Integrated Autoencoder-Level Set Method Outperforms Autoencoder for Novelty Detection

Integrated Autoencoder-Level Set Method Outperforms Autoencoder for Novelty Detection

Shuo Liu
Intelligent Systems Research Centre
University of Ulster
Derry~Londonderry
liu-s17@ulster.ac.uk

Xuemei Ding
Intelligent Systems Research Centre
University of Ulster
Derry~Londonderry
x.ding@ulster.ac.uk

Damien Coyle
Intelligent Systems Research Centre
University of Ulster
Derry~Londonderry
dh.coyle@ulster.ac.uk

for the Alzheimer's Disease
Neuroimaging Initiative*

***Abstract*— Novelty detection (ND) has gained attention in many applications for its effectiveness in dealing with imbalanced data. Many ND algorithms have been proposed. For example, the level set boundary description (LSBD) algorithm can accurately estimate a boundary around normal data which is subsequently used to detect novelties. However, the computational complexity and the convergence time of the LSBD algorithms increases substantially when data dimensionality increases. To solve those challenges, we propose an Integrated Autoencoder-Level Set Method (AE-LSM) for ND in this paper. The AE structure is employed to reduce the feature space with high dimensionality to a 3-dimensional (3D) space. The LSM algorithm is trained based on the compressed 3D data to identify the boundary of normal data. The AE-LSM has advantages of boundary control and good generalization performance. Experiments on 5 benchmark UCI datasets and an Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset demonstrate that the proposed AE-LSM present a 3%~14% significant improvement based on the average AUC ($p<0.05$) over the AE and LSBD algorithms across the six datasets.**

Keywords—novelty detection, level set methods, autoencoder, level set boundary description

I. INTRODUCTION

Novelty detection (ND), also known as one-class classification or anomaly/outlier detection, has attracted a lot of research interest across a range of applications, where abnormal data are limited or are extremely rare, e.g., credit card [26], mobile phone fraud detection [1], mobile robotics [2], sensor networks[3], rumor detection [4],[5], video surveillance [6], [7], [8] and healthcare [9],[10],[11] areas. ND aim to train the detectors with the target (normal, negative) class and then identify the deviated data as novelties (abnormal, positive) class using the trained detectors.

Many algorithms have been proposed for ND, such as K-Nearest Neighbors (KNN) [12], Extreme Learning Machine (ELM) [13], Self-Organizing Maps (SOMs) [14], one-class support vector machine [15], Ensemble methods [16],[17], autoencoder (AE) [18], and deep learning [19],[20]. Most of the ND algorithms identify the novelties according to an anomaly score. If the anomaly score of a data point is larger than the threshold, the point is considered to be abnormal data. Otherwise, it is normal data. Some ND algorithms identify the novelties by constructing a boundary around the normal data, such as the level set boundary description (LSBD) method [21]. Autoencoders (AEs) have been widely used in ND. The basic idea of AE for ND is that the reconstruction/prediction error is used as the anomaly score to identify whether a new data is normal or anomaly data [18]. Some variations of AE for ND have been proposed to improve the performance and robustness. More details of variations of AE can be found in [22],[23], [24] [25], [26], [27].

The basic idea behind ND algorithms is to construct boundaries enclosing the normal data in the training phase. The trained boundaries will be used to identify abnormal data in the testing phase. In real-world applications, the boundaries are usually nonlinear which is not easy to represent directly in the input space. Ding et al. [28] propose a kernel density estimation (KDE) based level set method (LSM) to construct a minimum volume surface around the normal data by dynamically controlling the boundary. They first learn the distribution of the given normal data with KDE to get the initial boundary from the zero-level set function (LSF). Then the boundary shrinks or expands in the normal direction with a constant velocity. By using the final decision boundary, a new coming data point will be classified into normal or abnormal according to whether it lies inside or outside the boundary. In [21], the approach to find data points outside of the boundary and the final decision is simplified in [28]. They propose a level set boundary description (LSBD) method, where the sign of the LSF determines the location of data points. To make the decision boundary more effective and efficient, they further propose a locally adaptive boundary description (LALSBD) method to evolve the boundary locally instead of globally [29]. The LALSBD can then evolve the boundary smoothly and describe data distribution with different shapes which perform better than the LSBD. However, the LALSBD only works well when less than 5% of normal data are permitted to be misclassified in the training phase.

There are many advantages of LSM-based algorithms: 1) complex shape boundaries in the input space [30] can be directly constructed; 2) more flexibility than parametric representations since the boundary can shrink/expand in a specific direction with a given speed [31]; 3) the motion of the boundaries can be managed, i.e., merging and splitting [21]. However, the implementation of LSBD for high-dimensional datasets is cumbersome since it is implemented on a discretized Cartesian grid where the values of LSF are maintained and updated. However, the grid nodes grow exponentially with increasing dimensions. Hence, in each iteration, a large number of points need to be stored and updated for high-dimensional datasets, which is time-consuming and computationally complex [31].

One direction to solve this challenge is dimensionality reduction, which can compress the high dimensional data into lower-dimensional space and subsequently employ the compressed data to learn the LSBD. The commonly used dimensionality reduction algorithms are linear principal

component analysis (PCA), kernel PCA and AE. Compared with linear PCA, AEs are able to extract nonlinear presentations of the features [32]. It has been proven that the performance of AE for dimensionality reduction is better than the PCA-based algorithms [18]. Moreover, the LSM has been combined with AEs for image segmentation [33], [34], [35], [36] and as we mentioned above, AE has been used extensively in ND. Nevertheless, we are unaware of any research that focuses on combining AE with LSM for ND in high-dimensional datasets. Hence to address the challenge experienced by LSBD algorithm for high dimensional data, we propose an Integrated Autoencoder-LSM (AE-LSM) algorithm which employs the AE to reduce the dimensions of the data and then the compressed features are used to train the LSBD algorithm to improve the performance of the LSBD algorithm on the datasets with more than three dimensions.

