Unsupervised Time Series Novelty Detection Using Clustering-based Local Autoencoders

Renan Fonteles Albuquerque
Department of Teleinformatics Engineering
Federal University of Ceará
Fortaleza, Brazil
renanfonteles@alu.ufc.br

Guilherme de Alencar Barreto
Department of Teleinformatics Engineering
Federal University of Ceará
Fortaleza, Brazil
gbarreto@ufc.br

***Abstract*—Novelty detection, also known as anomaly detection, plays a crucial role in identifying new or abnormal instances within a dataset. Traditional autoencoder models have been effective in learning compact representations of data, but they often struggle with capturing fine-grained local variations in complex and high-dimensional datasets. To address this limitation, we propose a novel learning method called Local Autoencoders (LAEs) for novelty detection. LAEs incorporate local information into the encoding and decoding processes, enabling more precise and detailed reconstructions. In this paper, we present preliminary results on evaluation of LAEs considering benchmark datasets for time series novelty detection and compare their performance against traditional (i.e., global) autoencoder and nearest neighbor learning method. The results demonstrate LAEs' competitive performance in detecting novel instances, surpassing (in several cases) traditional autoencoders. The proposed LAEs present a promising avenue for further exploration in the field of novelty detection, leading into new opportunities for research and practical applications.**

***Index Terms*—Local learning, Autoencoders, Novelty detection, Time series**

I. Introduction

In recent years, the field of artificial intelligence has witnessed remarkable advancements, especially in the area of deep learning. Autoencoders, a class of neural networks, have emerged as a powerful tool for unsupervised learning and dimensionality reduction [1]. They have been successfully applied in various domains, ranging from computer vision [2], [3] and natural language processing [4] to anomaly detection [3], [5]–[10]. Autoencoders have gained significant attention due to their ability to learn compact representations of input data by encoding and decoding it through a bottleneck layer (i.e., latent space). They are composed of two main components: an encoder network that maps the input data to a latent representation and a decoder network that reconstructs the input from the latent space. The training objective of an autoencoder is to minimize the reconstruction error.

Autoencoders have demonstrated exceptional performance in numerous applications. In computer vision, they have been employed for image restoration [11], image denoising [12], video surveillance [2], and feature extraction [1]. In natural language processing, autoencoders have been utilized for sentiment analysis [13] and document forgery detection [14]. Additionally, they have proven valuable in anomaly detection tasks, where they can learn to reconstruct normal instances and identify outliers in the data [1]. Despite their successes, traditional autoencoders face challenges when dealing with large-scale and high-dimensional datasets. The global nature of their encoding and decoding processes often leads to a loss of fine-grained details and fails to capture local variations in the data.

To address these limitations, we propose a new learning method called Local Autoencoders (LAEs). LAEs incorporate local information into the encoding and decoding processes, allowing for a more specific and less costly reconstruction of the input data. By leveraging local data clusters, LAEs enhance the expressive power of autoencoders and improve their performance in tasks like dimensionality reduction and novelty detection. Novelty detection consists of the task of identifying new or anomalous data in a data set. These new data, generally called anomalies or *outliers*, present divergent characteristics to a given set of data referred to as ordinary or normal. This paper presents preliminary findings of applying the Local Autoencoder (LAE) model in novelty detection. Additionally, the proposed method shows potential for exploration in diverse domains and applications.

The structure of this paper is organized as follows. Section II discusses the problem of novelty detection and the challenges associated with it. In addition, we introduce the autoencoder neural network, and discuss the state-of-the-art of autoencoders applied on novelty detection applications. Section III provides a comprehensive overview of the proposed Local Autoencoders (LAEs), describing their architecture and the incorporation of local information into the encoding and decoding processes. Section IV details the experimental setup and the dataset used to evaluate the performance of LAEs in the context of novelty detection. Section V presents the preliminary results obtained, demonstrating the effectiveness of the proposed model in detecting novel instances. Finally, Section VI concludes the paper, highlighting the contributions of this work and discussing avenues for future research.

This study was financed by the Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico - FUNCAP (BMD-0008-01413.01.13/19) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES (No. 001). The authors also thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (grant no. 309379/2019-9), for supporting this research.

