FEDERAL UNIVERSITY OF CEARÁ

CENTER OF TECHNOLOGY

DEPARTMENT OF TELEINFORMATICS ENGINEERING

GRADUATE PROGRAM IN TELEINFORMATICS ENGINEERING

DOCTORAL DEGREE IN TELEINFORMATICS ENGINEERING

RENAN FONTELES ALBUQUERQUE

ON LOCAL MODELS FOR NOVELTY DETECTION: NEW ALGORITHMS AND PRACTICAL APPLICATIONS

FORTALEZA

2024

RENAN FONTELES ALBUQUERQUE

ON LOCAL MODELS FOR NOVELTY DETECTION: NEW ALGORITHMS AND PRACTICAL APPLICATIONS

Thesis submitted to the Graduate Program in Teleinformatics Engineering of the Center of Technology of the Federal University of Ceará, as a partial requirement for obtaining the degree of Doctor in Teleinformatics Engineering. Concentration Area: Signals and Systems

Advisor: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A313o Albuquerque, Renan Fonteles.
On Local Models for Novelty Detection : New Algorithms and Practical Applications / Renan Fonteles Albuquerque. – 2024.
121 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2024.
Orientação: Prof. Dr. Guilherme de Alencar Barreto.

1. Aprendizagem local. 2. Análise de componentes principais via kernel. 3. Autoencoders. 4. Classificação de uma classe. 5. Detecção de novidade. I. Título.

CDD 621.38

RENAN FONTELES ALBUQUERQUE

ON LOCAL MODELS FOR NOVELTY DETECTION: NEW ALGORITHMS AND PRACTICAL APPLICATIONS

Thesis submitted to the Graduate Program in Teleinformatics Engineering of the Center of Technology of the Federal University of Ceará, as a partial requirement for obtaining the degree of Doctor in Teleinformatics Engineering. Concentration Area: Signals and Systems

Approved on: 29th January 2024

EXAMINATION BOARD

Prof. Dr. Guilherme de Alencar Barreto (Advisor)
Universidade Federal do Ceará (UFC)

Prof. Dr. Alberto Guillén Perales
Universidad de Granada (UGR)

Prof. Dr. Eduardo Furtado Simas Filho
Universidade Federal da Bahia (UFBA)

Prof. Dr. César Lincoln Cavalcante Mattos
Universidade Federal do Ceará (UFC)

Prof. Dr. Tarcisio Ferreira Maciel
Universidade Federal do Ceará (UFC)

For my beloved family.

ACKNOWLEDGEMENTS

I would like to dedicate this thesis to my beloved family, whose support, encouragement, and love have been vital throughout this journey. Their presence has been a constant source of strength and inspiration.

I extend my sincere gratitude to my esteemed supervisor, Prof. Dr. Guilherme de Alencar Barreto, for his unwavering support, guidance, and partnership throughout this research. His patience, expertise and mentorship have played an essential role in shaping this work.

I am also deeply grateful to all the professors and researchers who have generously contributed to this project. I extend my special appreciation to my fellow researchers at the Signal and Information Processing for Data Analysis and Learning Systems (SPIRAL) and the *Programa de Pós-Graduação em Engenharia de Teleinformática (PPGETI)*, with whom I have shared meaningful conversations, insightful contributions, and enjoyable moments.

Lastly, I would like to acknowledge the financial support provided by the *Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP)* through the scholarship that has made this research possible. This support has been instrumental in allowing me to fully dedicate myself to this endeavor.

To all those mentioned above and to countless others who have played a role in shaping this thesis, my sincere gratitude.

"It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment."

(Carl Friedrich Gauss)

ABSTRACT

Machine Learning (ML) has redefined problem-solving by enabling systems to autonomously learn and make decisions from data patterns. With the increasing complexity of data, there is a growing need for sophisticated ML techniques. In this regard, local learning has emerged as a promising approach that concentrates analysis on specific localized characteristics of the data, such as data proximity, feature subsets, or graph structures. Unlike global learning, which aims to build a single global model, local learning concentrates on smaller, potentially more interpretable problem subsets. This thesis explores the effectiveness, advantages, and limitations of this approach across different scenarios. Additionally, it introduces two novel local learning techniques designed specifically for one-class classification: local kernel principal component analysis (LKPCA) and local autoencoder (LAE). LKPCA leverages kernel methods to handle nonlinear data, while LAE utilizes deep autoencoders (DAEs). The proposed local learning techniques have the potential to reduce processing costs by leveraging localized representations, which makes them particularly efficient in handling imbalanced datasets and redundant data. Moreover, they are effective at removing noise and irrelevant data in sparse regions, enabling the model to focus on meaningful patterns and improve the detection performance. LKPCA variants were compared against global KPCA and state-of-the-art methods across 17 one-class datasets derived from 9 benchmark datasets. The results indicate that cooperative LKPCA generally outperforms global KPCA, while competitive LKPCA frequently presents lower performance. The cooperative LKPCA also demonstrated higher predictive power compared to state-of-the-art methods on several datasets. Regarding LAE, it was assessed on 7 time series datasets, revealing an improved F1-score over global autoencoders (AEs) in datasets such as BeetleFly, Wafer, and ItalyPowerDemand. These results show the potential of applying local learning structures in one-class classification problems.

Keywords: local learning; kernel principal component analysis; autoencoders; one-class classification; novelty detection.

RESUMO

O aprendizado de máquina (AM) redefiniu a resolução de problemas, permitindo que os sistemas aprendam e tomem decisões de forma autônoma a partir de padrões de dados. Com o aumento da complexidade dos dados, há uma crescente necessidade de técnicas sofisticadas de AM. Neste contexto, a aprendizagem local surgiu como uma abordagem promissora, em que a análise se concentra em aspectos específicos e localizados dos dados, como proximidade entre padrões, subconjuntos de características ou estruturas de grafos. Ao contrário da aprendizagem global, que visa construir um modelo único e global, a aprendizagem local se concentra em subconjuntos de problemas menores, potencialmente mais interpretáveis. Essa tese explora a eficácia, vantagens e limitações dessa abordagem em diferentes cenários. Além disso, este estudo apresenta duas novas técnicas de aprendizado local para classificação de uma classe: *local kernel principal component analysis (LKPCA)* e *local autoencoder (LAE)*. O LKPCA aproveita métodos de kernel para lidar com dados não lineares, enquanto o LAE utiliza deep autoencoders (DAEs). As técnicas de aprendizado local propostas têm o potencial de reduzir os custos de processamento ao utilizar representações localizadas, tornando-as especialmente eficientes no tratamento de conjuntos de dados desbalanceados e com redundância. Ademais, são eficazes na remoção de ruído e dados irrelevantes em regiões esparsas, permitindo que o modelo se concentre em padrões significativos e melhore o desempenho de detecção. As variantes LKPCA foram comparadas com o KPCA global e com métodos de referência em 17 conjuntos de dados de uma classe derivados de 9 conjuntos de dados de referência. Os resultados indicam que o LKPCA cooperativo geralmente supera o KPCA global, enquanto o LKPCA competitivo frequentemente apresenta desempenho inferior. O LKPCA cooperativo também demonstrou maior poder preditivo em comparação com métodos de referência em vários conjuntos de dados. Em relação ao LAE, ele foi avaliado em 7 conjuntos de dados de séries temporais, revelando um F1-*score* melhor em relação ao autoencoder (AE) global em conjuntos de dados como BeetleFly, Wafer e ItalyPowerDemand. Esses resultados mostram o potencial de aplicação de estruturas de aprendizagem local em problemas de classificação de uma classe.

Palavras-chave: aprendizagem local; análise de componentes principais via kernel; autoencoders; classificação de uma classe; detecção de novidade.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

1-NN	1-nearest neighbor
AD	anomaly detection
AE	autoencoder
AI	artificial intelligence
CLHP	cluster-based local modeling with hard partitioning
CNN	convolutional neural network
DAE	deep autoencoder
DL	deep learning
DNN	deep neural network
FL	federated learning
FPR	false positive rate
GAE	global autoencoder
GAN	generative adversarial network
GMM	Gaussian mixture model
HC	hierarchical clustering
HIS	hybrid intelligent systems
HSI	hyperspectral imagery
IF	isolation forest
IQR	interquartile range
KDE	kernel density estimator
KNN	k-nearest neighbour
KPCA	kernel principal component analysis
LAE	local autoencoder
LKPCA	local kernel principal component analysis
LKPCA-AVG	average weighted local kernel principal component analysis
LKPCA-ST	stacking local kernel principal component analysis
LKPCA-SW	similarity weighted local kernel principal component analysis
LKPCA-WTA	winner-take-all local kernel principal component analysis
LKPCA-WTA-MT	multi-threshold winner-take-all local kernel principal component analysis
LOF	local outlier factor
LPCA	local principal component analysis

LSTM	long short-term memory
LSTM-AE	long short-term memory autoencoder
LTSA	local tangent space alignment
MKL	multiple kernel learning
ML	machine learning
MLP	multi-layer perceptron
MSE	mean squared error
MTCC	multi-class classification
ND	novelty detection
NN	nearest neighbor
OCC	one-class classification
OCSVM	one-class classification support vector machine
OD	outlier detection
OPM	operator map
PC	principal component
PCA	principal component analysis
pdf	probability density function
PPV	positive predictive value
RBF	radial basis function
RC	regional classifier
RLTSA	robust local tangent space alignment
RM	regional model
ROC	receiver operating characteristic
ROC-AUC	area under the ROC curve
SOM	self-organizing map
SVDD	support vector data description
SVM	support vector machine
TNR	true negative rate
TPR	true positive rate
WTA	winner-takes-all

LIST OF SYMBOLS

$\boldsymbol{x}$	Input vector
y	Output class label
n	Number of samples in a dataset
d	Dimensionality of the input space
$d_{\mathcal{F}}$	Dimensionality of the feature space
$\boldsymbol{K}$	Kernel matrix
K_{ij}	Element in the i-th row and j-th column of the kernel matrix
$\mathcal{X}_{tr}$	A training dataset
$\mathcal{X}_{te}$	A test dataset
$\mathbb{R}^d$	A d dimensional real vector space
$\mathcal{F}$	A high dimensional feature space
$\mathcal{L}$	A loss function
$f(\cdot)$	A classification discriminant function
$g(\cdot)$	A global discriminant function derived from a combination of multiple local discriminant functions
$g_i(\cdot)$	The i-th local discriminant function
$h_i(\cdot)$	A locality function that weights i-th local model's contribution to a global discriminant function, based on data similarity
$s(\cdot,\cdot)$	A function that measures similarity between two vectors
$\psi(\cdot,\cdot)$	A function that measures dissimilarity between two vectors
$\|\|\cdot\|\|_2$	The Euclidean norm
K	The number of clusters to be formed by the K-means algorithm
C	A data local cluster
$\boldsymbol{c}$	A K-means cluster centroid or a prototype vector
$\emptyset$	The empty set
$\boldsymbol{\varphi}(\cdot)$	A feature mapping function that projects data from its original space into a higher-dimensional feature space

$k(\cdot,\cdot)$	A kernel function
$p_S^{(j)}(\cdot)$	The j-th *local spherical potential*
$\tilde{p}_S^{(j)}(\cdot)$	The j-th simplified *local spherical potential*
$p^{(j)}(\cdot)$	The j-th local KPCA novelty measure
$\boldsymbol{w}$	A weight vector for local KPCA models
$\mathcal{E}$	A matrix with dimensions $n \times K$ that contains the reconstruction errors obtained from K local KPCA models applied to a training set with n instances
θ	A novelty detection threshold
θ_{wta}	The novelty detection threshold for the LKPCA-WTA method
$\theta_{wta}^{(j)}$	The j-th local novelty detection threshold for the LKPCA-WTA-MT method
θ_{avg}	The novelty detection threshold for the LKPCA-AVG method
θ_{sw}	The novelty detection threshold for the LKPCA-SW method
θ_{st}	The novelty detection threshold for the LKPCA-ST method

CONTENTS

1 INTRODUCTION

Machine learning (ML), a subset of artificial intelligence (AI), delves into the exploration of algorithms and statistical models that enable systems to perform specific tasks without relying on explicit instructions; instead, they leverage on data patterns and inference. Essentially, it is about creating algorithms that allow computers to learn and make decisions from data (BISHOP, 2006; MURPHY, 2021). Within the broad field of ML, lies the specialized domain of pattern classification. Pattern classification is the process of categorizing input data into classes based on its features or patterns. Classification tasks can be grouped based on the number of different classes they comprise. Multi-class classification (MTCC) involves techniques focused on categorizing input data into one of several classes. An example of an MTCC problem is distinguishing between various types of fruits — such as apples, oranges, bananas, and grapes — based on features like color, texture, and shape (DUDA *et al.*, 2000). Conversely, one-class classification (OCC) focuses on modeling instances from a target class — representing normal, regular, or background data — to discriminate them from all other classes. Unlike traditional MTCC classifiers, whose require labeled examples from multiple classes for training, OCC focuses solely on the characteristics of a target class, training a discriminant function to detect deviations from that class. Consequently, OCC is well-suited for outlier, anomaly and novelty detection, as it aims to identify data points that significantly differ from the expected target class distribution (TAX, 2001; CARREÑO *et al.*, 2020; PERERA *et al.*, 2021).

OCC offers significant advantages in scenarios where normal data is abundant and deviations are rare. These classifiers operate on the assumption that anomalies, outliers, or novelties are exceptions to the expected data behavior (CHANDOLA *et al.*, 2009; NASSIF *et al.*, 2021; PERERA *et al.*, 2021). Consequently, OCC techniques are widely used to detect rare or unexpected events, serving as an essential tool in critical domains such as finance (PORWAL; MUKUND, 2019; HILAL *et al.*, 2022), healthcare (MUHAMMAD *et al.*, 2021), and cybersecurity (ANDRESINI *et al.*, 2021; WANG *et al.*, 2022), where unusual patterns may indicate fraud, disease outbreaks, or security breaches. The versatility of OCC has led to its extensive use across various detection fields, applications and real-world scenarios (AGGARWAL; YU, 2001; TAX, 2001; MARKOU; SINGH, 2003; LATECKI *et al.*, 2007; HOFFMANN, 2007; SHAHREZA *et al.*, 2011; TIMM; BARTH, 2012; FARIA *et al.*, 2016; AGUAYO; BARRETO, 2017; PARK, 2019; YANG *et al.*, 2019a; BOUKERCHE *et al.*, 2020; LI *et al.*, 2020; PU *et al.*, 2022; MAES *et al.*, 2022; FANG *et al.*, 2022; GÔLO *et al.*, 2023).

OCC learning methods compile an extensive set of techniques from distinct modeling approaches, such as: probabilistic or density-based, boundary-based, clustering-based, reconstruction-based, based on hybrid intelligent systems (HIS), among others (BOUKERCHE *et al.*, 2020; MARQUES *et al.*, 2023). Probabilistic models for OCC are based on estimating the probability density function (pdf) of the normal data. Anomalies are then identified as data points that fall outside of a specific probability threshold, which is determined from the estimated distribution. As for boundary-based methods, they excel at encapsulating the normal data within a defined boundary, that is, such models establish a perimeter around the regular data, classifying new instances inside this boundary as inliers and those outside as outliers (MARQUES *et al.*, 2023). In this context, two notable methods are the one-class classification support vector machine (OCSVM) (TAX, 2001) and the support vector data description (SVDD) (TAX; DUIN, 2004).

Clustering-based methods are unsupervised techniques that learn the typical data distribution of inliers. Novelties are then detected when new data points do not share a high degree of similarity with the existing clusters. Hence, such techniques typically depend on similarity or dissimilarity measures (MUÑOZ; MURUZÁBAL, 1998; ALBERTINI; MELLO, 2007; SHAHREZA *et al.*, 2011; AGUAYO; BARRETO, 2017; PISANO *et al.*, 2017; YANG *et al.*, 2019a). The self-organizing map (SOM), a noteworthy technique for learning the topological structure of data, has also been applied to OCC applications (MUÑOZ; MURUZÁBAL, 1998; LIU; XU, 2003; ALBERTINI; MELLO, 2007; HUANG *et al.*, 2008; SHAHREZA *et al.*, 2011; AGUAYO; BARRETO, 2017; YANG *et al.*, 2019a, 2019a).

In the domain of reconstruction error-based modeling, the emphasis lies on employing a learning architecture that transforms input data into a *latent space* and subsequently attempt to reconstruct it with minimal information loss. The underlying hypothesis is that anomalous instances tend to yield a higher reconstruction error compared to the normal instances upon which the model was trained. Models such as the principal component analysis (PCA) (MOON *et al.*, 2022) and its nonlinear extension, the kernel principal component analysis (KPCA) (SCHÖLKOPF *et al.*, 1997; HOFFMANN, 2007; ZHANG, 2009; DAS *et al.*, 2018), are relevant techniques within this context.

Autoencoders (AEs) constitute another extensively studied approach to OCC (GEDDES *et al.*, 2019; YU *et al.*, 2021; TSAI; JEN, 2021; ANDRESINI *et al.*, 2021; AMINI; ZHU, 2022; TAKHANOV *et al.*, 2023). An AE represents a form of self-supervised learning algorithm composed by two primary components: an encoder and a decoder. The encoder is designed to transform input data into a lower-dimensional latent representation. Conversely, the decoder is responsible for reconstructing the input data from the latent representation, aiming to minimize the associated reconstruction errors. In OCC problems, AEs are trained exclusively on normal data. This process generates a distribution of reconstruction errors, which is then used to establish a threshold to defining normal data behavior. A discriminant function classifies a data point as novelty if its reconstruction error surpasses this threshold (YU *et al.*, 2021).

Deep learning (DL) and deep neural networks (DNNs) have become central to OCC field (SCHLACHTER *et al.*, 2019; LI *et al.*, 2023). Within this context, notable research has explored novel methods, architectures and applications, including:

- **deep AEs** (FAN *et al.*, 2020; CHAO *et al.*, 2021; BANERJEE *et al.*, 2022);
- **convolutional neural networks (CNNs)** (PARK *et al.*, 2017; OZA; PATEL, 2019; DONG *et al.*, 2022; FANG *et al.*, 2022; JANA *et al.*, 2022);
- **generative adversarial networks (GANs)** (YANG *et al.*, 2019b; LIU *et al.*, 2019; AKCAY *et al.*, 2019; LI *et al.*, 2020; KIM *et al.*, 2021; LEE *et al.*, 2022; KATZEF *et al.*, 2022; PU *et al.*, 2022; XIA *et al.*, 2022); and
- **long short-term memory (LSTM)** (MALHOTRA *et al.*, 2015; LI *et al.*, 2023; MUNIR *et al.*, 2019; MALEKI *et al.*, 2021; LINDEMANN *et al.*, 2021; LIU *et al.*, 2022).

Choosing an OCC learning method depends on factors like data availability, the nature of the problem, and the application domain. As data patterns become more complex (i.e., high-dimensional, nonlinear and multimodal) there is a growing need for robust models that can handle these challenges. This has led to a growing interest in local learning, which can improve model performance by focusing on specific and localized tasks (ALPAYDIN; JORDAN, 1996; HOFFMANN *et al.*, 2009; HUANG *et al.*, 2012; DING *et al.*, 2018; ZHAO *et al.*, 2023). Various local learning methods have been proposed to define these contexts based on data proximity (DRUMOND *et al.*, 2022), data features (ARMANFARD *et al.*, 2016; PARK *et al.*, 2017; HU *et al.*, 2020), graph structures (SHENG *et al.*, 2024), data origin (NGUYEN *et al.*, 2021), or a combination of diverse learning models (MOON *et al.*, 2022).

This thesis examines local learning models, exploring their challenges and potential in the field of novelty detection. We address several important questions, including the challenges in training local-based learning methods, their advantages and disadvantages, and the criteria for defining the concept of locality for each of these methods. We specifically focus on a local learning framework known as cluster-based local modeling with hard partitioning (CLHP). This approach clusters the data space into distinct local partitions, and then trains a localized model for each partition, providing a tailored representation of local data characteristics, enhancing the model's ability to capture patterns while also reducing the computational complexity associated with fitting individual models (DRUMOND *et al.*, 2022). In addition, we introduce three novel OCC-based local learning techniques: the *competitive* LKPCA, the *cooperative* LKPCA and the LAE. The first two techniques are based on the kernel principal component analysis (KPCA) for novelty detection (HOFFMANN, 2007), while the last one explores the advantages of AEs in feature representation and reconstruction.

1.1 Objectives

This thesis contributes to the field of novelty detection by proposing innovative local learning methods. This work also provides a comprehensive review of the literature and evaluates the clustering-based local learning paradigm. This assessment highlights both the theoretical advancements and practical applications where these methodologies are most effective. The specific objectives of this thesis are summarized as follows.

1. To propose a local learning framework for novelty detection based on clustering.
2. To propose and implement OCC-based algorithms that leverage local learning principles, focusing on improving detection performance and computational efficiency, including:
   a) *Competitive* LKPCA;
   b) *Cooperative* LKPCA; and
   c) LAE.
3. To analyze and compare global and local learning OCC-based techniques applied on novelty detection problems, highlighting advantages and trade-offs.
4. To conduct experiments on standard novelty detection benchmark datasets to assess the effectiveness of the proposed methods compared to state-of-the-art baseline techniques.

1.2 Methodology

This thesis begins with a comprehensive literature review on pattern classification focusing on OCC and its associated use cases, such as outlier, anomaly, and novelty detection. We emphasized the current state of the art, reviewing key articles on applications, methodologies, frameworks, and surveys on the subject. Based on the literature assessment, we selected established algorithms — including isolation forest, one-class classification support vector machine, and local outlier factor — to serve as baseline methods for comparative analysis. Additionally, we conducted a survey regarding local learning methodologies within the domain of OCC and pattern classification.

This study also introduces three novel cluster-based local methods: the *competitive* LKPCA, the *cooperative* LKPCA, and the LAE. Using various benchmark datasets, we conducted a series of computational experiments to evaluate these methods against traditional OCC techniques. Performance was assessed using a range of indices, including accuracy, selectivity, sensitivity, and the F1-score, alongside graphical representations like receiver operating characteristic (ROC) curves and boxplots to analyze model performance and variability.

1.3 List of Publications

The present thesis has resulted in the following publications:

1. Drumond, R. B., **Albuquerque, R. F.**, & Barreto, G. A. (2019). Regional Classifiers: A Novel Framework for Pattern Classification. XVI Brazilian Symposium on Intelligent Automation (SBAI 2019). doi: 10.17648/sbai-2019-111444.
2. Drumond, R. B., **Albuquerque, R. F.**, Sousa, D. P., & Barreto, G. A. (2019). Classificação Local utilizando Least Square Support Vector Machine (LSSVM). In XIV Brazilian Congress on Computational Intelligence (CBIC 2019). doi: 10.21528/CBIC2019-98.
3. Drumond, R. B., **Albuquerque, R. F.**, Barreto, G. A., & Souza, A. H. (2022). Pattern classification based on regional models. **Applied Soft Computing**, 129, 109592. doi: 10.1016/j.asoc.2022.109592.
4. **Albuquerque, R. F.**, & Barreto, G. A. (2023). Unsupervised Time Series Novelty Detection Using Clustering-based Local Autoencoders. In XVI Brazilian Congress on Computational Intelligence (CBIC 2023). doi: 10.21528/CBIC2023-172.

1.4 Organization of the Thesis

The remaining chapters are organized as follows:

- Chapter 2 introduces the field of pattern classification and explores the fundamentals of one-class classification and its associated use cases, including the problems of outlier, anomaly and novelty detection. This chapter also reviews the OCC literature, highlighting its challenges, applications and relevant studies.
- Chapter 3 introduces the principles of local learning and present the current trends in the field. We also present notable works that have impacted the OCC field and delve into the strengths and limitations of the local learning approach. Finally, we formalize a local learning framework known as cluster-based local modeling with hard partitioning (CLHP) while discussing its advantages and drawbacks.
- Chapter 4 proposes an OCC method named local kernel principal component analysis (LKPCA). It begins by outlining the fundamentals on the KPCA for novelty detection, then provides a detailed description of the proposed LKPCA. Additionally, two *competitive* LKPCA variants are introduced: one based on a single threshold and another using local multi-thresholding. The chapter concludes with an analysis of the simulation findings on state-of-the-art OCC methods, along with both global KPCA and *competitive* LKPCA.
- Chapter 5 presents a set of *cooperative* variants based on LKPCA. The first one considers the average output of local models; the second variant relies on the similarity-weighted output of local models, and the third variant depends on stacking ensemble for training a *metamodel* that aggregates local novelty measures. The chapter details each cooperative variant, emphasizing its features and contributions to novelty detection. The chapter also presents and discusses the results of computational simulations comparing the global KPCA, *competitive* LKPCA and *cooperative* LKPCA.
- Chapter 6 proposes a novelty detection method based on autoencoders: the local autoencoder (LAE). This chapter explores the fundamental concepts and the implementation details of LAE, presenting and discussing simulation results for its application to time series novelty detection.
- Chapter 7 summarizes the thesis's contributions, discusses key concepts, and presents the conclusions drawn from the previous chapters. The chapter also addresses the limitations of the proposed methods and suggests avenues for future research.

2 ONE-CLASS CLASSIFICATION

This chapter formalizes the problem of pattern classification, discussing its challenges and outlining methods for performance assessment. It then explores the domain of OCC, introducing its fundamental concepts and presenting a taxonomy-based survey of its learning approaches. The chapter concludes with a comprehensive review of outlier, anomaly, and novelty detection in the context of OCC.

2.1 Classification Problem

Pattern classification is a crucial area of ML, centered on algorithms and techniques for data-driven decision-making. This field has evolved significantly, integrating insights from statistics, computer science, and engineering (BISHOP, 2006; HASTIE *et al.*, 2009; LOYOLA-GONZÁLEZ *et al.*, 2020; DRUMOND *et al.*, 2022; WANG *et al.*, 2023). In this section, we lead a discussion on the topic of pattern classification, starting with a formal definition of the problem and then exploring its methods, applications and inherent challenges.

Consider a training set $\mathcal{X}_{tr} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$, where n is the number of training samples; $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ is the i-th input vector of dimensionality d; and $y_i \in \mathcal{Y} = \{1, \ldots, n_c\}$ represents the i-th class label, where n_c symbolizes the total number of distinct classes. The essence of learning a classifier revolves around estimating a discriminant function $f(\cdot)$ that maps input variables to class labels as follows:

$$f : \mathcal{X} \rightarrow \mathcal{Y}, \tag{2.1}$$

where $\mathcal{X}$ represents the input space (i.e., feature space) that consists of a set of possible inputs that can be fed into the classification model; and $\mathcal{Y}$ is the output space (i.e., label space) that is the set of all possible labels that can be predicted by the classification model. In this sense, a well-fitted classifier is characterized by its ability to effectively distinguish between classes, demonstrating a high level of discriminative capacity. Beyond discriminative power, there are several other aspects that should be considered to ensure the effectiveness of a classifier such as generalization, robustness, computational efficiency and interpretability (BISHOP, 2006; WANG *et al.*, 2023).

2.1.1 Challenges and Issues

Pattern classification is an evolving field with diverse and critical challenges. The following items explore broad ML topics as well as specific challenges of developing classifiers in a dynamic data environment.

1. **High dimensionality** - Working with high-dimensional data — datasets with a large number of attributes — presents a set of challenges often referred to as "curse of dimensionality" (HASTIE *et al.*, 2009). This can lead to several issues that negatively impact ML models, including the need for a high volume of data instances to avoid overfitting; increases of computational efforts; difficulty in visualization and feature engineering; and diminishing usefulness of distance metrics.
2. **Data variability** - Data variability is a critical factor in ML model design. Data can vary due to factors such as noise, distortions, or changes in the environment, which can lead to a significant decrease in classification performance (KORYCKI; KRAWCZYK, 2021). The heterogeneity of data sources and the high dimensionality of features tend to aggravate this variability, making it difficult to identify significant patterns (BISHOP, 2006).
3. **Generalization** - Generalization is a model's ability to accurately predict on out-of-sample data (BISHOP, 2006). It is a fundamental goal in ML, ensuring a model is both reliable and effective in real-world scenarios (TAX, 2001; DUDA *et al.*, 2000). A well-generalizing model achieves a balance between learning the essential patterns in the training data and ignoring noise or non-representative anomalies (BISHOP, 2006). This is the core concept behind avoiding overfitting.
4. **Overfitting and underfitting** - Overfitting occurs when a model learns the training data's details and noise to the extent that affects negatively its performance on out-of-sample data. One can identify this by comparing the model's performance on the training set and independent test sets. If the training performance is significantly higher, the model is likely overfitted. Conversely, underfitting occurs when a model fails to capture the underlying data patterns, leading to poor performance on both the training and test sets.
5. **Data nonstationarity** - Data nonstationarity occurs when the statistical properties of the data — such as the mean, variance, correlation or the entire density distribution — change over time (BROWN; HWANG, 2012). This presents a major challenge for ML models, as a model trained on past data may no longer accurately represent its current state. In the context of data streams, this issue is commonly referred to as concept

drift (PARK, 2019; GRUHL *et al.*, 2021). Handling nonstationarity often involves using adaptive models that can continuously update themselves with new data. Other common approaches include employing time window-based methods to capture recent trends or applying transformation techniques to make the data stationary (GRUHL *et al.*, 2021; DING *et al.*, 2023). Addressing nonstationarity is crucial for building robust predictive models in dynamic environments, especially for novelty detection.

6. **Scalability and computational complexity** - As datasets grow in size and dimensionality, ML algorithms need to be scalable and computationally efficient to handle large-scale data in big data applications (GOODFELLOW *et al.*, 2016).
7. **Interpretability** - Interpretability, also termed as explainability, is increasingly recognized as a crucial attribute for ML models. It refers to the ability of a model to provide understandable and interpretable insights into its functioning and decision-making processes. This concept, while lacking a universally accepted definition, is grounded in the idea that users should be able to comprehend and trust the decisions made by artificial systems (DOSHI-VELEZ; KIM, 2017; RASHEED *et al.*, 2022).
8. **Ethical and Privacy Concerns** - Ethics and privacy present significant challenges in real-world applications, especially with the increasing use of personal data. Adopting ethical AI practices and robust privacy safeguards is crucial to ensure that ML models are ethical, fair and trustworthy (WU *et al.*, 2020).
9. **Imbalanced datasets** - A significant challenge in MTCC involves dealing with imbalanced datasets. The substantial disparity in the number of samples per class can cause classifiers to become biased toward the majority class, leading to poor performance on the less-represented minority classes (VUTTIPITTAYAMONGKOL *et al.*, 2021).
10. **Class overlap and noise** - In many real-world applications, classes are not perfectly separable and there is a significant overlap between classes, usually intensified by noise in the data. This overlap makes it challenging for classifiers to effectively distinguish between classes, frequently leading to nonlinearly separable classification problems (VUTTIPITTAYAMONGKOL *et al.*, 2021).
11. **Class drift** - Class drift is a form of concept drift where the distribution of classes changes over time (KORYCKI; KRAWCZYK, 2021). To handle this issue, models must be adaptive, either by being continuously retrained to adjust to the class changes or by employing online learning techniques.

2.1.2 Performance Evaluation

Evaluating pattern classification models is critical to ensure their reliability and effectiveness in practical applications. This assessment relies on several performance indices and methods. The confusion matrix, for example, is a powerful tool that provides a granular breakdown of a model's strengths and weaknesses in predicting positive and negative classes (STEHMAN, 1997). Several performance indices can be derived from this matrix. Table 1 presents the structure of a confusion matrix, where TP, TN, FP and FN denote True-Positive, True-Negative, False-Positive and False-Negative values, respectively.

Table 1 – Confusion Matrix.

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

Source: the author (2024).

The following equations define the main indices for evaluating classification models. Accuracy, a commonly used index, measures the proportion of correctly predicted instances over the total, as follows:

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{2.2}$$

It should be mentioned that accuracy is not adequate for evaluating classifiers trained from imbalanced datasets. In such scenarios, precision, specificity and sensitivity become essential, providing insights into the model's ability to correctly predict positive and negative classes while avoiding false positive and false negative errors (BRADLEY, 1997). The precision, also known as positive predictive value (PPV), quantifies the model's capability of correctly classifying positive instances while avoiding FP errors. It is computed as follows:

$$\text{precision} = \frac{TP}{TP+FP}. \tag{2.3}$$

In addition, the sensitivity — alternatively referred as true positive rate (TPR) or recall — determines the proportion of actual positive instances that were correctly detected out of the total positive class instances ($TP+FN$), reflecting the model's capacity of classifying positive cases, as described below.

$$\text{sensitivity} = \frac{TP}{TP+FN}. \tag{2.4}$$

Similarly, the specificity, also known as true negative rate (TNR) or selectivity, indicates the proportion of actual negative instances that were correctly identified out of the total negative class instances ($TN + FP$):

$$\text{specificity} = \frac{TN}{TN + FP}. \tag{2.5}$$

Alternatively, the F_1-score introduces a harmonic mean of *precision* and *recall*, offering a single metric for assessing problems with imbalanced classes. It is defined as

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}. \tag{2.6}$$

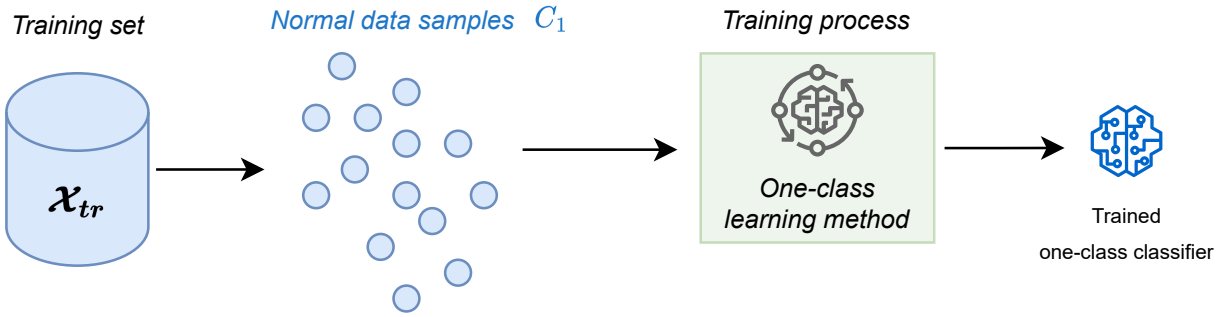
Beyond performance indices, the receiver operating characteristic (ROC) curve and its area under the ROC curve (ROC-AUC) provide a robust evaluation of a classifier's performance across various threshold settings, capturing the trade-off between true positive and false positive rates (BRADLEY, 1997; DAVIS; GOADRICH, 2006). Additionally, cross-validation techniques, such as k-fold cross-validation, are employed to assess a model's generalization capability by partitioning the data into different subsets for training and testing (BISHOP, 2006). Ultimately, the choice of evaluation indices is crucial, as misclassifications have varying impacts depending on the application. Neglecting this aspect can lead to misleading conclusions about a model's effectiveness, while a diverse set of indices provides a more detailed understanding of its performance.

2.2 One-class Classification

One-class classification (OCC), also known as single-class or *unary* classification, is an ML problem where a model is trained using data from only a single class, usually representing the normal behavior. The goal is to distinguish these normal patterns from all other potential nonconforming patterns (TAX; DUIN, 1999). Nonconforming patterns are frequently referred to as outliers, anomalies, novelties, discordant observations, exceptions, surprises or contaminants (MARKOU; SINGH, 2003; CHANDOLA *et al.*, 2009; PIMENTEL *et al.*, 2014). Similarly, the normal data is often referred to as background data, regular data, or inliers.

Figure 1 illustrates the training process of an OCC model, where a boundary is constructed around the data points of the normal class (C_1). Instances outside this boundary are classified as abnormal.

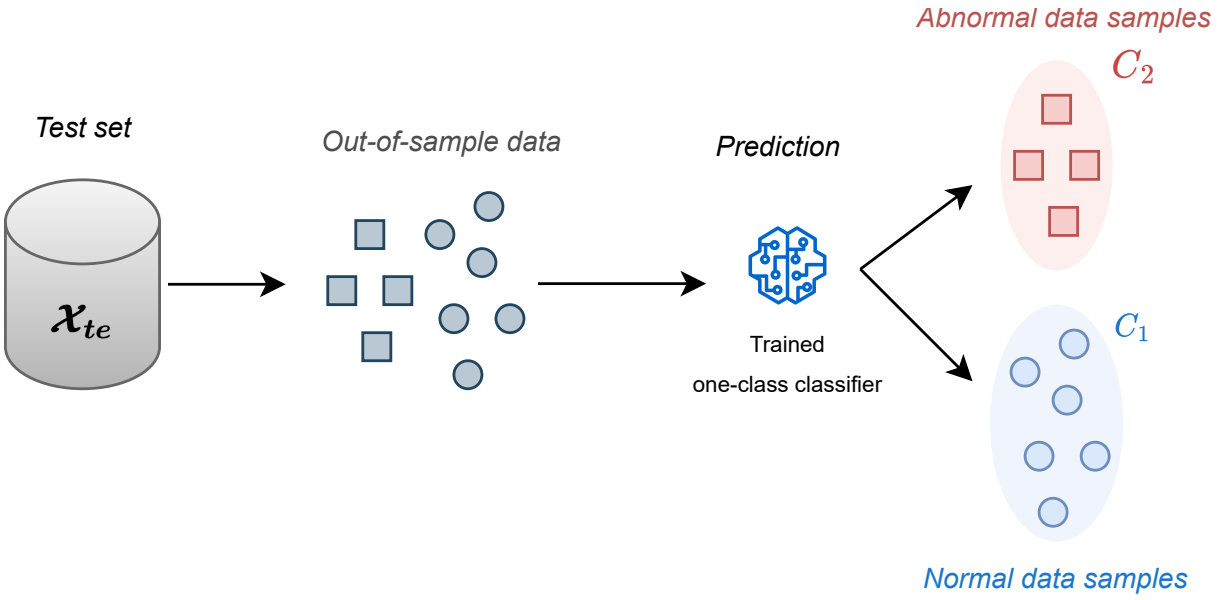
Figure 1 – One-class classifier training.



Source: the author (2024).

As for predicting out-of-sample data points, the trained model must be able to distinguish normal class (C_1, denoted by circles) against any potential deviating patterns, which are represented as squares under C_2, as illustrated in Figure 2.

Figure 2 – One-class classifier out-of-sample prediction.



Source: the author (2024).

