbol{\beta}$

Remark 4. Compared to two trade-off parameters in SS-ELM [35], SAUC-ELM only has one trade-off parameter γ to tune, which greatly reduces the computational cost for model selection. Besides, SS-ELM is motivated to improve the accuracy metric, while SAUC-ELM is motivated to directly optimize the AUC metric, which is more stable and robust for imbalanced learning. Therefore, as the first attempt, SAUC-ELM exhibits the great potential to deal with semi-supervised imbalanced learning tasks specifically.

Remark 5. In fact, let $\overline{\mathcal{D}}$ only represent the distribution of all the labeled samples, theorem 1 still holds true. Thus, we can easily obtain a variant of AUC-ELM, i.e., changing $\mathbf{B}$ in Eq. (31) into $\mathbf{I}+\frac{2\gamma}{N}\sum_{i=1}^{N^++N^-}\big(h(\boldsymbol{x}_i)-\boldsymbol{c}\big)\big(h(\boldsymbol{x}_i)-\boldsymbol{c}\big)^T$, where $\boldsymbol{c}$ is the mean vector of all the labeled samples. However, AUC-ELM introduced in the last section has its potential application value, especially for incremental or on-line learning. That is, as the sample size increases, we can determine the value of $\boldsymbol{c}^+$ and $\boldsymbol{c}^-$ based on the new sample labels, and perform the formula transformation using Eq. (14). Whereas, for the variant mentioned above, we must do the formula transformation using Eq. (29) on all the samples. Thus, apparently, the proposed AUC-ELM is more suitable for handling on-line or incremental learning. In the near future, we will report more experimental results about on-line AUC-ELM.

V. Results

We evaluated our proposed algorithms on various supervised and semi-supervised imbalance classification tasks. For supervised learning, comparisons were made with conventional and state-of-art learning algorithms, e.g., ELM, WELM, CCR-ELM and CS-ELM. In addition, owing to the theoretical and extensively experimental evidences of SVM in strong generalization capability, we also took SVM as the baseline method. For semi-supervised learning, we experimentally compared SAUC-ELM with ELM, SVM, and AUC-ELM. SAUC-ELM was not directly compared with the existing method SS-ELM

[35], considering that the applicability of SS-ELM is limited by too many parameters, i.e., both two trade-off parameters and Laplacian graph matrix for the adopted datasets. In addition, we evaluated on how SAUC-ELM leverages the useful information in a progressive way from unlabeled data by giving different percentages of labeled training data.

The proposed algorithms were implemented using MATLAB R2014a on a 2.40GHz machine with 8GB of memory.

A. Supervised learning results

1) Datasets and parameter settings: The proposed algorithm is evaluated on 38 public datasets from KEEL dataset repository [54]. The class imbalance ratios of these datasets vary. To quantify the imbalance degree of a dataset, the imbalance ration (IR) is used:

$$\text{IR} = \frac{\text{No. minority instances}}{\text{No. majority instances}} \tag{32}$$

The details of the adopted datasets are given in Table. I. In the data-preprocessing stage, features scaling is applied to normalize the attributes of the datasets to the range $[-1, 1]$.

For the parameter setting of AUC-ELM, owing to the use of random weights between the input and hidden layers, we followed [49] to use 5-fold cross-validation with grid search to find the optimal values of both the trade-off parameter γ and the number n_h of hidden nodes from the sets $\{2^{-18}, 2^{-16}, \cdots, 2^{50}\}$ and $\{10, 20, \cdots, 990, 1000\}$. Sigmoid nodes are taken for all the adopted ELM based algorithms. The experimental setup for ELM, WELM, CCR-ELM, and CS-ELM is same as that in [49]. For SVM, Gaussian kernel was taken due to the best performance in the experimental trial on each dataset. The kernel width δ and regularization parameter C were selected from $\{2e-12, 2e-11, \cdots, 2e11, 2e12\}$ and $\{1, 10, 50, 100, 150, 200, 250, 500\}$. Considering the weights between the input and hidden layers are randomly selected, which may fluctuate the performance, we reported the average results for ten individual runnings for each algorithm.

G-mean and AUC are used as evaluation metrics to measure the class imbalance learning performance. G-mean is defined as follows.

$$\text{G-mean} = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \tag{33}$$

where TP and TN are the correctly classified instances belonging to positive and negative class respectively, and FP and FN are the incorrectly classified instances belonging to negative and positive class, respectively. G-mean is like AUC which is comparatively robust to class imbalance. In addition, the other common evaluation metrics, such as accuracy, precision and recall [55] are recorded. Here, due to space limit, we listed these metrics results for semi-supervised learning experiments only.

TABLE I: Details of Imbalanced datasets

Datasets	No. instances	No. features	IR
abalone19	4174	8	0.0078
yeast6	1484	8	0.0255
ecoli0137vs26	281	7	0.0255
yeast5	1484	8	0.0305
yeast2vs8	482	8	0.0433
glass5	214	9	0.0438
shuttleC2vsC4	129	9	0.0488
glass016vs5	184	9	0.0514
abalone9vs18	731	8	0.0590
page-blocks13vs4	472	10	0.0631
glass4	214	9	0.0646
yeast1vs7	459	7	0.0724
shuttleC0vsC4	1829	9	0.0721
ecoli4	336	7	0.0735
cleveland0vs4	173	13	0.0813
glass2	214	9	0.0962
glass016vs2	192	9	0.0972
vowel0	988	13	0.0990
yeast05679vs4	528	8	0.1069
yeast2vs4	514	8	0.1101
ecoli034vs5	200	7	0.1111
page-blocks0	5472	10	0.1140
ecoli3	336	7	0.1200
yeast3	1484	8	0.1233
glass6	214	9	0.1567
segment0	2308	19	0.1662
ecoli2	336	7	0.1831
new-thyroid1	215	5	0.1946
ecoli1	336	7	0.2973
haberman	306	3	0.3731
vehicle1	846	18	0.3960
vehicle2	846	18	0.3960
yeast1	1484	8	0.4066
glass0	214	9	0.4854
pima	768	8	0.5356
wisconsin	683	9	0.5380
ecoli0vs1	220	7	0.5384
glass1	214	9	0.5495