II. Autoencoders in Novelty Detection

Novelty detection involves the task of identifying new or anomalous instances within a dataset. These novel data observations, often referred to as anomalies or outliers, exhibit distinctive characteristics compared to the normal or ordinary data points. Simply put, any observation that deviates from the group of normal observations is considered a novelty. It is worth noting that in the literature, the terms anomaly and outliers are symbolic and interchangeable. According to [15], outliers are observed data points that significantly differ from the distribution of other data and can be categorized as either noise or anomalies. Noise refers to irrelevant data that should be discarded as it may degrade the performance of the detection model. On the other hand, anomalies or news represent outliers that provide new and valuable information to the model, such as a new class or concept change. Autoencoders have been widely used for anomaly and novelty detection in various domains. In this section we present a brief review on the state-of-the-art of this research field.

A. Methods and frameworks

In this section, we highlight several research papers that have made contributions to the field by proposing techniques and frameworks based on autoencoders. Building upon these advancements, our work adds to the existing body of knowledge in novelty detection area, focusing specifically on the paradigm of local learning.

In [16] a new framework named as U-Transformer-based anomaly detection framework (UTRAD) is introduced. Deep pre-trained features are represented using transformer-based autoencoders, enabling stable training and precise anomaly detection and localization. UTRAD utilizes a multi-scale pyramidal hierarchy with skip connections and outperforms other state-of-the-art methods on industrial and medical datasets.

In the deep learning field, a notable work was conducted by [17]. This work introduces a novel deep learning framework for linear systems with time-invariant parameters to address faults in sensor data used for structural and infrastructural monitoring. By employing a Convolutional Neural Network (CNN) for fault detection and a suite of Convolutional Autoencoder (CAE) networks for data reconstruction, the framework achieves high accuracy in fault detection, localization, and data reconstruction for both single and multiple sensor faults. Furthermore, the authors in [18] address the use of deep learning techniques, particularly autoencoders (AEs), for fault detection and diagnosis (FDD). They highlight the common drawback of AEs in misclassifying faulty samples that resemble normal patterns. To overcome this, they propose a source-aware autoencoder (SAAE) that incorporates faulty samples during training, providing flexibility in balancing recall and precision, the ability to detect unseen faults, and applicability to imbalanced datasets. They design a fault detection network using a bidirectional long short-term memory (BiLSTM) with skip connections, and introduce a deep network combining BiLSTM and a residual neural network (ResNet) for fault diagnosis, aiming to avoid input feature order randomness. They present a comprehensive comparison between existing techniques and their SAAE-ResNet method using the Tennessee-Eastman process, demonstrating the superiority of their proposed FDD approach.

In the time series domain, a work carried out by [19] discusses the challenges in multivariate time series classification due to the reliance on hand-engineered features, which can be subjective and time-consuming. To address this, the authors propose a framework based on deep learning, specifically stacked LSTM Autoencoder Networks, for unsupervised feature extraction. The compressed representations obtained from the LSTM Autoencoders are then used for classification using Deep Feedforward Neural Networks. The framework is applied to sensor time series data in the process industry, focusing on detecting the quality of semi-finished products and predicting the next production process step. Real-world data from the steel industry is used to validate the effectiveness of the proposed approach.

B. Applications

The work conducted by [20] proposes the usage of unsupervised ensemble autoencoders connected to a Gaussian mixture model (GMM) for robust anomaly detection in high-dimensional and sparse data. The ensemble autoencoder captures attention-based latent representations and reconstructed features to identify outliers related to cyberattack anomalies. Another interesting work presented by [21] rely on autoencoder network models (including traditional, deep, and deep convolutional autoencoders) for automatic fish species identification.

In [22], the authors combine particle filters and autoencoders for structural damage detection and localization in the field of Structural Health Monitoring (SHM). Autoencoders capture damage-related features from vibration measurements, while particle filters estimate hidden states related to damages. The algorithm is robust to changing environmental conditions and offers a valuable tool for decision-making in structural health assessment.

The study outlined in [2] introduces a Gaussian Mixture Variational Autoencoder-based method for video anomaly detection and localization. The method learns feature representations of normal samples as a Gaussian Mixture Model, allowing the scoring of anomaly for test patches. Appearance and motion anomalies are combined using a two-stream network framework. The method demonstrates superiority over state-of-the-art approaches on popular benchmark datasets.

There are numerous other works in both application and technique proposals that have been introduced and are noteworthy to mention, such as:

- Deep Autoencoding Gaussian Mixture Model (DAGMM) on multi- or high- dimensional data [23];
- Stacked autoencoders-based adaptive subspace model (SAEASM) is proposed for hyperspectral anomaly detection [24]; and
- A framework named Adaptive Adversarial Latent Space (AALS) for novelty detection [25].