OCC problems have been extensively explored in the literature, comprising multiple reviews and surveys that summarize the state of the art in this field. For instance, Perera *et al.* (2021) provide a survey on OCC contributions covering from classical statistical techniques until more advanced ones, including DL methods. A further relevant review focuses on one-class classification support vector machine (OCSVM), covering algorithms, parameter estimation, feature selection strategies and application domains (ALAM *et al.*, 2020).

2.2.1 Outliers, Anomalies and Novelties

OCC problems that involve detecting unusual patterns are addressed using several terminologies, each with specific distinctions depending on the context. The terms outliers, anomalies, and novelties refer to patterns that deviate from normal behavior. However, they are frequently used interchangeably in the literature. As Carreño *et al.* (2020) discuss, this overlap and confusion in terminology highlights the lack of a standardized vocabulary for describing and consistently characterizing deviations in data. To clarify these detection problems, this thesis proposes key attributes to characterize the emphasis for each area. Table 2 presents a comparative analysis based on these attributes. The following subsections will explore the origin, definition, and main features of each term.

Table 2 – Characterization of outlier, anomaly, and novelty detection.

Characteristic	**Outlier**	**Anomaly**	**Novelty**
Involve detecting deviating patterns	✓	✓	✓
Broader term historically originated from statistic field	✓		
Emphasis on detecting errors or noises in data	✓		
Emphasis on detecting anomalies of interest (e.g., faults)		✓	
Emphasis on detecting new data patterns			✓
Widely used in real-time detection		✓	✓
Widely used in real-time nonstationary data streams			✓

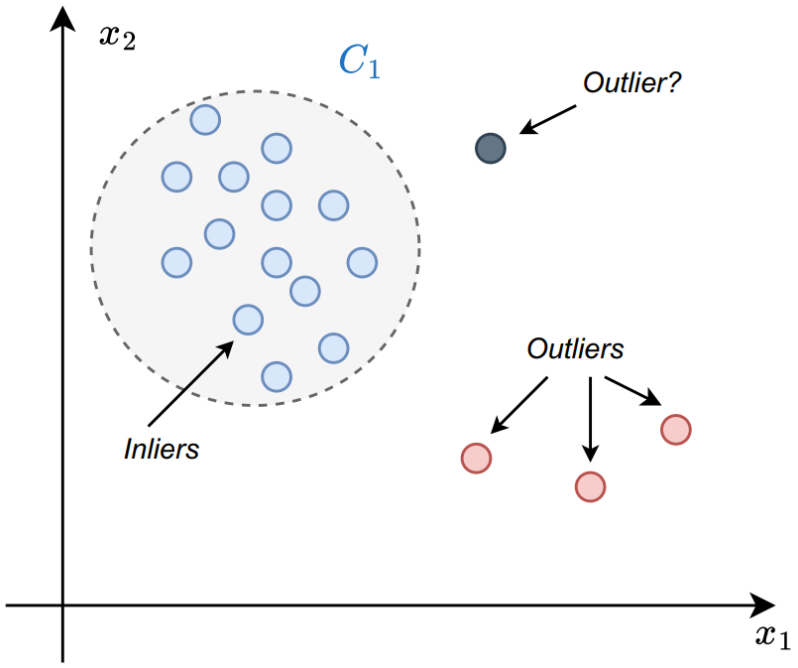
Source: the author (2024).

2.2.1.1 Outlier Detection

Outlier detection (OD) involves techniques for identifying data points that significantly deviate from other observations in a dataset (AGGARWAL; YU, 2001). These outliers can arise from a variety of causes, including human errors, malicious activities, or changes in environment (HAWKINS, 1980; HODGE; AUSTIN, 2004; CHANDOLA *et al.*, 2009). In certain fields, such as instrumentation and signal processing, these outliers are usually important, as they can indicate variability in measurements such as noise, experimental errors, or genuinely novel data instances (AGGARWAL; YU, 2001).

Figure 3 illustrates an OD problem in a two-dimensional data space (x_1 and x_2). A dense cluster, C_1, represents the inliers — the normal data points — enclosed by a dotted boundary. Three separate points, located far from the main cluster, are examples of outliers. The goal of an OD problem is to determine whether a new, out-of-sample data point, such as the gray one in the figure, is an inlier or an outlier.

Figure 3 – Illustrative example of an outlier detection problem.



Source: the author (2024).

The OD field has deep historical roots, with early explorations dating back to contributions such as the work on discordant observations by Edgeworth (1887). From a statistical perspective, OD typically assumes a specific data distributions for the normal data, and statistical tests are used to detect outliers (HAWKINS, 1980; BARNETT; LEWIS, 1994). A commonly used statistical rule for OD based on data distribution is the 3σ rule. In this approach, points deviating three times the *standard deviation* (σ) from the *mean* (μ) might be outliers (PUKELSHEIM, 1994). In other words, a data point x is considered an outlier if $x < \mu - 3\sigma$ or $x > \mu + 3\sigma$. This rule is formalized by the following discriminant function:

$$f_{3\sigma}(x) = \begin{cases} \textit{Outlier}, & \text{if } |x - \mu| > 3\sigma, \\ \textit{Inlier}, & \text{otherwise.} \end{cases} \tag{2.7}$$

Another classic statistical method involve using the interquartile range (IQR). It consists in a robust measure of variability in a dataset, defined as the difference between the third quartile ($Q3$), representing the 75th percentile, and the first quartile ($Q1$), representing the 25th percentile of the dataset, as described in the following expression:

$$IQR = Q_3 - Q_1. \tag{2.8}$$

The IQR effectively captures the middle 50% of the data, providing a sense of the spread of the central portion of the distribution. This metric is less susceptible to the influence of extreme values, which makes it particularly useful in outlier detection. Outliers can be determined by finding values that fall below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$, as these are typically indicative of deviation from the general trend of the data (TUKEY, 1977). This approach is also used in boxplot analysis, a graphical representation that relies on the IQR to visualize the dispersion of data and potential outliers (TUKEY, 1977; MCGILL *et al.*, 1978). A piecewise function that detects outliers using IQR rule is defined as follows:

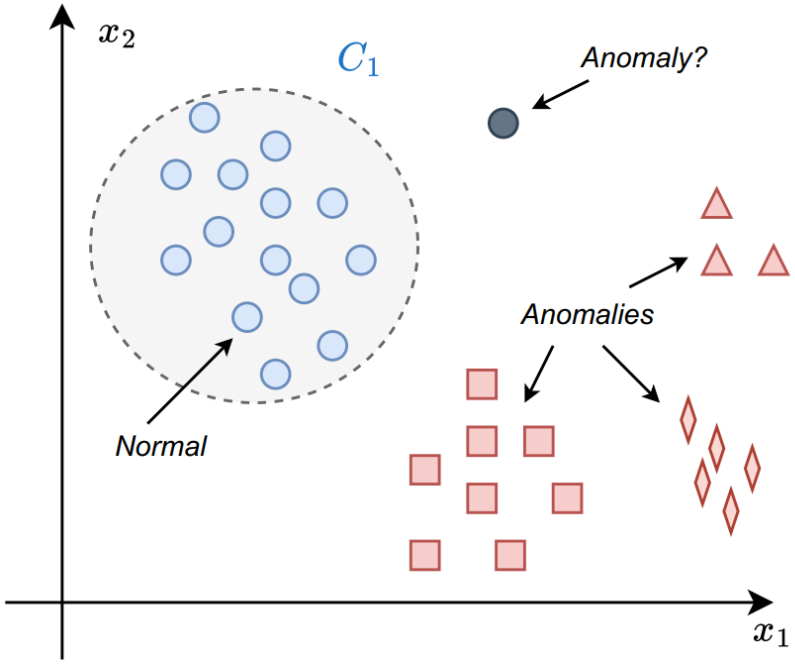
$$f_{IQR}(x) = \begin{cases} Outlier, & \text{if } x < Q_1 - 1.5 \times IQR \text{ or } x > Q_3 + 1.5 \times IQR, \\ Inlier, & \text{otherwise.} \end{cases} \tag{2.9}$$

In the literature of ML and statistics, numerous reviews and surveys have introduced and defined the main concepts and terminologies regarding outlier detection (HODGE; AUSTIN, 2004; ZIMEK *et al.*, 2012; GUPTA *et al.*, 2014; SOUIDEN *et al.*, 2017; ZIMEK; FILZMOSER, 2018; WANG *et al.*, 2019; PARK, 2019; BOUKERCHE *et al.*, 2020; ALGHUSHAIRY *et al.*, 2021; MARQUES *et al.*, 2023). For instance, Zimek *et al.* (2012) conduct a comprehensive review on the challenges and solutions associated with the identification of outliers in high-dimensional data within an Euclidean space, delving into the nuances of the *curse of dimensionality*, discussing its implications and relevant studies. In a more recent survey, Marques *et al.* (2023) focus on the comparison analysis involving OCC algorithms and unsupervised OD methods in a rigorous experimental setting. The survey presents an elucidation of the nomenclature pertinent to OCC and OD, alongside an extensive assessment of the various learning methodologies within the OCC domain and its related problems, including a comparative analysis on their performance.

2.2.1.2 *Anomaly Detection*

Anomaly detection (AD) involves identifying data patterns that deviates from normal behavior (CHANDOLA *et al.*, 2009; NASSIF *et al.*, 2021). Figure 4 depicts an illustrative example of an AD problem. Within the figure, a dense cluster, C_1, represents the inliers. Outside the cluster, various types of anomalies are depicted, illustrating their diverse origins.

Figure 4 – Illustrative example of an anomaly detection problem.



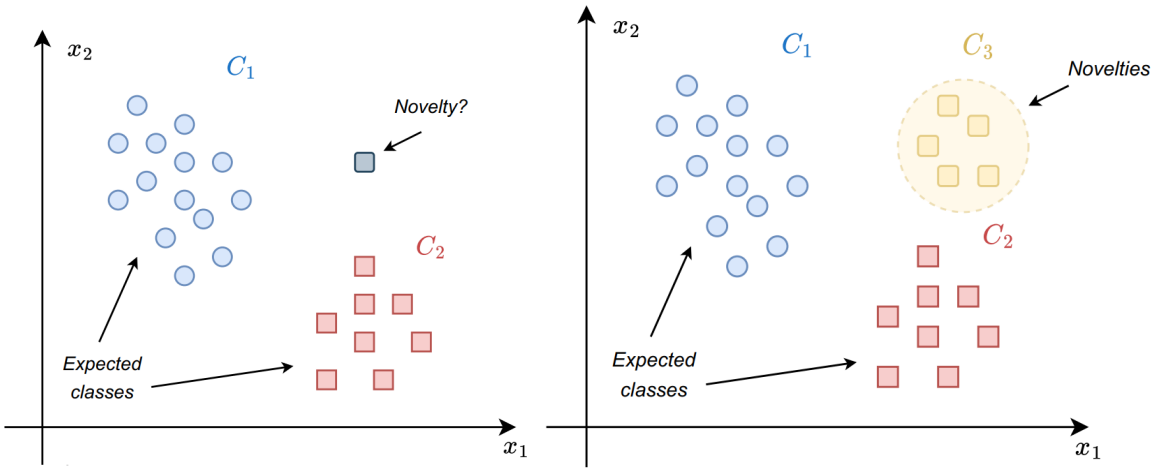
Source: the author (2024).

OD and AD are often used interchangeably, with AD considered a subset of OD where deviating data points are termed anomalies. The definition of an anomaly is context-dependent, but it generally refers to an unexpected data behavior that may represent an event of interest. Anomalies are frequently used for detecting critical events such as credit card fraud (PORWAL; MUKUND, 2019), epilepsy seizures (YOU *et al.*, 2022), heart arrhythmia (LIU *et al.*, 2022), surfaces defects (TSAI; JEN, 2021), network intrusions (GIACINTO *et al.*, 2008; CHEN *et al.*, 2021; ZHANG *et al.*, 2022; SARIKAYA *et al.*, 2023), and system failures (LAU *et al.*, 2013; YIN *et al.*, 2014; LIU; ZHANG, 2020; AMINI; ZHU, 2022; WANG *et al.*, 2022). Several reviews on AD have explored various techniques, challenges, and applications (AGGARWAL; YU, 2001; CHANDOLA *et al.*, 2009), while others have focused on domain-specific areas like time series (BLÁZQUEZ-GARCÍA *et al.*, 2021), data streams (PARK, 2019), financial fraud (HILAL *et al.*, 2022) and hyperspectral imagery (HSI) (HU *et al.*, 2022).

2.2.1.3 *Novelty Detection*

Novelty detection (ND) focuses on identifying new or previously unobserved patterns in the data. Figure 5 illustrates this concept in two dimensions. As shown in Figure 5a, there are two known classes (C_1 and C_2), and a new, unknown gray data point that may not belong to either classes. The goal of ND is to identify if this new data point belongs to a new class, such as C_3 in Figure 5b. A specific characteristic of ND is that novelties are frequently incorporated into ND models after detection, meaning that it actively learns from new patterns rather than treating them as noise or exceptions (CHANDOLA *et al.*, 2009).

Figure 5 – Illustrative example of a novelty detection problem.



(a) Unknown data pattern (b) Novel class detected

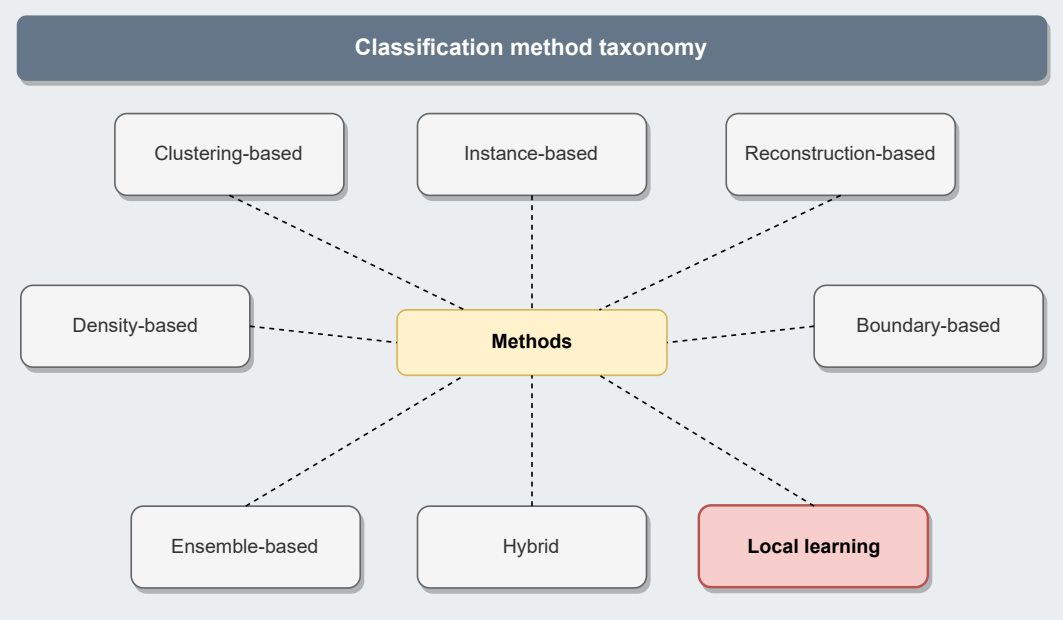
Source: the author (2024).

Novelty detection spans across various domains, including signal processing, computer vision, pattern recognition, data mining, and robotics (MARKOU; SINGH, 2003; PIMENTEL *et al.*, 2014). ND is particularly useful in applications where novelties are rare and unpredictable, making traditional MTCC methods unsuitable due to the data scarcity regarding unlabeled data. ND techniques encompass a variety of approaches, including statistics, information theory, support vector machines, kernel methods, and neural networks. These techniques typically compute a novelty score that is compared to a threshold to detect novelties (PIMENTEL *et al.*, 2014). For data streams, Faria *et al.* (2016) provide a comprehensive review of ND, highlighting key aspects such as the difference between offline and online detection; the number of classes; the efficacy of ensemble versus singular classification; and concept drift.

2.2.2 Modeling Approaches

The foundation of ML lies in systematic learning methods designed to extract patterns and knowledge from data (BISHOP, 2006). Within this context, OCC emerges as a multidisciplinary field with diverse applications and a broad spectrum of learning approaches (CARREÑO *et al.*, 2020; ALAM *et al.*, 2020; PERERA *et al.*, 2021). This subsection introduces a taxonomy to classify the most common OCC techniques, as shown in Figure 6. It then provides a review of each learning approach within this taxonomy. Local learning (highlighted in red) will be discussed in Chapter 3.

Figure 6 – Taxonomy of learning methods for one-class classification.



Source: the author (2024).

While not part of this taxonomy, DL methods are also widely used in OCC, often involving techniques such as

- **CNNs** (OZA; PATEL, 2019; DONG *et al.*, 2022; FANG *et al.*, 2022; JANA *et al.*, 2022);
- **AEs** (GEDDES *et al.*, 2019; YU *et al.*, 2021; TSAI; JEN, 2021; ANDRESINI *et al.*, 2021; TAKHANOV *et al.*, 2023);
- **LSTMs** (MALHOTRA *et al.*, 2015; MALEKI *et al.*, 2021; LINDEMANN *et al.*, 2021; LIU *et al.*, 2022); and
- **GANs** (YANG *et al.*, 2019b; LIU *et al.*, 2019; AKCAY *et al.*, 2019; LI *et al.*, 2020; KIM *et al.*, 2021; LEE *et al.*, 2022; KATZEF *et al.*, 2022; PU *et al.*, 2022; XIA *et al.*, 2022).

2.2.2.1 Density-based Models

Density-based modeling employs probabilistic techniques to estimate the density distribution of normal data. These methods focus on deriving the pdf of the data, enabling the establishment of a threshold with a probabilistic interpretation to distinguish abnormal instances from regular data. Simple density-based models typically assume a Gaussian distribution, using the 3σ rule or the IQR to identify novelties that lie outside a predefined threshold (CHANDOLA *et al.*, 2009). More advanced approaches include mixtures of parametric distributions, such as Gaussian mixture models (GMMs), which provide a more flexible framework for modeling complex data distributions (CHANDOLA *et al.*, 2009). GMMs have proven effective in a variety of AD applications, including cybernetic attacks (AN *et al.*, 2022), video monitoring (FAN *et al.*, 2020), energy systems (TANG *et al.*, 2015) and wind turbines (MATSUI *et al.*, 2022). Additionally, innovative adaptations of GMM have emerged. For example, the deep autoencoding GMM introduced by Zong *et al.* (2018) combines a deep AE for dimensionality reduction and reconstruction error computation with a GMM for anomaly detection. Similarly, Mohammadi-Ghazi *et al.* (2018) propose a boosted conditional GMM using forward stage-wise additive modeling and boosting techniques to estimate the conditional densities of observed variables, facilitating novelty detection.

Nonparametric approaches, such as kernel density estimator (KDE), are also widely employed in density-based modeling in OCC domain (LATECKI *et al.*, 2007; PAVLIDOU; ZIOUTAS, 2014). KDE estimates the density function by superimposing kernels (typically Gaussian) over each data point, resulting in a smooth and continuous representation of the data distribution (HASTIE *et al.*, 2009). For instance, Pavlidou and Zioutas (2014) propose a robust OD algorithm that uses a weighted KDE to identify outliers by removing points with the lowest robust pdf values, thereby refining the clean data subset.

The local outlier factor (LOF), introduced by Breunig *et al.* (2000), is another popular density-based model that measures the local density deviation of a data point relative to its neighbors. The idea is to identify how isolated the point is in relation to the surrounding neighborhood. A higher LOF value indicates that the data point is significantly different in density from its neighbors, suggesting it is an outlier. LOF-based learning have been widely explored in the literature of OD, AD and ND (BREUNIG *et al.*, 2000; POKRAJAC *et al.*, 2007; GAO *et al.*, 2011; SONG *et al.*, 2014; PAULAUSKAS; BAGDONAS, 2015; DING *et al.*, 2018; DENG; WANG, 2018; YANG *et al.*, 2019a; SU *et al.*, 2019; ALGHUSHAIRY *et al.*, 2021).

For instance, Gao *et al.* (2011) address two limitations of LOF-based models: firstly, the local density estimate of LOF lacks the required accuracy to identify outliers in intricate and sizable datasets; secondly, LOF's performance is contingent upon the parameter k, determining the scale of the local neighborhood. To address these challenges, the authors propose a method, known as robust kernel-based LOF, that incorporates a variable KDE to enhance accuracy and a weighted neighborhood density estimate to enhance robustness against variations in the parameter k. They also introduce a novel kernel function, the *Volcano kernel*, specifically designed for OD. Furthermore, Alghushairy *et al.* (2021) review LOF for static and stream data, highlighting gaps and future directions. Meanwhile, Blázquez-García *et al.* (2021) conduct a structured overview of unsupervised OD techniques for time series, introducing a detailed taxonomy that categorizes existing methods based on characteristics such as data structure, detection strategy, and the nature of the outliers being identified.

2.2.2.2 Clustering-based Models

Clustering involves grouping data into clusters where similar data instances are placed together. In the OCC and detection field, clustering-based algorithms often follow a two-step process: first, data is grouped using clustering algorithms, and then the deviation of data points from these clusters is analyzed (PIMENTEL *et al.*, 2014; BOUKERCHE *et al.*, 2020). In general, it relies on defining clusters for normal data and such clusters are used for detecting anomalies. Any data point that is not similar to the cluster representation (i.e., does not belong to the cluster) is then assigned as abnormal.

Another approach related to clustering-based methods is prototype-based learning. Prototype-based OCC methods use a set of prototypes to represent the normal class, assessing abnormality by calculating the minimum distance between a test point and its nearest prototype. To determine optimal prototype locations, various strategies are used, with K-means clustering being widely applied due to its simplicity (MARKOU; SINGH, 2003). Within the area of vector quantization and clustering analysis, the technique known as self-organizing map (SOM) has inspired the development of various approaches for detecting outliers, anomalies and novelties. Numerous studies have delved into exploring Kohonen's unsupervised methods in this context (MUÑOZ; MURUZÁBAL, 1998; ALBERTINI; MELLO, 2007; SHAHREZA *et al.*, 2011; AGUAYO; BARRETO, 2017; PISANO *et al.*, 2017; YANG *et al.*, 2019a; DRUMOND *et al.*, 2022). For instance, Frota *et al.* (2007) investigate AD in mobile communication networks,

treating it as a pattern recognition task aimed at identifying abnormal or unknown behavior within the monitored system. In their work, they propose a general procedure for determining decision thresholds, utilizing Kohonen's SOMs and incorporating nonparametric confidence intervals based on percentiles. Additionally, Albertini and Mello (2007) investigate adaptive models in ND, focusing on the identification of unexpected or unknown patterns. They introduce a neural network, called self-organizing novelty detection, specifically designed for online and unsupervised learning.

Cluster-based methods are also explored in a hybrid setting. Fang *et al.* (2022) introduce an AD method which combines hierarchical clustering (HC) with support vector machine (SVM) applied on diabetes data. The HCSVM method first categorizes diabetes data based on features using a clustering algorithm. It then segments the data into significantly abnormal and potentially abnormal parts. For the analysis of each segmented part, the authors employ a CNN to detect anomalies. Furthermore, Zhou *et al.* (2022) design an innovative approach namely clustering-based collective anomaly detection framework for continuous monitoring the collective anomaly in sliding windows over network traffic. Collective anomaly is characterized by a group of similar data instances that, when considered together, display anomalous behavior compared to the overall dataset, even if they may not appear unusual individually.

2.2.2.3 Instance-based Models

Instance-based learning, also referred as memory-based or lazy learning, consists on learning methods that rely on stored instances of training data, using them as a reference for classification based on similarity measures. It involves mainly nearest neighbor (NN) algorithms, and therefore, it may also be referred to as NN-based (CHANDOLA *et al.*, 2009; KHAN; MADDEN, 2014; PIMENTEL *et al.*, 2014). NN learning involves methods that classify new data points based on how similar they are to available data points. The k-nearest neighbour (KNN) algorithm is a classic example of instance-based learning. It operates by assessing the distance of a new instance from the closest points in the training set, with outliers typically being those instances that are significantly distant from these nearest neighbors. KNN is considered as a technique for estimating the density function of data, hence, it is sometimes referred in the literature to as a nonparametric density-based approach (MARKOU; SINGH, 2003).

Instance-based methods are valuable for their adaptability and flexibility, making them particularly useful in dynamic environments where data patterns change over time. Nevertheless, they face several challenges. A primary issue is the *curse of dimensionality*, which can make distance calculations less meaningful in high-dimensional spaces. Additionally, these methods are sensitive to the selection of parameters, such as the number of neighbors in the KNN algorithm, which can significantly affect performance. The approach can also be computationally intensive for large datasets due to the numerous distance calculations, and it may be sensitive to noise and outliers within the training data. Despite these challenges, instance-based learning has proven effective in various applications (GU *et al.*, 2019). For example, Idé *et al.* (2007) address the challenge of change analysis in correlated multi-sensor systems, with a focus on monitoring sensor performance in the automotive industry. The authors introduce the concept of a stochastic neighborhood, a novel approach that allows for the robust computation of anomaly scores for each sensor. Similarly, Guo and Shui (2020) proposed a KNN based AD method for sea-surface target detection.

2.2.2.4 Boundary-based Models

Boundary-based methods concentrate on establishing a decision surface around normal data, labeling anything beyond this boundary as anomalous. Training such models requires only normal class data and the learning process uses the normal training samples to establish a discriminant function that characterizes the normal class. Typically, this approach involves setting a threshold for the classification of new instances (MARQUES *et al.*, 2023). Despite their effectiveness, these methods face several challenges. They are vulnerable to outliers or noise in the training data, sensitive to feature scaling, and can have a high computational cost due to complex optimization tasks (MARQUES *et al.*, 2023).

One-class classification support vector machine (OCSVM) (SCHÖLKOPF *et al.*, 2001) and support vector data description (SVDD) (TAX; DUIN, 1999) are classic examples of boundary-based methods. OCSVM is a specialized version of the SVM designed for unsupervised learning, particularly useful for AD or OD (SCHÖLKOPF *et al.*, 2001). Unlike traditional SVMs that separate two classes using a hyperplane, OCSVM focuses on a single class: the normal data. It aims to find the optimal hyperplane that separates all the data points from the origin in a high-dimensional feature space, maximizing the margin between the hyperplane and the origin. This is achieved by mapping the data points into a higher-dimensional space using

a kernel function, typically the radial basis function (RBF), also known as Gaussian kernel. Alternatively, SVDD aims to minimize the volume of a sphere that covers the training data points, ensuring it tightly encompasses the normal data. SVDD also uses kernel functions to map the data into a high-dimensional feature space where this spherical boundary is constructed. Data points that fall outside of this sphere are classified as anomalies (TAX; DUIN, 1999). Both OCSVM and SVDD are widely explored in the literature (MANEVITZ; YOUSEF, 2001; SHIN *et al.*, 2005; WANG *et al.*, 2009; YIN *et al.*, 2014; CHEN *et al.*, 2015; ERFANI *et al.*, 2016; ALAM *et al.*, 2020; ZHANG; DENG, 2021; WU *et al.*, 2022).

2.2.2.5 Reconstruction-based Models

Reconstruction-based modeling is a crucial approach in ML, particularly effective in identifying novel or unusual patterns in data. Reconstruction methods aim to create a compressed representation of the training data while retaining most of its information. Once this compact representation is established, new instances can be reconstructed back into the original data space using this model. The reconstruction error, which is the discrepancy between the original and the reconstructed instance, serves as a measure of how closely a new instance resembles the training distribution (MARQUES *et al.*, 2023).

Principal component analysis (PCA), from Pearson (1901), and kernel principal component analysis (KPCA), introduced by Schölkopf *et al.* (1997), are two foundational techniques in this domain (HOFFMANN, 2007; LAU *et al.*, 2013; SONG *et al.*, 2014; WU *et al.*, 2019; MOON *et al.*, 2022). The linear PCA works by transforming the data into a set of linearly uncorrelated variables known as principal components (PCs), which are then used to reconstruct the input data. KPCA extends PCA to nonlinear data, using kernel methods to project data into a higher-dimensional space, enabling the capture of more complex structures (SCHÖLKOPF *et al.*, 1997; HOFFMANN, 2007). Both techniques depend on the reconstruction error as an indicator of novelty, where larger errors suggest novelty.

Autoencoders (AEs), another widely used method in reconstruction-based detection, operate by learning a compressed representation of the input data through a model composed of two parts: an encoder that reduces the data to a lower-dimensional space, and a decoder that reconstructs the data back to its original dimension (GOODFELLOW *et al.*, 2016). AEs have been widely applied in OCC problems across various applications and related fields, with recent research exploring various architectural enhancements and application domains. Researchers

have used AEs for automated defect detection in manufacturing (TSAI; JEN, 2021; WU *et al.*, 2022), structural health monitoring (JANA *et al.*, 2022), detecting anomalies in water levels (NICHOLAUS *et al.*, 2021), and forensic timelines (STUDIAWAN; SOHEL, 2021).

Several works have improved AE performance by addressing its limitations. For instance, ensemble methods have been used to mitigate sensitivity to noise and the need for large datasets (CHEN *et al.*, 2017; AN *et al.*, 2022). Other approaches have focused on enhancing robustness and feature learning through techniques such as incorporating beta-divergence to handle outliers (AKRAMI *et al.*, 2022), using adversarial training to learn more semantically meaningful features (SALEHI *et al.*, 2021; CHEN *et al.*, 2021; LIU *et al.*, 2021), or leveraging orthogonal projection constraints to separate normal and abnormal data (YU *et al.*, 2021). Additionally, novel architectures like discriminative compact AEs (PARK *et al.*, 2021), deep denoising AEs (LEE *et al.*, 2020), and graph convolutional AEs for multivariate time series (MIELE *et al.*, 2022) have been proposed to address specific issues like latent space collapse, class imbalance, and complex sensor networks.

2.2.2.6 Ensemble-based Models

Ensemble methods involve the strategic combination of multiple individual models with the goal of enhancing the overall performance of the system. Instead of relying on a single model, ensemble techniques leverage the diversity of multiple models to collectively make predictions or decisions (CHAN *et al.*, 2020). The idea is that multiple weak detectors can together form a more robust and accurate detector (HASTIE *et al.*, 2009). Ensemble learning offers several advantages, as it typically yields higher performance and robustness compared to individual models. In addition, ensembles are particularly effective in reducing overfitting, as they average out individual biases and variances, leading to improved generalization on unseen data (HASTIE *et al.*, 2009). Nevertheless, ensemble methods can be computationally intensive and time-consuming, as they require training multiple models. They also tend to be more complex, both in terms of implementation and interpretation of the results. This complexity can make the tuning of hyperparameters more challenging and can lead to difficulties in troubleshooting and maintenance (HASTIE *et al.*, 2009).

In the literature, several works have proposed ensemble-based models for OCC (GIACINTO *et al.*, 2008; KRAWCZYK *et al.*, 2014; CHEN *et al.*, 2017; CHAN *et al.*, 2020). For instance, Giacinto *et al.* (2008) propose a modular ensemble approach for network intrusion detection. Recognizing the complexity of network traffic, characterized by diverse protocols and services, they developed a modular classification system. Each module within this system is specialized to model a specific group of similar protocols or services, facilitating a more efficient and targeted analysis of network traffic's multifaceted nature. In Krawczyk *et al.* (2014), the authors propose a multi-class classifier based on weighted OCSVM in a clustered feature space. Their approach involves dividing the target class into smaller, atomic subsets, which are then utilized as inputs for individual one-class classifiers. Chen *et al.* (2017) introduce an AE ensemble for OD that diverges from the traditional use of fully connected AEs by relying on a variety of randomly connected AEs. These AEs differ in their structures and connection densities, which notably reduce computational complexity. Additionally, they integrate an adaptive sample size method within the ensemble framework, targeting both enhanced diversity among the models and reduced training time.

2.2.2.7 Hybrid Models

Hybrid learning models or HIS consist of methods formed by a combination of two or more distinct learning techniques or algorithms to improve performance, leveraging on the strengths of each constituent model or overcoming their individual weaknesses (WOŹNIAK *et al.*, 2014). For instance, Jove *et al.* (2021) propose a hybrid classifier based on one-class techniques for detecting anomalies in an industrial plant designed to control the water level in a tank. Additionally, Vos *et al.* (2022) introduce a deep learning method that combines LSTM and SVM for detecting anomalies in vibration signals from two applications: endurance test of a reduction gearbox and helicopter test flight.

An interesting hybrid deep learning method recently explored in the literature is known as long short-term memory autoencoder (LSTM-AE) (MALEKI *et al.*, 2021; YANG *et al.*, 2021; ZHANG *et al.*, 2022). An LSTM-AE is essentially a combination of LSTM networks and the AE architecture. In this setup, the LSTM layers are used in both the encoder and decoder of the AE. The encoder compresses the input sequence into a lower-dimensional representation, capturing the essential features of the data. This compression is particularly adept at preserving temporal dependencies and patterns due to the inherent design of LSTMs,

which are well-suited for learning from sequences with long-term dependencies. The decoder then attempts to reconstruct the original input sequence from this compressed representation. The reconstruction error is often used to identify anomalies. In this context, Zhang *et al.* (2022) introduce a novel AD technique based on LSTM stacked denoising autoencoder for detecting and diagnosing anomalies in wind turbines. Yang *et al.* (2021) explore LSTM-AE combined with logistic regression for detecting false data injection attack in alternating current power systems.

NN-based algorithms are also used in a form of hybrid systems. For instance, Muhammad *et al.* (2021) combine clustering and the NN techniques for effective OD. This method facilitates a comprehensive understanding and insightful visualization of data behavior in healthcare safety. In addition, Mesarcik *et al.* (2022) propose a hybrid model derived from both instanced-based and reconstruction-based learning approaches, called nearest-latent-neighbours.

Regarding LOF-based modeling, Yang *et al.* (2019a) propose the neighbor entropy LOF, a method that employs an optimized self-organizing feature map clustering algorithm to cluster the dataset, reducing the computation of the LOF model within smaller clusters. Additionally, the algorithm leverages the entropy of the relative K-distance neighborhood to redefine the LOF, thereby enhancing the accuracy of OD. By combining the advantages of multiple approaches, hybrid models aim to achieve better generalization, accuracy, or robustness than standalone models. The integration can be performed at different stages of the learning process, from data preprocessing to the decision-making phase. Further information about HIS can be found in the work of Woźniak *et al.* (2014), where the authors present a survey that highlights essential aspects of HIS, such as the methods of decision fusion and the importance of diversity in classification systems.

2.3 Summary

This chapter introduces the principles of pattern classification and OCC, followed by a broad literature review on the subject. Hereafter, as this thesis focuses on ND, the terms *outlier* and *anomaly* will be used interchangeably with *novelty*.

3 ON LOCAL LEARNING

This chapter provides an overview of the local learning paradigm, introducing its key concepts and conducting a literature review of studies and applications in the field. In addition, it presents the principles of cluster-based local modeling with hard partitioning (CLHP), providing the necessary background for understanding the learning methods proposed in this thesis.

3.1 An Overview About Local Learning

Local learning is an ML paradigm that leverages locality principles to enhance the performance of learning structures or provide a more accurate representation of data. This area has been a recurrent theme in the literature, with relevant contributions spanning several decades (ALPAYDIN; JORDAN, 1996; WEINGESSEL; HORNIK, 2000; KUMPULAINEN; HÄTÖNEN, 2008; ZHAO *et al.*, 2023; GÔLO *et al.*, 2023). The following sections will delve into related ML problems, learning approaches, locality criteria, and challenges.

3.1.1 Problems and Applications

Local learning has proven its utility in both MTCC and OCC problems. In MTCC, methods often focus on enhancing local relationships between data points. For instance, Cerri *et al.* (2014) introduce a local method for hierarchical multi-label classification by training a multi-layer perceptron (MLP) for each level of the hierarchy. In addition, Gou *et al.* (2019) propose two locality constrained representation-based KNN rules. The first one is a method that represents a test sample as a weighted linear combination of its K-nearest neighbors from each class. The second method depends on calculating K-local mean vectors from the NNs of each class. More recently, Gou *et al.* (2023) introduced the locality-constrained weighted collaborative-competitive representation-based classification, a supervised method that uses local information from both positive and negative nearest samples to enhance performance (GOU *et al.*, 2019). Regarding OCC local learning, Wang *et al.* (2018) present a OD model that combines graph representations with local data. Meanwhile, Dang *et al.* (2013) introduce a technique, named local outlier detection with interpretation, that employs a strategy that utilizes quadratic entropy for the dynamic selection of neighboring data points.

Other local approaches leverages specialized techniques, such as multiple kernel learning (MKL). Gautam *et al.* (2018) introduce an MKL-based method for AD which locally

adjusts kernel weights to capture the inherent locality of data. Similarly, Chen *et al.* (2011) propose an MKL-based data domain description algorithm with an elastic-net-type constraint on the kernel weights for dealing with issues regarding to model's sparsity and susceptibility to noise and redundant information.

Regarding local modeling based on Kohonen's SOM and competitive neural networks, Kumpulainen and Hätönen (2008) introduced an AD method using a SOM to represent normal states in communication networks, where deviations from the SOM nodes are identified as anomalies. Similarly, Barreto *et al.* (2005) proposed an unsupervised approach that employs competitive neural algorithms applied on cellular network condition monitoring. In this approach, the training phase involves utilizing state vectors that represent the normal operation of a network. Post-training, both global and local normality profiles are constructed based on the distribution of quantization errors from training state vectors and their components. The global normality profile assesses the overall condition of the cellular system, and in the event of abnormal behavior detection, local normality profiles are employed component-wise to identify abnormal state variables. More recently, Aguayo and Barreto (2017) conducted an extensive investigation involving the application of dynamic SOM neural networks to address the challenge of ND in time series. They assessed the impact of different short-term memory kernels (e.g., *Gamma*, *Gamma II*, and *Laguerre*) on the performance of nonrecurrent dynamic SOM; analyzed the competitive advantages of recurrent dynamic SOM over nonrecurrent architectures in ND for time series; and proposed an alternative approach for dynamic SOM-based ND by revisiting the operator map (OPM) introduced by Lampinen and Oja (1989).