2) Experimental results: Following the same organization of the experimental results as in [49], the performances of AUC-ELM and the comparative algorithms on the adopted datasets with $IR \leq 0.1111$ and $IR \geq 0.1111$ in terms of AUC and G-mean are reported in Tables II-V. The experimental results for ELM, WELM, CCR-ELM, and CS-ELM are taken from [49]. It can be seen that AUC-ELM has a superior advantage over other algorithms no matter with high or low imbalance ratios. We also carried out the Friedman ranking test followed by Holm post-hoc test [56] for the statistical comparison of all the algorithms over these 38 public datasets. The Friedman ranking test is to evaluate whether there are statistically significant differences among the algorithms considered over given sets of data. The null hypothesis is that all the algorithms perform equally well for a given level. If the p-value for this test is smaller than 0.05, the null hypothesis is rejected. From Tables VI and VIII, the null hypothesis is

rejected in terms of AUC ($p = 8.8730E-11$) and G-mean ($p = 1.0059E-10$). The Holm post-hoc test is to further verify whether there is a statistical performance difference between the best Friedman ranking algorithm and every other algorithm. In our case, we used the Holm post-hoc test to compare AUC-ELM with every other algorithm. We set $\alpha = 0.05$ as the level of confidence in all cases. According to the results from Tables VII and IX, AUC-ELM significantly outperformed the other ELM-based methods and the conventional SVM in terms of AUC and G-mean. The mean training time of AUC-ELM and other algorithms are given in Table X. It is observed that AUC-ELM took comparable or a little more training time than ELM, CCR-ELM, and CS-ELM in most cases, whereas WELM took far more time to train. In addition, we can experimentally observe how the trade-off parameter γ affects the performance of AUC-ELM on the adopted datasets. Due to the paper's space limitation, we reported the impact of the trade-off parameter γ on the testing performance of AUC-ELM with different numbers of hidden nodes n_h only for the Pima dataset in Fig. 1. Obviously, γ can significantly influence the testing performance of AUC-ELM on this dataset. How to determine an appropriate value of both γ and the number of hidden nodes for a dataset is still worthy to be studied in the future.

TABLE II: AUC(%)±std. for datasets with high imbalance ratio (IR≤0.1111)

Datasets	AUC-ELM	ELM	WELM	CCR-ELM	CS-ELM	SVM
abalone9vs18	**99.31±0.10**	75.93±1.58	94.91±0.66	78.22±0.95	94.26±0.66	80.77±4.18
glass5	98.95±1.59	93.60±2.01	99.17±0.17	93.17±2.24	**99.71±0.13**	98.39±1.09
glass2	**84.49±1.11**	64.18±1.94	83.51±0.18	83.72±2.26	83.86±0.85	81.34±6.63
clevelandOvs4	**97.71±3.72**	77.41±5.80	97.41±4.90	83.72±2.26	97.69±0.84	90.64±4.89
yeast6	**96.64±0.11**	72.53±1.69	91.37±1.21	72.42±0.64	91.02±0.37	93.59±5.91
ecoli0137vs26	**93.89±11.33**	75.00±6.31	80.90±7.14	75.00±4.35	83.51±9.43	90.26±6.79
yeast5	**99.12±0.12**	82.81±1.47	98.55±0.49	83.87±1.06	98.80±0.27	98.51±0.89
abalone19	77.59±7.55	53.50±1.23	78.08±0.16	56.21±1.78	**79.02±0.07**	60.39±3.02
glass016vs5	**99.61±0.81**	94.36±3.33	98.36±1.21	94.73±2.52	99.14±1.12	94.61±7.63
ecoli4	**100.00±0.00**	91.37±1.34	99.12±3.78	95.59±2.11	99.26±0.75	98.11±3.15
glass4	**98.76±0.10**	93.74±3.91	92.97±4.79	88.93±4.30	94.69±3.03	94.72±1.81
shuttleC0vsC4	**100±0.00**	99.23±1.08	100±0.13	99.47±1.76	100±0.34	100±0.00
shuttleC2vsC4	**100±0.00**	99.23±2.71	100±0.00	96.15±3.10	99.00±2.58	100±0.00
page-blocks13vs4	99.42±0.61	97.63±1.37	99.54±0.87	92.67±1.39	**99.54±0.23**	94.71±2.60
glass016vs2	**96.34±0.91**	81.16±3.01	81.16±1.23	80.80±2.79	83.11±0.66	80.55±3.05
vowel0	**100±0.00**	100±0.00	100±0.00	100±0.00	100±0.00	99.99±0.01
yeast1vs7	**82.57±0.29**	65.58±1.23	79.43±0.83	75.27±0.56	79.59±0.50	79.00±6.37
yeast2vs8	**84.72±0.29**	70.92±1.92	80.22±2.58	72.02±1.12	81.61±2.36	81.62±8.12
yeast2vs4	**96.81±0.440**	89.31±1.31	93.76±0.32	90.02±1.47	93.48±0.65	94.60±2.86
yeast05679vs4	**88.11±0.41**	81.37±1.93	84.74±2.16	80.02±1.68	84.26±0.79	85.37±6.27

B. Semi-supervised learning results

1) Datasets and parameter settings: We evaluated the SAUC-ELM on 6 public datasets from UCI respiratory [57]. All the adopted datasets are imbalanced and their specifications are given in Table. XI. In data pre-processing stage, feature scaling is applied to normalize the attributes to the range [-1,1]. The parameter setting for SSAUC-ELM and ELM is the same as AUC-ELM in section V-A. Each dataset was split into five folds, one of which was used as the testing dataset, and the other four folds were used as the training dataset. In order to observe how SAUC-ELM leverages the unlabeled data in a progressive way, the percentage of the labeled set in the training set is set to be 10%, 15%, 20%, 25%, 30% and 40% in the experimental setting.

