Complexity

Complex Methods Applied to Data Analysis, Processing, and Visualization

Lead Guest Editor: Jose Garcia-Rodriguez
Guest Editors: Anastassia Angelopoulou, David Tomás, and Andrew Lewis

Complex Methods Applied to Data Analysis, Processing, and Visualization

Complexity

Complex Methods Applied to Data Analysis, Processing, and Visualization

Lead Guest Editor: Jose Garcia-Rodriguez
Guest Editors: Anastassia Angelopoulou, David Tomás, and Andrew Lewis

Editorial Board

Contents

Hindawi
Complexity
Volume 2019, Article ID 9316123, 2 pages
https://doi.org/10.1155/2019/9316123

Editorial

Complex Methods Applied to Data Analysis, Processing, and Visualisation

Jose Garcia-Rodriguez,[1] **Anastasia Angelopoulou**,[2] **David Tomás**,[1] **and Andrew Lewis**[3]

[1]*University of Alicante, Alicante, Spain*
[2]*University of Westminster, London, UK*
[3]*Griffith University, Brisbane, Australia*

Correspondence should be addressed to Jose Garcia-Rodriguez; jgarcia@dtic.ua.es

Received 30 April 2019; Accepted 30 April 2019; Published 11 June 2019

The amount of data available every day is not only enormous but growing at an exponential rate. Over the last ten years there has been an increasing interest in using complex methods to analyse and visualise massive datasets, gathered from very different sources and including many different features: social networks, surveillance systems, smart cities, medical diagnosis systems, business information, cyberphysical systems, and digital media data. Nowadays, there are a large number of researchers working in complex methods to process, analyse, and visualise all this information, which can be applied to a wide variety of open problems in different domains. This special issue presents a collection of research papers addressing theoretical, methodological, and practical aspects of data processing, focusing on algorithms that use complex methods (e.g., chaos, genetic algorithms, cellular automata, neural networks, and evolutionary game theory) in a variety of domains (e.g., software engineering, digital media data, bioinformatics, health care, imaging and video, social networks, and natural language processing). A total of 27 papers were received from different research fields, but sharing a common feature: they presented complex systems that process, analyse, and visualise large amounts of data. After the review process, 8 papers were accepted for publication (around 30% of acceptance ratio).

These papers can be organised in different groups. The focus of the first group of articles is time series. The paper titled "LMC and SDL Complexity Measures: A Tool to Explore Time Series" by J. Piqueira and S. Mattos presented a generalisation of LMC (López-Ruiz, Mancini and Calbet) and SDL (Shiner, Davison and Landsberg) complexity measures, considering that the state of a system or process is represented by a continuous temporal series of a dynamical variable. As the two complexity measures are based on the calculation of informational entropy, an equivalent information source was defined by using partitions of the dynamical variable range. During the time intervals, the information associated with the measured dynamical variable was the seed to calculate instantaneous LMC and SDL measures. To show how the methodology worked generating indicators, two examples concerning meteorological data and economic data were presented and discussed. Another accepted work dealing with time series is "Improved Permutation Entropy for Measuring Complexity of Time Series under Noisy Condition", presented by Z. Chen et al. This paper proposes an improved permutation entropy method (IPE) as a tool to measure and analyse complexity of time series combining some advantages of previous modifications of PE. Its effectiveness was validated through both synthetic and experimental analysis, overcoming PE limitations such as its low performance under noisy conditions.

The second group of publications includes works dealing with sensing data and image recognition. The paper by J. Guo et al., entitled "Activity Feature Solving Based on TF-IDF for Activity Recognition in Smart Homes", presents an activity feature solving strategy based on TF-IDF. In smart homes based on the internet of things, daily activity recognition aims to know resident's daily activity in a noninvasive manner. The performance of daily activity recognition heavily depends on solving strategy of activity feature. However, the current common employed solving strategy based on statistical