Several studies have concentrated on the integration of ensemble and local learning approaches. For instance, Moon *et al.* (2022) present an ensemble learning strategy for AD on cyber physical system. This technique utilizes various PCs to capture subtle data variations. Instead of relying solely on PCs with high variances as in typical PCA methods, their technique also uses PCs with low variances to capture subtle variations in training data. The ensemble of classifiers utilizing different PC combinations enhances accuracy and the detection of varied unknown attacks, demonstrating its adaptability across complex cyber physical system environments. Alternatively, Abdallah *et al.* (2021) propose a method that first applies a clustering algorithm to the positive training data. Subsequently, it trains individual OCC classifiers on the data samples within each cluster. For prediction, a data point is considered normal only if all classifiers unanimously predict it as such; otherwise, it is labeled as abnormal.

Local learning in clustering problems, specifically multiview clustering, utilize multiple data sources to extract complementary information. For example, Kang *et al.* (2020) propose an unified multiview subspace clustering model which incorporates the graph learning from each view, the generation of basic partitions, and the fusion of consensus partition. Another example is the non-negative matrix tri-factorization, a collaborative clustering approach that mines the duality between samples and features to capture both their clustering structures (TANG; FENG, 2022). Alternatively, Zhou *et al.* (2020) developed an local learning-based multi-task clustering method that explores discriminative information in a low-dimensional subspace while simultaneously performing clustering for multiple views. This approach integrates transfer learning, subspace learning, local manifold learning, and clustering, introducing a joint projection of heterogeneous features to control shared features and facilitate knowledge transfer across tasks.

Local learning structures have also proven valuable in the fields of regression, dimensionality reduction, signal processing, and dynamic systems. For instance, Hoffmann *et al.* (2009) investigate the suitability of local linear dimensionality reduction alongside locally-weighted regression for handling high-dimensional nonlinear regression tasks. The authors evaluate linear dimensionality reduction techniques under three categories: input data dimensionality reduction, joint input-output data modeling, and projection-output data correlation optimization. In system identification domain, Júnior *et al.* (2015) propose the regional models (RMs), a method that combines aspects of both global and local modeling. This approach initially segments the input space using a SOM and then clusters the prototypes generated by the trained SOM. Rather than building models for each prototype, as in local modeling, RMs develop regression models based on the centroids of these prototypes. The authors tested this approach by building linear and nonlinear regional models. Regarding nonlinear process monitoring, Cui *et al.* (2019) introduce the ensemble local KPCA, a framework that combines single local KPCA models using ensemble learning. Alternatively, Wu *et al.* (2019) propose the local and global randomized PCA, a technique that offers a more refined representation of input data compared to conventional KPCA, making it highly compatible with real-time and expansive process monitoring. Regarding control systems, Macció and Cervellera (2012) introduced a local model based on Nadaraya-Watson framework, addressing the challenge of developing an automatic controller that leverages data collected during the operation of complex systems by a reference teacher (e.g., a human operator).

Local learning has been applied to various specialized domains. For example, Isomura and Toyoizumi (2016) explored a local learning rule for independent component analysis. For detecting fake news from textual features, Gôlo *et al.* (2023) investigate a multimodal method that represents texts through an OCC model. This model learns a new representation from a combination of text embeddings, topic, and linguistic information. For face alignment, Park *et al.* (2017) employed deep neural networks to combine local features and recurrent regression. For facial recognition and tracking, Wang *et al.* (2013) used an incremental local sparse representation to address issues like occlusion and illumination variation. For 3D modeling, Zhang *et al.* (2023) proposed a generative AE that handles component combinations for objects. This AE uses segmented input objects to create component volumes that have redundant components and random configurations. It then can be applied to either multiple object categories for structure hybrid or a single object category for shape completion. In image classification, Zhao *et al.* (2015) addressed the scarcity of labeled samples in automatic image annotation by proposing a graph-based semi-supervised technique that focuses on both local and global regression regularization. For medical imaging, particularly in the detection of tuberculosis from chest x-ray images, Ding *et al.* (2017) presented a fusion of local and global classifiers.

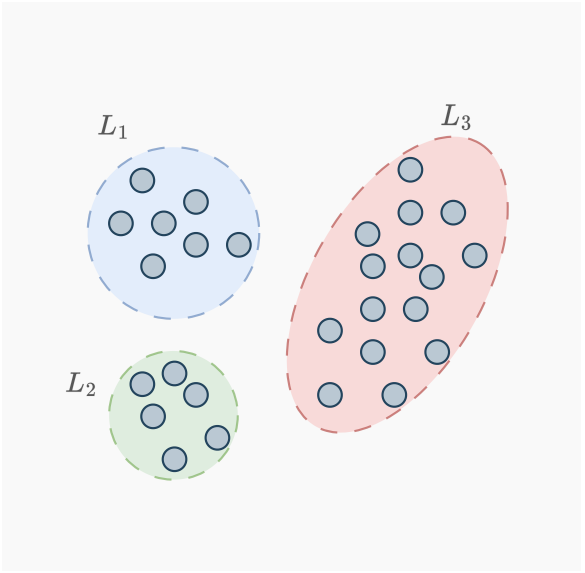
In the field of manifold learning, Zhan and Yin (2011) investigate the noise manifold learning problem, and propose a robust version of local tangent space alignment (LTSA) from the following aspects: firstly, a robust PCA algorithm based on iterative weighted PCA is employed instead of the standard Singular Value Decomposition (SVD) to reduce the influence of noise on local tangent space coordinates; secondly, robust local tangent space alignment (RLTSA) chooses neighborhoods that are well approximated by the local coordinates to align with the global coordinates; thirdly, in the alignment step, the influence of noise on embedding result is further reduced by endowing clean data points and noise data points with different weights into the local alignment errors. Experiments on both synthetic and real-world datasets demonstrate the effectiveness of the RLTSA when dealing with noise manifold (ZHAN; YIN, 2011).

In summary, the local learning paradigm offers a powerful and adaptable approach to a wide range of ML problems. By concentrating on localized data structures, these methods excel in scenarios where global models may fail to capture locality-dependent patterns.

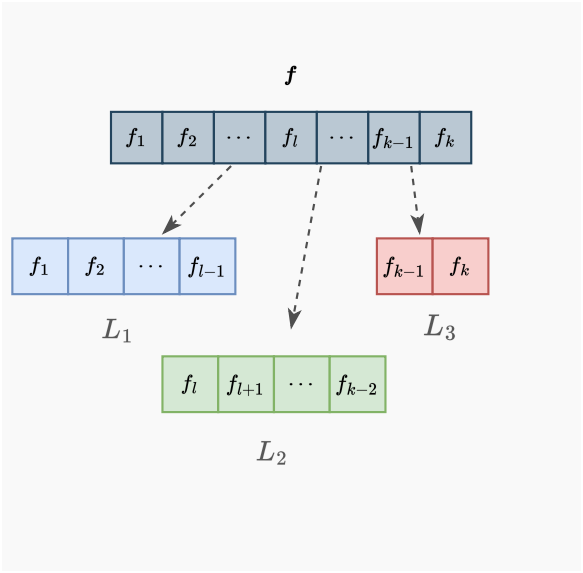
3.1.2 Locality Criteria

Local learning techniques are fundamentally defined by their locality criteria. The concept of "locality" is multifaceted, encompassing a diverse range of domains. Figure 7 illustrates four examples of locality criteria in local learning: data space, feature space, graph space and source space. These examples demonstrate how local learning models incorporate neighborhoods based on different notions of "closeness" for each domain. Each example comprises three different localities: L_1 (blue), L_2 (green) and L_3 (red).

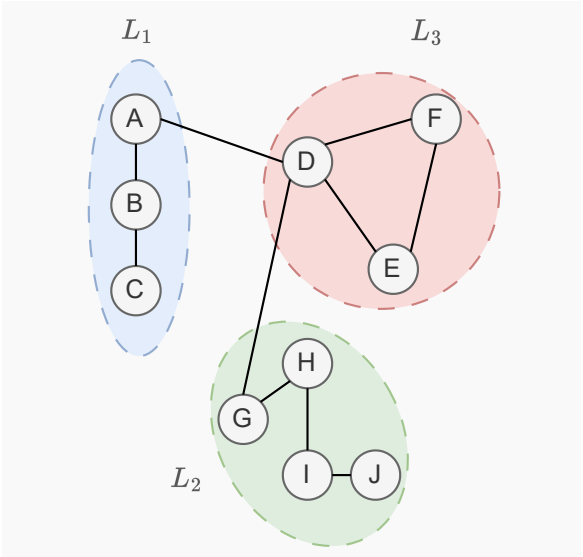
Figure 7 – Illustrative examples of different types of locality criteria.



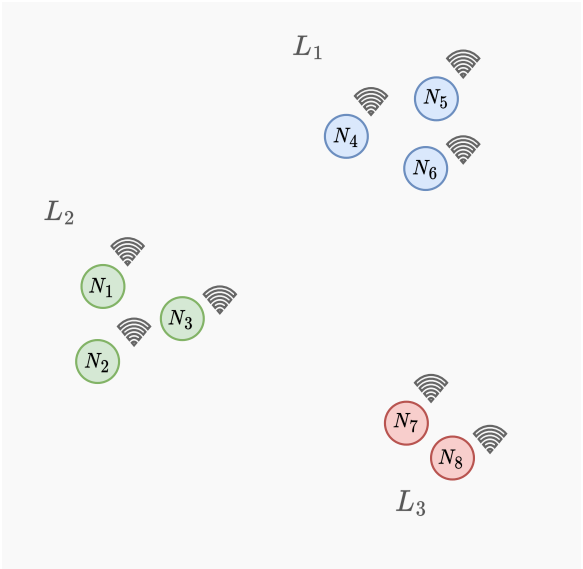
(a) Data space locality



(b) Feature space locality



(c) Graph space locality



(d) Source space locality

Source: the author (2024).

In *data space* locality, unlike traditional ML methods that trains a model using all observations, local techniques rely on specific data subsets. For example, Jacobs *et al.* (1991) introduce a supervised learning method for systems with multiple distinct networks, each specialized in a subset of the whole training data. Alternatively, Drumond *et al.* (2022) propose a novel classification method, the regional classifier (RC), which bridges the gap between global and local classification. RC is a two-level clustering-based method that create a refined input representation through a SOM, followed by the application of a clustering algorithm like K-means to the SOM units. Several other studies have also employed similar data space local learning strategies (HOFFMANN *et al.*, 2009; PERES; PEDREIRA, 2010; ZHAN; YIN, 2011; DANG *et al.*, 2013; JÚNIOR *et al.*, 2015; ABDALLAH *et al.*, 2021).

Other approaches focus on selecting or extracting localized *features* within the data (ARMANFARD *et al.*, 2016; PARK *et al.*, 2017; WU *et al.*, 2019; GRATI *et al.*, 2020; HU *et al.*, 2020; MOON *et al.*, 2022). For instance, Armanfard *et al.* (2016) present a localized feature selection methodology that uniquely tailors feature sets to each specific region of the sample space, ensuring an optimal fit to data space local characteristics. Similarly, Moon *et al.* (2022) introduce a novel ensemble method that extracts specific features optimal for AD. Unlike typical PCA that prioritizes high-variance PCs, this method also incorporates low-variance PCs to capture subtle variations in training data. In face recognition field, Grati *et al.* (2020) present a cutting-edge RGB-D method that leverages on local representations on facial patch description and matching. This approach innovates in the learning and integration of data-driven descriptors to characterize local patches around reference points in images. Additionally, Hu *et al.* (2020) propose a novel local-global feature extraction method that combines KPCA and an AE to autonomously extract discriminative features from the frequency spectrum of vibration signals.

The concept of graph locality, where nodes and edges depict localized relationships, has become increasingly relevant, especially within the domain of local learning. The study by Zhao *et al.* (2015) introduce a graph-based semi-supervised learning method for image classification that leverages both local and global regularization techniques. This method capitalizes on local classification functions to preserve the discriminative and geometric properties intrinsic to data, while simultaneously considering global discriminative features and computing projection matrices for the extrapolation of future data. Graph-based methods have been particularly effective in OD applications due to their ability to represent interdependencies among objects. However, a common shortfall of these methods is their neglect of local information around each

node, often resulting in high false-positive rates (WANG *et al.*, 2018). To address this, Wang *et al.* (2018) propose a model that integrates graph representation with local object information to form a local information graph, on which they implement random walk processes to calculate outlier scores.

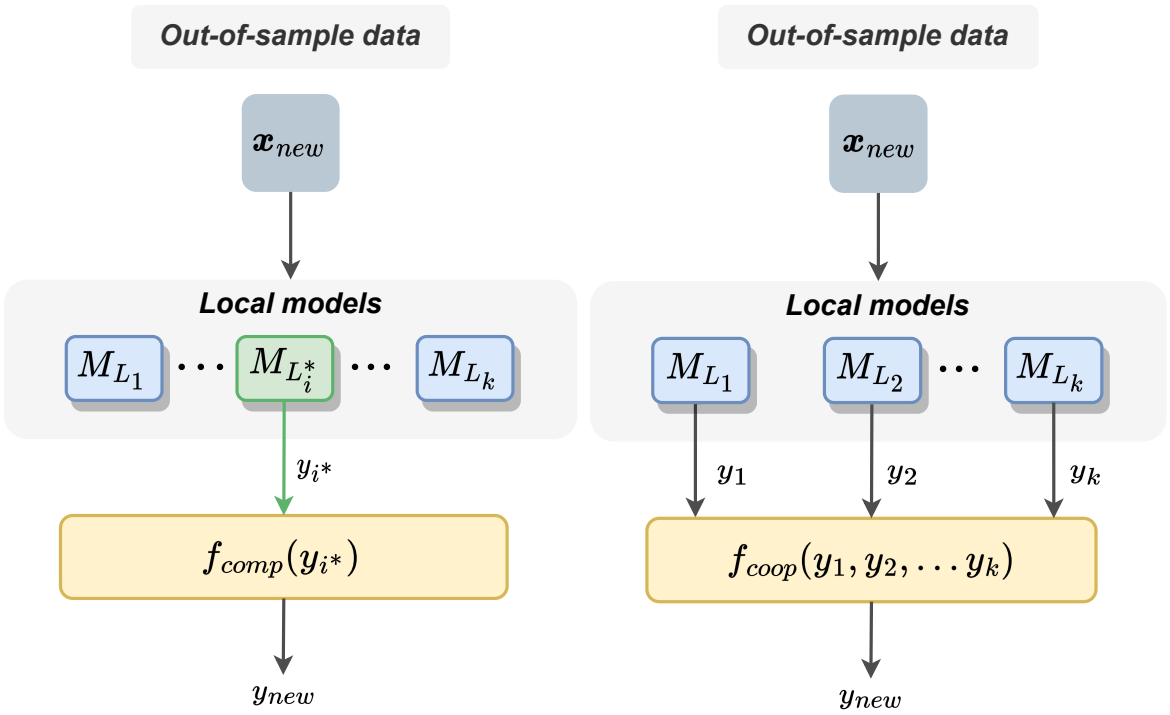
Methods that consider locality based on the origin of data represent a powerful paradigm in ML. An example of this approach is the federated learning (FL), a distributed approach that enables model training across many decentralized devices, each holding local data samples. In FL, each device, or a group of devices, is considered a unique source-based locality, where local models are trained, and they may be aggregated to refine a global model. This not only ensures the confidentiality of sensitive data by keeping it on the device but also leverages the collective knowledge from varied and scattered data sources (NGUYEN *et al.*, 2021). Recent advancements in this field include the hierarchical federated model embedding proposed by He *et al.* (2023), which establishes relationships between local data distributions. Additionally, Liu *et al.* (2023) present the local differential privacy federated learning, a method that utilizes an unbiased mean estimator based on maximum likelihood estimation for accurate global gradient computation to improve training precision and reducing variance. Figure 7d illustrates an example of this concept, showcasing several nodes (e.g., devices, systems) distributed across various localities representing different geographical areas or distinct computer networks.

In summary, locality criteria play a pivotal role in local learning as they impact on the learning process in many aspects. For instance, by segmenting complex problems into simpler, localized sub-problems can facilitate problem-solving as well as may reduce the computational cost (DRUMOND *et al.*, 2022). Furthermore, several locality criteria support the application of distributed learning techniques, allowing for simultaneous processing across different datasets or environments (NGUYEN *et al.*, 2021). Moreover, the localized approach may aid in a clearer understanding of the problem by providing contextual insights into specific sub-problems.

3.1.3 Competitive vs. Cooperative Approaches

Learning frameworks that incorporate multiple models, such as ensemble learning and local learning, rely on two main approaches for integrating model predictions: competitive and cooperative. Figure 8 illustrates both approaches.

Figure 8 – Illustration of competitive and cooperative output prediction.



(a) Competitive prediction (b) Cooperative prediction

Source: the author (2024).

Competitive learning (a.k.a. winner-take-all) involves a scenario where various local models compete against each other to get activated or respond to specific inputs (See Figure 8a). In essence, when predicting the output for a new, out-of-sample data pattern $\boldsymbol{x}_{new}$, the process exclusively involves a single local model. This can be interpreted as a linear combination of local model outputs and scalar weights where the dominant local model, represented by $M_{L_{i^*}}$, is assigned a weight of 1, whereas all other local models are assigned a weight of 0. Finding the winner typically relies on a similarity measure.

In contrast, cooperative learning embodies a collaborative strategy where multiple local models work together towards a shared objective. This method commonly involves a function that combines the outputs from multiple local models, as described in Figure 8b. Cooperative functions could involve several strategies, such as weighted averaging, voting, and stacking (ROKACH, 2019). This approach relies on strategies to combine the influence (i.e., weight) of the output for each local model in output prediction.

3.2 Cluster-based Local Modeling

Local modeling can be defined as a multimodel approach which creates a discriminant function of the form:

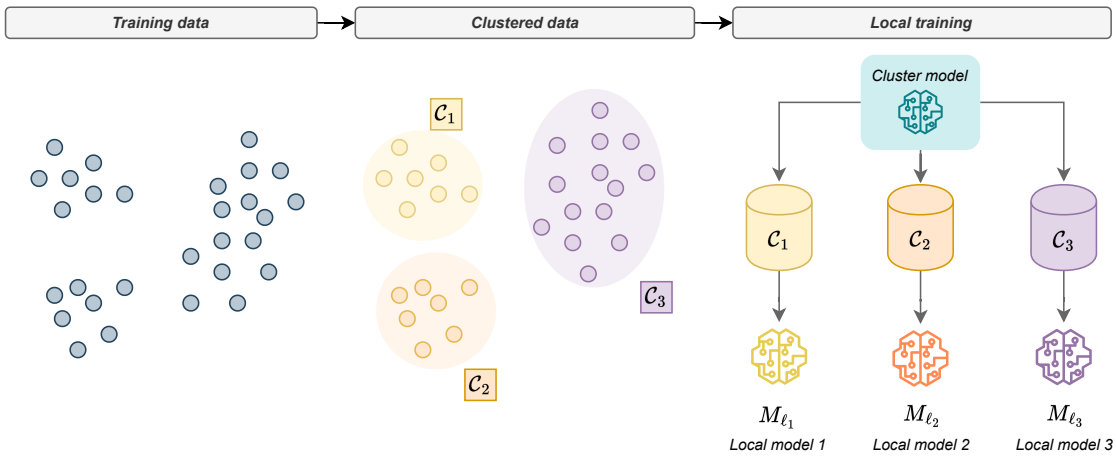
$$g(\mathbf{x}) = \sum_{i=1}^{K} h_i(\mathbf{x}) g_i(\mathbf{x}), \tag{3.1}$$

where K is the number of partitions; $h_i(\mathbf{x})$ is a function that gives a notion of locality for the i-th model; and $g_i(\mathbf{x})$ represents the i-th local model discriminant function (ALPAYDIN; JORDAN, 1996). The primary contribution of this thesis is the proposal of local learning methods based on the CLHP framework (DRUMOND *et al.*, 2022). This data space locality framework relies on clustering algorithms, such as K-means, to establish disjoint clusters within the training data where a local model is trained for each partition. K-means is a clustering algorithm that partitions n observations into K clusters by minimizing the within-cluster sum of squares. It aims to assign each observation to the cluster with the nearest centroid (BISHOP, 2006).

3.2.1 Training a Cluster-based Local Model

During the CLHP training phase, the whole dataset $\mathcal{X}_{tr}$ is divided into K partitions by a clustering algorithm (e.g., K-means). Then, for each cluster $C_i \subset \mathcal{X}_{tr}$ a local model is trained exclusively using the data of its associated partition. Considering that the subsets are disjoint, it can be said that CLHP does a hard partitioning of the dataset. Figure 9 illustrates the partition of data space into 3 local data partitions.

Figure 9 – Training a model using CLHP local learning framework.



Source: the author (2024).

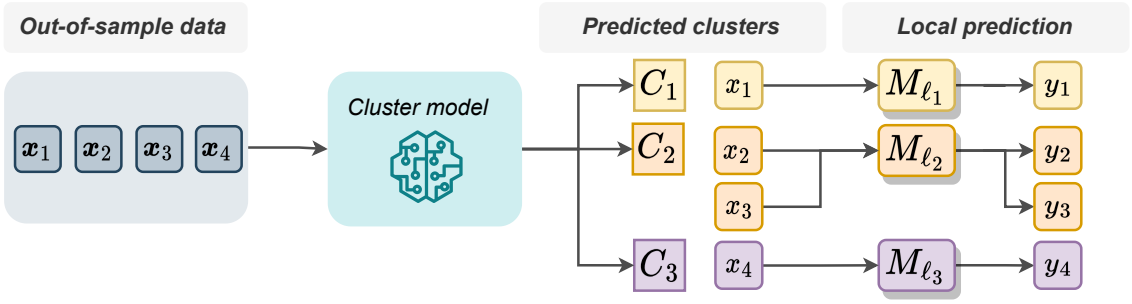
3.2.2 *Out-of-sample Class Prediction*

Predicting classes for out-of-sample data points through CLHP involve computing the dissimilarity between the novel instance and the local partitions defined in training phase. Considering a *competitive* prediction approach, for each new sample $\mathbf{x}$ the *winner* partition is found by searching for the closest K-means prototype using a dissimilarity measure (e.g., Euclidean distance). Considering i^* as the index corresponding to the nearest cluster to $\mathbf{x}$, indicating that $\mathbf{x}$ belongs to partition C_{i^*}, the most suitable model for class prediction is represented by the function $g_{i^*}(\mathbf{x})$. In this case, the neighborhood function has the following form:

$$h_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \psi(\mathbf{x}, \mathbf{c}_i) = \min\limits_{1 \leq j \leq K} \psi(\mathbf{x}, \mathbf{c}_j), \\ 0, & \text{otherwise,} \end{cases} \tag{3.2}$$

where $\psi(\cdot)$ is a function that measures dissimilarity between the sample $\mathbf{x}$ and the prototype $\mathbf{c}_i$. Equation (3.2) assumes that the clustering algorithm uses prototypes to represent the cluster. When a clustering algorithm does not provide prototypes, an alternative method should be employed to determine the winning local model for prediction. Figure 10 depicts the process of predicting out-of-sample data instances using the CLHP framework. First, the data patterns are submitted to a competitive process to define the closest cluster. As depicted in the Figure 10, $\boldsymbol{x}_1$ is closer to cluster C_1, $\{\boldsymbol{x}_2, \boldsymbol{x}_3\}$ to C_2, and $\boldsymbol{x}_4$ to C_3. Then, the respective local models are selected for predicting each new data instance, as described in the local prediction step.

Figure 10 – Output prediction using CLHP local learning framework.

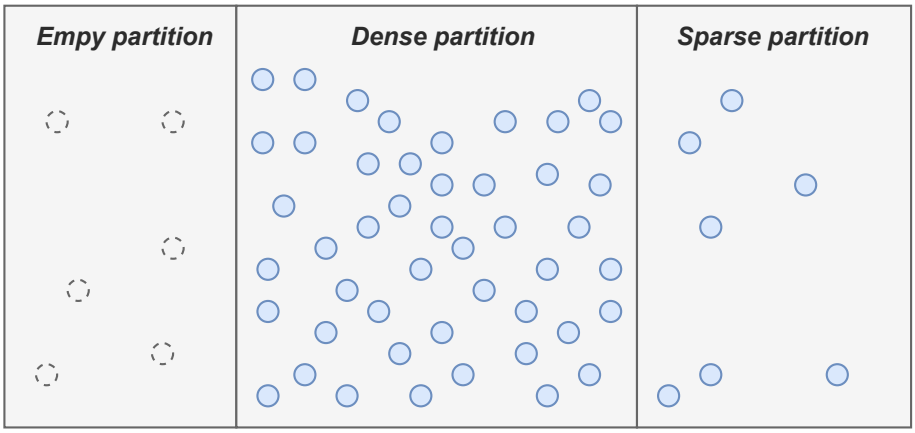


Source: the author (2024).

3.2.3 Local Partitions: Aspects and Challenges

Applying CLHP to pattern classification raises several issues related to data partitions. This section addresses the challenges of defining local partitions for both MTCC and OCC problems. As illustrated in Figure 11, three general partitions issues can arise: empty, dense, and sparse. It is important to mention that these partition issues are exclusively related to the quantity of data samples contained within the partition (i.e., cluster's density).

Figure 11 – Illustration of data partition issues in local modeling.

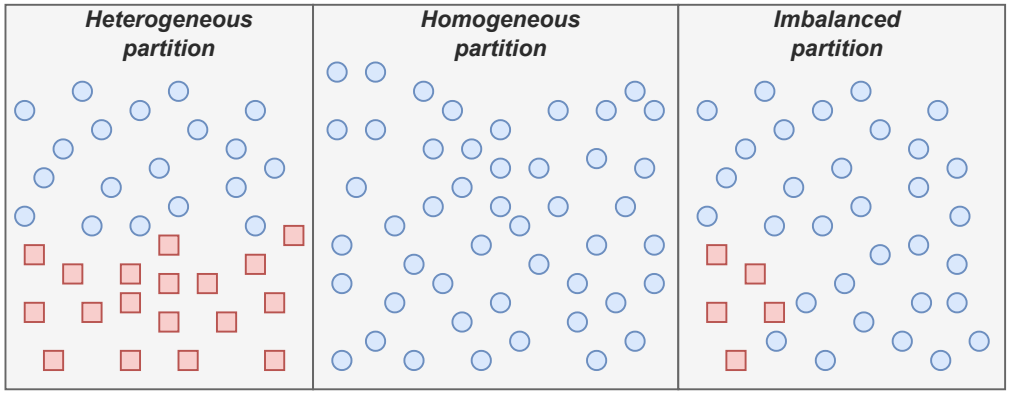


Source: the author (2024).

1. **Empty partitions** - Empty partitions arise from a cluster C_i that is empty. Therefore, there is no point building a classifier in such partition since there are no data samples to do so. This issue may be caused by poor clustering or due to selection of inadequate number of local partitions K.
2. **Dense partitions** - Dense partitions refer to clusters characterized by a substantial density of data samples (i.e., high volume of data within the cluster). In the context of local learning, the usual preference is for all partitions to exhibit high density, ensuring that each local partition is adequately dense for training a model.
3. **Sparse partitions** - Sparse partitions (or low density partitions) are clusters that present a low density in terms of data samples. In that case, the quantity of sample may be too few for fitting a local model. There are a few strategies that could be employed for solving this problem. A natural solution involve merging the sparse partition with the closest cluster. Alternatively, the data points from sparse partitions could be discarded.

The formation of local partitions is fundamentally an unsupervised learning process, meaning that these partitions are clustered based on the similarity of features rather than predefined labels. However, in the context of MTCC, especially those involving a large number of classes, it becomes crucial to examine three types of local partitions: homogeneous, heterogeneous, and imbalanced. Such categories are depicted in Figure 12.

Figure 12 – Illustration of data partition issues in MTCC local modeling.

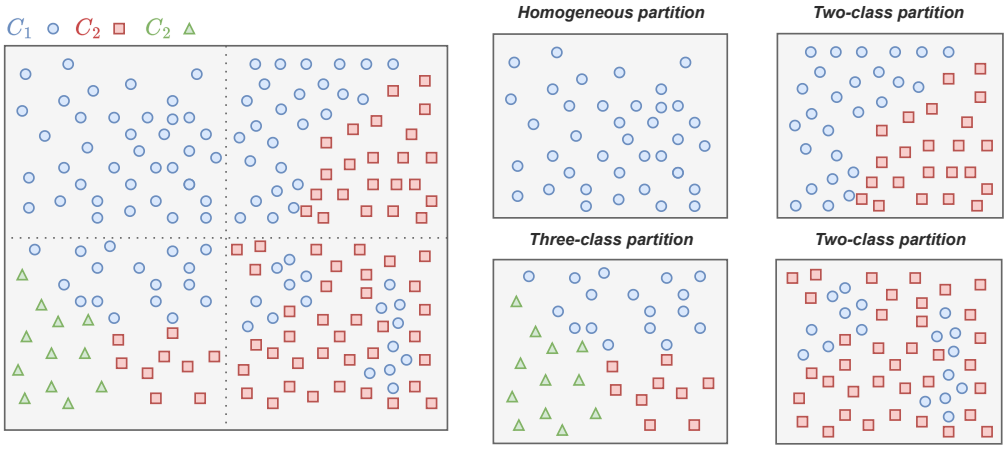


Source: the author (2024).

1. **Homogeneous partitions** - Homogeneous partitions arise when a cluster C_i is comprised solely of data points belonging to a single class. A pragmatic strategy to address this is to predict that any new data entering the partition space of C_i will be assigned the label L_i. It can be posited that the g_i model functions as a "bias model", given that they consistently produce a singular outcome (DRUMOND *et al.*, 2022).
2. **Heterogeneous partitions** - Heterogeneous partitions occur when a cluster C_i includes data points from various classes. Such partitions are typically intended for use in training models for supervised classification. These partitions can be further classified into two subtypes: balanced or imbalanced.
3. **Imbalanced partitions** - Imbalanced partitions refer to heterogeneous partitions where the distribution of data samples across classes is notably uneven.

The fundamental principle of local modeling involves segmenting the training dataset into numerous subsets, each to be addressed as an individual, more manageable subproblem (DRUMOND *et al.*, 2022). In MTCC, defining local partitions implies dividing the main classification problem into multiple sub-problems (one for each partition) with distinct complexity. For illustration, Figure 13 depicts a three-class classification problem. Here, the two-dimensional data space is partitioned into four quadrants, with each quadrant representing a local partition.

Figure 13 – Illustration of subproblems based on data partition in MTCC local modeling.



Source: the author (2024).

The top-right quadrant shows an example of a binary classification problem where a linear classifier would likely yield a satisfactory decision boundary. Conversely, in the lower-right quadrant, there is a binary classification problem with more intricate inter-class boundaries. The top-left quadrant shows a homogeneous partition, where all data points belong to a single class (represented by blue circles). Such partition can be modeled by using a bias model. Lastly, the lower-left quadrant presents a three-class classification problem that may require a more complex classifier.

3.3 Summary

In this chapter, we conducted a thorough examination on the domain of local learning, exploring recent advancements in local models for a wide range of ML problems. Furthermore, we introduced in details the CLHP framework, including the training and prediction steps. The subject discussed in this chapter provides essential background knowledge necessary for comprehending the local learning methodologies introduced in subsequent chapters.

4 LOCAL KERNEL PRINCIPAL COMPONENT ANALYSIS

Recent advancements in KPCA have yielded notable contributions in the field of OCC and ND (HOFFMANN, 2007; XIAO *et al.*, 2013; XIAO *et al.*, 2014; DAS *et al.*, 2018; HAMROUNI *et al.*, 2020). Remarkably, multiple studies have explored local learning methods and distributed models (DENG *et al.*, 2017; CUI *et al.*, 2019; HU *et al.*, 2020). This chapter introduces the LKPCA, a local learning method for OCC-based novelty detection problems. Initially, the fundamentals of LKPCA within a competitive setting are explored, with a focus on two thresholding strategies: single-thresholding and multi-thresholding. We detail the model's structure and training process, explaining how it handles thresholding and detects novelties in out-of-sample instances. The latter part of the chapter evaluates the performance of *competitive* LKPCA through simulations on various OCC benchmark datasets.

4.1 Kernel Principal Component Analysis

KPCA is a nonlinear extension of linear PCA (SCHÖLKOPF *et al.*, 1997). It performs PCA on data points after they have been mapped into a high-dimensional feature space $\mathcal{F}$. The core of KPCA is the *kernel trick*, which avoids the need to explicitly compute the high-dimensional mapping. Instead, it computes inner products in $\mathcal{F}$ using a kernel function defined in the original input space:

$$k(\boldsymbol{x},\boldsymbol{y}) = \langle \boldsymbol{\varphi}(\boldsymbol{x}), \boldsymbol{\varphi}(\boldsymbol{y}) \rangle, \tag{4.1}$$

where $\boldsymbol{\varphi} : \mathbb{R}^d \to \mathcal{F}$ is a mapping function that projects a data vector $\boldsymbol{x}$ into its feature-space counterpart $\boldsymbol{\varphi}(\boldsymbol{x})$. Kernel functions are essential to various kernel methods such as SVMs, OCSVMs and KPCAs. Table 3 describes commonly used kernel functions.

Table 3 – Frequently used kernel functions in machine learning models.

Kernel	**Equation**	**Hyperparameters**
Linear	$k(\boldsymbol{x},\boldsymbol{x}_i) = \boldsymbol{x}^T\boldsymbol{x}_i + c$	c is a constant.
Polynomial	$k(\boldsymbol{x},\boldsymbol{x}_i) = \left(\alpha \boldsymbol{x}^T\boldsymbol{x}_i + c\right)^p$	α and c are constants; p is the polynomial degree.
Gaussian	$k(\boldsymbol{x},\boldsymbol{x}_i) = \exp\left\{-\gamma\|\boldsymbol{x}-\boldsymbol{x}_i\|^2\right\}$	$\gamma = \frac{1}{\sigma^2}$, and $\sigma > 0$ is a constant.
Cauchy	$k(\boldsymbol{x},\boldsymbol{x}_i) = \dfrac{1}{1+\|\boldsymbol{x}-\boldsymbol{x}_i\|^2/\sigma^2}$	σ is a constant.
Log	$k(\boldsymbol{x},\boldsymbol{x}_i) = -\ln(\|\boldsymbol{x}-\boldsymbol{x}_i\|^c + 1)$	c is a constant.

Source: the author (2024).

To perform KPCA on a training dataset $\mathcal{X}_{tr} = \{\boldsymbol{x}_i\}_{i=1}^n$, we have to express the standard PCA in the feature space $\mathcal{F}$. Assuming the mapped training instances $\boldsymbol{\varphi}(\boldsymbol{x}_i) \in \mathcal{F}$ are centered (i.e., $\sum_{i=1}^n \boldsymbol{\varphi}(\boldsymbol{x}_i) = \mathbf{0}$), the covariance operator in $\mathcal{F}$ is defined as

$$\mathbf{C}_{\mathcal{F}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\varphi}(\boldsymbol{x}_i)\boldsymbol{\varphi}(\boldsymbol{x}_i)^T. \tag{4.2}$$

KPCA is then obtained by solving the eigenvalue problem in $\mathcal{F}$:

$$\mathbf{C}_{\mathcal{F}}\boldsymbol{v}_\ell = \lambda_\ell \boldsymbol{v}_\ell. \tag{4.3}$$

In KPCA, an eigenvector $\boldsymbol{v}_\ell \in \mathcal{F}$ corresponding to a nonzero eigenvalue ($\lambda_\ell \geq 0$) lies in the span of the mapped training points, so it can be written as

$$\boldsymbol{v}_\ell = \sum_{i=1}^{n} \alpha_{i\ell}\boldsymbol{\varphi}(\boldsymbol{x}_i), \tag{4.4}$$

for n coefficients $\alpha_{1\ell}, \ldots, \alpha_{n\ell}$. By substituting the expressions for $\boldsymbol{v}_\ell$ and $\mathbf{C}_{\mathcal{F}}$ into the eigenvalue problem and applying the kernel trick, the problem can be simplified to a linear algebra problem in the original space:

$$\boldsymbol{K}\boldsymbol{\alpha}_\ell = n\lambda_\ell \boldsymbol{\alpha}_\ell. \tag{4.5}$$

Here, the vectors $\boldsymbol{\alpha}_\ell = [\alpha_{1\ell}, \ldots, \alpha_{n\ell}]^T$ with $\ell = 1, \ldots, n$, are the eigenvectors of $\boldsymbol{K}$. The matrix $\boldsymbol{K}$ is the Gram matrix (also known as the kernel matrix), whose elements are the kernel functions applied to each pair of data points: $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The Gram matrix is described as

$$\boldsymbol{K} = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & \cdots & K_{nn} \end{bmatrix}. \tag{4.6}$$

As the premise of zero-mean data is generally not satisfied in real-world data, we centralize the points in feature space as follows:

$$\tilde{\boldsymbol{\varphi}}(\boldsymbol{x}_i) \;=\; \boldsymbol{\varphi}(\boldsymbol{x}_i) - \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{\varphi}(\boldsymbol{x}_j), \tag{4.7}$$

where $\tilde{\boldsymbol{\varphi}}(\boldsymbol{x}_i)$ denotes the i-th centered data point in $\mathcal{F}$. To obtain zero-mean data directly in terms of kernel functions, each entry of the centered Gram matrix $\tilde{\boldsymbol{K}}$ can be computed from the uncentered Gram matrix $\boldsymbol{K}$ as

$$\tilde{K}_{ij} \;=\; K_{ij} - \frac{1}{n}\sum_{r=1}^{n} K_{ir} - \frac{1}{n}\sum_{r=1}^{n} K_{rj} + \frac{1}{n^2}\sum_{r,s=1}^{n} K_{rs}, \tag{4.8}$$

where $\tilde{K}_{ij} = \tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denotes the centered kernel. In practice, the KPCA is solved on $\tilde{\boldsymbol{K}}$, ensuring that the mapped data in $\mathcal{F}$ have zero mean. Finally, for an out-of-sample data point $\boldsymbol{x}'$, the projection onto the ℓ-th eigenvector $\boldsymbol{v}_\ell$ in feature space is given by

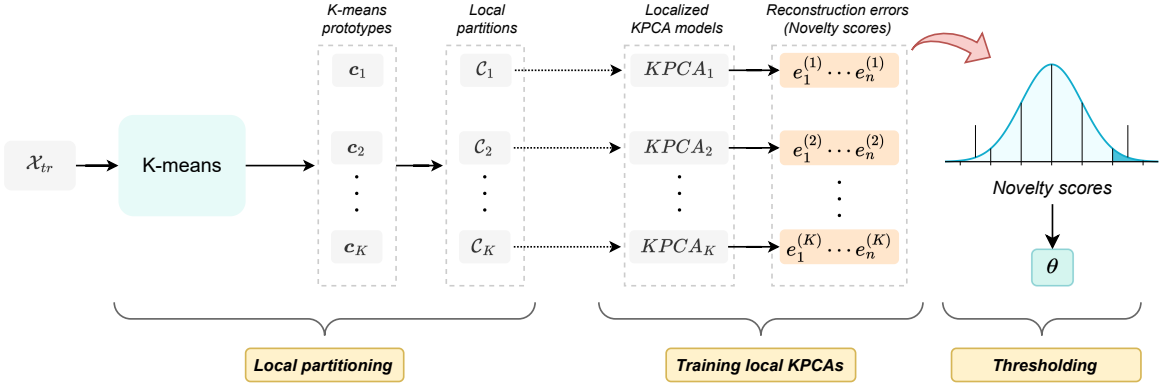
$$\langle \boldsymbol{v}_\ell, \tilde{\boldsymbol{\varphi}}(\boldsymbol{x}') \rangle = \sum_{i=1}^{n} \alpha_{i\ell}\, \tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}'), \tag{4.9}$$

where $\tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}')$ is the centered kernel value between training point $\boldsymbol{x}_i$ and the new point $\boldsymbol{x}'$.