TABLE III: AUC(%)±std. for datasets with low imbalance ratio (IR≥0.1111)

Datasets	AUC-ELM	ELM	WELM	CCR-ELM	CS-ELM	SVM
pima	**93.90±1.16**	73.60±0.21	78.72±0.29	73.87±0.26	79.02±0.20	81.12±2.57
wisconsin	**99.29±1.26**	97.81±0.81	98.44±0.14	97.88±0.13	98.92±0.17	97.67±0.08
haberman	**75.61±3.83**	58.51±0.11	67.63±0.16	58.83±1.34	67.63±0.26	62.93±2.10
yeast3	**99.15±0.17**	81.79±0.16	92.99±0.35	85.41±0.13	94.71±0.08	98.16±0.59
segment0	93.74±1.69	99.78±0.61	**99.83±0.43**	99.69±0.28	99.80±0.04	71.49±3.18
page-blocks0	87.64±1.26	92.86±0.31	94.47±0.20	93.89±0.17	**94.71±0.42**	54.67 ±3.37
new-thyoid1	**100±0.00**	98.47±0.21	100±0.09	98.51±0.28	100±0.03	99.69±0.44
ecoli0vs1	**100±0.00**	98.67±1.77	98.67±1.32	98.67±1.35	98.67±1.46	98.47±2.22
yeast1	**88.05±3.01**	67.63±1.21	76.14±0.17	65.68±0.16	76.62±0.14	79.79±1.79
ecoli1	**98.14±2.58**	88.67±1.27	93.69±0.53	89.29±1.50	93.80±0.44	92.71±3.68
ecoli2	**99.16±1.38**	92.23±0.37	94.94±0.78	89.71±0.34	95.11±0.44	94.35±2.83
glass0	**90.23±7.18**	75.78±1.56	79.45±1.83	76.88±2.42	82.35±0.76	84.49±4.80
glass1	**85.69±6.51**	75.07±1.36	78.99±1.05	78.79±1.59	80.64±0.73	82.23±6.23
glass6	**96.49±3.47**	92.58±2.05	92.49±0.90	92.38±1.87	92.75±1.99	93.19±2.14
ecoli3	**94.79±4.02**	91.15±1.37	92.79±0.38	91.18±0.83	92.85±0.08	94.05±3.18
ecoli034vs5	**98.17±3.17**	89.58±6.64	93.22±1.57	88.89±1.06	93.72±1.40	97.06±3.73
vehicle1	**94.02±1.41**	86.76±1.40	89.42±0.87	85.78±1.69	89.51±0.44	85.68±2.09
vehicle2	**99.92±0.17**	99.41±1.67	99.54±0.37	99.29±1.35	99.54±0.19	99.51±0.36

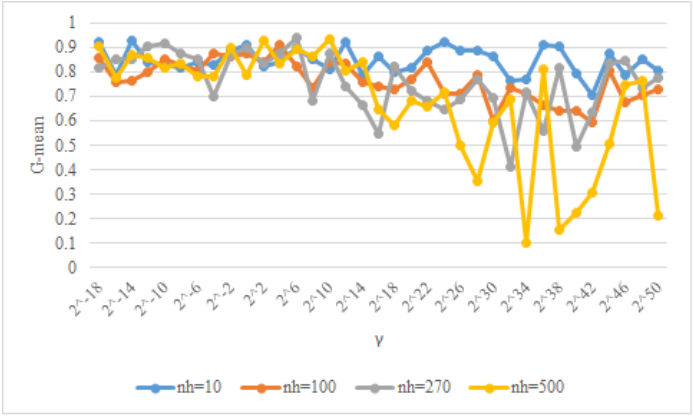

Fig. 1: Performance of AUC-ELM vs. γ and n_h for the Pima dataset, in which n_h denotes the number of hidden nodes

2) Experimental results: Tables XII and XIII present the mean training time and the average values of G-mean, AUC, accuracy, precision and recall, respectively obtained by the SAUC-ELM and comparative algorithms given 25% labeled set in the training set. ELM based algorithms have a much faster learning speed than SVM. From the results, we can conclude that SAUC-ELM obtained the best classification performance in terms of every evaluation metric on almost all the datasets using both labeled and unlabeled data, whereas ELM and SVM yielded less satisfactory results on most of the datasets using labeled data only. Fig. 2 shows the AUC performances of SAUC-ELM, AUC-ELM, ELM, and SVM on the UCI datasets with different percentages of labeled training data. In this experiment, all the settings keep the same as above, except that we varied the ratio of labeled and unlabeled data in the training set. We can observe that in most cases SAUC-ELM outperformed the other algorithms considerably when there is a low percentage of labeled data ($\leq 25\%$). Also,

TABLE IV: G-mean for datasets with high imbalance ratio (IR$\leq$0.1111)