information of individual activity does not support well the activity recognition. The proposal by Guo et al. exploits statistical information related to both individual activity and the whole of activities. Two distinct datasets were commissioned to mitigate the effects of coupling between datasets and sensor configuration. A number of traditional machine learning and deep learning techniques were evaluated to assess the performance of the method proposed for residents activity recognition. The second paper in this group is "MI-based Robust Waveform Design in Radar and Jammer Games", written by B. Wang et al. Due to the uncertainties of the radar target prior information in the actual scene, the waveform designed based on the radar target prior information cannot meet the needs of parameter estimation. To improve the performance of parameter estimation, Wang et al. presents a novel transmitted waveform design method under the hierarchical game model of radar and jammer. This approach maximises the mutual information between the radar target echo and the random target spectrum response. Another work in this group is "A Novel Semi-Supervised Learning Method Based on Fast Search and Density Peaks". This paper by F. Gao et al. address the problem of radar image recognition. Recognition algorithms achieve good classification results under the condition of sufficiently labelled samples, but labelled samples are scarce and costly to obtain. The main issue faced in this paper is how to use unlabelled samples to improve the performance of a recognition algorithm when the number of available labelled samples is limited. Unlike previous semisupervised learning methods, this work does not use unlabelled samples directly, but looks for safe and reliable samples before using them. The authors proposed two new semisupervised learning methods: one based on fast search and density peaks (S2DP) and the other on iterative S2DP. Finally, F. Zhao et al. propose in "Two-Phase Incremental Kernel PCA for Learning Massive or Online Datasets" a specific kernel PCA (KPCA) that can incorporate data into KPCA in an incremental way. This fact overcame typical drawbacks of KPCA when handling massive or online datasets. They tested their proposal in a synthesised dataset and in the classical MNIST database of handwritten digits images.

The last group of papers includes research in social impact domains such as economics and education. A. Herrero et al. present in "Hybrid Unsupervised Exploratory Plots: a Case Study of Analysing Foreign Direct Investment" a new visualisation technique, called HUEP. This proposal for descriptive data analysis combines the outputs of exploratory projection pursuit and clustering methods in a novel and informative way. As a case study, HUEP was validated in a real-world context for analysing the internationalisation strategy of companies by taking into account bilateral distance between home and host countries. As a multifaceted concept, distance encompasses multiple dimensions. Together with data from both the countries and the companies, various psychic distances were analyzed by means of HUEP, gaining deep knowledge about the internationalization strategy of large Spanish companies. Informative visualizations were obtained from the analyzed dataset, leading to useful business implications and decision making. The last paper in this issue, written by A. Hernández-Blanco et al., is focused on the educational domain. "A Systematic Review of Deep Learning Approaches to Educational Data Mining" surveys the research carried out in deep learning techniques applied to educational data mining (EDM) from its origins to the present day. The main goals of this study are to identify the EDM tasks that have benefited from deep learning techniques and those that are pending to be explored, to describe the main datasets used in this research area, to provide an overview of the key concepts, main architectures, and configurations of deep learning and its applications to EDM, and to discuss current state-of-the-art and future directions on this research field.

Conflicts of Interest

The authors declare that they have no conflicts of interest

Acknowledgments

This work has been funded by the Spanish Government TIN2016-76515-R grant for the COMBAHO project, supported with FEDER funds. We would like to thank Central University of Ecuador and in particular Jaime Salvador-Meneses and Zoila Ruiz for their participation and support in managing this special issue.

Jose Garcia-Rodriguez
Anastasia Angelopoulou
David Tomás
Andrew Lewis

Hindawi
Complexity
Volume 2019, Article ID 6271017, 14 pages
https://doi.org/10.1155/2019/6271017

Research Article

Hybrid Unsupervised Exploratory Plots: A Case Study of Analysing Foreign Direct Investment

Álvaro Herrero,[1] **Alfredo Jiménez**,[2] **and Secil Bayraktar**[3]

[1]*Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Ingeniería Civil, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos, Spain*
[2]*Department of Management, KEDGE Business School, 680 Cours de la Libération, 33405 Talence, Bordeaux, France*
[3]*Department of Corporate Social Responsibility and Human Resources, Toulouse Business School, 20 Boulevard Lascrosses, 31068 Toulouse, France*

Correspondence should be addressed to Álvaro Herrero; ahcosio@ubu.es

Received 20 December 2018; Accepted 29 January 2019; Published 2 June 2019

Guest Editor: Jose Garcia-Rodriguez

The curse of dimensionality has been an open issue for many years and still is, as finding nonobvious and previously unknown patterns in ever-increasing amounts of high-dimensional data is not an easy task. Advancing in descriptive data analysis, the present paper proposes Hybrid Unsupervised Exploratory Plots (HUEPs) as a new visualization technique to combine the outputs of Exploratory Projection Pursuit and Clustering methods in a novel and informative way. As a case study, HUEPs are validated in a real-world context for analysing the internationalization strategy of companies, by taking into account bilateral distance between home and host countries. As a multifaceted concept, distance encompasses multiple dimensions. Together with data from both the countries and the companies, various psychic distances are analysed by means of HUEPs, to gain deep knowledge of the internationalization strategy of large Spanish companies. Informative visualizations are obtained from the analysed dataset, leading to useful business implications and decision making.