4.2 Local KPCA for Novelty Detection

LKPCA is a local learning extension of the KPCA-based ND method proposed by Hoffmann (2007). It relies on three main stages. It begins by segmenting the data into distinct local partitions using the CLHP framework. Next, a localized KPCA model is trained on the data within each cluster. Finally, these models are combined using either competitive or cooperative strategies. The reconstruction errors from each localized KPCA are used to define a detection threshold, θ, which sets the boundary between normal and novel data points. A reconstruction error larger than this threshold indicates that a data point is likely a novelty. Figure 14 illustrates the LKPCA training process. The following sections provide the details of the steps involved in designing, training, and using LKPCA for ND.

Figure 14 – Illustration of the proposed LKPCA training steps.



Source: the author (2024).

4.2.1 Local Partitioning

The first step of the LKPCA involves clustering the training set $\mathcal{X}_{tr}$ into K non-overlapping local partitions using K-means. This process defines a set of K prototype vectors, $\{\boldsymbol{c}_j\}_{j=1}^{K}$, with each vector $\boldsymbol{c}_j$ forming a cluster C_j as follows:

$$C_j = \left\{ \boldsymbol{x} \in \mathcal{X}_{tr} \;\middle|\; \|\boldsymbol{x} - \boldsymbol{c}_j\|_2^2 = \min_{1 \le l \le K} \|\boldsymbol{x} - \boldsymbol{c}_l\|_2^2 \right\}. \tag{4.10}$$

The effectiveness of this stage is critical to the performance of LKPCA, as it directly impacts the novelty measure calculation. While K-means was used in this implementation, alternative clustering methods could also be employed.

4.2.2 Training a Local KPCA

A localized model is trained for each local partition using KPCA for ND (HOFFMANN, 2007). This process results in a set of K local one-class classifiers, where each local model yields a respective local novelty score. Given a data point $\boldsymbol{z} \in \mathcal{X}$, the novelty score is computed based on two measures: the local spherical potential and the projection of the input vector $\boldsymbol{z}$ onto the eigenvectors obtained from localized KPCA models.

4.2.2.1 Local Spherical Potential

The j-th *local spherical potential* of a data point $\boldsymbol{z}$, represented by $p_S^{(j)}(\boldsymbol{z})$, is the squared distance between the mapped data point $\boldsymbol{\varphi}(\boldsymbol{z})$ and its local cluster center in feature space, $\boldsymbol{\varphi}_0^{(j)} = \frac{1}{n_j}\sum_{r=1}^{n_j} \boldsymbol{\varphi}(\boldsymbol{x}_r)$:

$$\begin{aligned} p_S^{(j)}(\boldsymbol{z}) &= \|\boldsymbol{\varphi}(\boldsymbol{z}) - \boldsymbol{\varphi}_0^{(j)}\|_2^2 \\ &= k(\boldsymbol{z},\boldsymbol{z}) - \frac{2}{n_j}\sum_{i=1}^{n_j} k(\boldsymbol{z},\boldsymbol{x}_i) + \frac{1}{n_j^2}\sum_{r,s=1}^{n_j} k(\boldsymbol{x}_r,\boldsymbol{x}_s). \end{aligned} \tag{4.11}$$

The last term in Equation (4.11) is constant with respect to $\boldsymbol{z}$, and for Gaussian kernels, the first term is also constant ($k(\boldsymbol{z},\boldsymbol{z}) = 1$). Since constants do not affect the novelty score's predictive power, they can be omitted. Hence, a simplified form can be expressed as follows:

$$\tilde{p}_S^{(j)}(\boldsymbol{z}) = -\frac{2}{n_j}\sum_{i=1}^{n_j} k(\boldsymbol{z},\boldsymbol{x}_i). \tag{4.12}$$

4.2.2.2 *Projection onto Local Eigenvectors*

The second measure involves projecting the input vector $\boldsymbol{z}$ onto the localized eigenvectors obtained from each KPCA model, with one set of eigenvectors computed per local partition. For instance, in the j-th localized model, the eigenvectors $\{\boldsymbol{v}_\ell^{(j)}\}_{\ell=1}^{q_j}$ are ordered in decreasing order with respect to the eigenvalues, with $\boldsymbol{v}_1^{(j)}$ corresponding to the largest eigenvalue. In the j-th localized model, the projection onto the ℓ-th eigenvector is computed as follows:

$$\begin{aligned} f_\ell^{(j)}(\boldsymbol{z}) &= \langle \boldsymbol{\varphi}(\boldsymbol{z}), \boldsymbol{v}_\ell^{(j)} \rangle \\ &= \sum_{i=1}^{n_j} \alpha_{i\ell}^{(j)} \left[k(\boldsymbol{z}, \boldsymbol{x}_i) - \frac{1}{n_j} \sum_{r=1}^{n_j} k(\boldsymbol{x}_i, \boldsymbol{x}_r) - \frac{1}{n_j} \sum_{r=1}^{n_j} k(\boldsymbol{z}, \boldsymbol{x}_r) + \frac{1}{n_j^2} \sum_{r,s=1}^{n_j} k(\boldsymbol{x}_r, \boldsymbol{x}_s) \right]. \end{aligned} \tag{4.13}$$

4.2.2.3 *Local Novelty Score*

The j-th local novelty score is a combination of Equations (4.11) and (4.13):

$$p^{(j)}(\boldsymbol{z}) = p_S^{(j)}(\boldsymbol{z}) - \sum_{\ell=1}^{q_j} \left[f_\ell^{(j)}(\boldsymbol{z}) \right]^2, \tag{4.14}$$

where $p_S^{(j)}(\boldsymbol{z})$ is the j-th *local spherical potential*; and $f_\ell^{(j)}(\boldsymbol{z})$ is the projection of $\boldsymbol{z}$ onto the ℓ-th eigenvector of j-th local partition. The number of principal components, q_j, is a hyperparameter that can vary across localized models.

4.2.3 Thresholding Strategies

Thresholding in LKPCA can be performed using single or multiple thresholding strategies. Single-thresholding computes a single global threshold θ, while multi-thresholding defines a separate threshold $\theta^{(j)}$ for each local partition, forming the vector $\boldsymbol{\theta} = \begin{bmatrix} \theta^{(1)} & \theta^{(2)} & \cdots & \theta^{(K)} \end{bmatrix}$. Both strategies depend on computing the reconstruction errors for the entire training set using all local KPCA models. These errors can be arranged in the following $n \times K$ matrix:

$$\mathcal{E} = \begin{bmatrix} e_1^{(1)} & e_1^{(2)} & \cdots & e_1^{(K)} \\ e_2^{(1)} & e_2^{(2)} & \cdots & e_2^{(K)} \\ \vdots & \vdots & \ddots & \vdots \\ e_n^{(1)} & e_n^{(2)} & \cdots & e_n^{(K)} \end{bmatrix}, \tag{4.15}$$

where each column, $\boldsymbol{e}^{(j)} = \begin{bmatrix} e_1^{(j)} & e_2^{(j)} & \cdots & e_n^{(j)} \end{bmatrix}^T$, represents the reconstruction errors on the training set using the j-th local KPCA model, as defined by Equation (4.14).

Algorithm 1 outlines the three-step training process of LKPCA for ND, which is applicable to both the *competitive* and *cooperative* settings.

Algorithm 1: LKPCA fitting algorithm

Input: Training set $\mathcal{X}_{tr}$ containing only normal data.
Output: Trained LKPCA model with thresholds.

1. Local partitioning
Cluster the training set $\mathcal{X}_{tr}$ into K partitions using K-means algorithm.

$$\mathbf{C} = \{C_1, \ldots, C_K\}$$

2. Localized training
for *each cluster* C_j **do**
2.1. Train a localized j-th KPCA model on the subset C_j.
2.2. Compute the novelty score for all data points in C_j using Equation (4.14).

$$p^{(j)}(\boldsymbol{x}) = p_S^{(j)}(\boldsymbol{x}) - \sum_{\ell=1}^{q_j} \left[f_\ell^{(j)}(\boldsymbol{x}) \right]^2, \quad \forall \boldsymbol{x} \in C_j$$

end
3. Thresholding training
3.1 Compute a single global detection threshold θ; or
3.2 Compute multiple local thresholds $\boldsymbol{\theta} = \begin{bmatrix} \theta^{(1)} & \cdots & \theta^{(K)} \end{bmatrix}$.

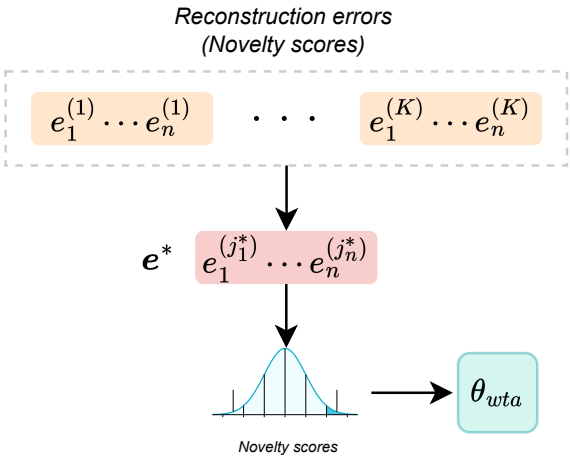
4.3 Competitive Local Kernel Principal Component Analysis

Competitive LKPCA employs the winner-takes-all (WTA) approach for both thresholding and prediction. This thesis defines two competitive LKPCA variants: winner-take-all local kernel principal component analysis (LKPCA-WTA), which uses a single global threshold; and a multi-threshold winner-take-all local kernel principal component analysis (LKPCA-WTA-MT), which relies on individual localized novelty thresholds.

4.3.1 Single Thresholding

Single-thresholding involves computing a global threshold, θ_{wta}, from the reconstruction errors of all training data points. For each data point, the nearest partition is identified, and its corresponding local model computes the reconstruction error. The threshold θ_{wta} is then derived from the distribution of these local reconstruction errors. Figure 15 illustrates the thresholding process employed by LKPCA-WTA.

Figure 15 – Single-thresholding strategy for the LKPCA-WTA.



Source: the author (2024).

Local reconstruction errors are arranged in the vector $\boldsymbol{e}^* = \begin{bmatrix} e_1^{(j_1^*)} & \cdots & e_n^{(j_n^*)} \end{bmatrix}^T$, where $e_i^{(j_i^*)}$ is the reconstruction error computed by the winning local KPCA model for the i-th training data point. Algorithm 2 details how θ_{wta} is computed. In this implementation, the threshold is selected as the maximum error, although other methods for computing the threshold can also be applied.

Algorithm 2: LKPCA-WTA single-thresholding algorithm

Input: Training set $\mathcal{X}_{tr}$ and a trained LKPCA model.
Output: Detection threshold θ_{wta}.

for *each data point* $\boldsymbol{x}_i$ **do**

1. Find the winner (i.e., closest) local $KPCA_{j^*}$ model:

$$j^* = arg\ min \|\boldsymbol{x}_i - \boldsymbol{c}_j\|_2^2,\ j = 1 \ldots K.$$

2. Compute the reconstruction error through $KPCA_{j^*}$ model using Equation (4.14):

$$e_i^{(j^*)} = p^{(j^*)}(\boldsymbol{x}_i).$$

end

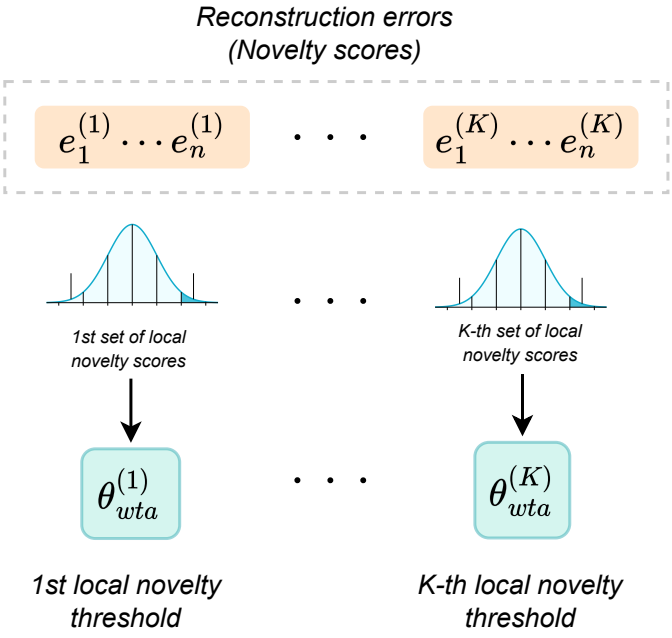
3. Compute the LKPCA-WTA threshold (θ_{wta}) by finding the maximum value in $\boldsymbol{e}^*$:

$$\theta_{wta} = max(\boldsymbol{e}^*).$$

4.3.2 Multiple Thresholding

In the LKPCA-WTA-MT approach, a separate novelty threshold is computed for each localized KPCA model, resulting in a vector of thresholds, $\boldsymbol{\theta}_{wta}$. As illustrated in Figure 16, each local KPCA yields a distribution of novelty scores used to define an associated threshold.

Figure 16 – Multi-thresholding strategy for the LKPCA-WTA-MT.



Source: the author (2024).

As detailed in Algorithm 3, each local threshold is determined by the maximum value among the novelty scores.

Algorithm 3: LKPCA-WTA multiple thresholds algorithm

Input: Training set $\mathcal{X}_{tr}$ and a trained LKPCA model.

Output: Vector of thresholds $\boldsymbol{\theta}_{wta} = \begin{bmatrix} \theta_{wta}^{(1)} & \cdots & \theta_{wta}^{(K)} \end{bmatrix}$.

for *each local model* $LKPCA_j$ **do**

Computes a threshold $\theta_{wta}^{(j)}$ for each *local reconstruction error* distribution.

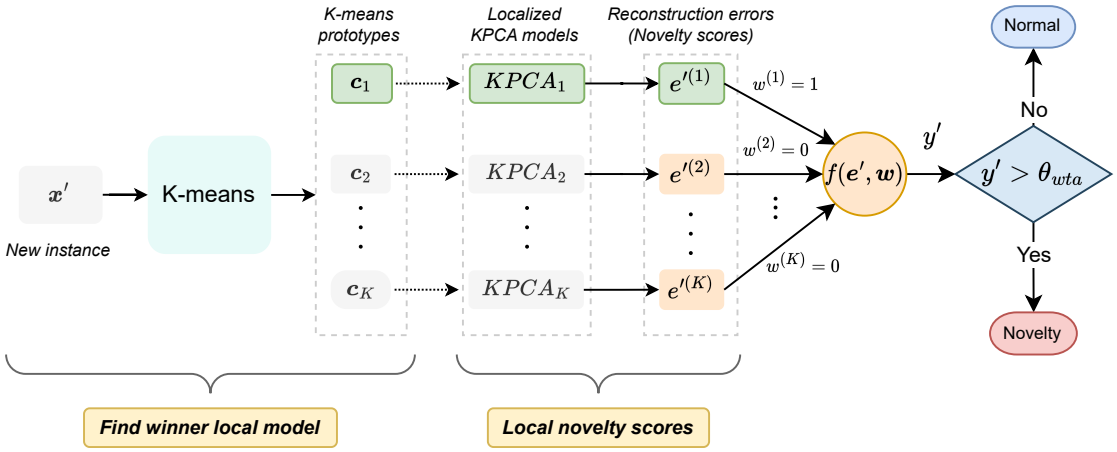
$$\theta_{wta}^{(j)} = \max_{\forall \boldsymbol{x} \in \mathcal{X}_{tr}} p^{(j)}(\boldsymbol{x})$$

end

4.3.3 *Competitive Local KPCA Prediction*

As illustrated in Figure 17, the LKPCA-WTA's out-of-sample prediction for a new instance, $\boldsymbol{x}'$, relies on a competitive approach, in which the model identifies the closest cluster and uses only the corresponding localized KPCA to compute the reconstruction error $e'^{(j^*)}$. All other localized models are then disregarded. This is equivalent to assigning a weight of 1 to the winning local model ($w^{(j^*)} = 1$), and 0 to all others ($w^{(j)} = 0$ for $j \neq j^*$). In the example, $j^* = 1$.

Figure 17 – Illustration of the LKPCA-WTA out-of-sample prediction.

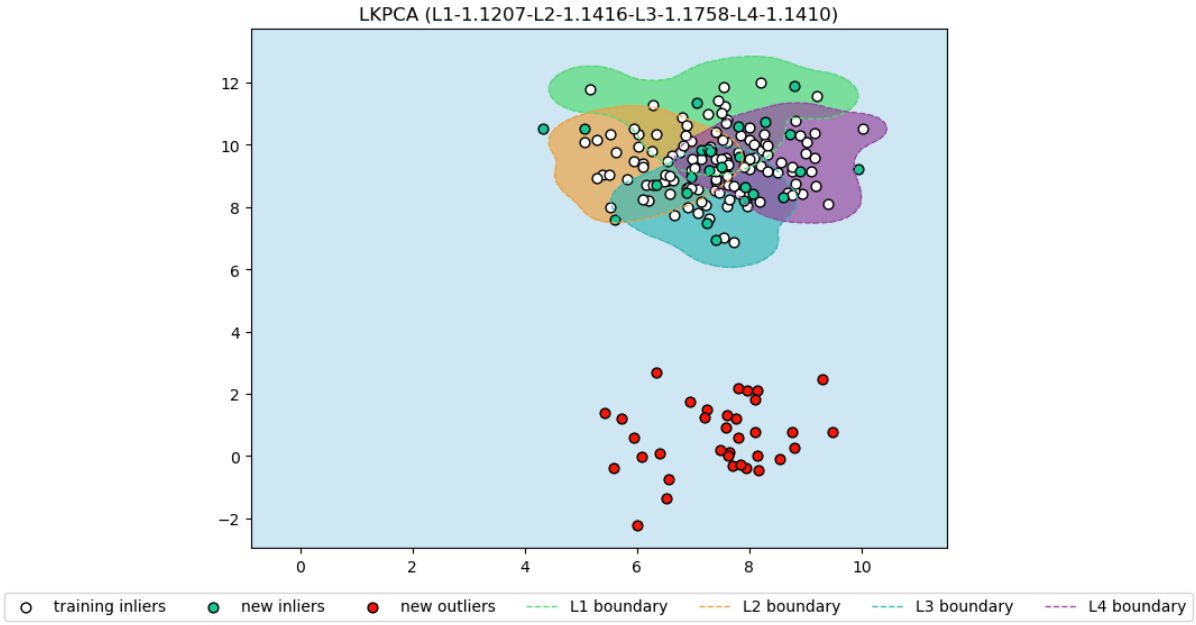


Source: the author (2024).

4.3.4 *A Word About Benefits and Challenges*

The LKPCA presents specific considerations that should be addressed. As highlighted in Section 3.2, the method's performance is vulnerable to poorly defined local partitions, which can result in sparsely clustered data. This occurs when a local model lacks a sufficient number of data points to effectively learn patterns within its designated partition. This issue is directly related to how the decision boundary is formed. Figure 18 illustrates how the combination of individual localized KPCA decision surfaces forms a unified LKPCA decision region. Each color represents the decision boundary of a different local model within the framework. This visualization demonstrates how LKPCA can segment the data space, allowing each partition to be handled by a specialized local model. This strategy aims to capture the complex structure of the data more effectively than a single global model. Nevertheless, the competitive step of selecting the "winning" local model for prediction introduces variability, particularly when a new instance is equidistant from multiple partitions.

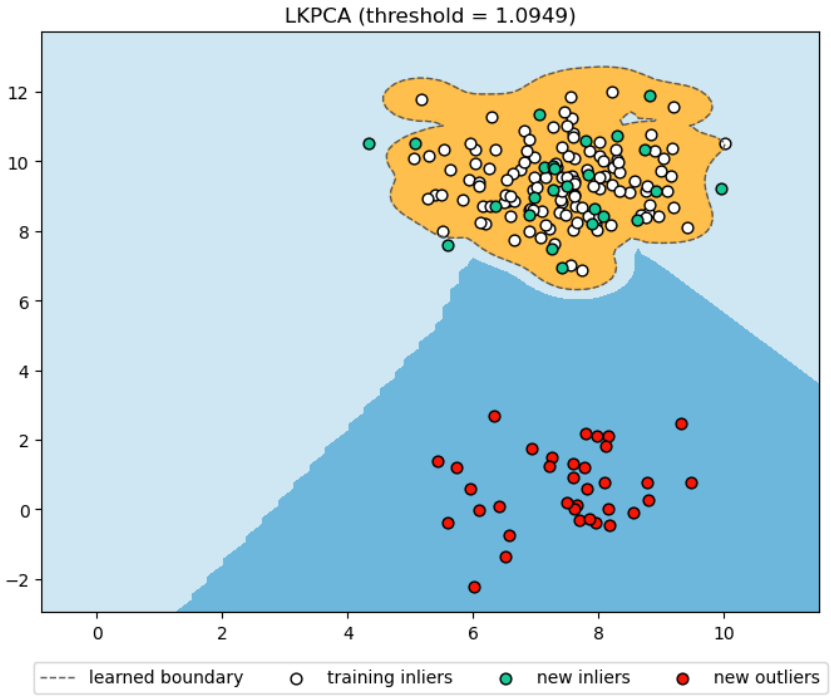
Figure 18 – Illustrative combination of decision surfaces for each localized KPCA.



Source: the author (2024).

Figure 19 presents the pooled LKPCA decision surface, which resembles the union of the localized KPCA surfaces from Figure 18.

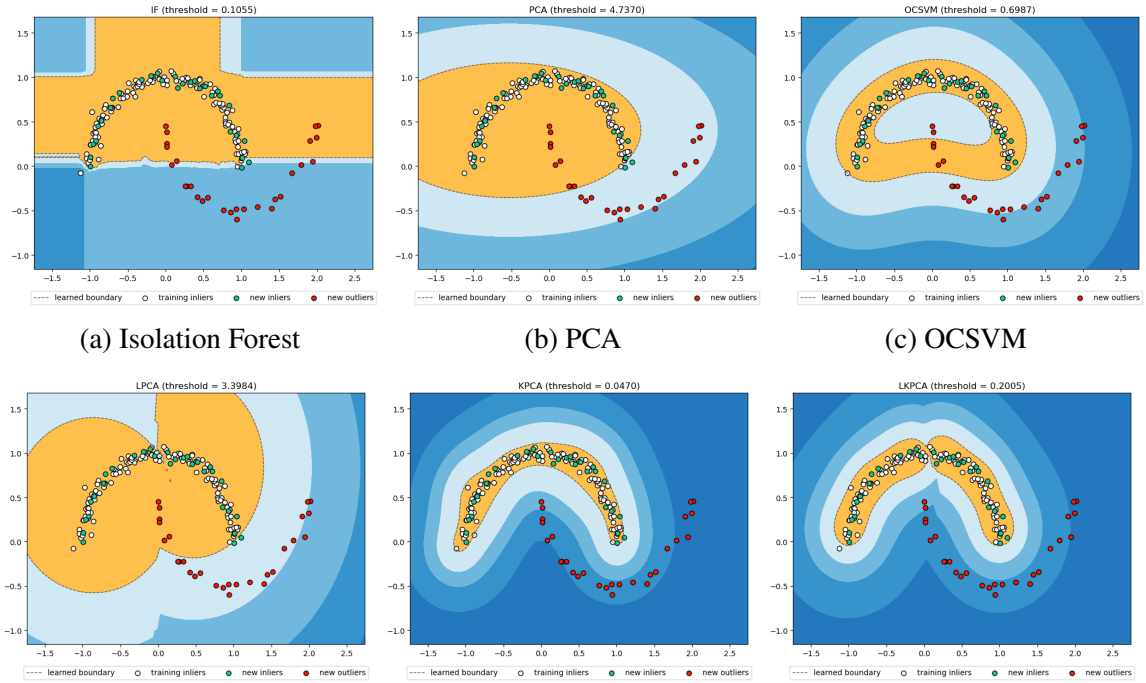
Figure 19 – Illustrative decision surface of the pooled LKPCA.



Source: the author (2024).

Figure 20 illustrates the decision surfaces of six OCC methods on the two moons dataset. The decision surface of each method reveals its unique approach to ND. The isolation forest (IF) creates a broad surface that may misclassify several novelties as normal. PCA forms a tight boundary but fails to capture the dataset's nonlinearity. The OCSVM produces a smoother, more curvilinear surface that fits the dataset's shape well, although it included a few novelties. The global KPCA exhibits a complex surface, effectively modeling the data's nonlinear patterns.

Figure 20 – Decision surfaces from six learning methods on the two moons dataset.



(a) Isolation Forest (b) PCA (c) OCSVM

(d) Local PCA ($K = 2$) (e) KPCA (f) Local KPCA ($K = 2$)

Source: the author (2024).

As for the local learning models, for both local principal component analysis (LPCA) and LKPCA, the two localized decision surfaces converge into a single boundary, demonstrating that for CLHP-based methods, the final decision surface is contingent on the number of local partitions. LPCA with $K = 2$ fits the normal data better than global PCA, even though its performance could likely improve with an increase of the number of partitions to fit more closely to the data. LKPCA combines two local decision surfaces into an unified boundary. This approach balances modeling the data's structure with achieving class separability, closely adhering to the dataset's inherent patterns.

4.3.5 Complexity Analysis

Evaluating the computational efficiency of LKPCA is essential for understanding its advantages over global KPCA. A key benefit of LKPCA is its ability to reduce computational cost by training multiple KPCA models on smaller data subsets, which also facilitates parallelization and distributed processing. Global KPCA's time complexity is dominated by two main steps: computing the kernel matrix and performing eigendecomposition. These have a time complexity of $O(n^2)$ and $O(n^3)$, respectively, where n is the dataset size. Consequently, global KPCA's overall complexity is $O(n^3)$. In contrast, LKPCA's time complexity involves two stages. First, the dataset is clustered into K local partitions using K-means, which has a time complexity of $O(InKd)$, where I is the number of iterations and d is the input dimensionality. In the second stage, KPCA is applied independently to each cluster, leading to a complexity of $\sum_{j=1}^{K} O(n_j^3)$, where n_j is the number of samples in cluster C_j. Since n is typically much larger than any individual n_j, partitioning the dataset into smaller clusters can significantly reduce the time complexity. This approach allows LKPCA to substantially improve computational and memory efficiency for large datasets. While the K-means clustering step also contributes to complexity (linear in n), LKPCA remains computationally advantageous over standard KPCA, especially for large datasets.

Table 4 summarizes the computational complexity of global KPCA, K-means and LKPCA. The prediction time complexities provided assume evaluation of a single data instance. Note that for *competitive* LKPCA the prediction only considers the closest localized model. In contrast, *cooperative* LKPCA uses all localized models for prediction.

Table 4 – Complexity analysis of KPCA, K-means and LKPCA.

Method	**Time Complexity**		**Memory Complexity**
	(Training)	**(Prediction)**	**(Trained Model)**
KPCA	$O(n^3)$	$O(n)$	$O(n^2)$
K-means	$O(InKd)$	$O(Kd)$	$O(Kd)$
LKPCA	$O(InKd) + \sum_{j=1}^{K} O(n_j^3)$	$O(Kd) + \underbrace{O(n_{j^*})}_{\text{Competitive}}$ $\underbrace{\sum_{j=1}^{K} O(n_j)}_{\text{Cooperative}}$	$O(Kd) + \sum_{j=1}^{K} O(n_j^2)$

Source: the author (2024).

4.4 Simulations and Results

This section presents the results of computational experiments on *competitive* LKPCA methods and discusses their implications. All simulations and algorithms, implemented in Python, are available in a GitHub repository[1]. For the simulation settings, we performed 50 independent train/test experiments by repeating the **10-fold** cross-validation technique 5 times. In OCC-based k-fold, only normal class data were considered for training. Prior to training, features were scaled to the range $[0,1]$ using min-max scaling. The CLHP method was optimized by tuning the number of local partitions K, over the range $K \in \{2,3,...,\lfloor\sqrt{n}\rfloor\}$, where n is the total number of samples. The optimized number of partitions, K_{opt}, was selected as the one yielding the highest Silhouette index (ROUSSEEUW, 1987). The Silhouette index quantifies how well each data point fits within its assigned cluster. For a data instance $\boldsymbol{x}_i$, the Silhouette index can be computed as:

$$S(\boldsymbol{x}_i) = \frac{b(\boldsymbol{x}_i) - a(\boldsymbol{x}_i)}{\max(a(\boldsymbol{x}_i), b(\boldsymbol{x}_i))}, \tag{4.16}$$

where $a(\boldsymbol{x}_i)$ is the average distance between $\boldsymbol{x}_i$ and all other points in the same cluster (i.e., average intra-cluster distance); and $b(\boldsymbol{x}_i)$ is the lowest average distance between $\boldsymbol{x}_i$ and points in the nearest neighboring cluster (i.e., lowest average inter-cluster distance). The score for the clustering solution is then the average of the Silhouette scores for all individual data points:

$$S = \frac{1}{n}\sum_{i=1}^{n} S(\boldsymbol{x}_i). \tag{4.17}$$

Hyperparameter optimization for each learning method was conducted using *random search* with **5-fold** cross-validation over 40 trials. For OCSVM and KPCA-based methods, a Gaussian kernel was employed, with γ selected from $[0.5,5]$. For IF, LOF, and KPCA, the proportion of *novelties* was chosen from $[0.05,0.1]$. For IF, the number of estimators is selected from $\{20,50\}$, and the sample size proportion is selected from the interval $[0.5,0.9]$. For OCSVM, the parameter ν, which controls the trade-off between training error and model complexity, is optimized within the range $[0.03,0.2]$. For LOF, the number of estimators varies between $\{5,50\}$. Regarding PCA, the number of retained principal components (q) is selected from $\{1,\ldots,d\}$, where d is the input space dimensionality. Similarly, for KPCA and LKPCA, q is chosen from $\{1,\ldots,30\}$.

[1] The LKPCA code can be accessed in the following link: <https://github.com/renanfonteles/local-kpca> .

The UCI Machine Learning Repository (DUA; GRAFF, 2017) was the source for the first set of datasets used in this experiment. Since these are multi-class problems, we created one-class versions using a one-versus-all approach. This method designates one class as the normal class and groups all remaining classes as novelty instances. Hence, a multi-class dataset with n_C classes produces n_C distinct one-class variants. The characteristics of these multi-class datasets are summarized in Table 5.

Table 5 – Benchmark datasets selected for LKPCA simulations.

Dataset	**No. of instances**	**No. of features**	**No. of classes**
Iris	150	4	3
Parkinson	195	22	2
Vertebral Column	310	6	3
Wall-following (2f)	5456	2	4

Source: the author (2024).

Additionally, several OCC benchmark datasets were selected from ODDs library (RAYANA, 2016), including *Annthyroid*, *Breastw*, *Cardio*, *Glass*, and *Ionosphere* datasets. Table 6 presents all one-class datasets used in computational simulations.

Table 6 – One-class datasets selected for LKPCA simulations.

Dataset	**No. instances**	**No. features**	**No. inliers**	**No. novelties**
Annthyroid	7200	6	6666	534
Breastw	683	9	444	239
Cardio	1831	21	1655	176
Glass	214	9	205	9
Ionosphere	351	33	225	126
Iris (0)	150	4	100	50
Iris (1)	150	4	100	50
Iris (2)	150	4	100	50
Parkinson (0)	195	22	147	48
Parkinson (1)	195	22	48	147
VC (0)	310	6	250	60
VC (1)	310	6	210	100
VC (2)	310	6	160	150
WF2F (0)	5456	2	3251	2205
WF2F (1)	5456	2	3359	2097
WF2F (2)	5456	2	5128	328
WF2F (3)	5456	2	4630	826

Source: the author (2024).

4.4.1 Competitive LKPCA Classification Performance

The first experiment evaluates the classification performance of the proposed competitive LKPCA variants — LKPCA-WTA and LKPCA-WTA-MT — against several established one-class classifiers. The primary benchmark for this analysis is the global KPCA for novelty detection (HOFFMANN, 2007). For a broader comparative analysis, we also include a set of state-of-the-art methods, including PCA, IF, OCSVM and LOF. The objective is to assess the detection effectiveness of the proposed methods, focusing on their ability to achieve a robust balance between TPR and false positive rate (FPR). Table 7 presents the classification *accuracy* results, measured as the mean (μ) ± standard deviation (σ) from 50 trials. Each trial consists of 5 realizations of a 10-fold cross-validation.

Table 7 – Classification accuracy of the *competitive* LKPCA variants and the baseline methods on benchmark datasets.

Dataset	Baseline					Proposed	
	IF	LOF	OCSVM	PCA	KPCA	LKPCA-WTA	LKPCA-WTA-MT
Annthyroid	91.14 ± 0.25	**93.23 ± 0.05**	85.09 ± 0.15	91.35 ± 0.10	89.88 ± 0.09	92.64 ± 0.03	92.70 ± 0.03
Breastw	67.66 ± 0.82	64.66 ± 0.17	**93.20 ± 0.32**	68.23 ± 1.28	92.65 ± 0.57	51.27 ± 15.81	86.07 ± 8.19
Cardio	92.91 ± 0.52	**95.17 ± 0.60**	89.94 ± 0.26	89.97 ± 0.27	93.43 ± 0.16	86.36 ± 8.63	90.83 ± 0.41
Glass	91.06 ± 0.55	94.68 ± 0.58	87.01 ± 1.02	90.98 ± 0.72	91.14 ± 0.56	77.64 ± 19.30	**95.69 ± 0.25**
Ionosphere	82.77 ± 2.33	73.39 ± 1.76	91.26 ± 0.98	82.25 ± 0.97	**92.47 ± 0.63**	60.52 ± 24.87	71.86 ± 2.77
Iris (0)	92.41 ± 2.09	67.80 ± 1.95	93.15 ± 1.02	66.68 ± 1.57	**95.31 ± 1.00**	76.21 ± 24.87	94.57 ± 6.77
Iris (1)	68.75 ± 4.96	74.69 ± 3.32	63.13 ± 2.03	63.95 ± 1.83	**80.63 ± 1.81**	78.13 ± 14.11	66.63 ± 0.21
Iris (2)	89.20 ± 4.79	**94.27 ± 2.20**	91.87 ± 1.03	86.40 ± 1.58	91.44 ± 1.46	88.33 ± 18.58	69.41 ± 8.32
Parkinson (0)	71.47 ± 0.77	**82.99 ± 1.12**	79.16 ± 0.95	79.45 ± 0.81	82.15 ± 1.23	70.65 ± 16.41	77.65 ± 3.56
Parkinson (1)	77.39 ± 2.56	86.44 ± 2.60	78.57 ± 1.45	74.91 ± 1.17	**93.57 ± 0.54**	85.20 ± 7.13	43.57 ± 7.95
VC (0)	77.48 ± 0.79	80.18 ± 0.46	75.15 ± 0.71	79.73 ± 0.48	77.83 ± 0.59	80.19 ± 0.85	**80.58 ± 0.13**
VC (1)	64.59 ± 0.68	66.54 ± 0.30	62.99 ± 1.00	64.40 ± 0.44	64.15 ± 0.57	65.44 ± 3.73	**67.66 ± 0.17**
VC (2)	73.90 ± 1.06	73.43 ± 1.03	80.60 ± 1.33	74.08 ± 0.99	**90.52 ± 0.83**	77.33 ± 20.80	56.02 ± 2.96
WF2F (0)	70.03 ± 0.31	**73.52 ± 0.76**	61.87 ± 0.14	62.06 ± 1.24	61.19 ± 0.30	59.44 ± 1.00	59.58 ± 0.01
WF2F (1)	61.89 ± 1.44	81.47 ± 0.41	93.40 ± 0.22	58.80 ± 0.15	**93.68 ± 1.05**	71.57 ± 11.71	61.57 ± 0.03
WF2F (2)	94.88 ± 0.23	**98.33 ± 0.15**	86.14 ± 0.19	91.77 ± 0.60	94.60 ± 0.13	15.93 ± 21.21	93.98 ± 0.01
WF2F (3)	84.57 ± 1.67	86.76 ± 0.85	77.23 ± 0.19	81.33 ± 0.11	**90.98 ± 0.43**	82.61 ± 3.96	84.88 ± 0.03

Source: the author (2024).