Datasets	G-mean											
	AUC-ELM		ELM		WELM		CCR-ELM		CS-ELM		SVM	
	(γ, n_h)	Testing result (%)	(C, n_h)	Testing result (%)	(C, n_h)	Testing result (%)	(C, n_h)	Testing result (%)	(C, n_h)	Testing result (%)	(γ, C)	Testing result (%)
abalone9vs18	(2^10, 300)	**99.31**	(2^40, 150)	75.29	(2^32, 20)	88.72	(2^24, 2^36, 180)	76.22	(2^6, 600)	91.99	(2e-1, 200)	64.98
glass5	(2^10, 30)	**98.94**	(2^20, 90)	90.81	(2^10, 110)	95.99	(2^-6, 2^-4, 200)	94.78	(2^4, 820)	97.36	(1, 200)	98.17
glass2	(2^14, 160)	**82.34**	(2^28, 110)	79.49	(2^22, 140)	80.33	(2^-4, 2^-12, 10)	79.40	(2^10, 80)	80.89	(2e-1, 100)	48.29
cleveland0vs4	(2^10, 50)	**97.63**	(2^22, 80)	71.45	(2^-8, 760)	93.35	(2^-4, 2^-12, 10)	74.40	(2^-18, 900)	93.69	(2e-1, 50)	78.93
yeast6	(2^5, 230)	**96.64**	(2^44, 335)	70.77	(2^14, 900)	88.29	(2^8, 2^-14, 40)	84.45	(2^8, 270)	88.55	(2e-1,50)	79.14
ecoli0137vs26	(2^2, 60)	**93.00**	(2^2, 600)	74.14	(2^4, 400)	75.29	(2^6, 2^4, 450)	74.41	(2^18, 120)	78.08	(1, 50)	76.18
yeast5	(2^32, 200)	**99.00**	(2^36, 900)	81.04	(2^30, 100)	95.39	(2^28, 2^32, 880)	83.08	(2^6, 380)	96.67	(2e-1, 100)	88.49
abalone19	(2^8, 140)	**79.53**	(2^42, 990)	47.52	(2^6, 150)	77.19	(2^-10, 2^4, 400)	64.21	(2^2, 100)	78.68	(1, 100)	0.00
glass016vs5	(2^24, 320)	**99.77**	(2^18, 660)	92.41	(2^6, 960)	98.70	(2^8, 2^24, 200)	97.70	(2^4, 920)	98.84	(1, 100)	77.58
ecoli4	(2^5, 10)	**100.00**	(2^22, 60)	91.96	(2^6, 180)	97.83	(2^-8, 2^-12, 10)	98.43	(2^40, 30)	98.56	(2e-1, 100)	89.43
glass4	(2^10, 120)	**99.25**	(2^34, 30)	85.72	(2^12, 120)	91.34	(2^-8, 2^-2, 40)	96.18	(2^38, 900)	96.21	(100, 1)	98.04
shuttleC0vsC4	(2^12, 80)	**100**	(2^14, 10)	100	(2^38, 10)	100	(2^6, 2^6, 20)	100	(2^6, 20)	100	(1, 100)	100
shuttleC2vsC4	(2^4, 40)	**100**	(2^40, 20)	93.54	(2^28, 10)	100	(2^-12, 2^-12, 10)	100	(2^12, 20)	100	(1, 10)	85.18
page-blocks13vs4	(2^10, 10)	**99.42**	(2^8, 660)	97.60	(2^14, 420)	98.16	(2^12, 2^12, 300)	97.33	(2^2, 900)	98.10	(1, 100)	96.98
glass016vs2	(2^30, 150)	**94.18**	(2^34, 150)	67.78	(2^14, 380)	83.77	(2^34, 2^30, 240)	76.44	(2^10, 280)	85.13	(2e-1, 100)	52.04
vowel0	(2^18, 220)	**100**	(2^28, 110)	100	(2^50, 120)	100	(2^-8, 2^-2, 400)	100	(2^8, 600)	100	(2e-1, 10)	99.44
yeast1vs7	(2^38, 820)	**80.14**	(2^40, 960)	65.58	(2^16, 550)	77.26	(2^18, 2^-2, 40)	75.27	(2^8, 260)	78.56	(2e-1, 10)	68.56
yeast2vs8	(2^4, 480)	**80.23**	(2^0, 290)	72.83	(2^8, 60)	76.01	(2^8, 2^-2, 200)	73.02	(2^8, 20)	78.11	(2e-1, 10)	75.75
yeast2vs4	(2^10, 90)	**96.67**	(2^36, 280)	86.25	(2^26, 940)	91.56	(2^-8, 2^24, 400)	90.02	(2^12, 920)	92.42	(10, 1)	87.70
yeast05679vs4	(2^6, 610)	**89.29**	(2^32, 390)	64.49	(2^-2, 150)	81.05	(2^-8, 2^-2, 200)	80.02	(2^-10, 460)	81.24	(10, 1)	82.80

TABLE V: G-mean for datasets with low imbalance ratio (IR$\geq$0.1111)

Dataset	G-mean											
	AUC-ELM		ELM		WELM		CCR-ELM		CS-ELM		SVM	
	(γ, n_h)	Testing result (%)	(C, n_h)	Testing result (%)	(C, n_h)	Testing result (%)	(C^+, C^-, n_h)	Testing result (%)	(C, n_h)	Testing result (%)	(γ, C)	Testing result (%)
Pima	(2^6, 270)	**93.88**	(2^32, 30)	70.10	(2^14, 20)	74.74	(2^2, 2^48, 280)	70.99	(2^8, 530)	75.73	(2e-1, 100)	70.09
wisconsin	(2^2, 120)	**99.30**	(2^34, 50)	96.32	(2^34, 60)	97.07	(2^2,2^2, 420)	96.94	(2^2, 450)	97.36	(2e1, 100)	98.23
haberman	(2^10, 290)	**75.52**	(2^44, 910)	49.16	(2^34, 10)	65.11	(2^36,2^34, 20)	49.81	(2^-18, 400)	65.71	(2e-2, 100)	53.45
yeast3	(2^4, 100)	**99.14**	(2^40, 100)	80.75	(2^16,700)	93.25	(2^-8, 2^6, 100)	91.11	(2^16, 60)	93.57	(1, 100)	87.94
segment0	(2^32, 90)	92.36	(2^8, 720)	99.24	(2^18, 30)	99.75	(2^8, 2^8, 800)	99.18	(2^6, 620)	**99.87**	(2e-2, 150)	55.60
page-blocks0	(2^6, 250)	87.63	(2^34, 830)	89.92	(2^24, 820)	**93.40**	(2^16, 2^24, 800)	90.89	(2^12, 500)	93.38	(2e-2, 100)	21.88
new-thyroid1	(2^10, 190)	**100**	(2^18, 180)	98.24	(2^18, 30)	99.44	(2^-14, 2^-18, 10)	99.24	(2^6,260)	99.44	(2e-1, 100)	96.62
ecoli0vs1	(2^28, 900)	**100**	(2^0, 80)	98.64	(2^14, 20)	98.51	(2^2, 2^2, 240)	98.64	(2^2,260)	98.47	(2e-1, 50)	97.33
yeast1	(2^50, 530)	**87.95**	(2^44, 300)	63.26	(2^26, 120)	72.57	(2^-8, 2^2, 400)	71.70	(2^12,110)	72.98	(2e,1, 50)	68.77
ecoli1	(2^0, 10)	**98.09**	(2^16, 140)	87.77	(2^4, 320)	90.69	(2^4, 2^2, 20)	89.06	(2^8,350)	91.73	(2e-1, 50)	85.48
ecoli2	(2^4, 90)	**99.15**	(2^36, 60)	91.17	(2^28, 40)	93.91	(2^-4, 2^-4, 20)	92.80	(2^10,50)	94.26	(2e-1, 50)	89.76
glass0	(2^16, 80)	**89.48**	(2^14, 950)	79.61	(2^22, 800)	81.17	(2^8, 2^-2, 880)	88.56	(2^14,500)	80.70	(2e-1, 100)	79.25
glass1	(2^38, 700)	**84.75**	(2^16, 440)	78.36	(2^22, 900)	78.31	(2^-10, 2^12, 70)	76.07	(2^-8,370)	79.64	(2e-2, 100)	74.26
glass6	(2^8, 160)	**96.42**	(2^46, 450)	94.96	(2^44, 30)	95.72	(2^-12, 2^-4, 20)	91.29	(2^16,300)	95.78	(1, 150)	90.59
ecoli3	(2^2, 280)	**94.71**	(2^44, 70)	77.38	(2^46, 10)	90.17	(2^-12, 2^-18, 10)	91.45	(2^44,60)	89.86	(1, 150)	74.22
ecoli034vs5	(2^2, 200)	**98.12**	(2^8, 480)	88.67	(2^44, 30)	95.72	(2^12, 2^4, 80)	89.29	(2^16,300)	95.78	(1, 100)	89.96
vehicle1	(2^10, 250)	**93.92**	(2^8, 570)	79.29	(2^14, 450)	85.30	(2^8, 2^16, 500)	79.60	(2^6, 640)	86.12	(2e-1, 100)	72.91
vehicle2	(2^10, 90)	**99.92**	(2^12, 600)	98.43	(216, 800)	99.12	(2^8, 2^-2, 900)	98.63	(2^8, 380)	99.37	(2e-1, 150)	97.75