III. Local Autoencoders

A. Local learning

Consider a training set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ composed of n data points, where each data point is a d-dimensional vector, that is, $\mathbf{x}_i \in \mathbb{R}^d$ represents de i-th vector. Local modeling is a multimodel approach which creates a discriminant function of the form [26]

$$g(\mathbf{x}) = \sum_{i=1}^{K} h_i(\mathbf{x}) g_i(\mathbf{x}), \tag{1}$$

where K is the number of partitions, as $h_i(\mathbf{x})$ is a function that gives a notion of locality for the i-th model, represented by the function $g_i(\mathbf{x})$. As mentioned in the previous section, there are several local learning approaches in the literature. Specifically, the type of local modeling addressed in this paper is called cluster-based local modeling with hard partitioning (CLHP) [27]. During CLHP training phase, the whole data set $\mathcal{X}$ is divided into K partitions by a clustering algorithm. Then, for each cluster $C_i \subset \mathcal{X}$ we build a classification model so that each model in the set $\{g_i\}_{i=1}^K$ has only seen the data of its associated partition. Considering that the subsets are disjoint, it can be said that CLHP does a hard partitioning of the data set. After fitting, the prediction applied on out-of-sample data points rely on first searching for the closest local region, and then using its respective local detector to predict the output.

Selecting regions in CHLP depend on prototypes that represent a local region, and we use dissimilarity measures (e.g., Euclidean distance) to the K-means prototypes for finding the closest local region.

B. Local autoencoders (LAE)

Local autoencoders (LAE) encompass a clustering-based local model characterized by a two-stage framework, namely local partitioning and local learning. The schematic representation of the LAE model training process is depicted in Figure 1, while Algorithm 1 delineates the steps for training a LAE model. As mentioned before, local partitioning involves the utilization of a prototype-based clustering algorithm to construct a model capable of grouping data based on their features' similarities. Subsequently, the phase of local learning commences, and each data point within each cluster serves as input for the training of a localized autoencoder model. The cluster-based model, also referred to as the local partitioning model, plays a pivotal role in orchestrating the training process by delineating k distinct local regions.

Note that the proposed technique can be explored in two applications: data compression and reconstruction; and novelty detection. Regarding data compression, it will be evaluated if the proposed local autoencoders affects the reconstruction error. As for the case of novelty detection, there is a further step that must be considered in training: the computation of a local threshold (θ). As described in Algorithm 1, for each local autoencoder LAE_i, a threshold θ_i must be computed. As shown in Figure 1, the thresholds are computed based on the localized reconstruction error which may be computed by the Euclidean distance between the original instance $\boldsymbol{x}_i$ and the reconstructed one, represented by is given by $\boldsymbol{x}_i'$ as described in Equation (2):

$$e_{ij} = ||\boldsymbol{x}_{ij} - \boldsymbol{x}_{ij}'||_2^2, \tag{2}$$

where e_{ij} is the reconstruction error of the j-th instance on i-th local cluster. Note that $j = \{1, \ldots, n_1\}$, where n_i denotes the number of instances in i-th local cluster.

Therefore, for each local cluster i a reconstruction error vector is given by

$$\boldsymbol{e}_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ \vdots \\ e_{in_i} \end{bmatrix}. \tag{3}$$

As for the threshold, it was computed as following

$$\theta_i = f(\boldsymbol{e}_i) = mean(\boldsymbol{e}_i) + std(\boldsymbol{e}_i), \tag{4}$$

where $mean(\boldsymbol{e}_i)$ and $std(\boldsymbol{e}_i)$ stand for the mean and standard deviation applied on the i-th local cluster reconstruction error.