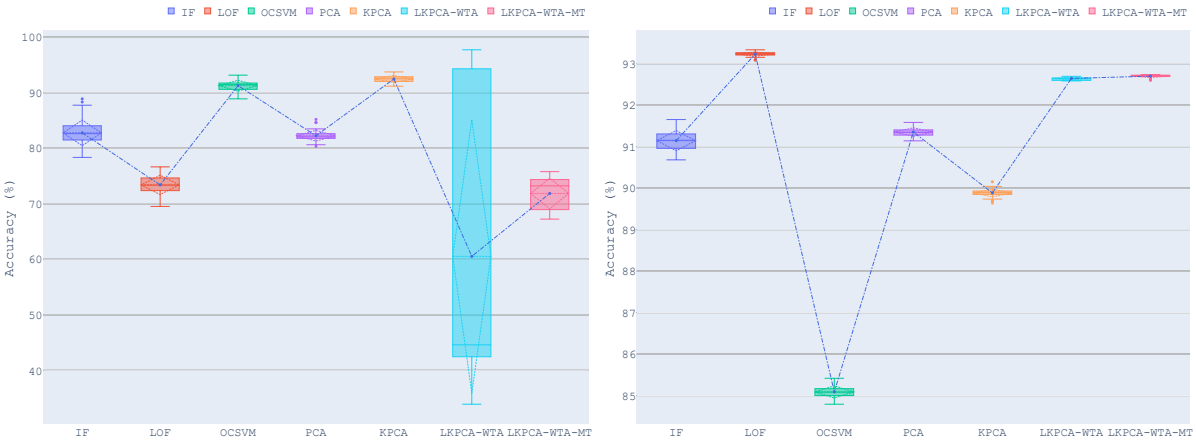
The classification accuracy results indicate that, although the proposed LKPCA methods are competitive, they do not consistently outperform all baseline models across datasets. Their varied performance across datasets highlights a significant distinction between the two variants. LKPCA-WTA, in particular, exhibits high variability, as indicated by its large standard deviations on several datasets (e.g., *Breastw*, *Cardio*, *Ionosphere*, *Glass*, *Iris*). The method's reliance on a single, global threshold for all local models contributes to this instability, as it may result in inadequate thresholding. In contrast, the LKPCA-WTA-MT demonstrates significantly higher stability and generally higher accuracy than LKPCA-WTA, as the usage of multiple thresholds, one for each local model, better captures the distinction between normal and novel instances within different data partitions. This improvement is particularly evident on datasets

like *WF2F (2)*, where LKPCA-WTA performs poorly (15.93% accuracy), while LKPCA-WTA-MT achieves a substantially higher accuracy of 93.98%.

The results also highlight the efficacy of baseline OCC methods. Global KPCA demonstrates excellent performance and stability across many datasets. It achieves the highest accuracy on 7 of the 17 datasets, including *Ionosphere* (92.47%), *Iris (0)* (95.31%), *Iris (1)* (80.63%), *Parkinson (1)* (93.57%), *VC (2)* (90.52%), *WF2F (1)* (93.68%) and *WF2F (3)* (90.98%). Its high accuracy and low standard deviation suggest that the global KPCA is effective for these specific data distributions. It is then a robust method for capturing complex, nonlinear structures when the normal class forms a cohesive cluster in the kernel space. Other baselines methods also perform well on specific datasets. LOF is particularly effective on datasets like *Annthyroid* (93.23%), *Cardio* (95.17%), and *WF2F (2)* (98.33%), while OCSVM is the top performer on the *Breastw* dataset (93.20%). Although IF provides competitive and stable results, it has not achieved the best accuracy on any single dataset.

Boxplots in Figures 21 to 24 are used to visualize and compare the distribution and dispersion of classification accuracy results for competitive LKPCA and baseline methods across benchmark datasets. These plots provide a concise summary of the median, quartiles, and potential outliers, facilitating a direct comparison of model performance.

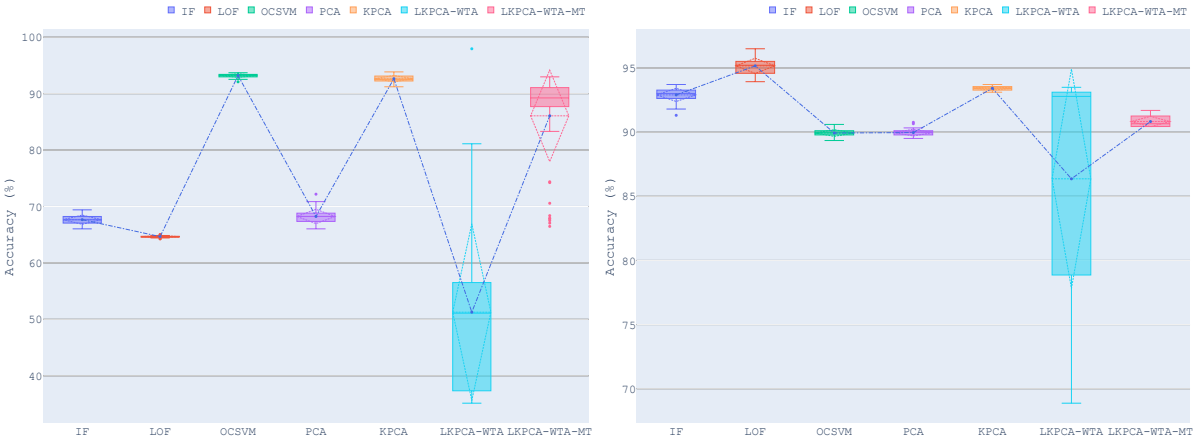
Figure 21 – Boxplot of classification accuracy for *competitive* LKPCA and baseline methods on the *Ionosphere* and *Annthyroid* datasets.



(a) *Ionosphere* (b) *Annthyroid*

Source: the author (2024).

Figure 22 – Boxplot of classification accuracy for *competitive* LKPCA and baseline methods on the *Breastw* and *Cardio* datasets.



(a) *Breastw* (b) *Cardio*

Source: the author (2024).

Figure 23 – Boxplot of classification accuracy for *competitive* LKPCA and baseline methods on the *Parkinson (0)* and *Vertebral Column (VC)* datasets.

(a) *Parkinson (0)* (b) *VC (0)*

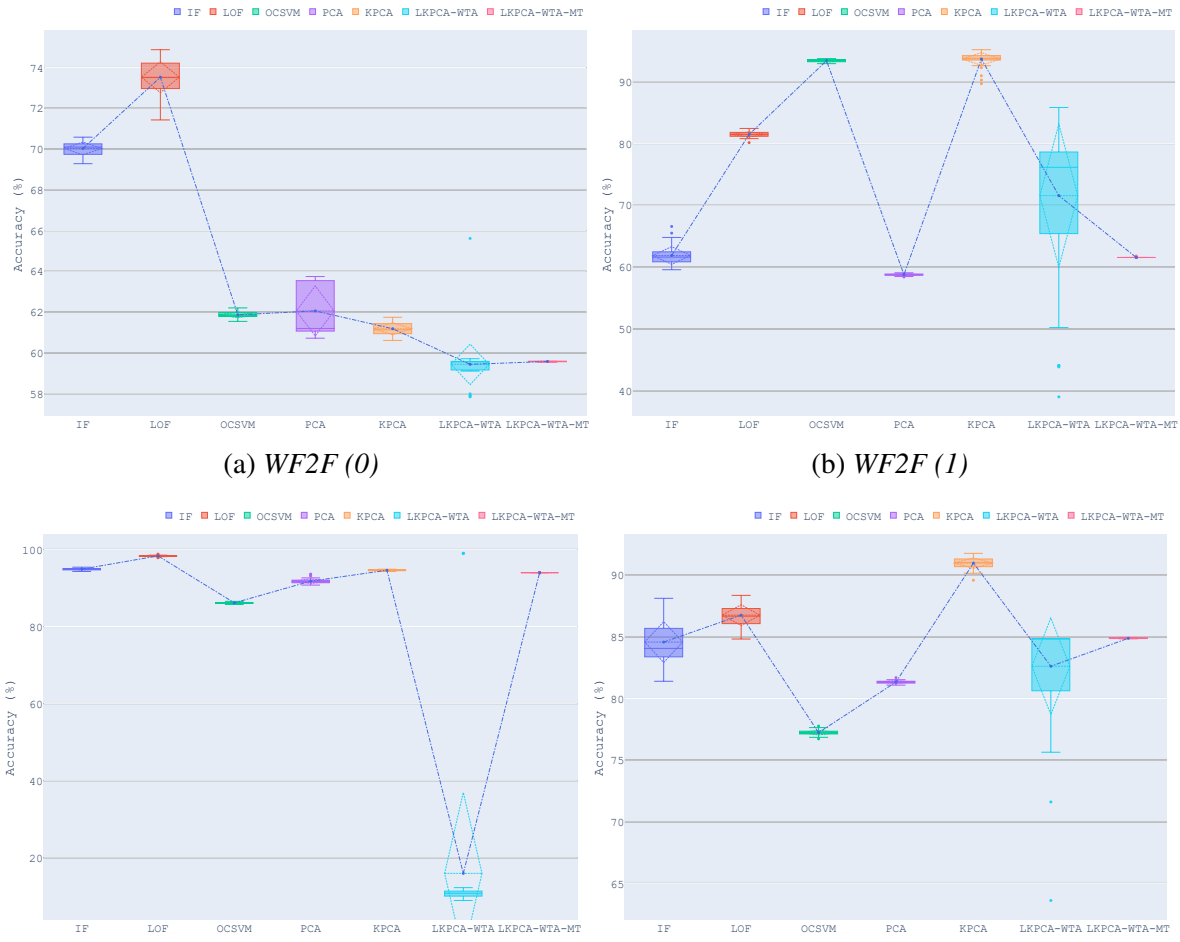
(c) *VC (1)* (d) *VC (2)*

Source: the author (2024).

The boxplots show that LKPCA-WTA generally performs poorly in terms of accuracy compared to the global KPCA on most datasets. Notable exceptions are seen with *Annthyroid*, *VC (0)* and *VC (1)* datasets, where LKPCA-WTA shows superior detection performance. As previously discussed, LKPCA-WTA's single thresholding method can lead to an imprecise threshold, causing significant variability in the results, as reflected by the wide spread in its boxplots. This variance is particularly pronounced in the *Ionosphere*, *Cardio*, *Parkinson (0)*, and *VC (2)* datasets.

For the *Wall-following* dataset, KPCA consistently showed higher accuracy for all derived one-class datasets. In this case, it seems the definition of local partitions has not improved the representation of data. This usually occurs when the separability between clusters is not clear and the partitions are not defined properly. Consequently, LKPCA models may underfit and produce suboptimal thresholds.

Figure 24 – Boxplot of classification accuracy for *competitive* LKPCA and baseline methods on the *Wall-following (2f)* dataset.



(a) *WF2F (0)* (b) *WF2F (1)*

(c) *WF2F (2)* (d) *WF2F (3)*

Source: the author (2024).

Table 8 present the models' performance based on the area under the ROC curve (ROC-AUC). This index, which measures the ability to distinguish between normal and novel instances across various thresholds, offers a more robust evaluation of generalization than the classification accuracy.

Table 8 – ROC-AUC of the *competitive* LKPCA variants and the baseline methods on benchmark datasets.

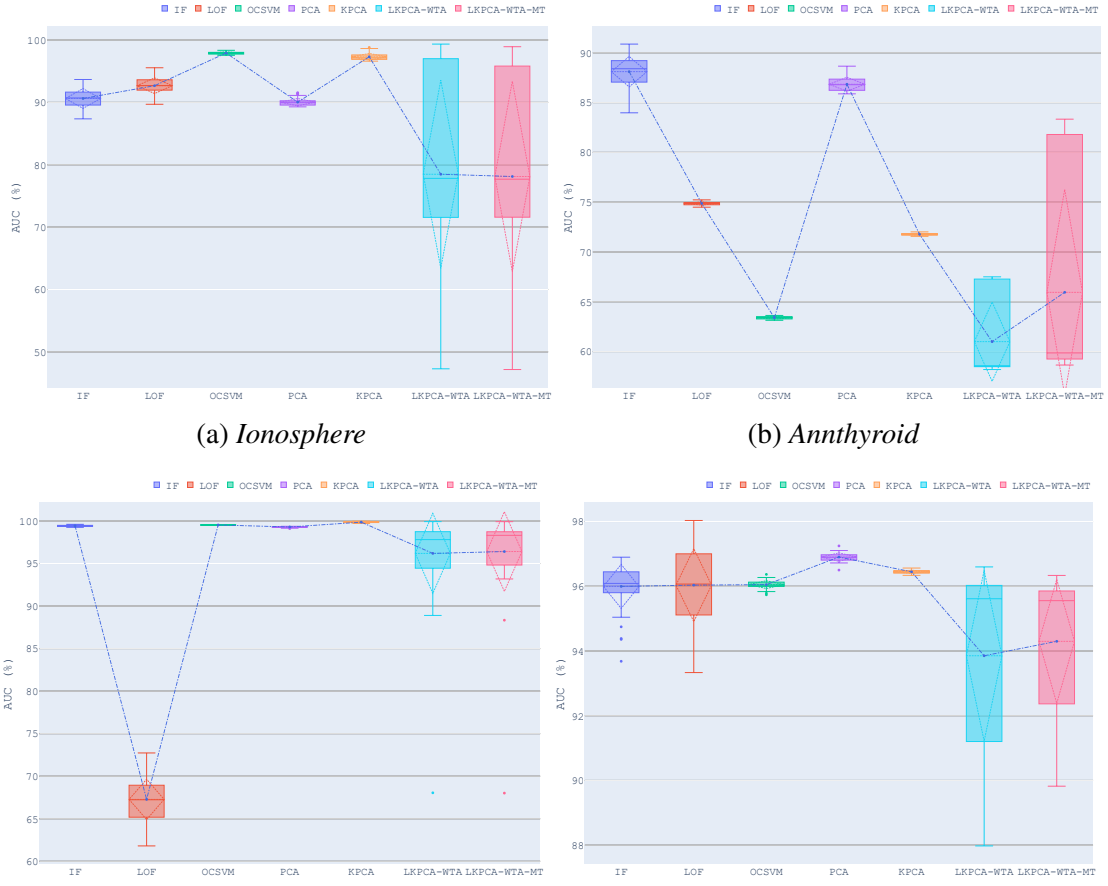
Dataset	Baseline					Proposed	
	IF	LOF	OCSVM	PCA	KPCA	LKPCA-WTA	LKPCA-WTA-MT
Annthyroid	**88.11 ± 1.54**	74.87 ± 0.16	63.40 ± 0.14	86.84 ± 0.71	71.78 ± 0.11	61.00 ± 4.02	65.95 ± 10.43
Breastw	99.43 ± 0.08	67.27 ± 2.39	99.53 ± 0.02	99.30 ± 0.06	**99.86 ± 0.06**	96.19 ± 4.76	96.41 ± 4.73
Cardio	96.00 ± 0.69	96.03 ± 1.14	96.05 ± 0.12	**96.90 ± 0.13**	96.45 ± 0.05	93.85 ± 2.66	94.30 ± 1.93
Glass	74.22 ± 2.36	80.63 ± 0.91	42.39 ± 2.94	48.61 ± 1.95	**84.68 ± 0.99**	52.13 ± 14.12	45.98 ± 21.75
Ionosphere	90.62 ± 1.60	92.68 ± 1.27	**97.92 ± 0.20**	90.04 ± 0.60	97.31 ± 0.55	78.48 ± 15.23	78.12 ± 15.42
Iris (0)	96.61 ± 2.16	**100.00 ± 0.00**	100.00 ± 0.00	99.97 ± 0.12	100.00 ± 0.00	100.00 ± 0.00	99.91 ± 0.41
Iris (1)	79.19 ± 8.38	**92.21 ± 1.36**	42.71 ± 11.38	13.67 ± 11.79	89.99 ± 0.79	88.93 ± 26.39	91.27 ± 26.85
Iris (2)	96.35 ± 2.03	98.96 ± 0.50	98.89 ± 0.39	93.76 ± 1.61	99.09 ± 0.41	**99.35 ± 0.28**	99.22 ± 0.18
Parkinson (0)	65.33 ± 4.20	81.99 ± 2.86	76.36 ± 1.28	79.03 ± 0.88	82.04 ± 1.44	78.28 ± 2.50	**85.80 ± 4.55**
Parkinson (1)	81.92 ± 2.34	90.15 ± 2.51	81.88 ± 1.30	79.66 ± 0.91	**98.74 ± 0.79**	73.38 ± 18.71	74.54 ± 17.85
VC (0)	57.54 ± 5.92	66.23 ± 1.98	56.99 ± 1.12	**74.18 ± 0.57**	53.02 ± 2.08	59.52 ± 1.48	62.14 ± 3.47
VC (1)	51.05 ± 5.98	40.50 ± 1.62	51.38 ± 3.96	**54.37 ± 3.63**	49.70 ± 1.33	50.29 ± 5.11	50.29 ± 5.11
VC (2)	96.66 ± 1.04	90.45 ± 0.31	88.05 ± 0.98	**98.47 ± 0.18**	98.20 ± 0.37	93.08 ± 6.29	85.87 ± 2.41
WF2F (0)	84.56 ± 1.31	**93.37 ± 0.34**	26.20 ± 0.01	41.80 ± 18.87	67.41 ± 0.70	75.90 ± 6.21	83.23 ± 7.38
WF2F (1)	85.04 ± 2.64	95.72 ± 0.26	96.59 ± 0.34	38.21 ± 0.34	**97.80 ± 0.22**	91.71 ± 10.74	79.89 ± 8.28
WF2F (2)	98.74 ± 0.34	**99.35 ± 0.04**	69.06 ± 0.94	84.58 ± 0.97	97.72 ± 0.04	64.83 ± 11.29	60.01 ± 8.35
WF2F (3)	87.52 ± 2.25	91.95 ± 0.65	26.68 ± 0.48	55.62 ± 0.81	**94.28 ± 0.27**	68.01 ± 26.38	73.13 ± 28.57

Source: the author (2024).

The ROC-AUC results reveal that the proposed LKPCA-WTA and LKPCA-WTA-MT variants have significant limitations in their generalization capacity and are generally outperformed by the baseline models. This suggests that while LKPCA can improve detection performance through data partitioning, it may also lead to underfitting to local structures, making it less stable in a competitive setting. Notably, LKPCA-WTA performs ineffectively on several datasets (*Annthyroid* and *Glass*), achieving low ROC-AUC scores. This indicates potential underfitting or poor thresholding due to its reliance on a single global threshold. Additionally, suboptimal partitioning (choice of K or poorly separated clusters) in these datasets may contribute to this underperformance. The high standard deviations observed for both LKPCA variants on many datasets further highlight their lack of stability. In contrast, the baseline models demonstrate superior performance and stability in most cases. Global KPCA is a leading performer, achieving the highest ROC-AUC on 6 datasets. Its consistent performance highlights its robustness for datasets where the normal class forms a cohesive structure in the kernel space. LOF also proves highly effective, winning on 4 datasets (e.g., *Iris (0)*, *Iris (1)*, *WF2F (0)*, and *WF2F (2)*). PCA stands out on *Cardio* and *VC* datasets, achieving the highest scores, while OCSVM and IF show the highest ROC-AUC on *Ionosphere* and *Annthyroid*, respectively.

Boxplots in Figures 25 to 28 are used to visualize and compare the distribution and dispersion of ROC-AUC for competitive LKPCA and baseline models on several benchmark datasets. The boxplots show that LKPCA-WTA models generally perform worse in terms of ROC-AUC than global KPCA on most datasets, except for *Iris (2)*, *Parkinson (0)*, *VC (2)* and *WF2F (0)*. For the LKPCA-WTA-MT variant, there is high variability in its ROC-AUC results for the *Ionosphere*, *Parkinson*, *Annthyroid*, *Cardio* and *WF2F (3)* datasets. This variability was not as evident in the accuracy analysis and suggests that even with multiple thresholds, the performance of LKPCA can vary due to its dependence on data partitioning and the competitive nature of its prediction process.

Figure 25 – Boxplots of ROC-AUC for *competitive* LKPCA and baseline methods on the *Ionosphere*, *Annthyroid*, *Breastw* and *Cardio* datasets.

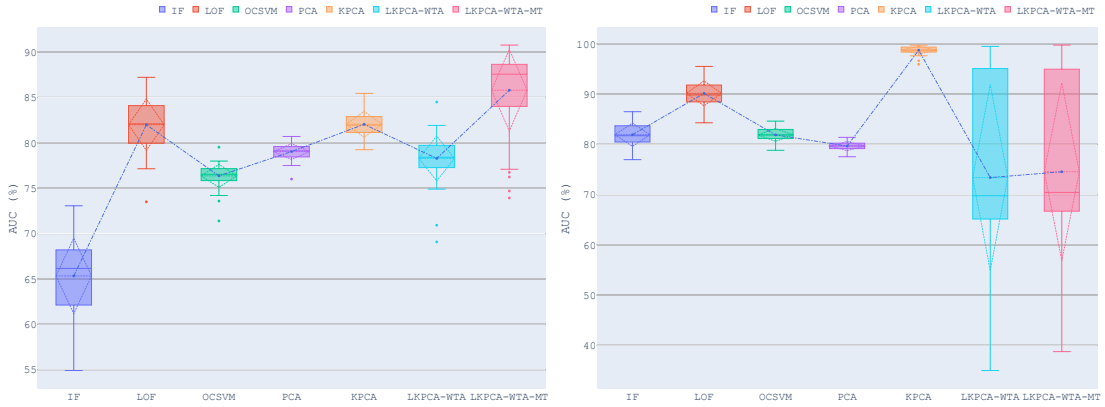


(a) *Ionosphere* (b) *Annthyroid*

(c) *Breastw* (d) *Cardio*

Source: the author (2024).

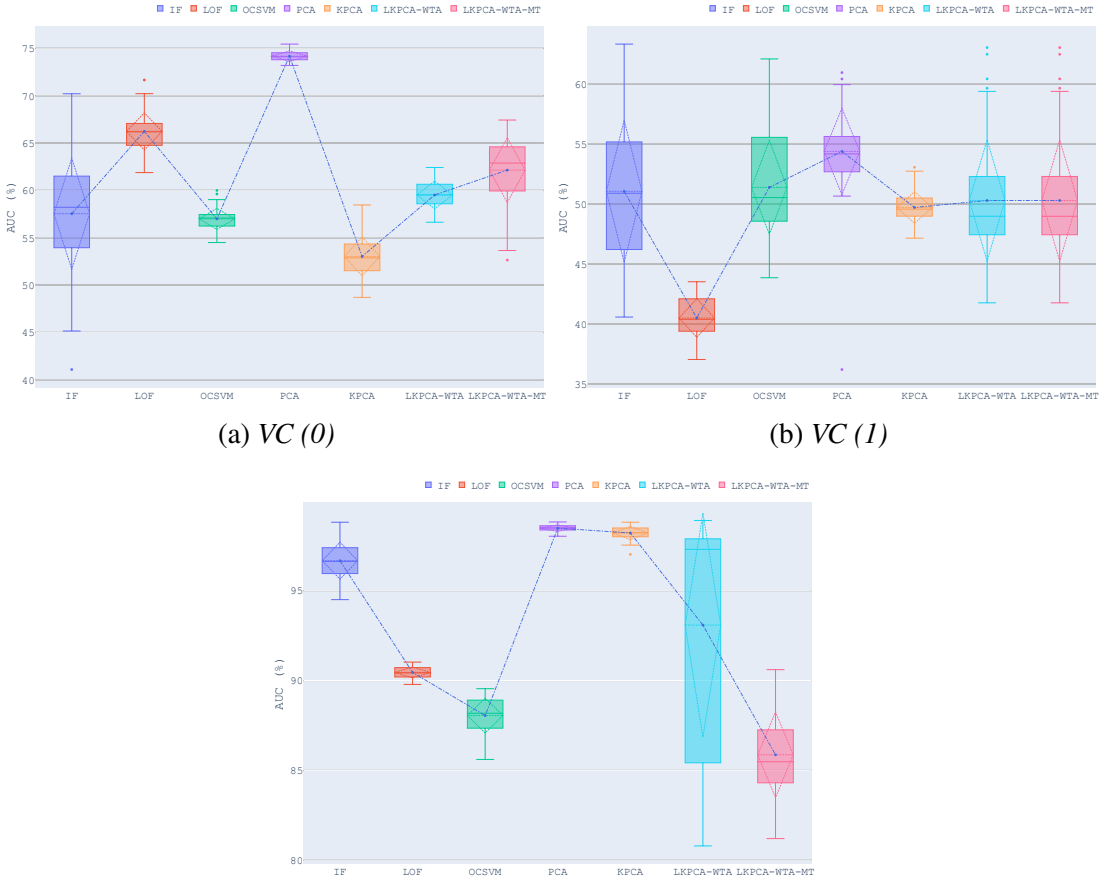
Figure 26 – Boxplots of ROC-AUC for *competitive* LKPCA and baseline methods on the *Parkinson* dataset.



(a) *Parkinson (0)* (b) *Parkinson (1)*

Source: the author (2024).

Figure 27 – Boxplots of ROC-AUC for *competitive* LKPCA and baseline methods on the *Vertebral Column* dataset.

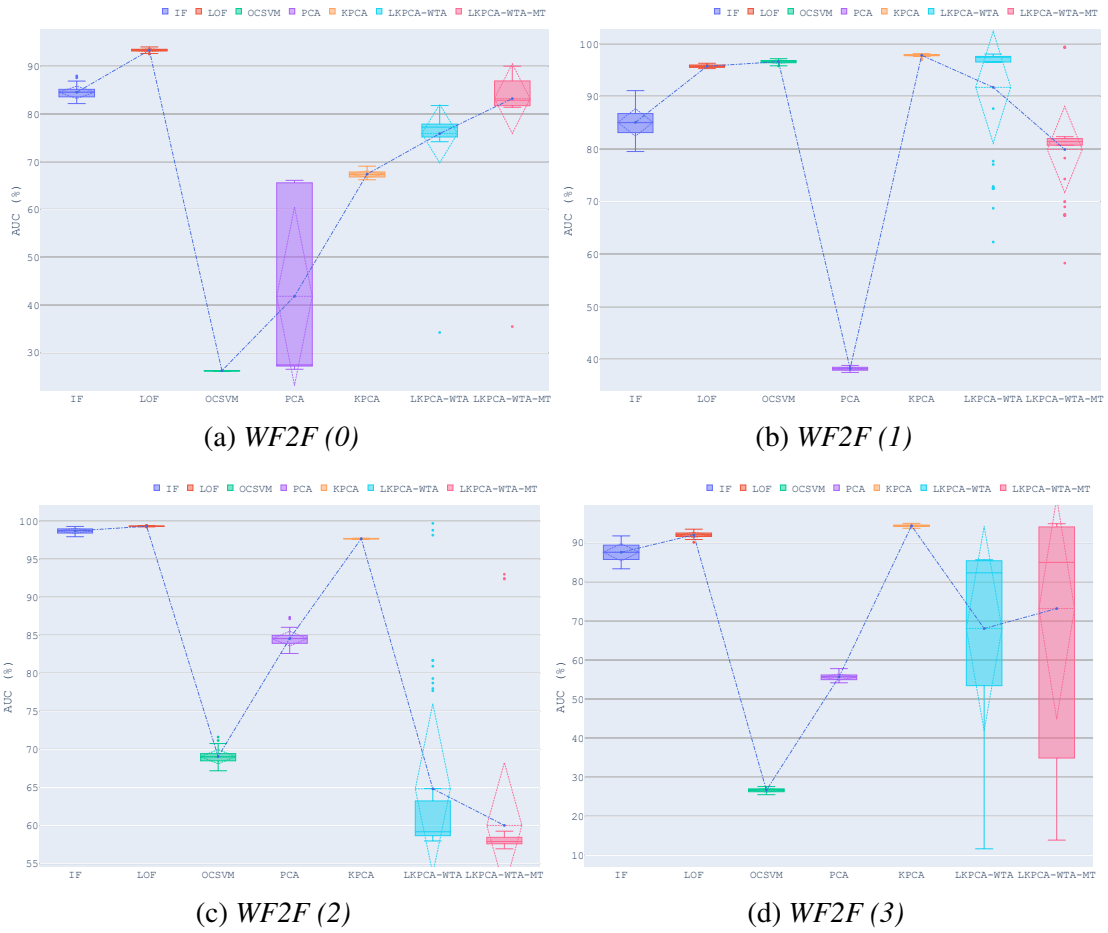


(a) *VC (0)* (b) *VC (1)*

(c) *VC (2)*

Source: the author (2024).

Figure 28 – Boxplots of ROC-AUC for *competitive* LKPCA and baseline methods on the *Wall-following (2f)* dataset.



(a) *WF2F (0)* (b) *WF2F (1)*

(c) *WF2F (2)* (d) *WF2F (3)*

Source: the author (2024).

4.5 Summary

This chapter introduced the LKPCA, a novel OCC approach that integrates the KPCA for novelty detection (HOFFMANN, 2007) with a local learning framework known as CLHP (DRUMOND *et al.*, 2022). This method involves three main steps: data partitioning with a clustering algorithm, training a local KPCA novelty detector in each partition; and establishing a ND threshold based on local reconstruction errors. Two competitive variants of LKPCA were presented: LKPCA-WTA and LKPCA-WTA-MT.

Simulations comparing competitive LKPCA with global KPCA and baseline methods like IF, OCSVM, and LOF showed promising results. Competitive LKPCA can outperform baselines in terms of both accuracy and ROC-AUC. However, its performance suffers from high variance due to its reliance on the data partitioning and thresholding stages.

5 COOPERATIVE LOCAL KERNEL PRINCIPAL COMPONENT ANALYSIS

Local learning is a powerful ML framework that combines multiple local models by using two main approaches: *competitive* and *cooperative*. Chapter 4 introduced the *competitive* LKPCA, where only the winning localized KPCA model is selected for output prediction. In contrast, cooperative learning combines the outputs of all local models into a single, unified decision. This chapter explores a *cooperative* approach to LKPCA, introducing three variants. The first uses the average of local models' output for thresholding and prediction. The second technique quantifies the influence of each localized KPCA model using similarity measures. The third variant employs a stacking ensemble that trains a *metamodel* with the local reconstruction errors from all localized KPCA models over the training set. The chapter concludes by presenting simulation results on benchmark datasets.

5.1 Cooperation Through Averaging

The average weighted local kernel principal component analysis (LKPCA-AVG) combines the averaged outputs of all local models to define its discriminant function and detection threshold. This method assigns equal importance to the contribution of each model. To establish the threshold, it first computes the average reconstruction error for all training data points with respect to each local KPCA model, creating a vector of localized error averages:

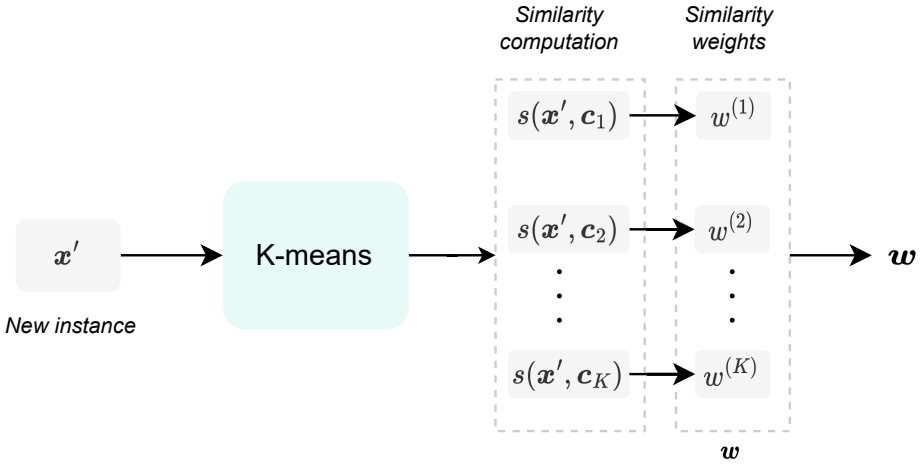
$$\bar{\boldsymbol{e}} = \begin{bmatrix} \bar{e}^{(1)} & \cdots & \bar{e}^{(K)} \end{bmatrix}, \tag{5.1}$$

where $\bar{e}^{(j)} = \frac{1}{n}\sum_{i=1}^{n} e_i^{(j)}$ is the average of reconstruction errors for the j-th local KPCA model. The detection threshold is then defined as the maximum value within the vector of averages, that is, $\theta_{avg} = \max(\bar{\boldsymbol{e}})$.

5.2 Cooperation Through Similarity

The similarity weighted local kernel principal component analysis (LKPCA-SW) method computes a weight vector based on the similarities between an input data ($\boldsymbol{x}'$) and the centroids from the K-means model (as illustrated in Figure 29). These weights are then used to aggregate the outputs from all local models, resulting in a single novelty score. This technique incorporates the proximity of an input instance to the local clusters, enabling a more selective contribution from each model to the overall prediction.

Figure 29 – Illustration of the similarity-based weight vector construction.



Source: the author (2024).

LKPCA-SW's adaptability is enhanced by its ability to weigh the contributions of its individual, localized KPCA models. Instead of treating all local models equally, it adjusts their influence based on how similar their corresponding cluster is to the data instance being predicted. To ensure consistency with the cluster definitions, the same dissimilarity measure used in the K-means algorithm must also be used to determine the weights for each local model. A common dissimilarity measure is the Euclidean distance, also known as the L_2-norm:

$$d_E(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2. \tag{5.2}$$

To convert this into a similarity weight, the following function can be used:

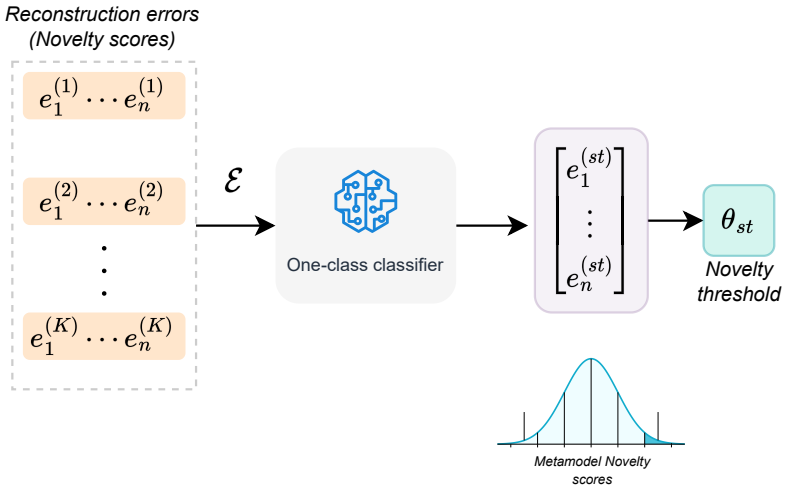
$$s(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{d(\boldsymbol{x}, \boldsymbol{y}) + \epsilon}, \tag{5.3}$$

where ϵ is a small constant that prevents division by zero when the distance tend to zero.

5.3 Cooperation Through Stacking

The stacking local kernel principal component analysis (LKPCA-ST) framework leverages a metamodel to learn a detection rule from the reconstrution errors of localized models. This metamodel uses the matrix of local reconstruction errors ($\mathcal{E}$) as training data, enabling it to distinguish between normal and novel instances based on the aggregated novelty scores from the localized models. This approach allows for a unified detection rule for identifying novel instances from the collective knowledge of multiple models. The process of training an LKPCA-ST model is illustrated in Figure 30.

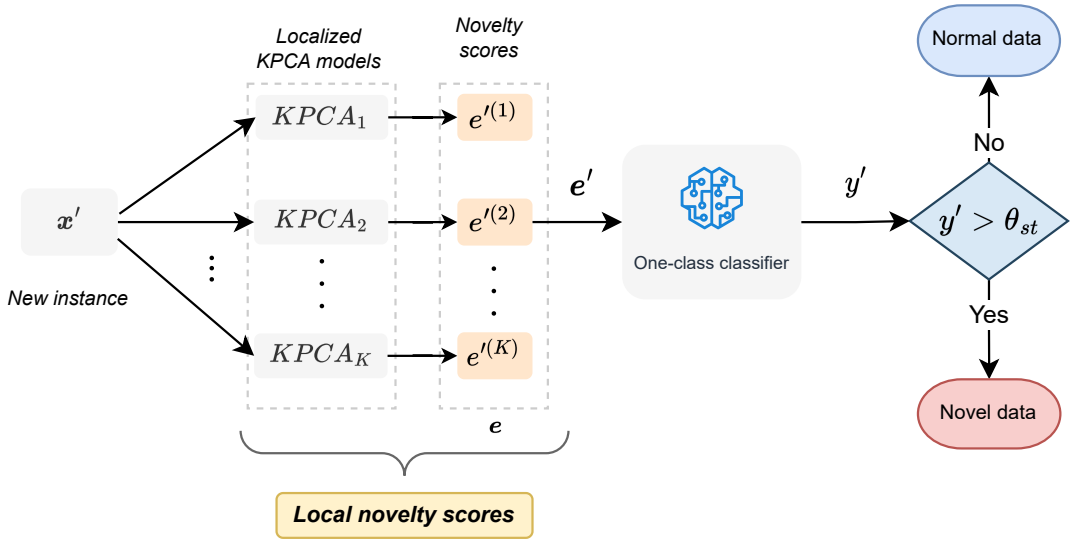
Figure 30 – Illustration of the thresholding strategy utilized in the LKPCA-ST.



Source: the author (2024).

For this approach, we use a 1-nearest neighbor (1-NN) model as the *metamodel*. This technique improves the overall ability to detect new instances by integrating both local insights and an intermediate classifier. Its primary detection capacity relies on the localized KPCA models, with the metamodel serving only to train a decision rule based on the localized novelty scores. The process of detecting novelties in new input instances is a two-step procedure. It first computes the local novelty scores for all K models in the LKPCA setting, and then it determines the metamodel novelty score, $e'^{(st)}$. This process is illustrated in Figure 31.

Figure 31 – Illustration of LKPCA-ST class prediction.



Source: the author (2024).

5.4 Simulations and Results

This section details the simulations and results for the *cooperative* LKPCA. The simulation settings, including cross-validation, hyperparameter optimization, and the datasets used, are consistent with those in the previous chapter; for further details, see Section 4.4. The experiments were designed to assess the classification performance of global KPCA and cooperative LKPCA variants across various benchmark datasets, using accuracy and ROC-AUC as performance indices. Table 9 and 10 present the results for classification accuracy and ROC-AUC, respectively. In each row of these tables, the highest performance index among all KPCA-based approaches are highlighted in bold.

Table 9 – Classification accuracy of the global KPCA and the LKPCA variants on benchmark datasets.