Our contributions are four-fold. First, we propose a hybrid ND algorithm by combining the AE and LSBD. Second, we propose a new strategy to control the evolution speed in the AE-LSM algorithm to make sure the algorithm can reach the final boundary. Third, we compare the performance of the proposed AE-LSM algorithm with the AE and LSBD algorithms to clarify the property of the proposed algorithm. We show that the proposed AE-LSM outperforms the other two algorithms on high dimensional (>3D) datasets. It is faster than the LSBD algorithm. Moreover, the proposed AE-LSM algorithm retains the advantages of LSBD. Finally, we visualize the evolution of boundaries on the high-dimensional data.

II. Related Works

In this section, the basic theory behind the AE and LSBD are introduced.

A. Autoencoder

Autoencoder is a particular type of unsupervised feed-forward neural network where the size of the input and output layer of network structure are equivalent, but the middle hidden layer presents a bottleneck which is an encoded version of the input. AE thus includes an encoder and decoder structure. Given a dataset $x=\{x_1, \ldots, x_N\}^d$, N and d denote the number of the samples and dimensions of the dataset, the encoder can be presented by the standard neural network function

$$Z=\sigma(Wx+b) \tag{1}$$

where Z is the latent dimension, σ denotes the transition function, W and b are the weight and bias between the input layer and bottleneck layer. The decoder can be written as,

$$\hat{x}=f\left(W^{'}Z+b^{'}\right) \tag{2}$$

where $\hat{x}$ denotes the output which is the reconstructed input, f is the transition function for the decoder, $W^{'}$ and $b^{'}$ denote the weight and bias between the bottleneck and output layers. The objective loss function is the mean squared error (MSE) which measures the reconstruction error.

$$\text{MSE}=\frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{x_i})^2 \tag{3}$$

The aim of AE is to minimize the MSE by updating the weight and bias. For ND, the MSE is used as the anomaly score. The MSE is small if the test data are normal data, while it becomes large with abnormal data [18]. For dimensionality reduction, the objective function is

$$\min \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2}\|\mathrm{x}_i-\hat{\mathrm{x}}_i\|^2\right)+\frac{\gamma}{2}\sum_{l=1}^{n_l}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(W_{ji}^{l})^2 \tag{4}$$

where n_l is the number of layers in the AE network and s_l denotes the number of nodes in the corresponding layer. γ is the regularization parameter.

B. Level set boundary description

LSBD algorithm is a density-based ND algorithm [28]. There are four steps to building LSBD: 1) constructing the level set function (LSF) and finding the initial boundaries; 2) evolving boundaries; 3) stopping the evolution; 4) evaluating the final boundary with testing data. The boundary evolution is controlled by a level set equation (LSE), i.e.

$$\frac{\partial\varphi}{\partial t}+\vec{V}\cdot\nabla\varphi=0 \tag{5}$$

where $\frac{\partial\varphi}{\partial t}$ denotes the partial derivative of the implicit LSF, φ, with respect to the time variable, t, i.e., the pseudo time, $\vec{V}$ is the velocity field and ∇ is the gradient operator. The $\vec{V}$ includes a normal component, V_n, and a tangent component V_t. While only the V_n influences the movement of the boundary; the LSE can be written as:

$$\varphi_t+V_n|\nabla\varphi|=0 \tag{6}$$

where, $|\nabla\varphi|$ is the norm of the gradient of φ.

The initial boundary usually is defined as a signed distance function with $|\nabla\varphi|=1$. The initial boundary is constructed based on the KDE, which is defined by:

$$f(x)=\frac{1}{Nh^d}\sum_{i=1}^{N}K(\frac{x-x_i}{h}) \tag{7}$$

where $K(\cdot)$ is a kernel function, h denotes the kernel bandwidth. Selecting h is very important for convergence. If h is too small, there will be many small boundaries around. While if it is too large, the boundary will be over-smoothed. The Gaussian kernel is used in this paper.

After estimating the density of the dataset, the Laplacian operator Δ is used to obtain the initial boundary, the zero-level set of φ. The initial boundary with Gaussian kernel is described as

$$\varphi=\Delta f(x)=\sum_{i=1}^{N}\frac{\|x-x_i\|^2-h^2}{Nh^{4+d}(2\pi)^{d/2}}\exp\left(-\frac{\|x-x_i\|^2}{h^2}\right) \tag{8}$$

An example of the implicit function (LSF) and the obtained initial boundaries on the two-dimensional (2D) XOR dataset is shown in Fig 1. The blue points are from the normal class. The grid surface represents the implicit φ. The two black closed curves are the initial boundaries.

Next, the boundary evolves according to the threshold λ which is a hyperparameter that must be set. In this algorithm, λ is the percentage of normal points rejected, called the expected false positive rate (FPr). For instance, if we permit 10% of training samples to be misclassified as the abnormal class, then λ=0.1. Although we know all the training data are from the normal class, rejecting a small fraction of training samples helps the classifier to learn the most representative model from the training data [37]. If the current FPr (λ_t) is smaller than λ, the boundaries will shrink by solving the LSE

*Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

to include more normal data. Otherwise, boundaries will expand to include less normal data. The evolution of the boundary will stop when $\lambda_t \in [\lambda-\varepsilon, \lambda+\varepsilon]$, (where ε is a small positive number) or when a maximum time threshold is exceeded. Finally, with the obtained boundaries, the features inside of the boundaries ($\varphi<0$) are classified as normal data, whilst features of the data lying outside of the boundaries ($\varphi>0$) are identified as abnormal data.