1. Introduction

As it is well known, there are many different ways of analysing unlabeled datasets in order to gain knowledge about them. A key challenge in the analysis of high-dimensional unknown data is to identify the patterns that exist across dimensional boundaries. Such patterns may become visible if a change is made to the basis of the space; however, an *a priori* decision as to which basis will reveal most patterns requires prior knowledge of the unknown patterns. This is the main idea behind Exploratory Projection Pursuit (EPP) [1]. As opposed to feature selection, EPP lies within the feature extraction paradigm, as the resulting dimensions are combinations (it could be linear or nonlinear) of the original features in the dataset. On the other hand, clustering [2] consists in the organization of a collection of data items or patterns into clusters based on similarity. Hence, patterns within the same cluster are more similar to each other than they are to a pattern belonging to a different cluster.

Both EPP and clustering methods have been widely applied and combined in previous work. Although it had been stated [3, 4] that dimensionality reduction may identify dimensions that do not enhance the results of a subsequent clustering, some authors have contributed to the main combination stream where dimensionality reduction and clustering methods are sequenced, namely, "tandem" approach. Furthermore, [5] pointed out that "cluster analysis is one of the most frequent contexts in which principal components are derived in order to reduce dimensionality prior to the use of a different multivariate technique". That is the case of [6] where a canonical transformation is applied to data in order to optimize k-means clustering results on functional data. Additionally, [7] have optimized EPP as an initial step and then applied some clustering methods (hierarchical, partitional, and density-based) attaining interesting results. More recently, [8] have proposed Extreme Learning Machine for Joint Embedding and Clustering as a first step to preserve the manifold structure of the data in the original space

while maximizing the class separability of the data in the embedded space at the same time. Similarly, unsupervised dimensionality reduction methods are proposed in [9–12] for subsequent clustering through k-means.

In [13, 14] EPP and clustering methods are also combined but from a different perspective; dimensionality reduction models, implemented as neural networks, have been applied to add the output of some clustering methods to the obtained projections, though different labels, colours, and symbols. As a result, 2D projections are generated, enriched with information about the number of the cluster assigned to each sample. In those previous works, data from the cybersecurity and environmental fields have been analysed, respectively.

In a third alternative approach, clustering and EPP methods interact. Projection Pursuit Clustering [15] has been proposed to recover clusters in lower dimensional subspaces of the data by simultaneously performing dimension reduction and clustering. The proposed methodology finds both an optimal clustering for a subspace of given dimension and an optimal subspace for this clustering. In order to do that, clustering and projection pursuit methods are adapted in order to interchange information during execution. In a similar way, [16] has proposed a projection pursuit index to identify clusters and other structures in multivariate data, which is obtained from the variance decompositions of the data's one-dimensional projections.

Finally, some other independent uses of these two kinds of methods have been proposed so far [17]. In [18], k-means clustering method is applied in order to compare its results with those obtained from EPP. Thus, no combination of such methods is proposed but the comparison of their results, instead. On the other hand, both dimensionality reduction and clustering are independently combined in [19] for different tasks under the frame of a hybrid recommender system.

Differentiating from previous work, the present paper proposes the independent application of EPP methods on the one hand and clustering ones on the other. Complete results of the two of them are then combined, together with the glyph metaphor, in a novel way, called Hybrid Unsupervised Exploratory Plots (HUEPs), to support decision making. When compared to the above-mentioned previous work, it can be said that the present paper's proposal is a far more general and simpler approach where any EPP and clustering methods could be combined to generate informative and intuitive 3D visualizations of high-dimensional data. In order to validate this proposal, HUEPs are applied and compared in a case study where internationalization strategies from Spanish Multinational Enterprises (MNEs) are analysed.

In today's business context, management of international operations has been a focal element of company strategies. However, while investing abroad, companies face numerous challenges that, if not taken into consideration, may significantly risk the success of their investment. "Distance" emerges as a major challenge among those. A clear understanding of the differences between idiosyncrasies of the host country and home country may provide opportunities, on one hand, or, just the contrary, ignoring such differences may lead to disruption of the company's activities overseas. Referring to the fundamental role of distance between countries in the field of international management, [20] has even explicitly stated that "*essentially, international management is management of distances*".