TABLE VI: Friedman mean-rankings of the algorithms in terms of AUC (p-value = 8.8730E-11)

Algorithm	Ranking
AUC-ELM	1.6447
ELM	5.1447
WELM	3.1053
CCR-ELM	5
CS-ELM	2.5658
SVM	3.5395

TABLE VII: Holm post-hoc comparison results in terms of AUC with $\alpha = 0.05$

i	algorithms	$z = (R_0 - R_i)/SE$	p	Holm
15	AUC-ELM vs. ELM	8.154753	0	0.003333
14	AUC-ELM vs. CCR-ELM	7.817527	0	0.003571
9	AUC-ELM vs. SVM	4.414603	0.00001	0.005556
6	AUC-ELM vs. WELM	3.402923	0.000667	0.008333
4	AUC-ELM vs. CS-ELM	2.145988	0.031874	0.0125

TABLE VIII: Friedman mean-rankings of the algorithms in terms of G-means (p-value = 1.0059E-10)

Algorithm	Ranking
AUC-ELM	5.9737
ELM	4.1053
WELM	2.2105
CCR-ELM	3.1579
CS-ELM	1.4737
SVM	4.0789

when dealing with small datasets ($N < 1000$), SAUC-ELM and ELM has a comparable training time. Once the size of the training set increases ($N > 1000$), SAUC-ELM falls behind ELM on the aspect of training time. Though SAUC-ELM is not as fast as ELM, its training is still efficient which took only seconds at the most on these datasets.

TABLE IX: Holm post-hoc comparison results in terms of G-means with $\alpha = 0.05$

i	algorithms	$z = (R_0 - R_i)/SE$	p	Holm
5	AUC-ELM vs. CS-ELM	10.484683	0	0.003333
4	AUC-ELM vs. WELM	8.767893	0	0.003571
3	AUC-ELM vs. CCR-ELM	6.560591	0	0.003846
2	AUC-ELM vs. SVM	4.414603	0.00001	0.005556
1	AUC-ELM vs. ELM	4.353289	0.000013	0.007143

TABLE X: Training time (in seconds) for imbalanced problems

Datasets	AUC-ELM	ELM	WELM	CCR-ELM	CS-ELM	SVM
page-blocks0	0.9693	0.7870	8.5239	0.8750	**0.7656**	9.7801
abalone19	0.6963	**0.5612**	4.7047	0.7658	0.5962	5.7418
segment0	0.4503	0.3543	1.8391	0.3602	**0.3510**	1.8747
shuttleC0vsC4	0.3435	0.2898	1.3208	0.2898	**0.2709**	0.6436
yeast1	0.2695	0.2466	1.0257	0.2810	**0.2416**	0.3813
yeast3	0.2558	0.2438	0.9970	0.2821	**0.2461**	0.4622
yeast6	0.2782	0.2665	0.9681	0.3229	**0.2455**	0.4472
yeast5	0.2615	0.2580	0.9309	0.2566	**0.2409**	0.4025
vowel0	0.1853	**0.1802**	0.6361	0.1905	0.1910	0.2431
vehicle1	0.1846	**0.1666**	0.5298	0.1745	0.1862	0.2240
vehicle2	0.1811	**0.1698**	0.5344	0.1812	0.1946	0.2390
pima	0.1762	0.1590	0.4283	0.1766	**0.1523**	0.2344
wisconsin	**0.1363**	0.1373	0.3847	0.1449	0.1417	0.3125
abalone9vs18	0.2703	0.1419	0.3850	0.1568	0.1666	**0.1135**
yeast2vs4	**0.0987**	0.1178	0.3508	0.1205	0.1313	0.1250
yeast05679vs4	**0.1123**	0.1186	0.2889	0.1305	0.1343	0.1875
yeast2vs8	**0.0928**	0.1139	0.2654	0.1221	0.1376	0.1875
yeast1vs7	**0.1036**	0.1134	0.2734	0.1211	0.1205	0.2031
page-blocks13vs4	**0.1172**	0.1318	0.2346	0.1451	0.1462	0.1250

TABLE XI: Details of semisupervised imbalanced datasets

Datasets	No. instances	No. features	IR
breast	683	9	0.5262
ilpd	579	10	0.3986
winequality	6497	12	0.3266
wilt	4839	10	0.4212
seismicbumps	2584	15	0.0704
magic	19020	10	0.5422