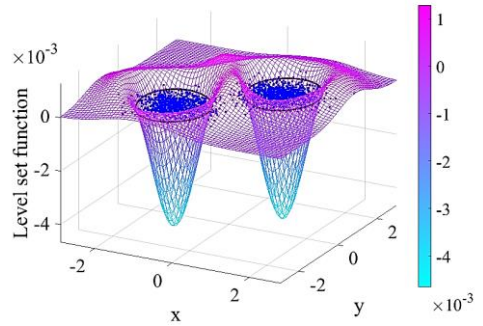

Fig 1. Implicit function φ and initial boundary based on Gaussian kernel density estimation on the XOR dataset.

In LSBD algorithm, the evolution speed is a very important parameter for capturing the final boundary. Ding et al. [21] set the initial speed as 0.25 and then dynamically changed the speed during the boundary evolution. If the current $\lambda_t < \lambda-\varepsilon$ while $\lambda_{t+1} > \lambda+\varepsilon$, then the speed V_t will be adjusted to $\frac{V_t}{2}$ to slow the boundary evolution. If the $\lambda_{t+1}=\lambda_t$, the speed V_t changes to $2*V_t$. If the final boundary cannot be found, then ε was increased by 10%. However, changing ε will change the performance of the trained boundary on the normal data. Moreover, for different datasets, the standards of the final boundary are different. In addition, the LSBD is time-consuming and determining the final boundary within the set time maybe not occur. Therefore, we propose the AE-LSM algorithm with speed adjust.

III. AUTOENCODER-BASED LEVEL SET METHOD

In this section, we introduced the structure of the proposed AE-LSM algorithms.

(a) Structure of the proposed AE_LSM

The proposed hybrid AE-LSM algorithm for ND includes two parts: dimensional reduction and discriminator. The AE is used to reduce the dimensions of high-dimensional datasets. The encoded data are used as the input of the LSBD algorithm which acts as a discriminator to detect the abnormal data.

The structure of the proposed AE-LSM algorithm is shown in Fig 2. First, the whole dataset is normalized with the mean value and standard deviation of the training data. Then the standardized training data are learned by AE with 3 hidden nodes in the hidden layer (bottleneck). The LSBD model is applied to the compressed 3D training data. In the testing step, the test data are compressed into 3D with the trained AE-LSM model. They are classified depending on the location of the data points. Data points outside of the boundaries are classified as abnormal data.

In this paper, we test the proposed AE-LSM on up to 18D datasets and compare it with the AE and LSBD algorithms alone on 6 benchmark datasets. Note that the structure of the AE and the proposed AE-LSM are different. For both algorithms, a 3-layer network with 3 hidden neurons is used. However, the transition functions and training algorithms are different. For example, for the proposed AE-LSM, the objective of using AE is to obtain nonlinear features representation to reduce dimensionality. Therefore, the nonlinear transition function is used for the encoder. The linear transition function is used for the decoder and the SCG backpropagation is used to train the network. However, the transition functions of encoder and decoder for the benchmark AE algorithm used alone is selected with nested cross validation (nCV) [38] which is detailed in the experiment section.

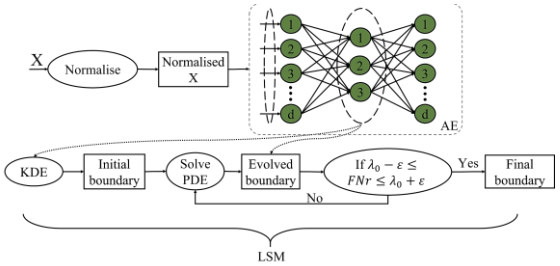

Fig 2. The structure of the proposed AE-LSM

b) Speed adjust algorithm

For the AE-LSM algorithm, instead of adjusting the speed with double or half the original speed or adapting ε, we adjust the speed based on the distance between λ_t and λ.

Given the initial speed V_0, λ, ε and the current FPr (λ_t) obtained from the initial boundary. We first calculate the difference e_t between λ_t and λ respectively. Then if the $\lambda_t \notin (\lambda-\varepsilon, \lambda+\varepsilon)$, the boundary needs to evolve. For each iteration, we calculate the new error between the current λ_t and λ and compare the signs of the new error e_t and e_{t-1}. If the signs of the error are the same, the current boundary and the previous boundary are at the same side of the target boundary. The boundary will evolve in the same direction. Otherwise, the boundary will evolve in the opposite direction. We set $\frac{1}{3}\lambda$ as a threshold to determine the change of the speed. If the error is larger than $\frac{1}{3}\lambda$, the speed will vary more to meet the termination criteria, while if the error is that less than $\frac{1}{3}\lambda$, the speed changes slowly to avoid missing the target boundary. With $|e_t| > \frac{1}{3}\lambda$, if the signs of the error are positive, the speed V_t changes to $2*V_{t-1}$. If the sign of the error is negative, the current speed is too large and cause a large step in the evolution but exceeds the target boundary. In this situation, we reset the boundary, speed and FPr using the boundary with the corresponding V_{t-1} and λ_{t-1} obtained before this evolution and then reduce the speed to $V_t = \frac{1}{2}V_{t-1}$. With $|e_t| < \frac{1}{3}\lambda$, if signs of the error are positive, the speed decreases to $V_t = \frac{1}{4}V_{t-1}$. We want the boundary to evolve slowly to get the final boundary. If signs of the error are negative, the speed $V_t = \frac{1}{2}V_{t-1}$. The details of the speed adjust algorithm is shown in Fig 3. In the experiments, we set V_0=0.15.

IV. EXPERIMENTS

A. Datasets

Five UCI benchmark classification datasets and one real Alzheimer's disease dataset, the ADNI dataset, are used in experiments. The dataset specifications are shown in TABLE

I. All the features are normalized in the range [0, 1]. For the ADNI dataset, we employ the stable mild cognitive impairment (sMCI) patients who remain MCI over the test time as the normal class. The MCI patients who develop AD within the following two years, called progressive MCI (pMCI) patients, are used as the abnormal class.