Dataset	Baseline	Proposed				
	KPCA	LKPCA-WTA	LKPCA-WTA-MT	LKPCA-AVG	LKPCA-SW	LKPCA-ST
Annthyroid	89.88 ± 0.09	92.64 ± 0.03	**92.70 ± 0.03**	92.60 ± 0.03	92.68 ± 0.02	87.13 ± 0.22
Breastw	92.65 ± 0.57	51.27 ± 15.81	86.07 ± 8.19	**98.56 ± 0.30**	97.15 ± 0.46	94.03 ± 0.77
Cardio	93.43 ± 0.16	86.36 ± 8.63	90.83 ± 0.41	92.64 ± 0.22	**93.65 ± 0.14**	90.37 ± 0.45
Glass	91.14 ± 0.56	77.64 ± 19.30	**95.69 ± 0.25**	95.61 ± 0.35	94.69 ± 0.62	89.47 ± 1.32
Ionosphere	92.47 ± 0.63	60.52 ± 24.87	71.86 ± 2.77	**93.48 ± 0.48**	82.24 ± 13.66	92.91 ± 1.11
Iris (0)	95.31 ± 1.00	76.21 ± 24.87	94.57 ± 6.77	**99.40 ± 0.70**	97.35 ± 2.48	94.61 ± 2.03
Iris (1)	80.63 ± 1.81	78.13 ± 14.11	66.63 ± 0.21	86.85 ± 3.72	85.04 ± 15.63	**91.29 ± 1.20**
Iris (2)	91.44 ± 1.46	88.33 ± 18.58	69.41 ± 8.32	**96.56 ± 0.43**	84.87 ± 15.56	92.53 ± 0.94
Parkinson (0)	**82.15 ± 1.23**	70.65 ± 16.41	77.65 ± 3.56	79.60 ± 2.84	76.81 ± 2.79	73.54 ± 3.69
Parkinson (1)	93.57 ± 0.54	85.20 ± 7.13	43.57 ± 7.95	**94.94 ± 1.76**	85.25 ± 7.19	94.06 ± 2.71
VC (0)	77.83 ± 0.59	80.19 ± 0.85	**80.58 ± 0.13**	80.54 ± 0.31	80.27 ± 0.22	76.97 ± 1.18
VC (1)	64.15 ± 0.57	65.44 ± 3.73	**67.66 ± 0.17**	67.65 ± 0.24	67.28 ± 0.33	64.87 ± 0.87
VC (2)	**90.52 ± 0.83**	77.33 ± 20.80	56.02 ± 2.96	68.17 ± 1.89	70.63 ± 1.73	84.96 ± 0.85
WF2F (0)	61.19 ± 0.30	59.44 ± 1.00	59.58 ± 0.01	59.70 ± 0.04	59.61 ± 0.05	**64.78 ± 0.77**
WF2F (1)	**93.68 ± 1.05**	71.57 ± 11.71	61.57 ± 0.03	61.60 ± 0.04	61.58 ± 0.05	85.53 ± 5.45
WF2F (2)	**94.60 ± 0.13**	15.93 ± 21.21	93.98 ± 0.01	94.00 ± 0.01	93.95 ± 0.05	91.91 ± 0.30
WF2F (3)	**90.98 ± 0.43**	82.61 ± 3.96	84.88 ± 0.03	84.86 ± 0.01	84.84 ± 0.02	85.55 ± 1.85

Source: the author (2024).

The results demonstrate that the proposed LKPCA variants generally outperform the global KPCA model in terms of classification accuracy. In most cases, one of the LKPCA methods achieves the highest accuracy score, with the winning model varying by dataset. For instance, LKPCA-WTA-MT and LKPCA-AVG achieved the highest accuracy on 5 and 6 datasets out of 17, respectively. In addition, LKPCA-SW stands out by achieving the top accuracy on the *Cardio* dataset. The large standard deviation for LKPCA-WTA and LKPCA-SW variants reveal their variability in performance on several datasets (e.g., *Breastw*, *Ionosphere*, and *Iris (0)*). This variability suggests that the effectiveness of these models is highly dependent on how well the dataset can be partitioned into distinct local clusters. The standard deviations for LKPCA-ST and LKPCA-AVG are consistently low across most datasets, make them consistent and dependable LKPCA variants.

The ROC-AUC results reveal that the proposed LKPCA variants generally perform on par with the global KPCA and, in several instances, surpass it. This is particularly true for LKPCA-ST and LKPCA-SW. LKPCA-ST demonstrates exceptional performance, achieving the highest ROC-AUC on 4 datasets (*Annthyroid*, *Glass*, *Ionosphere*, and *WF2F (2)*), while LKPCA-SW outperforms all other models on the *Cardio* and *VC (1)* datasets. The cooperative LKPCA exhibit a lower degree of variability, except for LKPCA-AVG and LKPCA-SW on *WF2F (3)* and *Iris (1)*, respectively. While LKPCA-ST demonstrates high ROC-AUC in various cases, it does not consistently achieve the highest accuracy, reflecting the trade-offs between sensitivity and specificity across models.

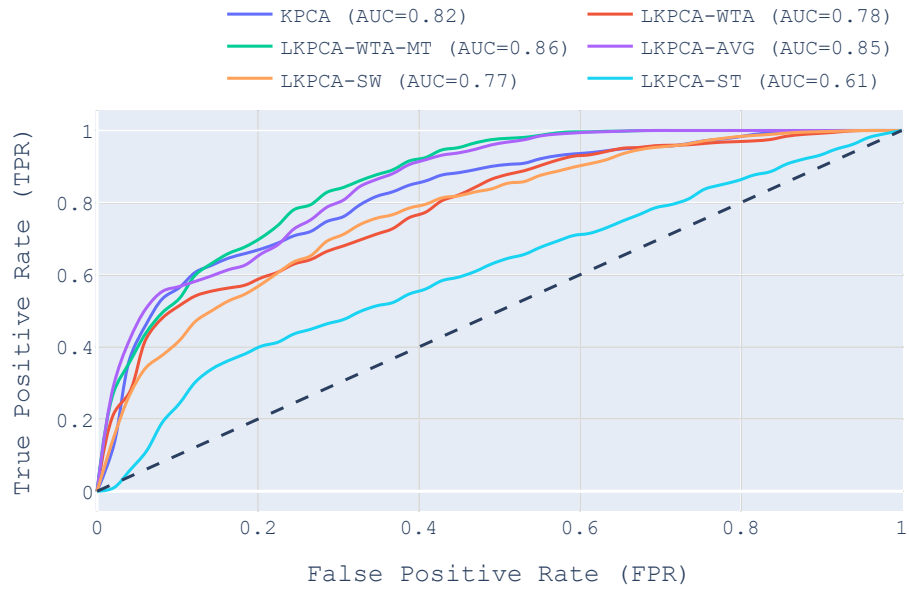
Table 10 – ROC-AUC of the global KPCA and LKPCA variants on benchmark datasets.

Dataset	Baseline	Proposed				
	KPCA	LKPCA-WTA	LKPCA-WTA-MT	LKPCA-AVG	LKPCA-SW	LKPCA-ST
Annthyroid	71.78 ± 0.11	61.00 ± 4.02	65.95 ± 10.43	61.53 ± 0.14	64.99 ± 0.66	**74.89 ± 0.61**
Breastw	**99.86 ± 0.06**	96.19 ± 4.76	96.41 ± 4.73	99.79 ± 0.05	99.74 ± 0.06	99.40 ± 0.29
Cardio	96.45 ± 0.05	93.85 ± 2.66	94.30 ± 1.93	96.09 ± 0.26	**97.23 ± 0.27**	94.09 ± 0.52
Glass	84.68 ± 0.99	52.13 ± 14.12	45.98 ± 21.75	30.68 ± 8.52	22.32 ± 8.83	**90.74 ± 1.61**
Ionosphere	97.31 ± 0.55	78.48 ± 15.23	78.12 ± 15.42	96.09 ± 1.32	95.21 ± 2.69	**98.59 ± 0.22**
Iris (0)	**100.00 ± 0.00**	100.00 ± 0.00	99.91 ± 0.41	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Iris (1)	89.99 ± 0.79	88.93 ± 26.39	91.27 ± 26.85	96.58 ± 1.04	91.71 ± 24.44	**98.34 ± 0.60**
Iris (2)	99.09 ± 0.41	99.35 ± 0.28	99.22 ± 0.18	**99.54 ± 0.10**	98.42 ± 0.51	97.85 ± 0.67
Parkinson (0)	82.04 ± 1.44	78.28 ± 2.50	**85.80 ± 4.55**	85.22 ± 1.40	77.36 ± 3.79	60.88 ± 4.42
Parkinson (1)	**98.74 ± 0.79**	73.38 ± 18.71	74.54 ± 17.85	97.54 ± 1.60	95.02 ± 3.32	98.46 ± 1.24
VC (0)	53.02 ± 2.08	59.52 ± 1.48	62.14 ± 3.47	61.44 ± 4.75	54.84 ± 1.43	**67.19 ± 6.03**
VC (1)	49.70 ± 1.33	50.29 ± 5.11	50.29 ± 5.11	51.07 ± 4.69	**52.36 ± 3.97**	51.13 ± 2.18
VC (2)	**98.20 ± 0.37**	93.08 ± 6.29	85.87 ± 2.41	86.32 ± 1.52	86.52 ± 1.61	92.11 ± 0.44
WF2F (0)	67.41 ± 0.70	75.90 ± 6.21	**83.23 ± 7.38**	32.03 ± 0.95	32.95 ± 0.79	71.01 ± 3.26
WF2F (1)	**97.80 ± 0.22**	91.71 ± 10.74	79.89 ± 8.28	93.10 ± 0.37	93.59 ± 1.00	91.89 ± 0.90
WF2F (2)	97.72 ± 0.04	64.83 ± 11.29	60.01 ± 8.35	64.70 ± 1.64	74.98 ± 3.44	**99.51 ± 0.06**
WF2F (3)	**94.28 ± 0.27**	68.01 ± 26.38	73.13 ± 28.57	52.94 ± 19.29	22.07 ± 7.19	87.32 ± 1.56

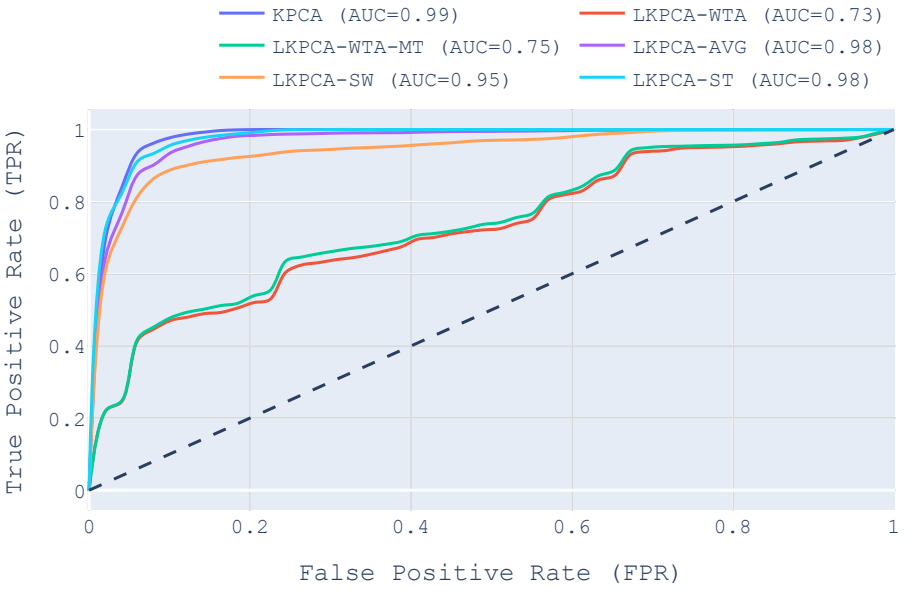
Source: the author (2024).

Overall, the cooperative LKPCA variants perform comparably to the global KPCA model, with the proposed methods often showing higher performance in both accuracy and ROC-AUC. An examination of the *mean* ROC curves provides a more detailed evaluation of these LKPCA variants. Figure 32 illustrates the mean ROC curves for the two versions of the *Parkinson* dataset. For *Parkinson (0)*, both LKPCA-WTA-MT and LKPCA-AVG outperform the global KPCA. Meanwhile, the poor performance of LKPCA-ST suggests that underfitting of localized KPCAs impacted the 1-NN metamodel. In contrast, for *Parkinson (1)*, the global KPCA surpasses all LKPCA variants, although LKPCA-ST and LKPCA-AVG demonstrate a nearly equivalent performance.

Figure 32 – Mean ROC curves for KPCA and LKPCA on the *Parkinson* dataset.



(a) *Parkinson (0)*

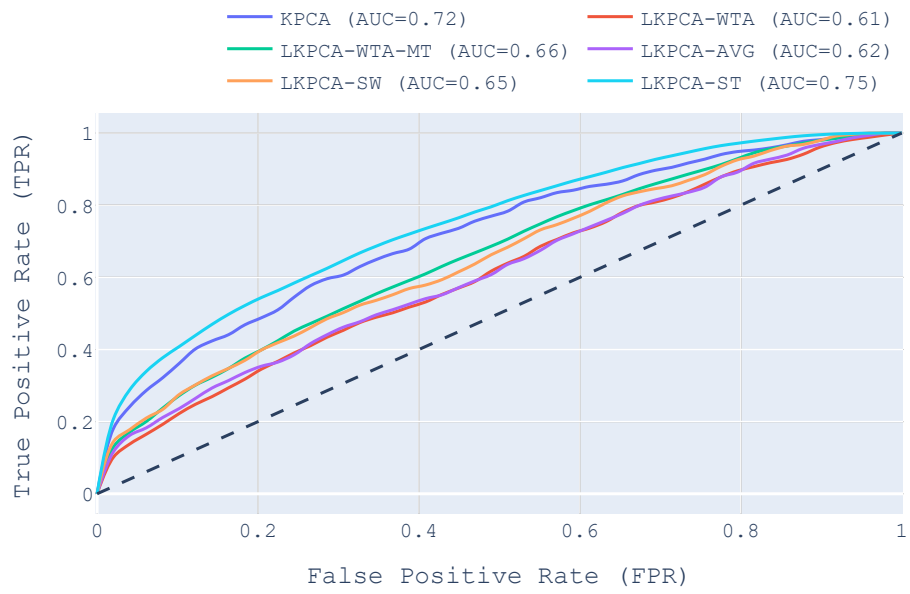


(b) *Parkinson (1)*

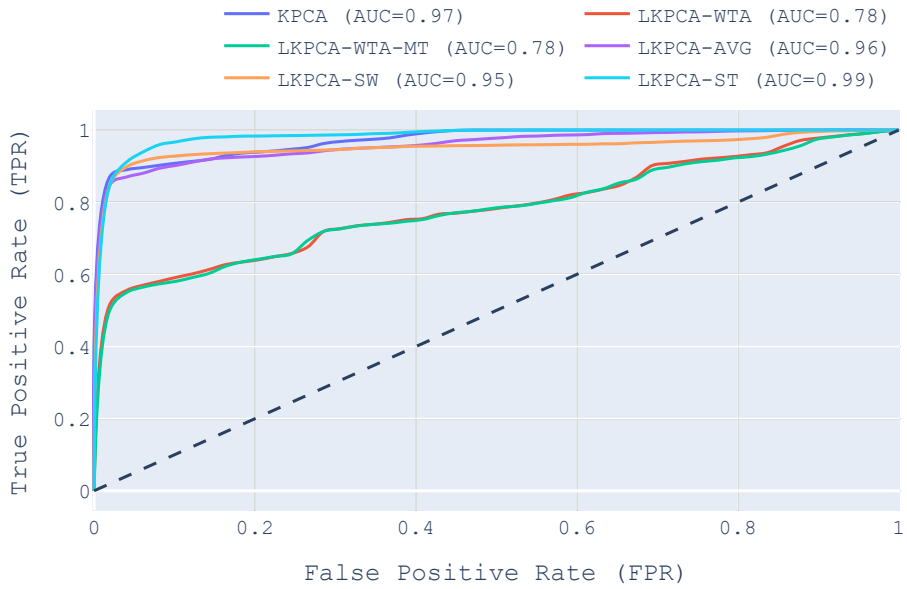
Source: the author (2024).

Figure 33 shows the *mean* ROC curves for *Annthyroid* and *Ionosphere* datasets. LKPCA-ST demonstrates a higher performance with respect to all KPCA-based variants in both *Annthyroid* and *Ionosphere*, achieving $AUC = 0.7489$, and $AUC = 0.9859$, respectively. In the *Ionosphere*, the competitive LKPCA exhibits poor performance. This indicates that detecting novelty through individual local models in a competitive setting does not improve the model's discriminant capability may even cause its degradation.

Figure 33 – Mean ROC curves for KPCA and LKPCA on the *Annthyroid* and *Ionosphere* datasets.



(a) *Annthyroid*

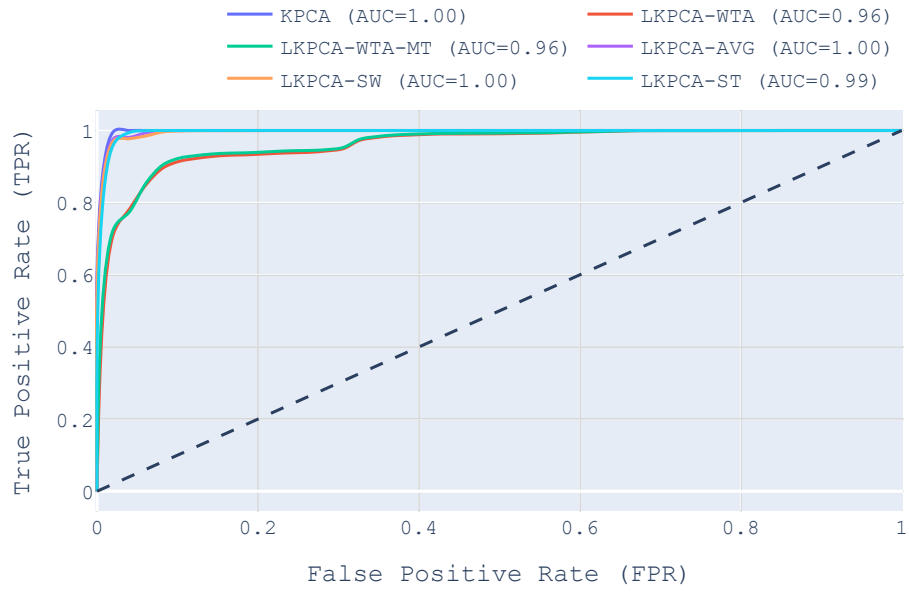


(b) *Ionosphere*

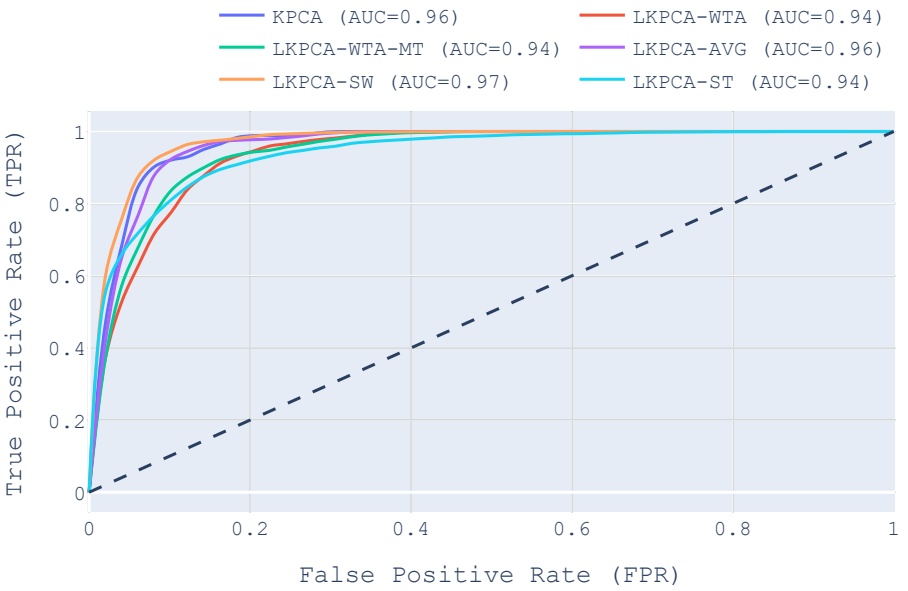
Source: the author (2024).

Concerning the *Breastw* and *Cardio* datasets, illustrated in Figure 34, various LKPCA variations exhibit similar outcomes with slight distinctions among them. In the *Breastw* dataset, KPCA, LKPCA-SW, and LKPCA-AVG showcase nearly optimal ROC curves, resulting in an $AUC \approx 1$. In the *Cardio* dataset, LKPCA-SW, LKPCA-ST, and KPCA exhibit the best results. Additionally, even though the distinction is not significant, the *competitive* variant of LKPCA exhibits lower outcomes when contrasted with other LKPCA variations.

Figure 34 – Mean ROC curves for KPCA and LKPCA on the *Breastw* and *Cardio* datasets.



(a) *Breastw*

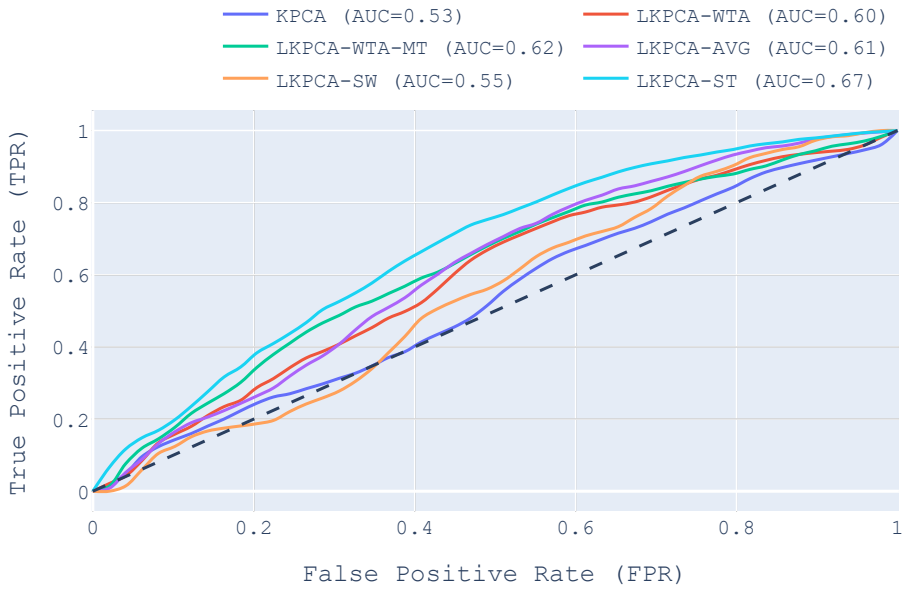


(b) *Cardio*

Source: the author (2024).

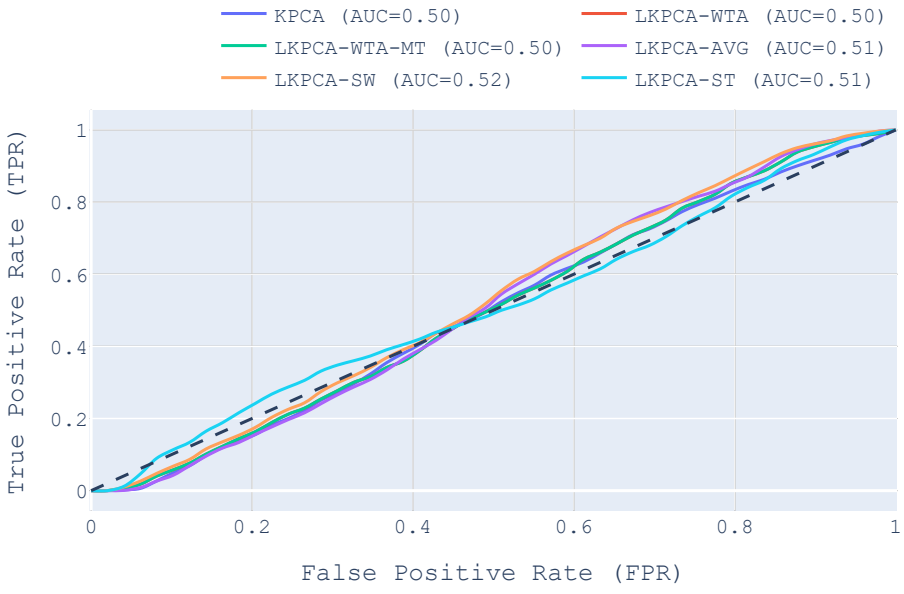
Figure 35 illustrates the mean ROC curves for the *vertebral column (VC)* datasets. In both *VC (0)* and *VC (1)*, all methods present ROC curves near the random model line, suggesting a significant overlap between normal and novelty classes. In *VC (0)*, all LKPCA variants outperformed the global KPCA, implying that employing local modeling may be beneficial for this dataset. Furthermore, LKPCA-ST exhibited the highest ROC-AUC, suggesting that simple rules like averaging may not sufficiently capture the intricacies of local models' representations.

Figure 35 – Mean ROC curves for KPCA and LKPCA on the *VC (0)* and *VC (1)* datasets



(a) *VC (0)*

.

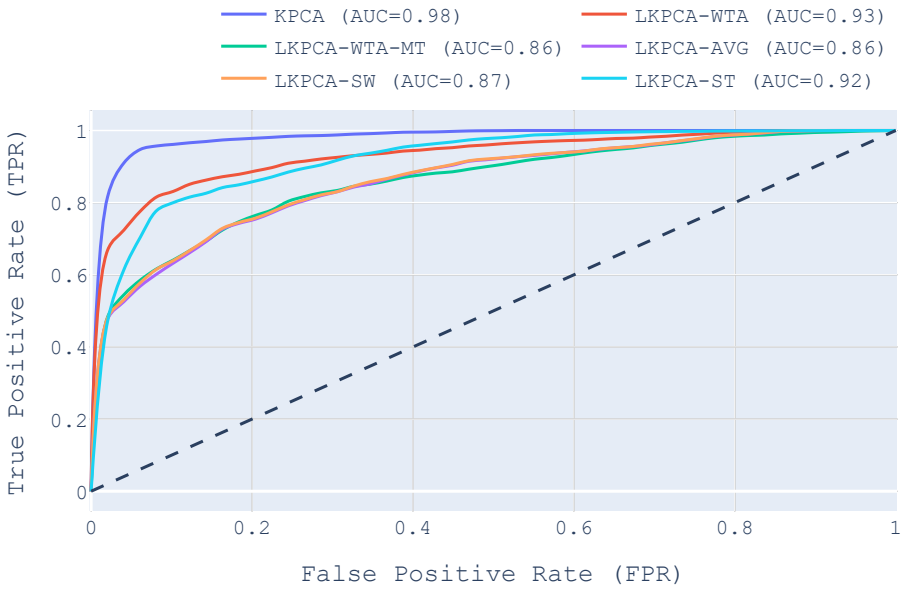


(b) *VC (1)*

Source: the author (2024).

The *VC (2)*, unlike *VC (0)* and *VC (1)*, appears to be a less complex problem, as all KPCA-based models achieved an $AUC > 0.85$ (see Figure 36). Among them, KPCA demonstrated the highest performance with an $AUC = 0.9820$, followed by LKPCA-WTA with $AUC = 0.9308$, and LKPCA-ST with $AUC = 0.9211$. The remaining variants showed comparable results, with ROC-AUC values around 0.86.

Figure 36 – Mean ROC curves for KPCA and LKPCA on the *VC (2)* dataset.



Source: the author (2024).

In the experimentation, we also compared the performance of LKPCA variants and global KPCA against several state-of-the-art baselines, including IF, OCSVM and LOF. The results of this performance comparison are summarized in Tables 11 and 12, where datasets on which an LKPCA variant performed best are highlighted in green.

Table 11 – Best classification accuracy on state-of-the-art methods and LKPCA variants.

Dataset	State-of-the-art		LKPCA variant	
Annthyroid	LOF	**93.23 ± 0.05**	WTA-MT	92.70 ± 0.03
Breastw	OCSVM	**93.20 ± 0.32**	AVG	86.07 ± 8.19
Cardio	LOF	**95.17 ± 0.60**	SW	90.83 ± 0.41
Glass	LOF	94.68 ± 0.58	WTA-MT	**95.69 ± 0.25**
Ionosphere	KPCA	**92.47 ± 0.63**	AVG	71.86 ± 2.77
Iris (0)	KPCA	**95.31 ± 1.00**	AVG	94.57 ± 6.77
Iris (1)	KPCA	**80.63 ± 1.81**	ST	78.13 ± 14.11
Iris (2)	LOF	**94.27 ± 2.20**	AVG	88.33 ± 18.58
Parkinson (0)	LOF	**82.99 ± 1.12**	AVG	79.60 ± 2.84
Parkinson (1)	KPCA	**93.57 ± 0.54**	AVG	**94.94 ± 1.76**
VC (0)	LOF	80.18 ± 0.46	WTA-MT	**80.58 ± 0.13**
VC (1)	LOF	66.54 ± 0.30	WTA-MT	**67.66 ± 0.17**
VC (2)	KPCA	**90.52 ± 0.83**	ST	84.96 ± 0.85
WF2F (0)	LOF	**73.52 ± 0.76**	ST	64.78 ± 0.77
WF2F (1)	KPCA	**93.68 ± 1.05**	ST	85.53 ± 5.45
WF2F (2)	LOF	**98.33 ± 0.15**	AVG	94.00 ± 0.01
WF2F (3)	KPCA	**90.98 ± 0.43**	ST	85.55 ± 1.85

Source: the author (2024).

The state-of-the-art baselines, particularly LOF and KPCA, achieved the highest accuracy on most datasets (13 of the 17). However, the LKPCA variants has shown competitive results, surpassing all baselines on 4 datasets: LKPCA-WTA-MT on *Glass*, *VC (0)*, and *VC (1)*, and LKPCA-AVG on *Parkinson (1)*. In terms of ROC-AUC, the LKPCA variants were highly competitive, outperforming baseline methods on 8 of 17 datasets. LKPCA-ST achieved the top score on *Glass*, *Ionosphere*, *Iris (1)*, and *WF2F (2)*, while LKPCA-SW has shown the best score on the *Cardio* dataset. The global KPCA and other baselines still demonstrate superior performance on *Breastw*, *Parkinson (1)*, and *WF2F (1)*. In summary, the results suggest that cooperative LKPCA variants are often capable of achieving a better balance between true positive and false positive rates compared to traditional methods.

Table 12 – Best ROC-AUC on state-of-the-art methods and LKPCA variants.

Dataset	State-of-the-art		LKPCA variant	
Annthyroid	IF	**88.11 ± 1.54**	ST	74.89 ± 0.61
Breastw	KPCA	**99.86 ± 0.06**	AVG	99.79 ± 0.05
Cardio	PCA	96.90 ± 0.13	SW	**97.23 ± 0.27**
Glass	KPCA	84.68 ± 0.99	ST	**90.74 ± 1.61**
Ionosphere	OCSVM	97.92 ± 0.20	ST	**98.59 ± 0.22**
Iris (0)	LOF/OCSVM	**100.00 ± 0.00**	WTA/AVG	**100.00 ± 0.00**
Iris (1)	LOF	92.21 ± 1.36	ST	**98.34 ± 0.60**
Iris (2)	KPCA	99.09 ± 0.41	AVG	**99.54 ± 0.10**
Parkinson (0)	KPCA	82.04 ± 1.44	WTA-MT	**85.80 ± 4.55**
Parkinson (1)	KPCA	**98.74 ± 0.79**	ST	98.46 ± 1.24
VC (0)	PCA	**74.18 ± 0.57**	ST	67.19 ± 6.03
VC (1)	PCA	**54.37 ± 3.63**	SW	52.36 ± 3.97
VC (2)	PCA	**98.47 ± 0.18**	ST	92.11 ± 0.44
WF2F (0)	LOF	**93.37 ± 0.34**	WTA-MT	83.23 ± 7.38
WF2F (1)	KPCA	**97.80 ± 0.22**	SW	93.59 ± 1.00
WF2F (2)	LOF	99.35 ± 0.04	ST	**99.51 ± 0.06**
WF2F (3)	KPCA	**94.28 ± 0.27**	ST	87.32 ± 1.56

Source: the author (2024).

5.5 Summary

This chapter introduces and details the three *cooperative* LKPCA variants for ND: LKPCA-AVG, LKPCA-SW, and LKPCA-ST, which differ in how they combine local model outputs — from simple averaging, through similarity-weighted aggregation, to a stacking ensemble that trains a *metamodel* to learn a decision rule from local novelty scores. We evaluate these variants on benchmark datasets using classification accuracy and ROC-AUC, and compare them with global KPCA, competitive LKPCA and established OCC methods. The findings highlight the efficacy of cooperative LKPCA, which demonstrates a consistent performance regarding accuracy and ROC-AUC.

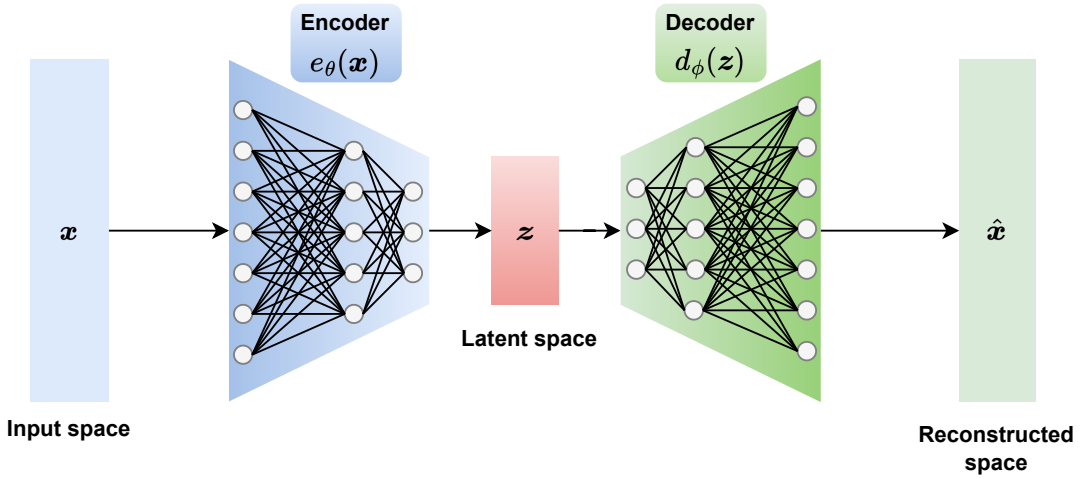
6 LOCAL AUTOENCODERS

Traditional autoencoders (AEs) have proven to be effective at learning compact data representations. However, these models may encounter difficulties in capturing local variations within complex and high-dimensional datasets. To address this limitation, we introduce a novel OCC method called local autoencoder (LAE). LAEs are designed to incorporate local information into the encoding and decoding processes, thereby enabling more precise and detailed reconstructions. This chapter first presents the fundamentals of LAEs. Subsequently, we present the findings from a series of simulations designed to evaluate the application of LAEs on time series ND problems. Finally, we provide a comprehensive discussion of the results, comparing the performance of LAEs against established methodologies, including global autoencoders (GAEs) and NN learning approaches on benchmark datasets.

6.1 A Brief Note on Autoencoders

Autoencoders are artificial neural networks designed to learn a compressed representation of input data. Their architecture consists of two main components: an encoder and a decoder. The encoder compresses high-dimensional input data into a lower-dimensional representation known as latent space. This latent space captures the essential features of the input data. Subsequently, the decoder reconstructs the original input from this compressed latent space (CHARTE *et al.*, 2020). Figure 37 illustrates the learning structure of an AE.

Figure 37 – Autoencoder architecture.



Source: the author (2024).

The decoder evaluates the efficacy of the latent space by analyzing the reconstruction error — the difference between the original and reconstructed data points. This error serves as a direct performance index, where a low value indicates that the latent space successfully captured the essential features of the data. Conversely, a high reconstruction error suggests that the model was either inadequately fitted or incapable of representing the data through its structure. The loss function to be minimized while training an AE are typically given by the expression:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\psi(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i), \tag{6.1}$$

where n is the number of training samples; $\boldsymbol{x}_i$ is the i-th input data sample; and $\hat{\boldsymbol{x}}_i$ is the i-th reconstructed data sample. The function $\psi(\cdot)$ quantifies the dissimilarity between the original and the reconstructed data, determining the model's ability of reconstructing input data samples without information loss. A generally used loss function for training AEs is the mean squared error (MSE) between the input data and the respective reconstructed ones:

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|_2^2. \tag{6.2}$$

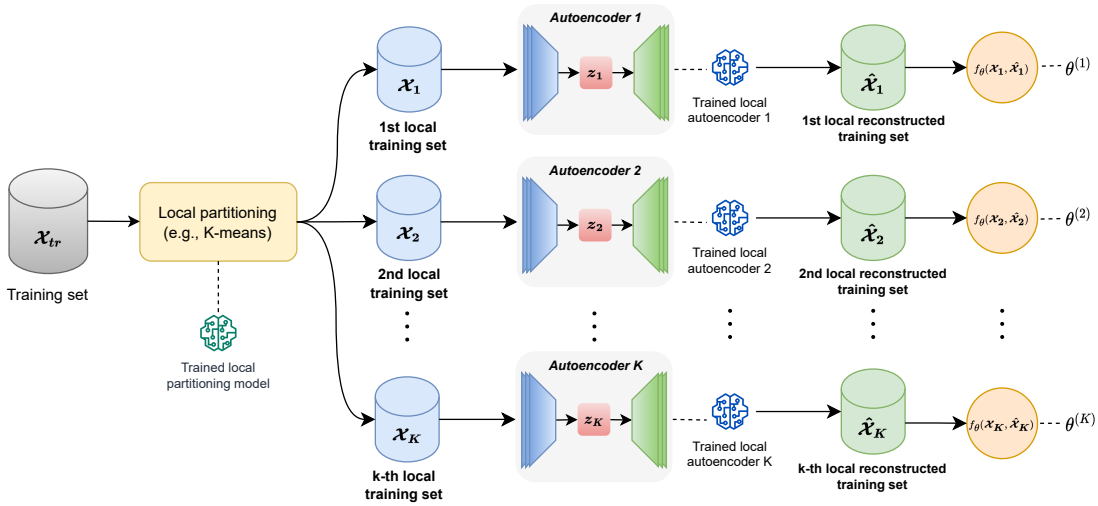
6.1.1 Autoencoders for Novelty Detection

In ND, AEs are trained exclusively on normal data, and a new data point is classified as a novelty if its reconstruction error exceeds a threshold determined by the training set's error distribution. The underlying assumption is that novelties will be poorly reconstructed by a model trained solely on normal data, yielding a higher error that exceeds the established threshold. This principle allows for the distinction between normal and novel samples. The following sections build upon this strategy by detailing the proposed method, which leverages this approach across multiple locally trained AEs.