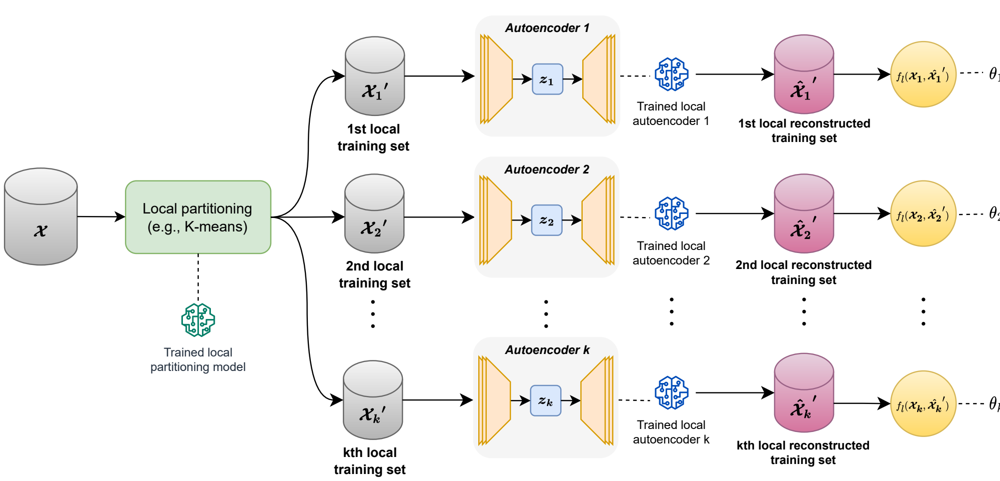


Fig. 1: Training process of local deep autoencoders.

Algorithm 1 Local autoencoder for novelty detection

Require: Training dataset $\boldsymbol{\mathcal{X}}$, number of local regions k
Ensure: Local-based Deep Autoencoder

Local partitioning:
  1. Train a clustering-based model to segment the training data into k subsets.
  2. Define the k local subsets: $\{\boldsymbol{\mathcal{X}_1}, \dots, \boldsymbol{\mathcal{X}_k}\}$.

Local training:
  for $i \leftarrow 1$ to k **do**
    1. Train ith local autoencoder (LAE_i) to fit on the $\boldsymbol{\mathcal{X}_i}$ subset.
    2. Reconstruct all instances from $\boldsymbol{\mathcal{X}_i}$ subset using the local trained model.
    3. Compute the reconstruction errors for all instances.
    4. Compute the novelty local threshold θ_i based on the reconstruction error distribution.
  end for

It is important to mention that we arbitrarily selected the Equation (4) for computing the threshold. Nevertheless, there are several other ways of computing it [28]. Furthermore, it is worth noting that for the novelty detection applications, the way in which the detection threshold is calculated is of paramount importance. However, as the focus of this work is to show preliminary results on the autoencoder architecture based on local learning (proposed in this paper), we decided to use the same threshold calculation for all local autoencoders.

IV. Experimentation settings

Within this section, we detail the experimental settings that were employed to assess the proposed method. In the experiment, we have evaluated the LAE considering several datasets from UCR repository [29]. The datasets used in this stage are described in Table I.

It is important to mention that each dataset described in Table I refers to a binary classification problem. For treating these datasets as an one-class classification problem (i.e., novelty detection) we have considered the class with greater quantity of samples as the background (or normal) class, as the other will be considered a novelty class. Additionally, a baseline multi-class based algorithm, known as k-nearest neighbor (KNN), was used in order to evaluate the overall performance of the autoencoder based models. In this phase of the experiment, our goal was to assess the novelty detection effectiveness of the proposed technique in comparison to the conventional autoencoder. Additionally, we aimed to compare its predictive capabilities with a baseline method: KNN. To address this, an evaluation was conducted employing detection measures including accuracy, precision, recall, and F1-score.

Moreover, we conducted an experiment to investigate how increasing the number of local regions in LAE impacts its performance. To achieve this objective, we trained several LAE models, systematically varying the parameter denoting the number of local regions (k) from 2 to 9. Subsequently, we computed metrics to observe the evolution of model performance with the increment in the number of regions.

This experiment was exclusively carried out on the *Wafer* dataset, as its purpose was to highlight a limitation associated with the use of local models. Specifically, it aimed to emphasize that local models are inherently influenced by the manner in which clusters are established, particularly with regard to the availability of an adequate number of samples within a cluster for effective learning model fitting.