Speed adjust algorithm

Initial speed V_0, Expected FPr λ,
Initial FPr λ_1 with the initial boundary
The boundaries, B

// For each iteration t
$e_t=\lambda_1-\lambda$ //The distance between the current λ_1 and the target λ_0.
$e_t=e_t$ // save the error
$\lambda_t=\lambda_1$ // save the current FPr
==========speed adjust==========
If $\lambda_1 \notin (\lambda-\varepsilon, \lambda+\varepsilon)$
  $e_t=\lambda_t-\lambda$ // The distance between the current λ_t and the target λ.
  $e\_s=e_t * e_{t-1}$ // check the sign change
                // between the two errors.
  If $|e_t|>\frac{1}{3}\lambda$
    if $e\_s>0$ // if the sign of the error is not change
      $V_t=2*V_{t-1}$
    elseif $e\_s<0$
      $B_t=B_{t-1}$
      $\lambda_t=\lambda_{t-1}$
      $V_t=\frac{1}{2}V_{t-1}$
    endif
  Elseif $|e_t|<\frac{1}{3}\lambda$
    if $e\_s>0$
      $V_t=\frac{1}{2}V_{t-1}$
    elseif $e\_s<0$
      $B_t=B_{t-1}$
      $\lambda_t=\lambda_{t-1}$
      $V_t=\frac{1}{2}V_{t-1}$
    endif
  Endif

Endif

Fig 3. Speed adjust algorithm

The data used in this study were obtained from Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu), the most frequently used dataset to develop computational approaches to predict the MCI subjects were at high risk for converting to AD early. The ADNI was launched in 2003 as a public-private partnership led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). For up-to-date information, see www.adni-info.org.

TABLE I. UCI datasets and ADNI dataset specifications

Datasets	Dim	Normal	Abnormal	Target class
Svmguide	4	4000	3089	1
Phoneme	5	3818	1586	1
Thyroid	6	3679	93	0
Breast cancer	9	186	77	1
Image	18	1188	898	1
ADNI	4	681	379	SMCI

B. *Experiment Setup*

It has been mentioned we employ nCV [38] to optimize the parameters for the AE ND algorithm to thoroughly validate the method without leaking between training and test data when optimizing hyperparameters and ensure proper generalization evaluation.

The nCV includes a 5-fold inner CV for tunning parameters and a 5-fold outer CV for evaluating the performances of trained AE. The experiment nCV structure is shown in Fig 4. There are three steps for splitting the dataset in nCV:

1) Splitting the whole dataset into normal data and abnormal data according to their labels and splitting the normal and abnormal data into 5 parts.

2) Selecting 20% of normal and abnormal data randomly as the independent testing dataset. The remaining 80% of the data are used for training and validation.

3) Splitting the 80% normal data into 5 folds for inner CV. In each iteration, one fold of normal data is combined with the 80% abnormal data as the validation dataset which is used to validate the performance of the algorithms trained with the remaining 4-fold normal data.

Constructing ND contains (1) Train and validation, s1.Train and s2.Validation in Fig 4. After training the algorithms with the training set, the validation set is used to validate the trained algorithms to evaluate their training performance for obtaining the optimized parameters. (2) Test (s3.Train and s4.Test in Fig 4). In this phase, the optimized parameters and 80% of all the normal data are used to train the corresponding algorithms, and the trained models are tested with the independent testing set. These two steps repeat 5 times with different parts of the training data for eliminating bias.

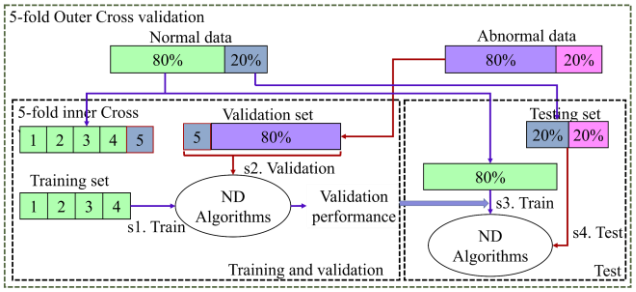

Fig 4. The structure of nested cross validation.

In the experiments, only the AE algorithm employs the nCV structure to find the optimal transfer functions for encoder and decoder and the best training algorithm. The transition function of the encoder is selected in the sigmoid and saturating liner (satlin) functions. The transition function of the decoder is chosen in the functions of [sigmoid, satlin, tan sigmoid, pure linear]. Training functions are used to update the weight and bias of the network. In this paper, training functions are chosen from the BFGS quasi-Newton backpropagation (BFG), Scaled conjugate gradient backpropagation (SCG), Gradient descent with momentum and adaptive learning rate backpropagation, and Levenberg-Marquardt backpropagation. The learning rate is set as 0.1, the maximum number of training epochs is 1000, γ=0.001, λ=0.1, and ε=0.01 . For the LSBD and the proposed AE-LSM algorithms, only the 5-fold outer CV is conducted in the

experiments since all hyperparameters are set and are not required to be optimized in these two algorithms.