TABLE XII: Training time (in seconds) for semi-supervised imbalance problems

Datasets	SAUC-ELM	ELM	SVM
breast	**0.0625**	0.0781	0.5625
ILPD	**0.0463**	0.0625	3.5781
winequality	5.0313	**0.1875**	49.640
WILT	4.4531	**0.1563**	54.734
seismicbumps	2.4531	**0.1250**	13.140
magic	14.096	**1.1250**	958.984

VI. Conclusions and future work

In this paper, we proposed two algorithms, i.e., AUC-ELM and SAUC-ELM, to adapt the traditional ELM for supervised and semi-supervised imbalanced learning respectively by directly integrating AUC metric optimization. AUC-ELM is theoretically derived to be equivalent to the corresponding ELM on the transformed input space. Therefore, it is expected to inherit the traditional ELM's outperforming generalization capability and training efficiency. With respect to the semi-supervised version SAUC-ELM, to best of our knowledge, it is the first extension of ELM to solve semi-supervised imbalanced learning task by direct AUC maximization. Compared to existing ELM variants for imbalance learning, such as CCR-ELM and SS-ELM, the proposed algorithms only have one trade-off parameter to tune, which reduces the computational cost for model selection. The experimental results show that AUC-ELM consistently outperformed the other ELM-based methods and SVM with various imbalance ratios. SAUC-ELM also exhibited superior classification performances compared with pure supervised learning algorithms such as ELM and SVM.

In future, there are several issues that need to be further studied. First, it is valuable to develop the extensions of AUC-ELM and SAUC-ELM to fit massive datasets and online learning situations. Second, in this study we only focus on binary classification, it is also necessary to further explore the algorithms to solve imbalanced multiclass classification tasks. To achieve such goal, we will refer to the definition of the total AUC for multiclass classification in [58] and modify the relevant objective functions accordingly. Third, it is worthwhile to extend the direct AUC optimization to different algorithms in the family of kernel ridge regression to further improve the classification performances. Fourth, except for AUC, other metrics such as recall and precision may also be used for imbalanced learning. As such, how to select a metric from the theoretical and practical perspectives and how to integrate these metrics into the ELM framework for imbalanced classification tasks will be an interesting topic in near future.

TABLE XIII: G-mean(±STD.) & AUC(±STD.) performances for UCI datasets given 25% labeled training data

Dataset	Metric	(γ, n_h)	SAUC-ELM	(γ, n_h)	AUC-ELM	(C, n_h)	ELM	(δ, C)	SVM
breast	G-mean	(2^2, 10)	**99.68±0.47**	(2^0, 10)	97.75±0.21	(2^0, 230)	95.55±1.38	(1, 50)	96.57±1.86
	AUC		**99.9±0.25**		98.49±0.30		93.69±1.34		98.52±1.23
	Accuracy		99.75±0.39		99.57±0.81		95.39±1.85		96.29±3.03
	Precision		**99.52±0.76**		98.88±0.35		94.81±3.90		97.25±3.15
	Recall		**100.0±0.00**		99.27±1.63		95.95±2.71		96.85±2.90
ILPD	G-mean	(2^0, 100)	**73.25±6.54**	(2^4, 140)	72.49±5.13	(2^8, 270)	55.02±4.16	(2e-1, 10)	58.51±0.41
	AUC		**74.29±5.34**		71.5±7.42		56.86±2.86		65.54±3.45
	Accuracy		**74.84±9.35**		71.59±7.43		46.39±7.57		65.52±8.29
	Precision		84.02±11.34		81.09±9.45		**84.36**±5.44		44.63±16.45
	Recall		**58.79±13.00**		53.15±12.2		38.04±8.49		50.38±13.83
winequality	G-mean	(2^5, 260)	**99.87±0.20**	(2^8, 50)	99.82±0.06	(2^10, 300)	99.17±0.08	(2e-2, 50)	97.76±0.47
	AUC		**99.71±0.20**		99.70±0.35		99.07±0.08		99.33±0.09
	Accuracy		**99.84±0.10**		99.74±0.14		99.10±0.47		98.43±0.45
	Precision		**99.98±0.04**		99.96±0.04		98.37±0.94		99.11±0.44
	Recall		**99.71±0.20**		99.50±0.26		**99.84±0.15**		98.83±0.36
WILT	G-mean	(2^10, 260)	**98.29±0.58**	(2^10, 40)	98.03±0.77	(2^10, 300)	55.07±4.37	(2e-2, 100)	86.37±2.02
	AUC		**98.71±0.57**		97.78±1.00		65.91±2.70		96.43±1.08
	Accuracy		**99.10±0.31**		93.06±4.23		51.73±6.46		96.49±1.64
	Precision		**98.42±0.50**		91.42±4.37		34.81±5.72		98.16±1.16
	Recall		**100.0±0.00**		99.04±1.03		99.84±0.14		98.12±1.31
seismicbumps	G-mean	(2^0, 250)	**66.35±7.25**	(2^0, 180)	59.78±4.09	(2^10, 300)	54.87±3.31	(2e-2, 10)	42.97±8.24
	AUC		**75.31±8.22**		69.55±5.05		57.69±3.31		57.95±6.21
	Accuracy		**85.38±5.51**		76.24±7.27		36.34±7.15		79.15±1.68
	Precision		**84.49±5.31**		76.82±7.57		10.74±5.84		83.76±1.44
	Recall		**96.93±1.30**		93.78±1.81		87.34±12.64		92.73±2.11
magic	G-mean	(2^10, 290)	**95.26±0.50**	(2^10, 300)	95.51±0.45	(210, 300)	81.72±0.37	(2e-1, 50)	81.43±0.15
	AUC		**95.62±0.49**		95.27±0.62		82.18±0.31		88.08±0.24
	Accuracy		**95.6±0.50**		95.25±0.54		80.86±1.09		85.19±1.34
	Precision		**97.52±0.29**		96.66±0.46		70.87±1.52		89.35±1.45
	Recall		**93.15±1.01**		93.00±0.79		94.54±0.76		88.13±0.82