6.2 Local Autoencoders

Local autoencoders (LAEs) encompass a clustering-based local model characterized by a two-stage framework. The first stage involves the utilization of clustering algorithms to define data partitions based on feature similarities. The second stage involves training a localized AE using the data points within each cluster. The schematic representation of the LAE model training process is depicted in Figure 38.

Figure 38 – Illustration of the learning process of the local autoencoder.



Source: the author (2024).

The clustering step in LAE divides the data into K distinct local partitions. This enables the model to compute multiple data representations, one for each localized AE, which are then used for novelty detection. The process involves computing the reconstruction errors and defining a set of local thresholds: $\theta^{(j)}$, where $j = 1, \ldots, K$. The localized reconstruction errors can be computed using the Euclidean distance between the original instance $\boldsymbol{x}_r^{(j)}$ and the reconstructed counterpart, $\hat{\boldsymbol{x}}_r^{(j)}$, as follows:

$$e_r^{(j)} = ||\boldsymbol{x}_r^{(j)} - \hat{\boldsymbol{x}}_r^{(j)}||_2^2, \tag{6.3}$$

where $e_r^{(j)}$ is the reconstruction error of the r-th instance on j-th local partition, with $r = 1, \ldots, n_j$, and n_j denotes the number of instances in j-th local cluster. Therefore, for each local partition j a reconstruction error vector is given by

$$\boldsymbol{e}^{(j)} = \begin{bmatrix} e_1^{(j)} \\ e_2^{(j)} \\ \vdots \\ e_{n_j}^{(j)} \end{bmatrix}. \tag{6.4}$$

The threshold is then computed by the following equation:

$$\theta^{(j)} = mean(\boldsymbol{e}^{(j)}) + std(\boldsymbol{e}^{(j)}), \tag{6.5}$$

where $mean(\boldsymbol{e}^{(j)})$ and $std(\boldsymbol{e}^{(j)})$ stand for the mean and standard deviation applied on the reconstruction error associated with the j-th local partition. While we have chosen this method for computing the threshold, a variety of other thresholding approaches exist in the literature (YANG *et al.*, 2019a).

Algorithm 4 delineates the steps for training a LAE.

Algorithm 4: Local autoencoder for novelty detection

Data: Training dataset $\mathcal{X}$, number of local partitions K
Result: Local deep autoencoder
Local partitioning:
1. Train a clustering-based model to segment the training data into K subsets.
2. Define the K local subsets: $\{\mathcal{X}_1, \ldots, \mathcal{X}_K\}$.

Local training:
for $j \leftarrow 1$ **to** K **do**
1. Train j-th local autoencoder (LAE_j) on data subset $\mathcal{X}_j$.
2. Reconstruct all instances in $\mathcal{X}_j$ using LAE_j.
3. Compute reconstruction errors for all instances.
4. Compute the local novelty threshold $\theta^{(j)}$ from the error distribution.

end

6.2.1 A Word About Benefits and Challenges

Local learning models divide the data into subsets, focusing on smaller, more representative portions of the dataset (DRUMOND *et al.*, 2022). This localized training strategy may enhance the model's ability to generalize to new, unseen data, as each local autoencoder learns features specific to its subset. In addition, training on individual data partitions facilitates a localized analysis of the learned representations, thereby improving the overall interpretability of the model's internal features. Regarding *time complexity*, LAEs, due to its multiple model nature, can optimize the training process by parallelization. Parallelizing the training process across subsets not only contributes to scalability but also may result in a significant reduction in training time. This reduction is crucial for handling large-scale datasets and resource-intensive models, making the use of LAEs a practical and efficient approach. LAEs also facilitate the adaptation of loss functions to the characteristics of each subset during training. This flexibility allows researchers to tailor loss functions, enhancing the model's performance in specific scenarios and improving its adaptability to diverse data characteristics. Nevertheless, it should be mentioned that LAE depends directly on the data partition quality, as discussed in Section 3.2.3.

6.2.2 Complexity Analysis

Evaluating the computational efficiency of LAE is important to identify the scenarios where it can be computationally cost effective when compared to global AE. Similarly to LKPCA, the LAE offers potential advantages when compared to global AE, as training multiple AEs on small data subsets can reduce overall computational demands. The time and memory complexity of a global AE depend on the network architecture (layers and neurons), input data size, and training process. The training time complexity is $O(EnLh)$, where E is the number of epochs, n is the number of data points, L is the number of layers, and h is the number of neurons per layer. This complexity arises from the forward and backward computations. The memory complexity is $O(Lh)$, dominated by the weights. Table 13 summarizes the computational complexity of the global AE and LAE.

Table 13 – Complexity analysis of global and local autoencoders.

Method	**Time**		**Memory**
	(Training)	**(Prediction)**	**(Trained Model)**
Global AE	$O(EnLh)$	$O(Lh)$	$O(Lh)$
Local AE	$O(InKd) + \sum_{j=1}^{K} O(E_j n_j L_j h_j)$	$O(Kd) + O(L_{j^*} h_{j^*})$	$O(Kd) + \sum_{j=1}^{K} O(L_j h_j)$

Source: the author (2024).

LAE training has two stages. The first stage partitions the dataset into K clusters using K-means, a process with a time complexity of $O(InKd)$, where I is the number of K-means iterations, n is the number of data points, K is the number of clusters, and d is the input dimensionality. The second stage involves independently training a local AEs per cluster. The training complexity for cluster j is given by $O(E_j n_j L_j h_j)$, where E_j is the number of training epochs for the j-th LAE, n_j is the number of data points in cluster C_j, L_j is the number of layers in the j-th LAE, and h_j is the number of neurons per layer. Consequently, the total training complexity for all K local AEs is $\sum_{j=1}^{K} O(E_j n_j L_j h_j)$. The memory complexity of the trained LAE model is $O(Kd) + \sum_{j=1}^{K} O(L_j h_j)$, accounting for the K cluster centroids and the weights of all local autoencoders. Training an independent AE on each cluster offers computational advantages, especially for small clusters, as the complexity of training an AE depends linearly on the number of samples. For prediction, the overall time complexity is $O(Kd + L_{j^*} h_{j^*})$. This involves finding the appropriate cluster for a new data point and performing a forward pass through the corresponding LAE, where j^* indicates the index of the chosen LAE model.

6.3 Simulations and Results

In this section, we detail the experiments and simulations conducted to validate and assess the effectiveness of LAEs. Through a quantitative and qualitative analysis of these simulations, we highlight the strengths and potential areas for improvement of the proposed method. All algorithms were implemented using Python language[1]. We have evaluated the LAE considering several datasets from UCR repository (CHEN *et al.*, 2015). Since these datasets are inherently binary classification problems, we adapted them for OCC and novelty detection. The majority class was designated as the normal class, while the minority class was treated as the novel class. The characteristics of these datasets are detailed in Table 14.

Table 14 – UCR Datasets used in LAE simulations.

Dataset	#instances	#features	#normal data	#novelties	#train	#train data	#test
BeetleFly	40	512	20	20	20	10	20
Coffee	56	286	29	27	28	14	28
ECGFiveDays	884	136	442	442	23	14	861
ItalyPowerDemand	1096	24	549	547	67	34	1029
ProximalPhalanxOutlineCorrect	891	80	605	286	600	194	291
Wine	111	234	54	57	57	30	54
Wafer	7164	152	6402	762	1000	97	6164

Source: the author (2024).

6.3.1 LAE Classification Performance

In this first simulation, we compare the performance of LAEs against global AE and the 1-NN baseline model. Although comparing an MTCC against OCC methods may seem unfair, 1-NN was selected as a reference because it is the top-performing model for these datasets according to the UCR repository (CHEN *et al.*, 2015). Our goal is to assess the ND effectiveness of LAE using classification performance indices, including accuracy, precision, recall and F1-score.

Table 15 summarizes the simulation results. It shows that the 1-NN algorithm outperforms both LAE and GAE in F1-score across all datasets. This is expected, as 1-NN is a multi-class classifier, unlike GAE and LAE that rely on data from a single class. The observed variations in performance can be attributed to the inherent differences in the methodologies of 1-NN and AE-based methods.

[1] The code can be accessed in the following link: <https://github.com/renanfonteles/local-autoencoder> .

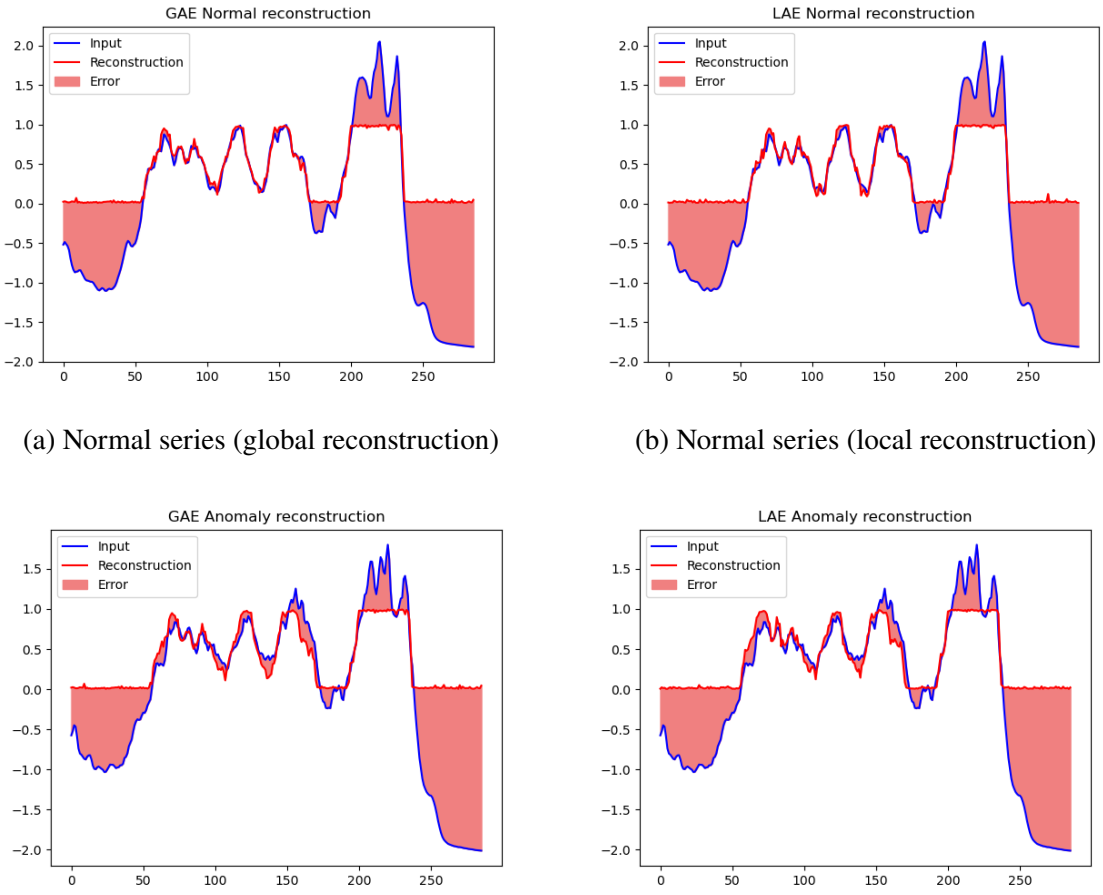
Table 15 – LAE performance on UCR time series datasets.

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^2	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

Source: the author (2024).

While LAE does not achieve the same high score as 1-NN, it consistently outperforms GAE in most scenarios. Comparing the two AE-based models directly, LAE indicates better ability to balance precision and recall. This is particularly evident on datasets *BeetleFly*, *ItalyPowerDemand* and *Wafer*, where the LAE shows a significant performance gain. To assess if there is any disparity in reconstruction outcomes between GAE and LAE, a comparison was made by plotting the difference between their respective reconstructions, as depicted in Figure 39. The visual examination reveals slight distinctions in the reconstructed data. However, it becomes challenging to quantitatively evaluate the extent of these differences and their implications in terms of novelty detection.

Figure 39 – Global vs. local autoencoder time series reconstruction on the Coffee dataset.



(a) Normal series (global reconstruction)

(b) Normal series (local reconstruction)

(c) Novel series (global reconstruction)

(d) Novel series (local reconstruction)

Source: the author (2024).

6.3.2 Detection Performance vs. Number of Local Partitions

For the second set of simulations, we investigated the impact of increasing the number of local partitions on LAE performance. We trained several LAE models on the *Wafer* dataset, systematically varying the number of local partitions (K) from 2 to 9, and then analyzed the evolution of their performance indices with the increment in the number of partitions. This experiment was conducted to highlight a specific characteristic of local models: their performance is significantly influenced by how clusters are formed. Specifically, it emphasizes that a sufficient number of samples must be available within each cluster for the local models to be trained effectively.

Table 16 presents the results of the simulation comparing the GAE and LAE (with a varying number of local partitions) across several performance indices.

Table 16 – LAE performance vs. number of local partitions K 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

Source: the author (2024).

From the results, the 1-NN algorithm demonstrates high performance, with accuracy, precision, recall, and F1-score exceeding 99%. This highlights the effectiveness of instance-based methods for this dataset. In contrast, GAE model performs significantly worse, with accuracy of 76.25%. Its lower scores for precision, recall, and F1-score indicate a notable number of misclassifications. The LAE models, with the number of local partitions (K) ranging from 2 to 9, consistently outperformed the GAE. Their performance indices showed considerable improvement: accuracy rom 82.28% to 90.07%, precision from 80.56% to 89.33%, recall from 99.38% to 99.61%, and F1-score from 89.03% to 94.14%. These results suggest that increasing the number of partitions can improve the LAE's detection performance. However, this comes with a trade-off: as granularity increases, the amount of data available for each local model decreases, potentially reducing its ability to learn effectively with the available data.

6.4 Summary

In this chapter, we introduced a novel AE-based local learning technique. The methodology is structured in two primary stages: employing a clustering algorithm to define local data partitions and subsequently training local AEs. Through experimentation on benchmark datasets, we observe its potential in detecting novel data in comparison to traditional AEs. LAEs demonstrate competitive results and enhanced prediction performance. Although these preliminary findings suggest that LAEs are promising, further studies on large and more diverse datasets are essential to comprehensively assess their performance and efficacy.

7 CONCLUSION AND FURTHER WORK

This thesis begins with a systematic introduction to the principles of pattern classification and one-class classification, followed by a comprehensive literature review. Subsequently, it delves into the main concepts of the local learning paradigm, highlighting its relevance in ML and pattern classification fields and discussing recent studies and applications. Afterwards, the CLHP local learning framework is introduced in details, which underpins the development of three novel OCC methods, as cited below.

1. **Competitive** LKPCA, including:
    - LKPCA-WTA; and
    - LKPCA-WTA-MT.
2. **Cooperative** LKPCA, including:
    - LKPCA-AVG;
    - LKPCA-SW; and
    - LKPCA-ST.
3. Local autoencoder (LAE).

The first contribution of this thesis is the LKPCA, a method that combines the Hoffman's KPCA for novelty detection (HOFFMANN, 2007) with the CLHP framework (DRUMOND *et al.*, 2022). LKPCA is developed in two primary settings: competitive and cooperative. In the LKPCA *competitive* settings, LKPCA-WTA defines a single threshold, while LKPCA-WTA-MT employs a localized multithresholding strategy. In both *competitive* variants, out-of-sample prediction uses a WTA approach, where only one local model is selected for class prediction. For the *cooperative* setting, three variants are proposed: LKPCA-AVG, which defines a single threshold by averaging the reconstruction error from all local models; LKPCA-SW, which relies on similarity measures to define the weights applied for each local model output; and finally, LKPCA-ST, a stacking ensemble model built by concatenating the reconstruction errors for each local model as input data for training a *metamodel* that learns the decision rule for detecting novelties. Simulations reveal that both LKPCA approaches yield promising results, though *competitive* LKPCA methods show significant sensitivity to thresholding issues, particularly in LKPCA-WTA. The effectiveness of LKPCA varies depending on the formation of local partitions, a critical factor for real-world applications. Despite these limitations, the LKPCA variants have surpassed KPCA in performance across several datasets. For instance, LKPCA-ST has consistently produced the best results in terms of *accuracy* and ROC-AUC.

This thesis also introduces a novel AE-based local learning technique, referred to as the LAE. This method operates in two stages: first, a clustering algorithm partitions the data into distinct local subsets; second, a localized AE is trained on each of these partitions. Through experimentation on benchmark time series ND datasets, LAE demonstrated a higher performance in identifying novel data compared to traditional AEs in several datasets, including *BeetleFly*, *Wafer* and *ItalyPowerDemand*. This observation underscores the model's potential and efficacy in discerning unique patterns within diverse datasets. The simulations showed that increasing the number of local partitions in LAE can enhance its performance, offering a notable improvement over its global counterpart.

In summary, this thesis underscores the potential of local modeling paradigm to construct nonlinear decision boundaries using a set of localized trained models. LKPCA and LAE demonstrated competitive performance in specific use cases compared to global counterparts. The findings suggest that this local learning paradigm is a powerful approach for distributed learning, as it can both enhance detection performance and provide a more computationally efficient training process.

7.1 Further Work

Considering avenues for future research, numerous promising ideas emerge:

1. Develop specific cluster validation metrics tailored for CLHP and local paradigms, as traditional clustering indices appear unsuitable for determining the optimal K.
2. Employ the CLHP framework on other OCC models and conduct comparative performance analyzes with respect to state-of-the-art local modeling approaches (MOON *et al.*, 2022; HE *et al.*, 2023; WANG *et al.*, 2023; ZHAO *et al.*, 2023).
3. Explore *cooperative* learning approaches using LAEs (YUAN *et al.*, 2018; WANG *et al.*, 2020; CHI *et al.*, 2020; GOU *et al.*, 2023).
4. Explore extensions to the LKPCA and LAE methods to operate in online, growing, and adaptive scenarios (DING *et al.*, 2010; GAUTAM *et al.*, 2020; JIANG *et al.*, 2020).
5. Explore the integration of federated learning with LKPCA and LAE, focusing on applications in distributed learning environments (NGUYEN *et al.*, 2021).
6. Extend LKPCA and LAE to regional classifier (RC) setting (DRUMOND *et al.*, 2022).
7. Conduct experimentation using LKPCAs and LAEs on a broader set of datasets, including well-established benchmarks such as ADBench (HAN *et al.*, 2022).

REFERENCES

ABDALLAH, L.; BADARNA, M.; KHALIFA, W.; YOUSEF, M. MultiKOC: Multi-One-Class Classifier Based K-Means Clustering. **Algorithms**, v. 14, n. 5, p. 134, Apr. 2021. ISSN 1999-4893. Available at: <https://www.mdpi.com/1999-4893/14/5/134>.

AGGARWAL, C. C.; YU, P. S. Outlier Detection for High Dimensional Data. **Journal of Information & Knowledge Management**, v. 19, 2001.

AGUAYO, L.; BARRETO, G. A. Novelty Detection in Time Series Using Self-Organizing Neural Networks: A Comprehensive Evaluation. **Neural Processing Letters**, Aug. 2017. ISSN 1370-4621, 1573-773X. Available at: <http://link.springer.com/10.1007/s11063-017-9679-2>.

AKCAY, S.; ATAPOUR-ABARGHOUEI, A.; BRECKON, T. P. GANomaly: Semi-supervised Anomaly Detection via Adversarial Training. In: JAWAHAR, C. V.; LI, H.; MORI, G.; SCHINDLER, K. (Ed.). **Computer Vision – ACCV 2018**. Cham: Springer International Publishing, 2019. v. 11363, p. 622–637. ISBN 978-3-030-20892-9 978-3-030-20893-6. Series Title: Lecture Notes in Computer Science. Available at: <http://link.springer.com/10.1007/978-3-030-20893-6_39>.

AKRAMI, H.; JOSHI, A. A.; LI, J.; AYDöRE, S.; LEAHY, R. M. A robust variational autoencoder using beta divergence. **Knowledge-Based Systems**, v. 238, p. 107886, Feb. 2022. ISSN 09507051. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0950705121010534>.

ALAM, S.; SONBHADRA, S. K.; AGARWAL, S.; NAGABHUSHAN, P. One-class support vector classifiers: A survey. **Knowledge-Based Systems**, v. 196, p. 105754, 2020. ISSN 09507051. Publisher: Elsevier B.V. Available at: <https://doi.org/10.1016/j.knosys.2020.105754>.

ALBERTINI, M. K.; MELLO, R. F. D. A self-organizing neural network for detecting novelties. In: **Proceedings of the 2007 ACM symposium on Applied computing**. Seoul Korea: ACM, 2007. p. 462–466. ISBN 978-1-59593-480-2. Available at: <https://dl.acm.org/doi/10.1145/1244002.1244110>.

ALGHUSHAIRY, O.; ALSINI, R.; SOULE, T.; MA, X. A review of local outlier factor algorithms for outlier detection in big data streams. **Big Data and Cognitive Computing**, v. 5, n. 1, p. 1–24, 2021. ISSN 25042289.

ALPAYDIN, E.; JORDAN, M. I. Local linear perceptrons for classification. **IEEE Transactions on Neural Networks**, v. 7, n. 3, p. 788–792, 1996. ISSN 10459227.

AMINI, N.; ZHU, Q. Fault detection and diagnosis with a novel source-aware autoencoder and deep residual neural network. **Neurocomputing**, v. 488, p. 618–633, Jun. 2022. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231221017446>.

AN, P.; WANG, Z.; ZHANG, C. Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection. **Information Processing & Management**, v. 59, n. 2, p. 102844, Mar. 2022. ISSN 03064573. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0306457321003162>.

ANDRESINI, G.; APPICE, A.; MALERBA, D. Autoencoder-based deep metric learning for network intrusion detection. **Information Sciences**, v. 569, p. 706–727, Aug. 2021. ISSN 00200255. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S002002552100462X>.

ARMANFARD, N.; REILLY, J. P.; KOMEILI, M. Local Feature Selection for Data Classification. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 38, n. 6, p. 1217–1227, Jun. 2016. ISSN 0162-8828, 2160-9292. Available at: <http://ieeexplore.ieee.org/document/7265078/>.

BANERJEE, A.; DAS, A.; BEHRA, S.; BHATTACHARJEE, D.; SRINIVASAN, N. T.; NASIPURI, M.; DAS, N. Carp-DCAE: Deep convolutional autoencoder for carp fish classification. **Computers and Electronics in Agriculture**, v. 196, p. 106810, May 2022. ISSN 01681699. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0168169922001272>.

BARNETT, V.; LEWIS, T. **Outliers in Statistical Data**. 3. ed. Chichester, England: Wiley, 1994. (Wiley Series in Probability and Statistics). ISBN 978-0-471-93094-5.

BARRETO, G.; MOTA, J.; SOUZA, L.; FROTA, R.; AGUAYO, L. Condition monitoring of 3G cellular networks through competitive neural models. **IEEE Transactions on Neural Networks**, v. 16, n. 5, p. 1064–1075, Sep. 2005. ISSN 1045-9227, 1941-0093. Available at: <http://ieeexplore.ieee.org/document/1510710/>.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York, NY, USA: Springer, 2006. (Information science and statistics). ISBN 978-0387-31073-2.

BLÁZQUEZ-GARCÍA, A.; CONDE, A.; MORI, U.; LOZANO, J. A. A Review on Outlier/Anomaly Detection in Time Series Data. **ACM Computing Surveys**, v. 54, n. 3, 2021. ISSN 15577341. ArXiv: 2002.04236.

BOUKERCHE, A.; ZHENG, L.; ALFANDI, O. Outlier Detection: Methods, Models, and Classification. **ACM Computing Surveys**, v. 53, n. 3, 2020. ISSN 15577341.

BRADLEY, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. **Pattern Recognition**, v. 30, n. 7, p. 1145–1159, Jul. 1997. ISSN 00313203. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0031320396001422>.

BREUNIG, M. M.; KRIEGEL, H.-P.; NG, R. T.; SANDER, J. Lof: Identifying density-based local outliers. **SIGMOD Rec.**, Association for Computing Machinery, New York, NY, USA, v. 29, n. 2, p. 93–104, may 2000. ISSN 0163-5808. Available at: <https://doi.org/10.1145/335191.335388>.

BROWN, R. G.; HWANG, P. Y. **Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises**. 4ed.. ed. New York, NY, USA: Wiley, 2012. ISBN 978-0-470-60969-9.

CARREÑO, A.; INZA, I.; LOZANO, J. A. Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. **Artificial Intelligence Review**, v. 53, n. 5, p. 3575–3594, Jun. 2020. ISSN 0269-2821, 1573-7462. Available at: <http://link.springer.com/10.1007/s10462-019-09771-y>.

CERRI, R.; BARROS, R. C.; CARVALHO, A. C. D. Hierarchical multi-label classification using local neural networks. **Journal of Computer and System Sciences**, v. 80, n. 1, p.

39–56, Feb. 2014. ISSN 00220000. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0022000013000718>.

CHAN, F. T.; WANG, Z.; PATNAIK, S.; TIWARI, M.; WANG, X.; RUAN, J. Ensemble-learning based neural networks for novelty detection in multi-class systems. **Applied Soft Computing**, v. 93, p. 106396, Aug. 2020. ISSN 15684946. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1568494620303367>.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Survey of Anomaly Detection. **ACM Computing Survey (CSUR)**, v. 41, n. 3, p. 1–72, 2009. ISSN 0360-0300. ArXiv: 1011.1669v3 ISBN: 0818663359.

CHAO, M. A.; ADEY, B. T.; FINK, O. Implicit supervision for fault detection and segmentation of emerging fault types with Deep Variational Autoencoders. **Neurocomputing**, v. 454, p. 324–338, Sep. 2021. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231221007001>.

CHARTE, D.; CHARTE, F.; JESUS, M. J. D.; HERRERA, F. An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges. **Neurocomputing**, v. 404, p. 93–107, Sep. 2020. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S092523122030624X>.

CHEN, G.; ZHANG, X.; WANG, Z. J.; LI, F. Robust support vector data description for outlier detection with noise or uncertain data. **Knowledge-Based Systems**, v. 90, p. 129–137, Dec. 2015. ISSN 09507051. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0950705115003640>.

CHEN, J.; SATHE, S.; AGGARWAL, C.; TURAGA, D. Outlier detection with autoencoder ensembles. **Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017**, p. 90–98, 2017. ISBN: 9781611974874.

CHEN, J.; WU, D.; ZHAO, Y.; SHARMA, N.; BLUMENSTEIN, M.; YU, S. Fooling intrusion detection systems using adversarially autoencoder. **Digital Communications and Networks**, v. 7, n. 3, p. 453–460, Aug. 2021. ISSN 23528648. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S2352864820302868>.

CHEN, X.; MA, Y.; CHANG, L.; CHEN, G. Variable Sparse Multiple Kernels Learning for Novelty Detection. In: LEE, G. (Ed.). **Advances in Automation and Robotics, Vol.1**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. v. 122, p. 115–124. ISBN 978-3-642-25552-6 978-3-642-25553-3. Series Title: Lecture Notes in Electrical Engineering. Available at: <http://link.springer.com/10.1007/978-3-642-25553-3_16>.

CHEN, Y.; KEOGH, E.; HU, B.; BEGUM, N.; BAGNALL, A.; MUEEN, A.; BATISTA, G. **The UCR Time Series Classification Archive**. 2015.

CHI, H.; XIA, H.; ZHANG, L.; ZHANG, C.; TANG, X. Competitive and collaborative representation for classification. **Pattern Recognition Letters**, v. 132, p. 46–55, Apr. 2020. ISSN 01678655. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0167865518302642>.

CUI, P.; ZHAN, C.; YANG, Y. Improved nonlinear process monitoring based on ensemble KPCA with local structure analysis. **Chemical Engineering Research and Design**, v. 142, p. 355–368, 2019. ISSN 02638762. Publisher: Institution of Chemical Engineers. Available at: <https://doi.org/10.1016/j.cherd.2018.12.028>.

DANG, X. H.; MICENKOVÁ, B.; ASSENT, I.; NG, R. T. Local Outlier Detection with Interpretation. In: **Advanced Information Systems Engineering**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. v. 7908, p. 304–320. ISBN 978-3-642-38708-1 978-3-642-38709-8. Series Title: Lecture Notes in Computer Science. Available at: <http://link.springer.com/10.1007/978-3-642-40994-3_20>.

DAS, R.; GOLATKAR, A.; AWATE, S. P. Sparse Kernel PCA for Outlier Detection. In: **2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)**. Orlando, FL: IEEE, 2018. p. 152–157. ISBN 978-1-5386-6805-4. Available at: <https://ieeexplore.ieee.org/document/8614055/>.

DAVIS, J.; GOADRICH, M. The relationship between Precision-Recall and ROC curves. In: **Proceedings of the 23rd international conference on Machine learning - ICML '06**. Pittsburgh, Pennsylvania: ACM Press, 2006. p. 233–240. ISBN 978-1-59593-383-6. Available at: <http://portal.acm.org/citation.cfm?doid=1143844.1143874>.

DENG, X.; WANG, L. Modified kernel principal component analysis using double-weighted local outlier factor and its application to nonlinear process monitoring. **ISA Transactions**, v. 72, p. 218–228, Jan. 2018. ISSN 00190578. Publisher: Elsevier Ltd. Available at: <http://dx.doi.org/10.1016/j.isatra.2017.09.015>.

DENG, X.; ZHONG, N.; WANG, L. Nonlinear Multimode Industrial Process Fault Detection Using Modified Kernel Principal Component Analysis. **IEEE Access**, v. 5, p. 23121–23132, 2017. ISSN 21693536.

DING, C.; ZHAO, J.; SUN, S. Concept Drift Adaptation for Time Series Anomaly Detection via Transformer. **Neural Processing Letters**, v. 55, n. 3, p. 2081–2101, Jun. 2023. ISSN 1370-4621, 1573-773X. Available at: <https://link.springer.com/10.1007/s11063-022-11015-0>.

DING, H.; DING, K.; ZHANG, J.; WANG, Y.; GAO, L.; LI, Y.; CHEN, F.; SHAO, Z.; LAI, W. Local outlier factor-based fault detection and evaluation of photovoltaic system. **Solar Energy**, v. 164, n. November 2017, p. 139–148, 2018. ISSN 0038092X. Publisher: Elsevier. Available at: <https://doi.org/10.1016/j.solener.2018.01.049>.

DING, M.; ANTANI, S.; JAEGER, S.; XUE, Z.; CANDEMIR, S.; KOHLI, M.; THOMA, G. Local-global classifier fusion for screening chest radiographs. In: COOK, T. S.; ZHANG, J. (Ed.). Orlando, Florida, United States: SPIE, 2017. p. 101380A. Available at: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2252459>.

DING, M.; TIAN, Z.; XU, H. Adaptive kernel principal component analysis. **Signal Processing**, v. 90, n. 5, p. 1542–1553, 2010. ISSN 01651684. Publisher: Elsevier. Available at: <http://dx.doi.org/10.1016/j.sigpro.2009.11.001>.

DONG, X.; TAYLOR, C. J.; COOTES, T. F. Defect Classification and Detection Using a Multitask Deep One-Class CNN. **IEEE Transactions on Automation Science and Engineering**, v. 19, n. 3, p. 1719–1730, Jul. 2022. ISSN 1545-5955, 1558-3783. Available at: <https://ieeexplore.ieee.org/document/9537583/>.

DOSHI-VELEZ, F.; KIM, B. Towards A Rigorous Science of Interpretable Machine Learning. In: . arXiv, 2017. ArXiv:1702.08608 [cs, stat]. Available at: <http://arxiv.org/abs/1702.08608>.

DRUMOND, R. B.; ALBUQUERQUE, R. F.; BARRETO, G. A.; SOUZA, A. H. Pattern classification based on regional models. **Applied Soft Computing**, v. 129, p. 109592, Nov. 2022. ISSN 15684946. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S156849462200641X>.

DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2017. Available at: <http://archive.ics.uci.edu/ml>.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2. ed. New York, NY, USA: Wiley-Interscience, 2000. Hardcover. ISBN 0471056693.

EDGEWORTH, F. XLI. *On discordant observations*. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, v. 23, n. 143, p. 364–375, Apr. 1887. ISSN 1941-5982, 1941-5990. Available at: <https://www.tandfonline.com/doi/full/10.1080/14786448708628471>.

ERFANI, S. M.; RAJASEGARAR, S.; KARUNASEKERA, S.; LECKIE, C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. **Pattern Recognition**, v. 58, p. 121–134, Oct. 2016. ISSN 00313203. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0031320316300267>.

FAN, Y.; WEN, G.; LI, D.; QIU, S.; LEVINE, M. D.; XIAO, F. Video anomaly detection and localization via Gaussian Mixture Fully Convolutional Variational Autoencoder. **Computer Vision and Image Understanding**, v. 195, p. 102920, Jun. 2020. ISSN 10773142. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1077314218302674>.

FANG, J.; XIE, Z.; CHENG, H.; FAN, B.; XU, H.; LI, P. Anomaly detection of diabetes data based on hierarchical clustering and CNN. **Procedia Computer Science**, v. 199, p. 71–78, 2022. ISSN 18770509. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1877050922000102>.

FARIA, E. R.; GONçALVES, I. J.; CARVALHO, A. C. de; GAMA, J. Novelty detection in data streams. **Artificial Intelligence Review**, v. 45, n. 2, p. 235–269, 2016. ISSN 15737462. Publisher: Springer Netherlands.

FROTA, R. A.; BARRETO, G. A.; MOTA, J. C. Anomaly detection in mobile communication networks using the self-organizing map. **Journal of Intelligent & Fuzzy Systems**, IOS Press, v. 18, p. 493–500, 2007. 5.

GAO, J.; HU, W.; ZHANG, Z.; ZHANG, X.; WU, O. RKOF: Robust Kernel-Based Local Outlier Detection. n. 60825204, p. 270–283, 2011. Available at: <http://link.springer.com/10.1007/978-3-642-20847-8_23>.

GAUTAM, C.; BALAJI, R.; SUDHARSAN, K.; TIWARI, A.; AHUJA, K. Localized Multiple Kernel Learning for Anomaly Detection: One-class Classification. **Knowledge-Based Systems**, v. 165, p. 241–252, May 2018. ISSN 09507051. ArXiv: 1805.07892 Publisher: Elsevier B.V. Available at: <http://www.ijabmr.org/text.asp?2019/9/4/226/268943>.

GAUTAM, C.; TIWARI, A.; SURESH, S.; AHUJA, K. Adaptive Online Learning With Regularized Kernel for One-Class Classification. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, p. 1–16, 2020. ISSN 2168-2216, 2168-2232. Available at: <https://ieeexplore.ieee.org/document/8692763/>.

GEDDES, T. A.; KIM, T.; NAN, L.; BURCHFIELD, J. G.; YANG, J. Y. H.; TAO, D.; YANG, P. Autoencoder-based cluster ensembles for single-cell RNA-seq data analysis. **BMC Bioinformatics**, v. 20, n. S19, p. 660, Dec. 2019. ISSN 1471-2105. Available at: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3179-5>.

GIACINTO, G.; PERDISCI, R.; RIO, M. D.; ROLI, F. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. **Information Fusion**, v. 9, n. 1, p. 69–82, Jan. 2008. ISSN 15662535. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1566253506000765>.

GÔLO, M. P. S.; SOUZA, M. C. D.; ROSSI, R. G.; REZENDE, S. O.; NOGUEIRA, B. M.; MARCACINI, R. M. One-class learning for fake news detection through multimodal variational autoencoders. **Engineering Applications of Artificial Intelligence**, v. 122, p. 106088, Jun. 2023. ISSN 09521976. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0952197623002725>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>. ISBN 978-0262035613.

GOU, J.; QIU, W.; YI, Z.; SHEN, X.; ZHAN, Y.; OU, W. Locality constrained representation-based K-nearest neighbor classification. **Knowledge-Based Systems**, v. 167, p. 38–52, Mar. 2019. ISSN 09507051. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0950705119300152>.

GOU, J.; XIONG, X.; WU, H.; DU, L.; ZENG, S.; YUAN, Y.; OU, W. Locality-constrained weighted collaborative-competitive representation for classification. **International Journal of Machine Learning and Cybernetics**, v. 14, n. 2, p. 363–376, Feb. 2023. ISSN 1868-8071, 1868-808X. Available at: <https://link.springer.com/10.1007/s13042-021-01461-y>.

GRATI, N.; BEN-HAMADOU, A.; HAMMAMI, M. Learning local representations for scalable RGB-D face recognition. **Expert Systems with Applications**, v. 150, 2020. ISSN 09574174. Publisher: Elsevier Ltd.

GRUHL, C.; SICK, B.; TOMFORDE, S. Novelty detection in continuously changing environments. **Future Generation Computer Systems**, v. 114, p. 138–154, 2021. ISSN 0167739X. Publisher: Elsevier B.V. Available at: <https://doi.org/10.1016/j.future.2020.07.037>.

GU, X.; AKOGLU, L.; RINALDO, A. Statistical Analysis of Nearest Neighbor Methods for Anomaly Detection. 2019.

GUO, Z.-X.; SHUI, P.-L. Anomaly Based Sea-Surface Small Target Detection Using K-Nearest Neighbor Classification. **IEEE Transactions on Aerospace and Electronic Systems**, v. 56, n. 6, p. 4947–4964, Dec. 2020. ISSN 0018-9251, 1557-9603, 2371-9877. Available at: <https://ieeexplore.ieee.org/document/9149659/>.

GUPTA, M.; GAO, J.; AGGARWAL, C. C.; HAN, J. Outlier Detection for Temporal Data: A Survey. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 9, p. 2250–2267, 2014. ISSN 10414347.

HAMROUNI, I.; LAHDHIRI, H.; ABDELLAFOU, K. B.; TAOUALI, O. Fault detection of uncertain nonlinear process using reduced interval kernel principal component analysis

(RIKPCA). **The International Journal of Advanced Manufacturing Technology**, v. 106, n. 9-10, p. 4567–4576, Feb. 2020. ISSN 0268-3768, 1433-3015. Available at: <http://link.springer.com/10.1007/s00170-019-04889-3>.

HAN, S.; HU, X.; HUANG, H.; JIANG, M.; ZHAO, Y. Adbench: Anomaly detection benchmark. In: **Neural Information Processing Systems (NeurIPS)**. New Orleans, LA, USA: NIPS, 2022.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. New York, NY: Springer Series in Statistics, 2009.

HAWKINS, D. M. **Identification of Outliers**. New York, NY, USA: Chapman and Hall, 1980.