C. Evaluation Measures

The area under the receiver operating characteristic curve (AUC) and adjusted F1 score (AGF) [39] are used as the measurements for performance evaluation. The definitions of different measures are shown below.

a) AUC. The AUC calculates the area under the receiver operating characteristic (ROC) curve showing the FPr and True positive rate across a continuum of thresholds [40]. It is a fundamental metric for evaluating a diagnostic model and has been commonly used in biomedical research to assess the classification and prediction performance for disease diagnosis and prognosis [41]. The AUC can be calculated with (10) [42].

$$\text{AUC} = 0.5*(\text{Sens}+\text{Spec}) \tag{10}$$

where

$$\text{Sens}=\text{Sensitivity}=\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{11}$$

$$\text{Spec}=\text{Specificity}=\frac{\text{TN}}{\text{TN}+\text{FP}} \tag{12}$$

TP is the true positive, TN is true negative, FP is the false positive and FN is false negative.

b) AGF. The AGF is an improved F-measure which ignores the true negative rate in the calculation. However, the AGF considers all the four elements of the confusion matrix to make the results more reliable [39]. The AGF first calculate the F_2 with the real labels and then the $F_{0.5}$ is calculated after the inversion of labels (*Inv* $F_{0.5}$). The AGF provides more weight to correctly classify patterns in the positive class (abnormal class). Higher AGF indicates better performance in identifying abnormal data.

$$\text{AGF}=\sqrt{F_2\times Inv\,F_{0.5}} \tag{13}$$

where,

$$F_2=5\times\frac{\text{Sens}\times\text{Prec}}{4\times\text{Prec}+\text{Sens}} \tag{14}$$

$$Inv\,F_{0.5}=\frac{5}{4}\times\frac{\text{Sens}\times\text{Prec}}{0.5^2\times\text{Prec}+\text{Sens}} \tag{15}$$

$$\text{Prec}=Precision=\frac{TP}{TP+FP} \tag{16}$$

All the experiments are carried out in the MATLAB R2019a environment using the Ian Mitchell toolbox [43] for the implementation of LSBD.

V. RESULTS AND DISCUSSION

In this paper, we select different encoder and decoder transition functions and network training functions to find the best functions for training AE for ND. After getting the optimized parameters, we train the proposed AE-LSM algorithm with the optimized parameters. However, even though the optimized parameters get the best classification performance with AE, they are unsuitable for the AE-LSM. Many compressed normal and abnormal data points obtained from the encoder with the best encoder transition function overlap Fig 5(A). Then we employ the sigmoid function as the transition function of the encoder since we aim to get a nonlinear representation of the data. The normal data and abnormal data are separated well (Fig 5 (B)). Therefore, to get the nonlinear transition of the original data, we used the sigmoid function as the transition function of the encoder.

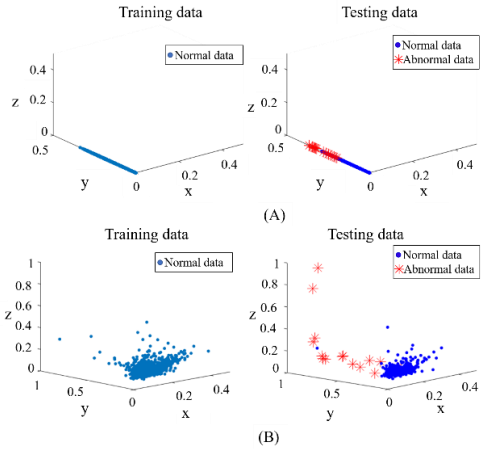

Fig 5. The example of Thyroid dataset after encoder with different transfer function and training algorithms. (A) The Saturating linear (satlin) as the encoder transfer function, Sigma function as the decoder transfer function and the BFG as the training. (B) The sigmoid function is the encoder transfer function, linear function as the decoder transfer function and the SCG as the training.

The average values of AUC (±std) of the outer CV is shown in TABLE II. The proposed AE-LSBD algorithm obtains a significantly higher AUC on 5 datasets, whilst the AUC of AE (0.9334±0.0189) is slightly higher than the AUC of AE-LSM (0.9229±0.0386) on the thyroid dataset. For Svmguide, ADNI, BreastCancer and Image datasets, the AE-LSM provides around 3%~14% increase on the AUC over the highest results obtained by LSBD and AE.

For the BreastCancer and Image datasets, the LSBD fails to find the final boundaries with speed suggested in [28]. We adjust the speed in our experiments to find the final boundary, but it still fails; however, with AE applied for dimensionality reduction, the AE-LSBD performs well.

TABLE II. The average AUC±std over 5-fold outer CV for the LSBD, AE and AE-LSM algorithms.

Datasets	LSBD	AE	AE-LSM
Svmguide	0.7582±0.0142	0.7798±0.0248	**0.9187±0.011**
Phoneme	0.6470±0.0234	0.6038±0.0363	**0.6471±0.0212**
Thyroid	0.6201±0.122	**0.9334±0.0189**	0.9229±0.0386
ADNI	0.6715±0.0221	0.6532±0.0656	**0.7143±0.0189**
BreastCancer	--	0.5634±0.0637	**0.6216±0.0540**
Image	--	0.6569±0.0428	**0.7847±0.0495**

The t-test based on the average AUC is conducted (TABLE III). It can be seen that the proposed AE-LSM significantly outperforms AE ($p=0.0139 < 0.05$) based on all the datasets and tends towards significantly outperforming LSBD ($p=0.0794$) based on the first four datasets.

TABLE III. The p-Value of the t-test based on the average AUC

p-Value	AE	AE-LSM
LSBD	0.2345	0.0794
AE		0.0139

Besides the AUC, the average values of the AGF over the 5-fold outer CV is shown in Fig 6. The proposed AE-LSM wins the highest AGF on 4 out 6 datasets. The AE gets the best AGF (0.3548) on the Thyroid dataset. The LSBD performs slightly better than the proposed AE-LSM on the ADNI

dataset with 0.6248 and 0.6228 AGF, respectively. The t-test results show that the proposed AE-LSM is significantly better than the AE on all the datasets ($p = 0.0045 < 0.05$) and, again, although not significant ($p = 0.0721$), achieves better than the LSBD on the first four datasets.