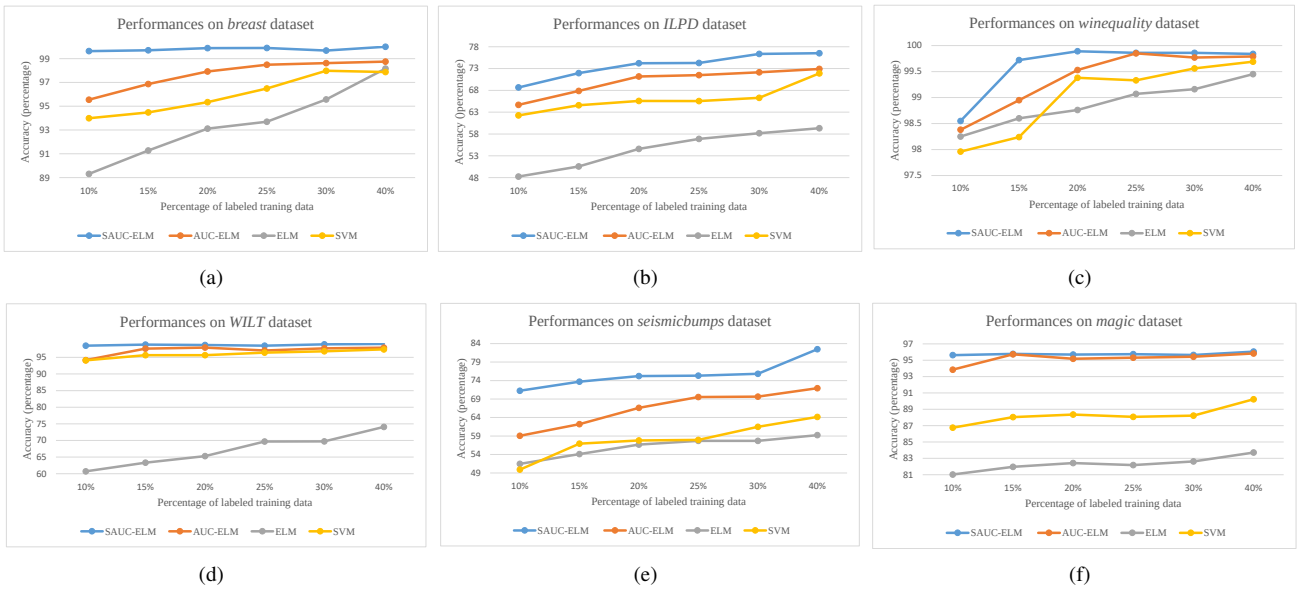

Fig. 2: AUC performances with respect to different percentages of labeled training data

ACKNOWLEDGMENT

We thank all the anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions. Guanjin Wang would like to thank School of Engineering and Information Technology in Murdoch University for the New Staff Start-up Grant (SEIT NSSG).

REFERENCES

[1] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via

learning discriminative cnns," *IEEE transactions on geoscience and remote sensing*, vol. 56, no. 5, pp. 2811–2821, 2018.
[2] A. Kumar, V. Bajaj, G. K. Singh *et al.*, "An improved fuzzy min-max neural network for data classification," *IEEE Transactions on Fuzzy Systems*, 2019.
[3] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.
[4] M. Gen and L. Lin, "Genetic algorithms," *Wiley Encyclopedia of Computer Science and Engineering*, pp. 1–15, 2007.
[5] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[6] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew *et al.*, "Extreme learning machine: a new learning scheme of feedforward neural networks," *Neural Networks*, vol. 2, pp. 985–990, 2004.
[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.
[8] Z. Liu, Y. Liu, J. Dezert, and F. Cuzzolin, "Evidence combination based on credal belief redistribution for pattern classification," *IEEE Transactions on Fuzzy Systems*, 2019.
[9] D. E. Rumelhart, G. E. Hinton, R. J. Williams *et al.*, "Learning representations by back-propagating errors," *Cognitive Modeling*, vol. 5, no. 3, p. 1, 1988.
[10] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.
[11] C. Chen, K. Li, A. Ouyang, Z. Tang, and K. Li, "GPU-accelerated parallel hierarchical extreme learning machine on flink for big data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2740–2753, 2017.
[12] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra, N. Mazzino, and D. Anguita, "Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2754–2767, 2017.
[13] J. Duan, Y. Ou, J. Hu, Z. Wang, S. Jin, and C. Xu, "Fast and stable learning of dynamical systems based on extreme learning machine," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, pp. 1175–1185, 2017.
[14] Y. Jia, S. Kwong, and R. Wang, "Applying exponential family distribution to generalized extreme learning machine," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, on-line available.
[15] J. Cao, K. Zhang, H. Yong, X. Lai, B. Chen, and Z. Lin, "Extreme learning machine with affine transformation inputs in an activation function," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 2093–2107, 2019.
[16] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.
[17] A. Ali, S. M. Shamsuddin, A. L. Ralescu *et al.*, "Classification with class imbalance problem: a review," *Int. J. Advance Soft Compu. Appl*, vol. 7, no. 3, pp. 176–204, 2015.
[18] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2008.
[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[20] C. Elkan, "The foundations of cost-sensitive learning," in *International Joint Conference on Artificial Intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.
[21] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.
[22] K. Li, X. Kong, Z. Lu, L. Wenyin, and J. Yin, "Boosting weighted elm for imbalanced learning," *Neurocomputing*, vol. 128, pp. 15–21, 2014.
[23] Y. Zhang, B. Liu, J. Cai, and S. Zhang, "Ensemble weighted extreme learning machine for imbalanced data classification based on differential evolution," *Neural Computing and Applications*, vol. 28, no. 1, pp. 259–267, 2017.
[24] W. Xiao, J. Zhang, Y. Li, S. Zhang, and W. Yang, "Class-specific cost regulation extreme learning machine for imbalanced classification," *Neurocomputing*, vol. 261, pp. 70–82, 2017.
[25] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
[26] C. Cortes and M. Mohri, "AUC optimization vs. error rate minimization," in *Advances in Neural Information Processing Systems*, 2004, pp. 313–320.
[27] Z. Yang, T. Zhang, J. Lu, D. Zhang, and D. Kalui, "Optimizing area under the ROC curve via extreme learning machines," *Knowledge-Based Systems*, vol. 130, pp. 74–89, 2017.
[28] U. Brefeld and T. Scheffer, "AUC maximizing support vector learning," in *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning*, 2005.
[29] A. Holst *et al.*, "Efficient auc maximization with regularized least-squares," in *Tenth Scandinavian Conference on Artificial Intelligence: SCAI 2008*, vol. 173. IOS Press, 2008, p. 12.
[30] L. Zhou, K. K. Lai, and J. Yen, "Credit scoring models with auc maximization based on weighted svm," *International journal of information technology & decision making*, vol. 8, no. 04, pp. 677–696, 2009.
[31] P. Zhao, S. C. Hoi, R. Jin, and T. YANG, "Online AUC maximization," 2011.
[32] J. Hu, H. Yang, M. R. Lyu, I. King, and A. M.-C. So, "Online nonlinear auc maximization for imbalanced data sets," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 882–895, 2017.
[33] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.
[34] J. Liu, Y. Chen, M. Liu, and Z. Zhao, "Selm: semi-supervised elm with application in sparse calibrated location estimation," *Neurocomputing*, vol. 74, no. 16, pp. 2566–2572, 2011.
[35] G. Huang, S. Song, J. N. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2405–2417, 2014.
[36] Y. Gu, Y. Chen, J. Liu, and X. Jiang, "Semi-supervised deep extreme learning machine for wi-fi based localization," *Neurocomputing*, vol. 166, pp. 282–293, 2015.
[37] Y. Zhou, B. Liu, S. Xia, and B. Liu, "Semi-supervised extreme learning machine with manifold and pairwise constraints regularization," *Neurocomputing*, vol. 149, pp. 180–186, 2015.
[38] S. Liu, L. Feng, H. Wang, and Y. Xiao, "Extend semi-supervised elm and a frame work," *Neural Computing and Applications*, vol. 27, no. 1, pp. 205–213, 2016.
[39] X. Jia, R. Wang, J. Liu, and D. M. Powers, "A semi-supervised online sequential extreme learning machine method," *Neurocomputing*, vol. 174, pp. 168–178, 2016.
[40] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 31–34, 2013.
[41] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, 2015.
[42] F. Du, J. Zhang, N. Ji, G. Shi, and C. Zhang, "An effective hierarchical extreme learning machine based multimodal fusion framework," *Neurocomputing*, vol. 322, pp. 141–150, 2018.
[43] H. Yu, X. Yang, S. Zheng, and C. Sun, "Active learning from imbalanced data: a solution of online weighted extreme learning machine," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 4, pp. 1088–1103, 2018.
[44] L. Hu, Y. Chen, J. Wang, C. Hu, and X. Jiang, "OKRELM: online kernelized and regularized extreme learning machine for wearable-based activity recognition," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 9, pp. 1577–1590, 2018.
[45] X. Yuan, L. Xie, and M. Abouelenien, "A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data," *Pattern Recognition*, vol. 77, pp. 160–172, 2018.
[46] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.
[47] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, 2013.
[48] S. Shukla and R. N. Yadav, "Regularized weighted circular complex-valued extreme learning machine for imbalanced learning," *IEEE Access*, vol. 3, pp. 3048–3057, 2015.
[49] B. S. Raghuwanshi and S. Shukla, "Class-specific extreme learning machine for handling binary class imbalance problem," *Neural Networks*, vol. 105, pp. 206–217, 2018.
[50] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, no. Nov, pp. 2399–2434, 2006.