HE, Y.; YAN, D.; CHEN, F. Hierarchical federated learning with local model embedding. **Engineering Applications of Artificial Intelligence**, v. 123, p. 106148, Aug. 2023. ISSN 09521976. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0952197623003329>.

HILAL, W.; GADSDEN, S. A.; YAWNEY, J. Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. **Expert Systems with Applications**, v. 193, p. 116429, 2022. ISSN 09574174. Publisher: Elsevier Ltd. Available at: <https://doi.org/10.1016/j.eswa.2021.116429>.

HODGE, V. J.; AUSTIN, J. A Survey of Outlier Detection Methodologies. **Artificial Intelligence Review**, v. 22, n. 2, p. 85–126, Oct. 2004. ISSN 0269-2821. Available at: <http://link.springer.com/10.1007/s10462-004-4304-y>.

HOFFMANN, H. Kernel PCA for novelty detection. **Pattern Recognition**, v. 40, n. 3, p. 863–874, 2007. ISSN 00313203.

HOFFMANN, H.; SCHAAL, S.; VIJAYAKUMAR, S. Local Dimensionality Reduction for Non-Parametric Regression. **Neural Processing Letters**, v. 29, n. 2, p. 109–131, Apr. 2009. ISSN 1370-4621, 1573-773X. Available at: <http://link.springer.com/10.1007/s11063-009-9098-0>.

HU, X.; XIAO, Z.; LIU, D.; TANG, Y.; MALIK, O. P.; XIA, X. KPCA and AE Based Local-Global Feature Extraction Method for Vibration Signals of Rotating Machinery. **Mathematical Problems in Engineering**, v. 2020, p. 1–17, Jun. 2020. ISSN 1024-123X, 1563-5147. Available at: <https://www.hindawi.com/journals/mpe/2020/5804509/>.

HU, X.; XIE, C.; FAN, Z.; DUAN, Q.; ZHANG, D.; JIANG, L.; WEI, X.; HONG, D.; LI, G.; ZENG, X.; CHEN, W.; WU, D.; CHANUSSOT, J. Hyperspectral Anomaly Detection Using Deep Learning: A Review. **Remote Sensing**, v. 14, n. 9, p. 1973, Apr. 2022. ISSN 2072-4292. Available at: <https://www.mdpi.com/2072-4292/14/9/1973>.

HUANG, D.; YI, Z.; PU, X. A new local PCA-SOM algorithm. **Neurocomputing**, v. 71, n. 16-18, p. 3544–3552, 2008. ISSN 09252312.

HUANG, H.; QIN, H.; YOO, S.; YU, D. Local anomaly descriptor. In: **Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12**. New York, New York, USA: ACM Press, 2012. v. 1, p. 405. ISBN 978-1-4503-1156-4. Available at: <http://dl.acm.org/citation.cfm?doid=2396761.2396815>.

IDÉ, T.; PAPADIMITRIOU, S.; VLACHOS, M. Computing Correlation Anomaly Scores Using Stochastic Nearest Neighbors. In: **Seventh IEEE International Conference on Data Mining (ICDM 2007)**. Omaha, NE, USA: IEEE, 2007. p. 523–528. ISBN 978-0-7695-3018-5. Available at: <http://ieeexplore.ieee.org/document/4470284/>.

ISOMURA, T.; TOYOIZUMI, T. A Local Learning Rule for Independent Component Analysis. **Scientific Reports**, v. 6, n. 1, p. 28073, Jun. 2016. ISSN 2045-2322. Available at: <https://www.nature.com/articles/srep28073>.

JACOBS, R. A.; JORDAN, M. I.; NOWLAN, S. J.; HINTON, G. E. Adaptive Mixtures of Local Experts. **Neural Computation**, v. 3, n. 1, p. 79–87, 1991. ISSN 0899-7667.

JANA, D.; PATIL, J.; HERKAL, S.; NAGARAJAIAH, S.; DUENAS-OSORIO, L. CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction. **Mechanical Systems and Signal Processing**, v. 169, p. 108723, Apr. 2022. ISSN 08883270. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0888327021010414>.

JIANG, B.; XU, F.; HUANG, Y.; YANG, C.; HUANG, W.; XIA, J. Adaptive Adversarial Latent Space for Novelty Detection. **IEEE Access**, v. 8, p. 205088–205098, 2020. ISSN 2169-3536. Available at: <https://ieeexplore.ieee.org/document/9256333/>.

JOVE, E.; CASADO-VARA, R.; CASTELEIRO-ROCA, J.-L.; PéREZ, J. A. M.; VALE, Z.; CALVO-ROLLE, J. L. A hybrid intelligent classifier for anomaly detection. **Neurocomputing**, v. 452, p. 498–507, Sep. 2021. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231220316490>.

JÚNIOR, A. H. S.; BARRETO, G. A.; CORONA, F. Regional models: A new approach for nonlinear system identification via clustering of the self-organizing map. **Neurocomputing**, v. 147, p. 31–46, Jan. 2015. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231214007085>.

KANG, Z.; ZHAO, X.; PENG, C.; ZHU, H.; ZHOU, J. T.; PENG, X.; CHEN, W.; XU, Z. Partition level multiview subspace clustering. **Neural Networks**, v. 122, p. 279–288, Feb. 2020. ISSN 08936080. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0893608019303326>.

KATZEF, M.; CULLEN, A. C.; ALPCAN, T.; LECKIE, C. Generative Adversarial Networks for anomaly detection on decentralised data. **Annual Reviews in Control**, v. 53, p. 329–337, 2022. ISSN 13675788. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1367578821000778>.

KHAN, S. S.; MADDEN, M. G. One-class classification: Taxonomy of study and review of techniques. **Knowledge Engineering Review**, v. 29, n. 3, p. 345–374, 2014. ISSN 14698005. ArXiv: 1312.0049.

KIM, H.; PARK, J.; MIN, K.; HUH, K. Anomaly Monitoring Framework in Lane Detection With a Generative Adversarial Network. **IEEE Transactions on Intelligent Transportation Systems**, v. 22, n. 3, p. 1603–1615, Mar. 2021. ISSN 1524-9050, 1558-0016. Available at: <https://ieeexplore.ieee.org/document/9005205/>.

KORYCKI, L.; KRAWCZYK, B. **Concept Drift Detection from Multi-Class Imbalanced Data Streams**. arXiv, 2021. ArXiv:2104.10228 [cs]. Available at: <http://arxiv.org/abs/2104.10228>.

KRAWCZYK, B.; WOŹNIAK, M.; CYGANEK, B. Clustering-based ensembles for one-class classification. **Information Sciences**, v. 264, p. 182–195, Apr. 2014. ISSN 00200255. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0020025513008694>.

KUMPULAINEN, P.; HÄTÖNEN, K. Local anomaly detection for mobile network monitoring. **Information Sciences**, v. 178, n. 20, p. 3840–3859, Oct. 2008. ISSN 00200255. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0020025508001886>.

LAMPINEN, J.; OJA, E. Self-Organizing Maps for Spatial and Temporal AR Models. **Proc. 6th SCIA, Scandinavian Conference on Image Analysis**, p. 120–127, 1989.

LATECKI, L. J.; LAZAREVIC, A.; POKRAJAC, D. Outlier Detection with Kernel Density Functions. In: PERNER, P. (Ed.). **Machine Learning and Data Mining in Pattern Recognition**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. v. 4571, p. 61–75. ISBN 978-3-540-73498-7 978-3-540-73499-4. ISSN: 0302-9743, 1611-3349 Series Title: Lecture Notes in Computer Science. Available at: <http://link.springer.com/10.1007/978-3-540-73499-4_6>.

LAU, C.; GHOSH, K.; HUSSAIN, M.; HASSAN, C. C. Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS. **Chemometrics and Intelligent Laboratory Systems**, v. 120, p. 1–14, Jan. 2013. ISSN 01697439. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0169743912002080>.

LEE, C.-K.; CHEON, Y.-J.; HWANG, W.-Y. Least Squares Generative Adversarial Networks-Based Anomaly Detection. **IEEE Access**, v. 10, p. 26920–26930, 2022. ISSN 2169-3536. Available at: <https://ieeexplore.ieee.org/document/9732473/>.

LEE, S.; KIM, H. J.; KIM, S. B. Dynamic dispatching system using a deep denoising autoencoder for semiconductor manufacturing. **Applied Soft Computing**, v. 86, p. 105904, Jan. 2020. ISSN 15684946. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1568494619306854>.

LI, D.; WANG, H.; ZHOU, J. Novelty Detection for Multimode Process Using GANs with Learning Disentangled Representation. In: **2020 Chinese Control And Decision Conference (CCDC)**. Hefei, China: IEEE, 2020. p. 2536–2541. ISBN 978-1-72815-855-6. Available at: <https://ieeexplore.ieee.org/document/9164262/>.

LI, P.; PEI, Y.; LI, J. A comprehensive survey on design and application of autoencoder in deep learning. **Applied Soft Computing**, v. 138, p. 110176, May 2023. ISSN 15684946. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1568494623001941>.

LINDEMANN, B.; MASCHLER, B.; SAHLAB, N.; WEYRICH, M. A survey on anomaly detection for technical systems using LSTM networks. **Computers in Industry**, v. 131, p. 103498, Oct. 2021. ISSN 01663615. Publisher: Elsevier B.V. Available at: <https://doi.org/10.1016/j.compind.2021.103498>.

LIU, C.; TIAN, Y.; TANG, J.; DANG, S.; CHEN, G. A novel local differential privacy federated learning under multi-privacy regimes. **Expert Systems with Applications**, v. 227, p. 120266, Oct. 2023. ISSN 09574174. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0957417423007686>.

LIU, K.; LI, A.; WEN, X.; CHEN, H.; YANG, P. Steel Surface Defect Detection Using GAN and One-Class Classifier. In: **2019 25th International Conference on Automation**

and Computing (ICAC). IEEE, 2019. p. 1–6. ISBN 978-1-86137-665-7. Issue: September. Available at: <https://ieeexplore.ieee.org/document/8895110/>.

LIU, P.; SUN, X.; HAN, Y.; HE, Z.; ZHANG, W.; WU, C. Arrhythmia classification of LSTM autoencoder based on time series anomaly detection. **Biomedical Signal Processing and Control**, v. 71, p. 103228, Jan. 2022. ISSN 17468094. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1746809421008259>.

LIU, Y.; LIN, Y.; XIAO, Q.; HU, G.; WANG, J. Self-adversarial variational autoencoder with spectral residual for time series anomaly detection. **Neurocomputing**, v. 458, p. 349–363, Oct. 2021. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231221009346>.

LIU, Z.; ZHANG, L. A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings. **Measurement**, v. 149, p. 107002, Jan. 2020. ISSN 02632241. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0263224119308681>.

LIU, Z. Y.; XU, L. Topological local principal component analysis. **Neurocomputing**, v. 55, n. 3-4, p. 739–745, 2003. ISSN 09252312.

LOYOLA-GONZÁLEZ, O.; MEDINA-PÉREZ, M. A.; CHOO, K.-K. R. A Review of Supervised Classification based on Contrast Patterns: Applications, Trends, and Challenges. **Journal of Grid Computing**, v. 18, n. 4, p. 797–845, Dec. 2020. ISSN 1570-7873, 1572-9184. Available at: <https://link.springer.com/10.1007/s10723-020-09526-y>.

MACCIÓ, D.; CERVELLERA, C. Local Models for data-driven learning of control policies for complex systems. **Expert Systems with Applications**, v. 39, n. 18, p. 13399–13408, 2012. ISSN 09574174. ISBN: 0106475835.

MAES, K.; SALENS, W.; FEREMANS, G.; SEGHER, K.; FRANçOIS, S. Anomaly detection in long-term tunnel deformation monitoring. **Engineering Structures**, v. 250, p. 113383, Jan. 2022. ISSN 01410296. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0141029621014930>.

MALEKI, S.; MALEKI, S.; JENNINGS, N. R. Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. **Applied Soft Computing**, v. 108, p. 107443, Sep. 2021. ISSN 15684946. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1568494621003665>.

MALHOTRA, P.; VIG, L.; SHROff, G.; AGARWAL, P. Long Short Term Memory Networks for Anomaly Detection in Time Series. In: **European Symposiym on Artificial Neural Networks, Computational Intelligence**. Bruges, Belgium: ESANN, 2015.

MANEVITZ, L. M.; YOUSEF, M. One-Class SVMs for Document Classification. 2001.

MARKOU, M.; SINGH, S. Novelty detection: A review - Part 1: Statistical approaches. **Signal Processing**, v. 83, n. 12, p. 2481–2497, 2003. ISSN 01651684.

MARQUES, H. O.; SWERSKY, L.; SANDER, J.; CAMPELLO, R. J. G. B.; ZIMEK, A. On the evaluation of outlier detection and one-class classification: a comparative study of algorithms, model selection, and ensembles. **Data Mining and Knowledge Discovery**, v. 37, n. 4, p. 1473–1517, Jul. 2023. ISSN 1384-5810, 1573-756X. Available at: <https://link.springer.com/10.1007/s10618-023-00931-x>.

MATSUI, T.; YAMAMOTO, K.; OGATA, J. Anomaly detection for wind turbine damaged due to lightning strike. **Electric Power Systems Research**, v. 209, n. April, p. 107918, 2022. ISSN 03787796. Publisher: Elsevier B.V. Available at: <https://doi.org/10.1016/j.epsr.2022.107918>.

MCGILL, R.; TUKEY, J. W.; LARSEN, W. A. Variations of box plots. **The American Statistician**, Taylor & Francis, v. 32, n. 1, p. 12–16, 1978.

MESARCIK, M.; RANGUELOVA, E.; BOONSTRA, A.-J.; NIEUWPOORT, R. V. V. Improving novelty detection using the reconstructions of nearest neighbours. **Array**, v. 14, p. 100182, Jul. 2022. ISSN 25900056. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S2590005622000388>.

MIELE, E. S.; BONACINA, F.; CORSINI, A. Deep anomaly detection in horizontal axis wind turbines using Graph Convolutional Autoencoders for Multivariate Time series. **Energy and AI**, v. 8, p. 100145, May 2022. ISSN 26665468. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S2666546822000076>.

MOHAMMADI-GHAZI, R.; MARZOUK, Y. M.; BÜYÜKÖZTÜRK, O. Conditional classifiers and boosted conditional Gaussian mixture model for novelty detection. **Pattern Recognition**, v. 81, p. 601–614, Sep. 2018. ISSN 00313203. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0031320318301122>.

MOON, J. H.; YU, J. H.; SOHN, K. A. An ensemble approach to anomaly detection using high- and low-variance principal components. **Computers and Electrical Engineering**, v. 99, n. May 2020, p. 107773, 2022. ISSN 00457906. Publisher: Elsevier Ltd. Available at: <https://doi.org/10.1016/j.compeleceng.2022.107773>.

MUHAMMAD, G.; ALSHEHRI, F.; KARRAY, F.; SADDIK, A. E.; ALSULAIMAN, M.; FALK, T. H. A comprehensive survey on multimodal medical signals fusion for smart healthcare systems. **Information Fusion**, v. 76, n. November 2020, p. 355–375, 2021. ISSN 15662535. Publisher: Elsevier B.V. Available at: <https://doi.org/10.1016/j.inffus.2021.06.007>.

MUNIR, M.; SIDDIQUI, S. A.; DENGEL, A.; AHMED, S. DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. **IEEE Access**, v. 7, p. 1991–2005, 2019. ISSN 2169-3536. Available at: <https://ieeexplore.ieee.org/document/8581424/>.

MUÑOZ, A.; MURUZÁBAL, J. Self-organizing maps for outlier detection. **Neurocomputing**, v. 18, n. 1-3, p. 33–60, 1998. ISSN 09252312.

MURPHY, K. **Machine Learning: A Probabilistic Perspective**. MIT Press, 2021. ISBN 9780262044660. Available at: <https://books.google.com.br/books?id=dAhkzQEACAAJ>.

NASSIF, A. B.; TALIB, M. A.; NASIR, Q.; DAKALBAB, F. M. Machine Learning for Anomaly Detection: A Systematic Review. **IEEE Access**, v. 9, p. 78658–78700, 2021. ISSN 21693536.

NGUYEN, D. C.; DING, M.; PATHIRANA, P. N.; SENEVIRATNE, A.; LI, J.; POOR, H. V. Federated Learning for Internet of Things: A Comprehensive Survey. **IEEE Communications Surveys & Tutorials**, v. 23, n. 3, p. 1622–1658, 2021. ISSN 1553-877X, 2373-745X. ArXiv:2104.07914 [eess]. Available at: <http://arxiv.org/abs/2104.07914>.

NICHOLAUS, I. T.; PARK, J. R.; JUNG, K.; LEE, J. S.; KANG, D.-K. Anomaly Detection of Water Level Using Deep Autoencoder. **Sensors**, v. 21, n. 19, p. 6679, Oct. 2021. ISSN 1424-8220. Available at: <https://www.mdpi.com/1424-8220/21/19/6679>.

OZA, P.; PATEL, V. M. One-Class Convolutional Neural Network. **IEEE Signal Processing Letters**, v. 26, n. 2, p. 277–281, Feb. 2019. ISSN 1070-9908, 1558-2361. Available at: <https://ieeexplore.ieee.org/document/8586962/>.

PARK, B. H.; OH, S. Y.; KIM, I. J. Face alignment using a deep neural network with local feature learning and recurrent regression. **Expert Systems with Applications**, v. 89, p. 66–80, 2017. ISSN 09574174. Publisher: Elsevier Ltd. Available at: <http://dx.doi.org/10.1016/j.eswa.2017.07.018>.

PARK, C. H. Outlier and anomaly pattern detection on data streams. **Journal of Supercomputing**, v. 75, n. 9, p. 6118–6128, 2019. ISSN 15730484. Publisher: Springer US. Available at: <https://doi.org/10.1007/s11227-018-2674-1>.

PARK, J.; JUNG, Y. G.; TEOH, A. B. J. Discriminative Multi-level Reconstruction under Compact Latent Space for One-Class Novelty Detection. In: **2020 25th International Conference on Pattern Recognition (ICPR)**. Milan, Italy: IEEE, 2021. p. 7095–7102. ISBN 978-1-72818-808-9. Available at: <https://ieeexplore.ieee.org/document/9413248/>.

PAULAUSKAS, N.; BAGDONAS, Ą. F. Local outlier factor use for the network flow anomaly detection: Local outlier factor use for the network flow anomaly detection. **Security and Communication Networks**, v. 8, n. 18, p. 4203–4212, Dec. 2015. ISSN 19390114. Available at: <https://onlinelibrary.wiley.com/doi/10.1002/sec.1335>.

PAVLIDOU, M.; ZIOUTAS, G. Kernel Density Outlier Detector. In: AKRITAS, M. G.; LAHIRI, S. N.; POLITIS, D. N. (Ed.). **Topics in Nonparametric Statistics**. New York, NY: Springer New York, 2014. v. 74, p. 241–250. ISBN 978-1-4939-0568-3 978-1-4939-0569-0. Series Title: Springer Proceedings in Mathematics & Statistics. Available at: <https://link.springer.com/10.1007/978-1-4939-0569-0_22>.

PEARSON, K. LIII. *On lines and planes of closest fit to systems of points in space*. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, v. 2, n. 11, p. 559–572, Nov. 1901. ISSN 1941-5982, 1941-5990. Available at: <https://www.tandfonline.com/doi/full/10.1080/14786440109462720>.

PERERA, P.; OZA, P.; PATEL, V. M. One-Class Classification: A Survey. p. 1–19, 2021. ArXiv: 2101.03064. Available at: <http://arxiv.org/abs/2101.03064>.

PERES, R.; PEDREIRA, C. A new local–global approach for classification. **Neural Networks**, v. 23, n. 7, p. 887–891, Sep. 2010. ISSN 08936080. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0893608010000936>.

PIMENTEL, M. A.; CLIFTON, D. A.; CLIFTON, L.; TARASSENKO, L. A review of novelty detection. **Signal Processing**, v. 99, p. 215–249, 2014. ISSN 01651684. Publisher: Elsevier. Available at: <http://dx.doi.org/10.1016/j.sigpro.2013.12.026>.

PISANO, B.; FANNI, A.; TEIXEIRA, C. A.; DOURADO, A. Application of self organizing map to identify nocturnal epileptic seizures. **12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization, WSOM 2017 - Proceedings**, 2017. ISBN: 9781509066384.

POKRAJAC, D.; LAZAREVIC, A.; LATECKI, L. J. Incremental Local Outlier Detection for Data Streams. In: **2007 IEEE Symposium on Computational Intelligence and Data Mining**. Honolulu, HI, USA: IEEE, 2007. p. 504–515. ISBN 978-1-4244-0705-7. Available at: <http://ieeexplore.ieee.org/document/4221341/>.

PORWAL, U.; MUKUND, S. **Credit Card Fraud Detection in e-Commerce: An Outlier Detection Approach**. arXiv, 2019. ArXiv:1811.02196 [cs, stat]. Available at: <http://arxiv.org/abs/1811.02196>.

PU, Z.; CABRERA, D.; BAI, Y.; LI, C. A One-Class Generative Adversarial Detection Framework for Multifunctional Fault Diagnoses. **IEEE Transactions on Industrial Electronics**, v. 69, n. 8, p. 8411–8419, Aug. 2022. ISSN 0278-0046, 1557-9948. Available at: <https://ieeexplore.ieee.org/document/9529073/>.

PUKELSHEIM, F. The three sigma rule. **The American Statistician**, Taylor & Francis, v. 48, n. 2, p. 88–91, 1994.

RASHEED, K.; QAYYUM, A.; GHALY, M.; AL-FUQAHA, A.; RAZI, A.; QADIR, J. Explainable, trustworthy, and ethical machine learning for healthcare: A survey. **Computers in Biology and Medicine**, v. 149, p. 106043, Oct. 2022. ISSN 00104825. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0010482522007569>.

RAYANA, S. **ODDS Library**. 2016. Available at: <https://odds.cs.stonybrook.edu>.

ROKACH, L. **Ensemble Learning: Pattern Classification Using Ensemble Methods**. 2. ed. World Scientific Publishing, 2019. ISBN 9811201951,9789811201950. Available at: <http://gen.lib.rus.ec/book/index.php?md5=1FF413BBE2CC3D9B25A6013E14422FDD>.

ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics**, v. 20, p. 53–65, Nov. 1987. ISSN 03770427. Available at: <https://linkinghub.elsevier.com/retrieve/pii/0377042787901257>.

SALEHI, M.; ARYA, A.; PAJOUM, B.; OTOOFI, M.; SHAEIRI, A.; ROHBAN, M. H.; RABIEE, H. R. ARAE: Adversarially robust training of autoencoders improves novelty detection. **Neural Networks**, v. 144, p. 726–736, Dec. 2021. ISSN 08936080. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0893608021003646>.

SARIKAYA, A.; KILIç, B. G.; DEMIRCI, M. RAIDS: Robust autoencoder-based intrusion detection system model against adversarial attacks. **Computers & Security**, v. 135, p. 103483, Dec. 2023. ISSN 01674048. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0167404823003930>.

SCHLACHTER, P.; LIAO, Y.; YANG, B. Deep One-Class Classification Using Intra-Class Splitting. In: **2019 IEEE Data Science Workshop (DSW)**. IEEE, 2019. p. 100–104. ISBN 978-1-72810-708-0. Available at: <https://ieeexplore.ieee.org/document/8755576/>.

SCHÖLKOPF, B.; PLATT, J. C.; SHAWE-TAYLOR, J.; SMOLA, A. J.; WILLIAMSON, R. C. Estimating the support of a high-dimensional distribution. **Neural Computation**, v. 13, n. 7, p. 1443–1471, 2001. ISSN 08997667.

SCHÖLKOPF, B.; SMOLA, A. J.; MÜLLER, K.-R. Kernel principal component analysis. In: **Proceedings of the 7th International Conference on Artificial Neural Networks**. Berlin, Heidelberg: Springer-Verlag, 1997. (ICANN '97), p. 583–588. ISBN 3540636315.

SHAHREZA, M. L.; MOAZZAMI, D.; MOSHIRI, B.; DELAVAR, M. Anomaly detection using a self-organizing map and particle swarm optimization. **Scientia Iranica**, v. 18, n. 6, p. 1460–1468, Dec. 2011. ISSN 10263098. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1026309811001751>.

SHENG, S.; GUO, X.; YU, K.; WU, X. Local causal structure learning with missing data. **Expert Systems with Applications**, v. 238, p. 121831, Mar. 2024. ISSN 09574174. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0957417423023333>.

SHIN, H. J.; EOM, D. H.; KIM, S. S. One-class support vector machines - An application in machine fault detection and classification. **Computers and Industrial Engineering**, v. 48, n. 2, p. 395–408, 2005. ISSN 03608352.

SONG, B.; SHI, H.; MA, Y.; WANG, J. Multisubspace Principal Component Analysis with Local Outlier Factor for Multimode Process Monitoring. **Industrial & Engineering Chemistry Research**, v. 53, n. 42, p. 16453–16464, Oct. 2014. ISSN 0888-5885, 1520-5045. Available at: <https://pubs.acs.org/doi/10.1021/ie502344q>.

SOUIDEN, I.; BRAHMI, Z.; TOUMI, H. A survey on outlier detection in the context of stream mining: Review of existing approaches and recommadations. **Advances in Intelligent Systems and Computing**, v. 557, p. 372–383, 2017. ISSN 21945357. ISBN: 9783319534794.

STEHMAN, S. V. Selecting and interpreting measures of thematic classification accuracy. **Remote Sensing of Environment**, v. 62, n. 1, p. 77–89, Oct. 1997. ISSN 00344257. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0034425797000837>.

STUDIAWAN, H.; SOHEL, F. Anomaly detection in a forensic timeline with deep autoencoders. **Journal of Information Security and Applications**, v. 63, p. 103002, Dec. 2021. ISSN 22142126. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S2214212621002076>.

SU, S.; XIAO, L.; RUAN, L.; GU, F.; LI, S.; WANG, Z.; XU, R. An Efficient Density-Based Local Outlier Detection Approach for Scattered Data. **IEEE Access**, v. 7, p. 1006–1020, 2019. ISSN 2169-3536. Available at: <https://ieeexplore.ieee.org/document/8572736/>.

TAKHANOV, R.; ABYLKAIROV, Y. S.; TEZEKBAYEV, M. Autoencoders for a manifold learning problem with a jacobian rank constraint. **Pattern Recognition**, v. 143, p. 109777, Nov. 2023. ISSN 00313203. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0031320323004752>.

TANG, J.; FENG, H. Robust collaborative clustering approach with adaptive local structure learning. **Knowledge-Based Systems**, v. 251, p. 109222, Sep. 2022. ISSN 09507051. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0950705122006086>.

TANG, X.-m.; YUAN, R.-x.; CHEN, J. Outlier Detection in Energy Disaggregation Using Subspace Learning and Gaussian Mixture Model. **International Journal of Control and Automation**, v. 8, n. 8, p. 161–170, Aug. 2015. ISSN 20054297, 20054297. Available at: <http://article.nadiapub.com/IJCA/vol8_no8/17.pdf>.

TAX, D. M.; DUIN, R. P. Support vector domain description. **Pattern Recognition Letters**, v. 20, n. 11-13, p. 1191–1199, 1999. ISSN 01678655.

TAX, D. M.; DUIN, R. P. Support Vector Data Description. **Machine Learning**, v. 54, n. 1, p. 45–66, Jan. 2004. ISSN 0885-6125. Available at: <http://link.springer.com/10.1023/B:MACH.0000008084.60811.49>.

TAX, D. M. J. **One-class classification: Concept learning in the absence of counter-examples**. Phd Thesis (PhD Thesis) — Technische Universiteit Delft, 2001. Available at: <http://proquest.umi.com/pqdweb?did=728104171&Fmt=2&clientId=36097&RQT=309&VName=PQD>.

TIMM, F.; BARTH, E. Novelty detection for the inspection of light-emitting diodes. **Expert Systems with Applications**, v. 39, n. 3, p. 3413–3422, Feb. 2012. ISSN 09574174. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0957417411013388>.

TSAI, D.-M.; JEN, P.-H. Autoencoder-based anomaly detection for surface defect inspection. **Advanced Engineering Informatics**, v. 48, p. 101272, Apr. 2021. ISSN 14740346. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1474034621000276>.

TUKEY, J. W. **Exploratory Data Analysis**. Addison-Wesley Publishing Company, 1977. (Addison-Wesley series in behavioral science, v. 2). ISBN 9780201076165. Available at: <https://books.google.com.br/books?id=UT9dAAAAIAAJ>.

VOS, K.; PENG, Z.; JENKINS, C.; SHAHRIAR, M. R.; BORGHESANI, P.; WANG, W. Vibration-based anomaly detection using LSTM/SVM approaches. **Mechanical Systems and Signal Processing**, v. 169, p. 108752, Apr. 2022. ISSN 08883270. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0888327021010682>.

VUTTIPITTAYAMONGKOL, P.; ELYAN, E.; PETROVSKI, A. On the class overlap problem in imbalanced data classification. **Knowledge-Based Systems**, v. 212, p. 106631, Jan. 2021. ISSN 09507051. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0950705120307607>.

WANG, C.; GAO, H.; LIU, Z.; FU, Y. A New Outlier Detection Model Using Random Walk on Local Information Graph. **IEEE Access**, v. 6, p. 75531–75544, 2018. ISSN 2169-3536. Available at: <https://ieeexplore.ieee.org/document/8548614/>.

WANG, C.; WANG, Y.; ZHANG, Z.; WANG, Y. Face Tracking and Recognition via Incremental Local Sparse Representation. In: **2013 Seventh International Conference on Image and Graphics**. Qingdao, China: IEEE, 2013. p. 493–498. ISBN 978-0-7695-5050-3. Available at: <http://ieeexplore.ieee.org/document/6643722/>.

WANG, C.; ZHOU, H.; HAO, Z.; HU, S.; LI, J.; ZHANG, X.; JIANG, B.; CHEN, X. Network traffic analysis over clustering-based collective anomaly detection. **Computer Networks**, v. 205, n. January 2021, 2022. ISSN 13891286. Publisher: Elsevier B.V.

WANG, C. K.; TING, Y.; LIU, Y. H.; HARIYANTO, G. A novel approach to generate artificial outliers for support vector data description. **IEEE International Symposium on Industrial Electronics**, n. ISlE, p. 2202–2207, 2009. ISBN: 9781424443499.

WANG, H.; BAH, M. J.; HAMMAD, M. Progress in Outlier Detection Techniques: A Survey. **IEEE Access**, v. 7, p. 107964–108000, 2019. ISSN 21693536.

WANG, W.; WANG, C.; WANG, Z.; YUAN, M.; LUO, X.; KURTHS, J.; GAO, Y. Abnormal detection technology of industrial control system based on transfer learning. **Applied Mathematics and Computation**, v. 412, n. 6, p. 821–832, 2022. ISSN 00963003.

WANG, W.; WANG, M.; DONG, X.; LAN, L.; ZU, Q.; ZHANG, X.; WANG, C. Class-specific and self-learning local manifold structure for domain adaptation. **Pattern Recognition**, v. 142, p. 109654, Oct. 2023. ISSN 00313203. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0031320323003552>.

WANG, Z.; ZHU, Z.; LI, D. Collaborative and geometric multi-kernel learning for multi-class classification. **Pattern Recognition**, v. 99, p. 107050, Mar. 2020. ISSN 00313203. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0031320319303528>.

WEINGESSEL, A.; HORNIK, K. Local PCA algorithms. **IEEE Transactions on Neural Networks**, v. 11, n. 6, p. 1242–1250, 2000. ISSN 10459227.

WOŹNIAK, M.; nA, M. G.; CORCHADO, E. A survey of multiple classifier systems as hybrid systems. **Information Fusion**, v. 16, p. 3–17, Mar. 2014. ISSN 15662535. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S156625351300047X>.

WU, P.; GUO, L.; LOU, S.; GAO, J. Local and Global Randomized Principal Component Analysis for Nonlinear Process Monitoring. **IEEE Access**, v. 7, p. 25547–25562, 2019. ISSN 21693536. Publisher: IEEE.

WU, Q.; LU, W.; YAN, X. Process monitoring of nonlinear uncertain systems based on Part Interval Stacked Autoencoder and Support Vector Data Description. **Applied Soft Computing**, v. 129, p. 109570, Nov. 2022. ISSN 15684946. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1568494622006305>.

WU, W.; HUANG, T.; GONG, K. Ethical Principles and Governance Technology Development of AI in China. **Engineering**, v. 6, n. 3, p. 302–309, Mar. 2020. ISSN 20958099. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S2095809920300011>.

XIA, X.; PAN, X.; LI, N.; HE, X.; MA, L.; ZHANG, X.; DING, N. GAN-based anomaly detection: A review. **Neurocomputing**, v. 493, p. 497–535, Jul. 2022. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231221019482>.

XIAO, Y.; WANG, H.; XU, W. Model selection of Gaussian kernel PCA for novelty detection. **Chemometrics and Intelligent Laboratory Systems**, v. 136, p. 164–172, 2014. ISSN 18733239. Publisher: Elsevier B.V. Available at: <http://dx.doi.org/10.1016/j.chemolab.2014.05.015>.

XIAO, Y.; WANG, H.; XU, W.; ZHOU, J. L1 norm based KPCA for novelty detection. **Pattern Recognition**, v. 46, n. 1, p. 389–396, Jan. 2013. ISSN 00313203. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0031320312002877>.

YANG, L.; ZHAI, Y.; LI, Z. Deep learning for online AC False Data Injection Attack detection in smart grids: An approach using LSTM-Autoencoder. **Journal of Network and Computer Applications**, v. 193, p. 103178, Nov. 2021. ISSN 10848045. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1084804521001880>.

YANG, P.; WANG, D.; WEI, Z.; DU, X.; LI, T. An Outlier Detection Approach Based on Improved Self-Organizing Feature Map Clustering Algorithm. **IEEE Access**, v. 7, n. c, p. 115914–115925, 2019. ISSN 21693536. Publisher: IEEE.

YANG, Y.; HOU, C.; LANG, Y.; YUE, G.; HE, Y. One-Class Classification Using Generative Adversarial Networks. **IEEE Access**, v. 7, p. 37970–37979, 2019. ISSN 2169-3536. Available at: <https://ieeexplore.ieee.org/document/8673546/>.

YIN, S.; ZHU, X.; JING, C. Fault detection based on a robust one class support vector machine. **Neurocomputing**, v. 145, p. 263–268, Dec. 2014. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S092523121400681X>.

YOU, S.; CHO, B. H.; SHON, Y.-M.; SEO, D.-W.; KIM, I. Y. Semi-supervised automatic seizure detection using personalized anomaly detecting variational autoencoder with behind-the-ear EEG. **Computer Methods and Programs in Biomedicine**, v. 213, p. 106542, Jan. 2022. ISSN 01692607. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0169260721006167>.

YU, Q.; KAVITHA, M.; KURITA, T. Autoencoder framework based on orthogonal projection constraints improves anomalies detection. **Neurocomputing**, v. 450, p. 372–388, Aug. 2021. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231221005609>.

YUAN, H.; LI, X.; XU, F.; WANG, Y.; LAI, L. L.; TANG, Y. Y. A collaborative-competitive representation based classifier model. **Neurocomputing**, v. 275, p. 627–635, Jan. 2018. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231217315278>.

ZHAN, Y.; YIN, J. Robust local tangent space alignment via iterative weighted PCA. **Neurocomputing**, v. 74, n. 11, p. 1985–1993, May 2011. ISSN 09252312. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0925231211000397>.

ZHANG, C.; HU, D.; YANG, T. Anomaly detection and diagnosis for wind turbines using long short-term memory-based stacked denoising autoencoders and XGBoost. **Reliability Engineering & System Safety**, v. 222, p. 108445, Jun. 2022. ISSN 09518320. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0951832022001107>.

ZHANG, F.; FU, Q.; LIU, Y.; LI, X. Component-aware generative autoencoder for structure hybrid and shape completion. **Graphical Models**, v. 129, p. 101185, Oct. 2023. ISSN 15240703. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1524070323000152>.

ZHANG, Y. Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM. **Chemical Engineering Science**, v. 64, n. 5, p. 801–811, 2009. ISSN 00092509.

ZHANG, Z.; DENG, X. Anomaly detection using improved deep SVDD model with data structure preservation. **Pattern Recognition Letters**, v. 148, p. 1–6, Aug. 2021. ISSN 01678655. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0167865521001598>.

ZHAO, M.; ZHAN, C.; WU, Z.; TANG, P. Semi-Supervised Image Classification Based on Local and Global Regression. **IEEE Signal Processing Letters**, v. 22, n. 10, p. 1666–1670, Oct. 2015. ISSN 1070-9908, 1558-2361. Available at: <http://ieeexplore.ieee.org/document/7091059/>.

ZHAO, W.; XU, Y.; LI, L.; YANG, H. Global-and-Local sampling for efficient hybrid task self-supervised learning. **Knowledge-Based Systems**, v. 268, p. 110479, May 2023. ISSN 09507051. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0950705123002290>.

ZHOU, J.; HE, Z.; LIU, X.; WANG, Y.; WANG, S.; LIU, Q. Transformed denoising autoencoder prior for image restoration. **Journal of Visual Communication and Image Representation**, v. 72, p. 102927, Oct. 2020. ISSN 10473203. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S1047320320301590>.

ZHOU, Y.; REN, H.; LI, Z.; PEDRYCZ, W. Anomaly detection based on a granular Markov model. **Expert Systems with Applications**, v. 187, n. August 2021, p. 115744, 2022. ISSN 09574174. Publisher: Elsevier Ltd. Available at: <https://doi.org/10.1016/j.eswa.2021.115744>.

ZIMEK, A.; FILZMOSER, P. There and back again: Outlier detection between statistical reasoning and data mining algorithms. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 8, n. 6, p. 1–26, 2018. ISSN 19424795.

ZIMEK, A.; SCHUBERT, E.; KRIEGEL, H. P. A survey on unsupervised outlier detection in high-dimensional numerical data. **Statistical Analysis and Data Mining**, v. 5, n. 5, p. 363–387, 2012. ISSN 19321872.

ZONG, B.; SONG, Q.; MIN, M. R.; CHENG, W.; LUMEZANU, C.; CHO, D.; CHEN, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: **International Conference on Learning Representations**. ICLR, 2018. Available at: <https://openreview.net/forum?id=BJJLHbb0->.