V. Results

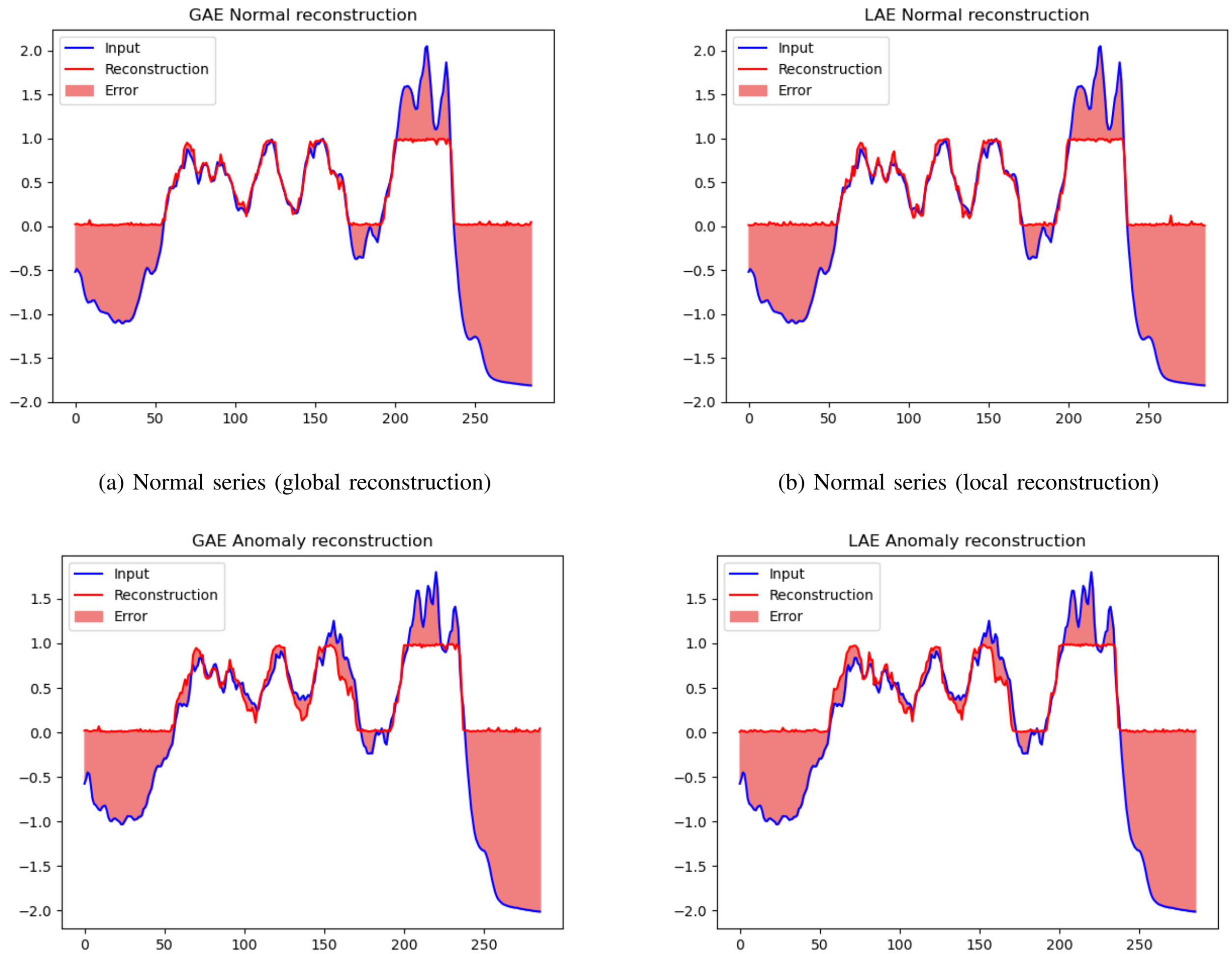
In this section, we present and discuss the results obtained from computer experiments. To assess the disparity in reconstruction outcomes between GAE and LAE, a comparison was made by plotting the difference between their respective reconstructions, as depicted in Figure 2. The visual examination reveals noticeable distinctions in the reconstructed data. However, it becomes challenging to quantitatively gauge the extent of these differences and their implications in terms of anomaly detection.

To assess the predictive performance, we have utilized detection metrics encompassing accuracy, precision, recall, and the F1-score. These metrics were utilized to quantify the performance of GAE and LAE in terms of their ability to identify anomalies. By employing these measures, a more comprehensive understanding of the comparative effectiveness of GAE and LAE in anomaly detection can be obtained.

The results presented in Table II we can observe that across the evaluated datasets, the consistent trend indicates that 1-NN outperforms both LAE and GAE in terms of F1-score. It is important to note that 1-NN is a multi-class classification method, while GAE and LAE are one-class classifiers, relying solely on data from one class. The 1-NN algorithm consistently demonstrates strong performance, as evidenced by its consistently high F1-scores across the datasets. In comparison, LAE exhibits relatively lower F1-scores than 1-NN but demonstrates better performance than GAE in most cases. GAE, on the other hand, generally exhibits the lowest F1-scores among the three algorithms.