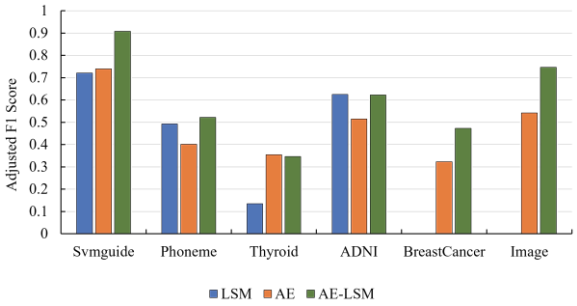

Fig 6. Comparisons of the adjusted F1 score (AGF).

An example of the boundary's evolution on a one-fold compressed Thyroid dataset is shown in Fig 7. The blue points are the normal points. The red stars are the abnormal points in the testing set. The red circles denote the normal data points out of the boundary, i.e. misclassified normal data. The black circles denote the abnormal data inside the boundary, i.e., misclassified abnormal data. The closed surfaces are boundaries. With the initial boundary (Fig 7. (a)), around 3.6345% of normal data is located outside the boundaries (FPr = 0.036345). Since the FPr is smaller than λ-ε, boundaries need to shrink to include more training data (see Fig 7. (b)~(c)). The boundaries become smaller and smaller until FPr = 0.10632 (Fig 7. (d)) that meets the termination condition ($FPr \in [\lambda-\varepsilon, \lambda+\varepsilon]$), the boundaries stop evolving. The final boundaries classify the test set in the test phase. The test results of the final boundary on the normal and abnormal class of the test data are present in Fig 7. (e) and (f), respectively. Around 9.6599% of normal data are misclassified as abnormal data, and 16.667% of abnormal data are misclassified as normal data.

VI. DISCUSSION

The experimental results show that the proposed AE-LSM has significantly better performance than LSBD and AE algorithms on most datasets. It improves the performance of LSBD on the datasets of higher three dimensions by compressing the data into 3D space with a 3-layer AE.

The results show that the LSBD algorithm fails to find the final boundaries on the BreastCancer and Image datasets. It may be because the LSBD evolves with a specific speed, the boundaries expand and shrink many times around the expected boundaries, but the evolution stops due to the time restriction imposed. For LSBD, setting proper running time and evolution speed is very important. However, it is challenging to optimize the two parameters. Compared with the LSBD, the proposed AE-LSM behaves better since we fix the dimensions of the dataset and the lower-dimensional data boundary is more manageable whilst the speed adjusts according to the distance between the target FPr and the current FPr. It can find the final boundaries with several evolutions faster than the LSBD algorithm.

Moreover, the boundaries evolve globally in the LSBD algorithm and the proposed AE-LSM algorithm in the experiment, which means that all the boundaries evolve with the same speed in each iteration (see Fig 7). Global evolution constructs the boundaries not only at high-density regions of the normal data but also the data at low-density regions. The boundaries around the low-density regions may misclassify more abnormal data. For example, the boundary enclosing 2 normal points in the final boundary (the first figure on the second row) in Fig 7 makes two abnormal data misclassified in the test phase (the last figure on the second row). Local evolution of the boundary helps to get more effective boundaries. Ding et al. [29] proposed an approach to evolve the boundaries of LSBD locally. However, their method only works well when the value of λ is very small, e.g., 0.001. When the λ is large, their method does not work. Therefore, developing a local-evolving algorithm is important.

The limitation of the proposed algorithm is that it does not work well on very high-dimensional datasets since reducing the very high-dimensional data into three neurons bottleneck may lose a lot of important information. To solve this limitation, we can use the deep AE [24] [25] to reduce dimensionality for high-dimensional data or apply LSBD on the dimensions associated with the optimal bottleneck for the AE.

VII. CONCLUSION

In this paper, we propose an Autoencoder-based level set method (AE-LSM) for unsupervised novelty detection. AE-LSM consists of two major components: AE network and LSBD detector, where the AE network projects samples into a low-dimensional space that preserves the key information for ND, and the detector is used to detect the abnormal data from new data points. We also develop a new speed adjust algorithm to help to find the final boundaries and speed up the algorithm.

The experimental results show that the proposed AE-LSM demonstrates superior performance over the AE and LSBD algorithms on public benchmark datasets and the ADNI dataset with up to 14% improvement on the standard AUC score. The advantages of the proposed AE-LSM: 1) it is efficient and time-saving. 2) it does not require large memory resources for high-dimensional data, and 3) it retains all the advantages of LSBD.

It is a known fact that there is no one algorithm performing best for all datasets. The performance of algorithms may change a lot depending on the data characteristics due to biases [44]. The AE-LSBD provides a new option when choosing an algorithm for ND. For further work, we will employ deep AE to work on higher dimensional datasets and improve the LSM algorithm to make it more effective, e.g., by employing locally evolving boundary estimation.