Recent work [21] has conceptualized distance as a multifaceted construct. Along similar lines, various frameworks have investigated the multiple dimensions of distance that may influence a company's international operations. For example the well-known CAGE framework [22] proposed distance to constitute cultural, administrative, geographic, and economic facets. Another framework further posited ten dimensions to capture distance between nations [23]. Moreover, some researchers have dissected these dimensions of distance into further subdimensions with the aim of comprehending this phenomenon better. For instance, cultural distance was proposed to include six dimensions (power distance, uncertainty avoidance, individualism, masculinity, long-term orientation, and indulgence) by the influential work by [24, 25]. The vast number of citations proves that this framework and its operationalization as a single construct [26] became widely popular.

Nevertheless, some recent criticism has raised that an important type of distance, psychic distance, cannot entirely be captured or measured by the current cultural dimensions even though it is a crucial variable influencing managerial decisions in international business [27]. Psychic distance is an extensive framework that goes beyond culture and entails multiple dimensions of distance [28, 29]. It is useful to understand the context in which a manager's perceptions are formed while making a decision. Reference [28] suggests that six macro factors called psychic distance stimuli shape that context [30]. These factors measure the national differences between language, industrial development, social system, democracy, education, and religion [28, 31]. Previous studies have shown that these stimuli significantly impact market selection, performance, entry mode choice, Foreign Direct Investments (FDI), online internationalization, and trade flows [32].

On the other hand, even though researchers [33, 34] confer that combining multiple stimuli into one single construct is problematic in the sense that it may cause an inaccurate view that all components are equally significant, many studies still follow this aggregation approach. This paper, being aware of this potential problem, focuses on one particular stimulus concerning the political system differences between countries, namely, democracy distance, in order to avoid the probable confounding effects of the other stimuli.

While all stimuli may have an important role, we focus on democracy because previous research has emphasized the critical impact of political institutions on FDI decisions [35–39]. The democracy distance variable indicates the level of political rights, civil liberties and checks and balances existing in the country to prevent any opportunistic behavior by the local government to unilaterally modify the rules and laws [28].

According to what has been explained above, the challenging task of analysing the internationalization strategy of companies requires advanced data analysis tools. Up to now, little effort has been devoted to support decision makers with means of getting deep knowledge from such datasets. The