TABLE I: UCR Datasets used in experiment

Dataset	n instances	n features	n classes	target indices	n normal instances	n anomaly instances	n train	n train normal	n test
BeetleFly	40	512	2	[1]	20	20	20	10	20
Coffee	56	286	2	[1]	29	27	28	14	28
ECGFiveDays	884	136	2	[1]	442	442	23	14	861
ItalyPowerDemand	1096	24	2	[1]	549	547	67	34	1029
ProximalPhalanxOutlineCorrect	891	80	2	[0]	605	286	600	194	291
Wine	111	234	2	[1]	54	57	57	30	54
Wafer	7164	152	2	[-1]	6402	762	1000	97	6164



(a) Normal series (global reconstruction)

(b) Normal series (local reconstruction)

(c) Abnormal series (global reconstruction)

(d) Abnormal series (local reconstruction)

Fig. 2: Global vs. local time series reconstruction on Coffee dataset.

These findings highlight the effectiveness of the 1-NN algorithm for anomaly detection tasks, benefiting from its multi-class classification approach. However, it is worth considering the inherent differences in the methodologies employed by 1-NN, LAE, and GAE. The observed variations in performance can be attributed to the contrasting characteristics and underlying principles of these algorithms.

When comparing LAE with GAE, it can be observed that LAE generally outperformed GAE in terms of F1-score. LAE (k=2) consistently achieved higher F1-scores across the datasets, indicating its better ability to balance precision and recall in anomaly detection. On the other hand, GAE showed lower F1-scores, suggesting that it may struggle in accurately identifying anomalies compared to LAE (k=2). The performance difference between LAE and GAE can be attributed to their underlying algorithms. LAE utilizes a local learning approach, which takes into account the density of data points in each local cluster. This allows for better discrimination between normal and anomalous instances.

The Table III presents the performance metrics for various algorithms based on their accuracy, precision, recall, F1 score, and error rate.

TABLE II: Model performance on UCR time series datasets

Dataset	Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	Error (%)
BeetleFly	1-NN	75.00	100.00	66.67	80.00	25.00
	GAE	55.00	10.00	100.00	18.18	45.00
	LAE (k=2)	65.00	30.00	100.00	46.15	35.00
Coffee	1-NN	100.00	100.00	100.00	100.00	0.00
	GAE	67.86	46.15	75.00	57.14	32.14
	LAE (k=2)	60.71	38.46	62.50	47.62	39.29
ECGFiveDays	1-NN	79.67	91.59	73.82	81.75	20.33
	GAE	43.67	82.01	46.25	59.14	56.33
	LAE (k=2)	42.97	80.84	45.83	58.50	57.03
ItalyPowerDemand	1-NN	95.53	94.93	96.06	95.49	4.47
	GAE	61.61	82.26	58.13	68.12	38.39
	LAE (k=2)	68.61	80.51	64.94	71.89	31.39
ProximalPhalanx[1]	1-NN	80.76	54.35	78.12	64.10	19.24
	GAE	54.64	94.57	40.65	56.86	45.36
	LAE (k=2)	52.92	95.65	39.82	56.23	47.08
Wafer	1-NN	99.55	98.65	97.19	97.91	0.45
	GAE	64.65	72.03	19.38	30.54	35.35
	LAE (k=2)	89.68	79.85	51.40	62.54	10.32
Wine	1-NN	61.11	59.26	61.54	60.38	38.89
	GAE	53.70	85.19	52.27	64.79	46.30
	LAE (k=2)	48.15	74.07	48.78	58.82	51.85

TABLE III: Model performance on Wafer dataset

Algorithm	Acc. (%)	Precision (%)	Recall (%)	F1 (%)	Error (%)
1-NN	99.55	99.65	99.84	99.75	0.45
GAE	76.25	75.29	97.53	84.98	23.75
LAE (k=2)	83.86	82.16	99.69	90.08	16.14
LAE (k=3)	82.97	81.41	99.38	89.50	17.03
LAE (k=4)	82.28	80.56	99.48	89.03	17.72
LAE (k=5)	88.03	86.96	99.56	92.84	11.97
LAE (k=6)	90.07	89.33	99.49	94.14	9.93
LAE (k=7)	89.23	88.27	99.61	93.60	10.77
LAE (k=8)	86.26	84.94	99.59	91.69	13.74
LAE (k=9)	87.62	86.49	99.58	92.57	12.38

In summary, the results reveals that:

- The 1-NN algorithm demonstrates high accuracy, precision, recall, and F1 score, all above 99%. It achieves excellent performance with a low error rate of 0.45%. This suggests that instance-based algorithms perform remarkably well in classifying these dataset.
- The GAE algorithm shows lower overall performance compared to 1-NN. Although it achieves a reasonable accuracy of 76.25%, its precision, recall, and F1 score are comparatively lower, indicating some misclassifications.
- LAE (k=2) to LAE (k=9): These algorithms, with varying values of k (from 2 to 9), demonstrate consistent performance improvement compared to GAE. They achieve accuracy ranging from 82.28% to 90.07%, precision ranging from 80.56% to 89.33%, recall ranging from 99.38% to 99.61%, and F1 score ranging from 89.03% to 94.14%. These results indicate that increasing the number of regions may improve the detection performance of a local model. Nonetheless, as the granularity of local regions increases, there is a proportional reduction in the quantity of data within each respective region, impacting the model's ability to learn with the available data.

Overall, the LAE algorithms outperform the GAE algorithm in terms of accuracy, precision, recall, F1-score, and error rate. However, it is worth noting that further evaluation and comparison with other algorithms are necessary to determine the significance of these results. Additionally, it would be beneficial to investigate the computational complexity and scalability of these algorithms to assess their practical applicability in larger datasets.

It is worth mention that these results should be interpreted in the context of the specific classification problem and the data used for evaluation. Further analysis, including consideration of other metrics and validation methods, is recommended for a comprehensive evaluation of the model's performance.

VI. Conclusions

This paper presents a novel autoencoder method based on a local learning approach. The method consists of two main steps: utilizing a cluster-based algorithm to define local data regions and training local autoencoders. Initially, the proposed method is evaluated through experiments conducted on benchmark datasets, specifically in the domains of data compression and novelty detection. The obtained results showcase the method's competitive performance compared to conventional autoencoders, its effectiveness in mitigating the adverse effects of noisy data, and its ability to enhance predictive accuracy. However, further investigation is necessary to assess the method's performance on larger datasets and address potential limitations associated with defining local regions.

It should be noted that this paper provides preliminary results and lacks comprehensive experimentation incorporating cross-validation techniques and larger datasets. Additionally, we intend to explore alternative approaches for defining local clusters and examine various methods of combining the outputs of local models to compute the reconstruction. Moreover, future research efforts will focus on optimizing the determination of local partitions and devising appropriate validation metrics for local models. Furthermore, our intention is to subject the proposed method to evaluation within more realistic scenarios, applying it to address genuine problems involving time series novelty detection.

REFERENCES

[1] P. Li, Y. Pei, and J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," *Applied Soft Computing*, vol. 138, p. 110176, May 2023.

[2] Y. Fan, G. Wen, D. Li, S. Qiu, M. D. Levine, and F. Xiao, "Video anomaly detection and localization via Gaussian Mixture Fully Convolutional Variational Autoencoder," *Computer Vision and Image Understanding*, vol. 195, p. 102920, Jun. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1077314218302674

[3] D.-M. Tsai and P.-H. Jen, "Autoencoder-based anomaly detection for surface defect inspection," *Advanced Engineering Informatics*, vol. 48, p. 101272, Apr. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1474034621000276

[4] R. Chen, J. Wang, L.-C. Yu, and X. Zhang, "Decoupled variational autoencoder with interactive attention for affective text generation," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106447, Aug. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0952197623006310

[5] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," *Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017*, pp. 90–98, 2017, iSBN: 9781611974874.

[6] Q. Yu, M. Kavitha, and T. Kurita, "Autoencoder framework based on orthogonal projection constraints improves anomalies detection," *Neurocomputing*, vol. 450, pp. 372–388, Aug. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231221005609

[7] J. Von Schleinitz, M. Graf, W. Trutschnig, and A. Schröder, "VASP: An autoencoder-based approach for multivariate anomaly detection and robust time series prediction with application in motorsport," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104354, Sep. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0952197621002025

[8] H. Studiawan and F. Sohel, "Anomaly detection in a forensic timeline with deep autoencoders," *Journal of Information Security and Applications*, vol. 63, p. 103002, Dec. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214212621002076

[9] I. T. Nicholaus, J. R. Park, K. Jung, J. S. Lee, and D.-K. Kang, "Anomaly Detection of Water Level Using Deep Autoencoder," *Sensors*, vol. 21, no. 19, p. 6679, Oct. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/19/6679