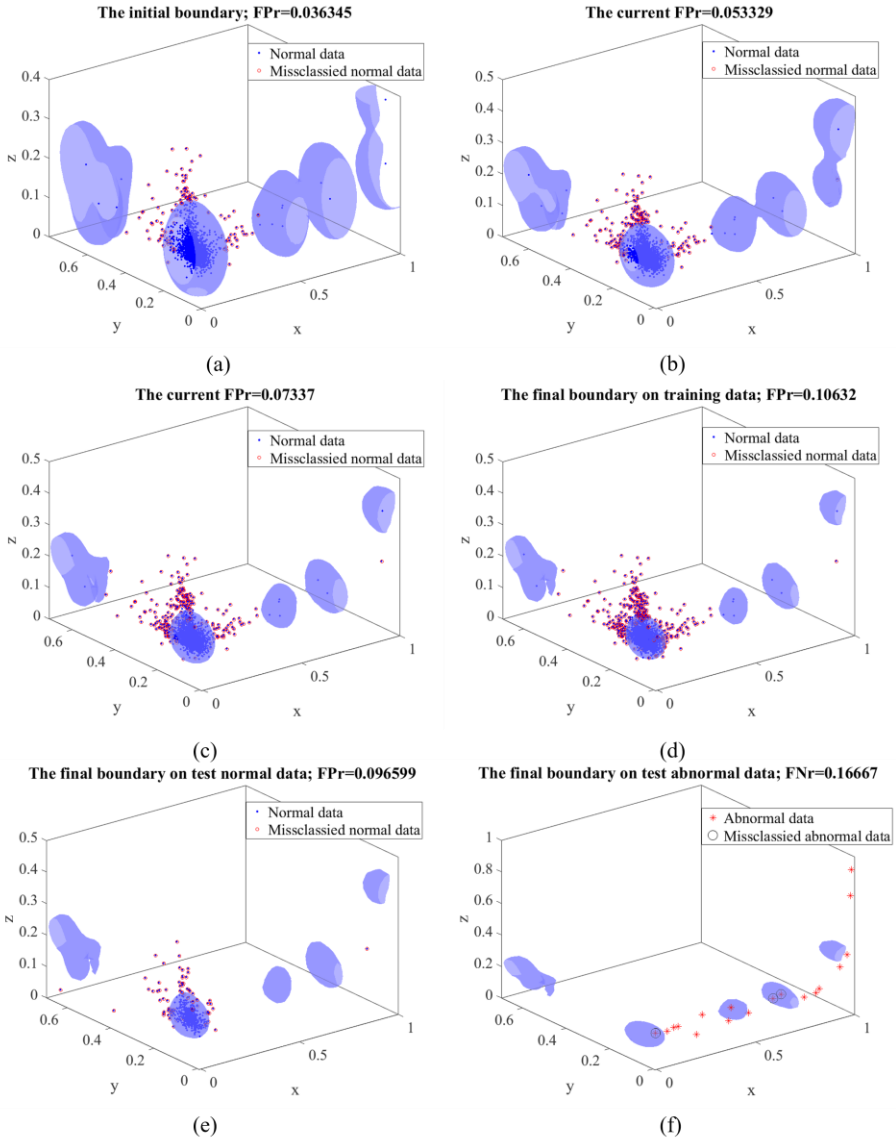

Fig 7. The boundaries evolution on the compressed Thyroid dataset. (a) Initial boundary; (b)~(c) boundary evolution; (d) final boundary on the training dataset; (e) and (f) final boundary on the normal class and abnormal class of testing data. The surfaces are the boundaries constructed by LSM and red circles denote the points out of the boundaries of the normal data, the black circles denote the misclassified abnormal points.

ACKNOWLEDGMENT

We are grateful for access to the Tier 2 High Performance Computing resources provided by the Northern Ireland High-Performance Computing (NI-HPC) facility funded by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant Nos. EP/T022175/ and EP/W03204X/1. DC is grateful for the UKRI Turing AI Fellowship 2021-2025 funded by the EPSRC (grant number EP/V025724/1). This project was supported by Vice Chancellor Research Scholarship, Ulster University, the Alzheimer's Research UK NI Networking, and the Global Challenges Research Fund Networking.

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

REFERENCES

[1] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A Review of Novelty Detection," Signal Processing, vol. 99, pp. 215–249, 2014.

[2] J. Ni, L. Wu, X. Fan, and S. X. Yang, "Bioinspired intelligent algorithm and its applications for mobile robot control: A survey," Comput. Intell. Neurosci., vol. 2016, no. January, 2016.

[3] D. Smolyakov, N. Sviridenko, E. Burikov, and E. Burnaev, "Anomaly pattern recognition with privileged information for sensor fault detection," Artif. Neural Networks Pattern Recognit., vol. 11081, pp. 320–332, 2018.

[4] A. Ebrahimi Fard, M. Mohammadi, Y. Chen, and B. Van de Walle, "Computational Rumor Detection Without Non-Rumor: A One-Class Classification Approach," IEEE Trans. Comput. Soc. Syst., vol. 6, no. 5, pp. 830–846, 2019.

[5] A. Bondielli and F. Marcelloni, "A Survey on Fake News and Rumour Detection Techniques," Inf. Sci. (Ny)., vol. 497, pp. 38–55, 2019.

[6] B. R. Kiran, D. M. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," J. Imaging, vol. 4, no. 2, pp. 1–15, 2018.

[7] L. K. Nair, "Prediction of Anomalous Activities in a Video," Int. Res. J. Eng. Technol., pp. 3470–3475, 2018.

[8] N. Patil and P. K. Biswas, "A survey of video datasets for anomaly detection in automated surveillance," Proc. - 2016 6th Int. Symp. Embed. Comput. Syst. Des. ISED 2016, pp. 43–48, 2017.

[9] C. Medrano, R. Igual, I. García-Magariño, I. Plaza, and G. Azuara, "Combining novelty detectors to improve accelerometer-based fall detection," Med. Biol. Eng. Comput., vol. 55, no. 10, pp. 1849–1858, 2017.

[10] S. W. Yahaya, A. Lotfi, and M. Mahmud, "A Consensus Novelty Detection Ensemble Approach for Anomaly Detection in Activities of Daily Living," Appl. Soft Comput. J., vol. 83, p. 105613, 2019.

[11] N. M. Rad, T. van Laarhoven, C. Furlanello, and E. Marchiori, "Novelty detection using deep normative modeling for imu-based abnormal movement monitoring in parkinson's disease and autism spectrum disorders," Sensors (Basel), vol. 18, no. 10, p. 3533, 2018.

[12] G. O. Campos et al., On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study, vol. 30, no. 4. Springer US, 2016.

[13] M. A. A. Albadr and S. Tiun, "Extreme Learning Machine: A Review," Int. J. Appl. Eng. Res., vol. 12, no. 14, pp. 4610–4623, 2017.

[14] L. Aguayo and G. A. Barreto, "Novelty Detection in Time Series Using Self-Organizing Neural Networks: A Comprehensive Evaluation," Neural Process. Lett., vol. 47, no. 2, pp. 717–744, 2018.