[51] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32–48, 2015.
[52] H. Pei, K. Wang, Q. Lin, and P. Zhong, "Robust semi-supervised extreme learning machine," *Knowledge-Based Systems*, vol. 159, pp. 203–220, 2018.
[53] S. J. Mason and N. E. Graham, "Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation," *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, vol. 128, no. 584, pp. 2145–2166, 2002.
[54] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.
[55] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011.
[56] N. Settouti, M. E. A. Bechar, and M. A. Chikh, "Statistical comparisons of the top 10 algorithms in data mining for classification task," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 1, pp. 46–51, 2016.
[57] A. Asuncion and D. Newman, "UCI machine learning repository," 2007.
[58] D. Díaz-Vico and J. R. Dorronsoro, "Deep least squares fisher discriminant analysis," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

Guanjin Wang received the join Ph.D degree in software engineering from University of Technology Sydney and The Hong Kong Polytechnic University. She is currently a lecturer in Information Technology with Discipline of Information Technology, Mathematics and Statistics, Murdoch University, Perth, Australia. Her current research interest lies in the areas of machine learning and health informatics.

Kok Wai Wong is an Associate Professor with the Discipline of Information Technology, Mathematics and Statistics at the College of Science, Health, Engineering and Education at Murdoch University in Western Australia. He is the current Vice President (Membership) for The Asia Pacific Neural Network Society (APNNS). He is a Senior Member of Institute of Electrical and Electronics Engineers (IEEE), a Senior member of Australia Computer Society (ACS), and Certified Professional of ACS. He is also the current chapter chair for IEEE Computer Intelligence Society (WA Chapter). Kok Wai Wong has involved in the editorial boards for a number of international journals and in many international conference organising committees. His current research interests include Intelligent Data Mining, Artificial Intelligence and Machine Learning.

Jie Lu is a Distinguished Professor, the Director of the Center for Artificial Intelligence, and the Associate Dean (Research Excellence) with the Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia. She received her Ph.D. in information systems from the Curtin University of Technology, Australia, in 2000. Her research expertise spans transfer learning, artificial intelligence, recommender systems, concept drift, and their applications in e-business. She has published 10 research books and over 400 papers in refereed journals and conference proceedings, with over 170 papers in IEEE Transactions and other international journals. She has been awarded eight Australian Research Council (ARC) Discovery Project grants and many other research grants. She is a member of the ARC College of Experts. She serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier), Editor-In-Chief for the International Journal on Computational Intelligence Systems (Atlantis), Associate Editor for IEEE Transactions on Fuzzy Systems, Editor for a book series on Intelligent Information Systems (WorldScientific), and has served as a guest editor of 12 special issues, general/PC/organization chairs for ten international conferences as well as having delivered 20 keynote/plenaryspeeches at IEEE and other international conferences. She is a Fellow of IEEE and Fellow of IFSA.