[10] E. S. Miele, F. Bonacina, and A. Corsini, "Deep anomaly detection in horizontal axis wind turbines using Graph Convolutional Autoencoders for Multivariate Time series," *Energy and AI*, vol. 8, p. 100145, May 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2666546822000076

[11] J. Zhou, Z. He, X. Liu, Y. Wang, S. Wang, and Q. Liu, "Transformed denoising autoencoder prior for image restoration," *Journal of Visual Communication and Image Representation*, vol. 72, p. 102927, Oct. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1047320320301590

[12] A. Saleh Ahmed, W. H. El-Behaidy, and A. A. Youssif, "Medical image denoising system based on stacked convolutional autoencoder for enhancing 2-dimensional gel electrophoresis noise reduction," *Biomedical Signal Processing and Control*, vol. 69, p. 102842, Aug. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1746809421004390

[13] S. Yang, K. Yu, F. Cao, L. Liu, H. Wang, and J. Li, "Learning Causal Representations for Robust Domain Adaptation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9566788/

[14] G. Jaiswal, A. Sharma, and S. K. Yadav, "Deep feature extraction for document forgery detection with convolutional autoencoders," *Computers and Electrical Engineering*, vol. 99, p. 107770, Apr. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0045790622000738

[15] C. C. Aggarwal, *An Introduction to Outlier Analysis*. New York, NY: Springer New York, 2013, pp. 1–40. [Online]. Available: https://doi.org/10.1007/978-1-4614-6396-2_1

[16] L. Chen, Z. You, N. Zhang, J. Xi, and X. Le, "UTRAD: Anomaly detection and localization with U-Transformer," *Neural Networks*, vol. 147, pp. 53–62, Mar. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0893608021004810

[17] D. Jana, J. Patil, S. Herkal, S. Nagarajaiah, and L. Duenas-Osorio, "CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction," *Mechanical Systems and Signal Processing*, vol. 169, p. 108723, Apr. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0888327021010414

[18] N. Amini and Q. Zhu, "Fault detection and diagnosis with a novel source-aware autoencoder and deep residual neural network," *Neurocomputing*, vol. 488, pp. 618–633, Jun. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231221017446

[19] N. Mehdiyev, J. Lahann, A. Emrich, D. Enke, P. Fettke, and P. Loos, "Time Series Classification using Deep Learning for Process Planning: A Case from the Process Industry," *Procedia Computer Science*, vol. 114, pp. 242–249, 2017.

[20] P. An, Z. Wang, and C. Zhang, "Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection," *Information Processing & Management*, vol. 59, no. 2, p. 102844, Mar. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306457321003162

[21] A. Banerjee, A. Das, S. Behra, D. Bhattacharjee, N. T. Srinivasan, M. Nasipuri, and N. Das, "Carp-DCAE: Deep convolutional autoencoder for carp fish classification," *Computers and Electronics in Agriculture*, vol. 196, p. 106810, May 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0168169922001272

[22] F. Cadini, L. Lomazzi, M. Ferrater Roca, C. Sbarufatti, and M. Giglio, "Neutralization of temperature effects in damage diagnosis of MDOF systems by combinations of autoencoders and particle filters," *Mechanical Systems and Signal Processing*, vol. 162, p. 108048, Jan. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0888327021004386

[23] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL FOR UNSUPERVISED ANOMALY DETECTION," 2018.

[24] L. Zhang and B. Cheng, "A stacked autoencoders-based adaptive subspace model for hyperspectral anomaly detection," *Infrared Physics & Technology*, vol. 96, pp. 52–60, Jan. 2019.

[25] B. Jiang, F. Xu, Y. Huang, C. Yang, W. Huang, and J. Xia, "Adaptive Adversarial Latent Space for Novelty Detection," *IEEE Access*, vol. 8, pp. 205 088–205 098, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9256333/

[26] E. Alpaydin and M. I. Jordan, "Local linear perceptrons for classification," *IEEE Transactions on Neural Networks*, vol. 7, no. 3, pp. 788–792, 1996.

[27] R. B. Drumond, R. F. Albuquerque, G. A. Barreto, and A. H. Souza, "Pattern classification based on regional models," *Applied Soft Computing*, vol. 129, p. 109592, Nov. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S156849462200641X

[28] J. Yang, S. Rahardja, and P. Fränti, "Outlier detection: how to threshold outlier scores?" in *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*. Sanya China: ACM, Dec. 2019, pp. 1–6.

[29] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," July 2015.