[15] S. Alam, S. K. Sonbhadra, S. Agarwal, and P. Nagabhushan, "One-Class Support Cector Classifiers: A Survey," Knowledge-Based Syst., vol. 196, p. 105754, 2020.

[16] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble Learning for Sata Stream Analysis: A Survey," Inf. Fusion, vol. 37, pp. 132–156, 2017.

[17] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," Comput. Secur., vol. 65, pp. 135–152, 2017.

[18] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," Proc. MLSDA 2014 2nd Work. Mach. Learn. Sens. data Anal., pp. 4–11, 2014.

[19] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," ArXiv, pp. 1–50, 2019.

[20] T. Czimmermann et al., "Visual-based defect detection and classification approaches for industrial applications—A SURVEY," Sensors (Switzerland), vol. 20, no. 5, pp. 1–25, 2020.

[21] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "Novelty Detection Using Level Set Methods," IEEE Trans. Neural Networks Learn. Syst., vol. 26, no. 3, pp. 576–588, 2015.

[22] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier Detection with Autoencoder Ensembles," in Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017, 2017, pp. 90–98.

[23] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly Detection using One-Class Neural Networks," ArXiv, 2018.

[24] T. Amarbayasgalan, B. Jargalsaikhan, and K. H. Ryu, "Unsupervised novelty detection using deep autoencoders with density based clustering," Appl. Sci., vol. 8, no. 9, 2018.

[25] B. Zong et al., "Deep autoencoding Gaussian mixture model for unspervised anomaly detection," Iclr2018, pp. 1–13, 2018.

[26] B. Eiteneuer, N. Hranisavljevic, and O. Niggemann, "Dimensionality reduction and anomaly detection for cpps data using autoencoder," Proc. IEEE Int. Conf. Ind. Technol., vol. 2019-Febru, pp. 1286–1292, 2019.

[27] Y. Liu et al., "Generative Adversarial Active Learning for Unsupervised Outlier Detection," IEEE Trans. Knowl. Data Eng., vol. 32, no. 8, pp. 1517–1528, 2020.

[28] X. Ding, Y. Li, A. Belatreche, and L. Maguire, "Constructing minimum volume surfaces using level set methods for novelty detection," Proc. Int. Jt. Conf. Neural Networks, pp. 10–15, 2012.

[29] X. Ding, Y. Li, A. Belatreche, and L. Maguire, "A Locally Adaptive Boundary Evolution Algorithm for Novelty Detection using Level Set Methods," in 2014 International Joint Conference on Neural Networks (IJCNN), 2014, pp. 1870–1876.

[30] X. Cai and A. Sowmya, "Level Learning Set : A Novel Classifier Based on Active Contour Models," in "Machine Learning: ECML 2007, 18th European Conference on Machine Learning, 2007, pp. 79–90.

[31] K. R. Varshney and A. S. Willsky, "Supervised learning of classifiers via level set segmentation," Proc. 2008 IEEE Work. Mach. Learn. Signal Process. MLSP 2008, pp. 115–120, 2008.

[32] J. A. Martinez-Heras and A. Donati, "Novelty Detection with Deep Learning," in 15th International Conference on Space Operations, 2018, 2018, no. June, pp. 1–6.

[33] K. Oda, R. Kawamata, S. Wakao, and N. Murata, "Fast Multi-Objective Optimization of Magnetic Shield Shape by Combining Auto-Encoder and Level-Set Method," IEEE Trans. Magn., vol. 57, no. 7, 2021.

[34] T. H. N. Le, K. G. Quach, K. Luu, C. N. Duong, and M. Savvides, "Reformulating Level Sets as Deep Recurrent Neural Network Approach to Semantic Segmentation," IEEE Trans. Image Process., vol. 27, no. 5, pp. 2393–2407, 2018.

[35] Z. Deng, Q. Guo, Z. Zhu, and D. Zhang, "Dynamic Regulation of Level Set Parameters Using 3D Convolutional Neural Network for Liver Tumor Segmentation," J. Healthc. Eng., vol. 2019, 2019.

[36] Y. Zhang et al., "Deep Learning Initialized and Gradient Enhanced Level-Set Based Segmentation for Liver Tumor from CT Images," IEEE Access, vol. 8, pp. 76056–76068, 2020.

[37] Q. Leng, H. Qi, J. Miao, W. Zhu, and G. Su, "One-Class Classification with Extreme Learning Machine," Math. Probl. Eng., vol. 2015, 2015.

[38] S. Parvandeh, H. W. Yeh, M. P. Paulus, and B. A. McKinney, "Consensus features nested cross-validation," Bioinformatics, vol. 36, no. 10, pp. 3093–3098, 2020.

[39] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted F-measure and kernel scaling for imbalanced data learning," Inf. Sci. (Ny)., vol. 257, pp. 331–341, 2014.

[40] D. J. Hand and R. J. Till, "A Simple Generalization of the Area Under the ROC Curve for Multiple Class Classification Problems," Mach. Learn., vol. 45, no. 2, pp. 171–186, 2001.

[41] K. Hajian-Tilaki, "Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation," Casp. J. Intern. Med., vol. 4, no. 2, pp. 627–635, 2013.

[42] C. Gautam, A. Tiwari, and Q. Leng, "On the construction of extreme learning machine for online and offline one-class classification—An expanded toolbox," Neurocomputing, vol. 261, pp. 126–143, 2017.

[43] I. M. Mitchell, "The flexible, extensible and efficient toolbox of level set methods," J. Sci. Comput., vol. 35, no. 2–3, pp. 300–329, 2008.

[44] K. R. Varshney and A. S. Willsky, "Classification using geometric level sets," J. Mach. Learn. Res., vol. 11, pp. 491–516, 2010.