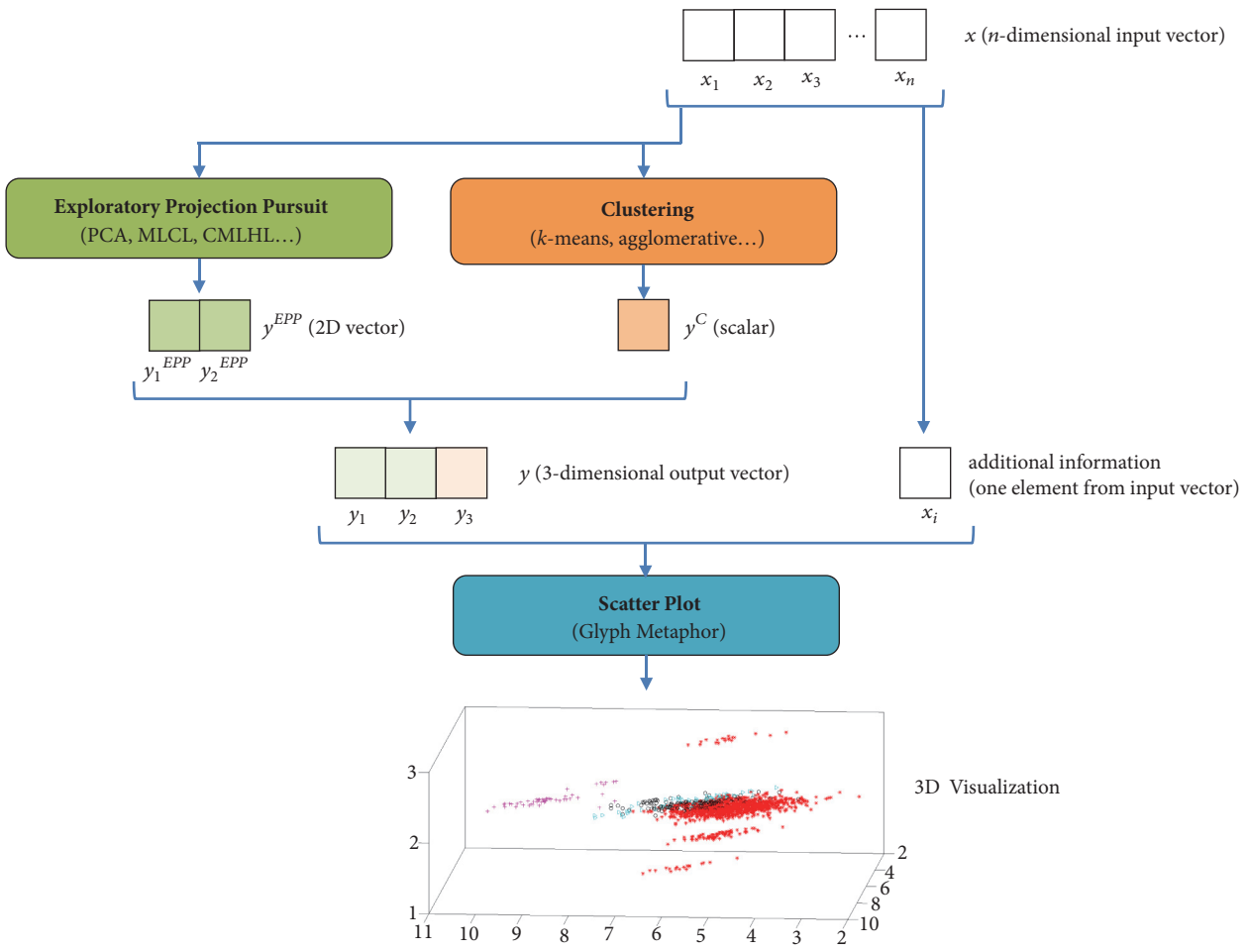


Figure 1: Process to obtain a HUEP.

present paper advances previous work by proposing a new visualization tool to ease the analysis of multidimensional datasets related to internationalization. The rest of this paper is organized as follows: the proposed HUEPs and their components are described in Section 2, the case study where HUEPs are validated is introduced in Section 3, together with its associated results that are presented in Section 4 and the conclusions of present study that are stated in Section 4.

2. Hybrid Unsupervised Exploratory Plots

As we humans are able to detect anomalies and to recognize different features or patterns through visual inspection, visualization techniques are a viable solution to information seeking. This idea is based on the ability to visualize high-dimensional datasets in a consistent and low-dimensional representation where those anomalies, features, or patterns can be identified. Such depiction of high-dimensional data through visual displays is not easy and cannot be performed immediately in most cases. The difficulty lies in converting raw big-size data into a graphical format that provides a useful insight into the visualized dataset [40]. As previously mentioned, Hybrid Unsupervised Exploratory Plots (HUEPs) are proposed as a new way of intuitively visualizing data within the field of descriptive datamining.

Visualization techniques have been widely covered in the literature; two of the most relevant works are [41, 42]. Among the wide variety of such techniques, a very popular one is scatter plots, which represent 2D or 3D data as points, with coordinates that correspond to their values. These plots still are one of the most popular and widely used visual representations for multidimensional data [43], due to their simplicity. However, there are some drawbacks, the two main ones being the required low dimensionality of the data to be displayed and the problem of overplotting.

A HUEP is proposed as a scatter plot where each data is considered as a 3D vector. These three-dimensional vectors are obtained from (raw) original data by means of an EPP method and a clustering one, according to what is shown in Figure 1.

As can be seen in Figure 1, a HUEP can be described as a mapping of vectors x onto vectors y in an output space. Vectors from the input space (n-dimensional, being $n \geq 3$) S_I are then mapped into a 3D output space S_O, according to H nonlinear transformation:

$$H\colon S_O \longrightarrow S_I, \quad \text{for any } x \in S_O \tag{1}$$

Resulting vectors y, from the S_O space, are defined as

$$y = (y_1, y_2, y_3) \tag{2}$$

y_1, y_2 are the output vectors of an EPP method (y_1^{EPP}, y_2^{EPP}) and y_3 the output (scalar) of a clustering method. Once obtained, output y vectors are then plotted in 3D scatter plots. Furthermore, the visualization of each vector is enriched thanks to the glyph metaphor, as can be seen in Section 3 and adding additional information from one of the input features (x_i). The widely used glyphs (or multidimensional icons) can be defined as graphical objects that are designed to convey multiple data values [44]. By using different symbols and colours, further information can be added to the 3D visualization of each data point.

Proposed HUEPs are hybrid as they combine both exploratory (dimensionality reduction) methods as well as clustering ones. On the other hand, they are unsupervised as both kinds of methods implement this kind of learning (no target class or value is provided to be reproduced).

The main steps to obtain HUEPs are described in the following subsections.

2.1. Exploratory Projection Pursuit. The well-known Exploratory Projection Pursuit (EPP) [1] was proposed as a method to identify structure in a given high-dimensional data. In the case of EPP, this general task is performed by projecting the data onto a low-dimensional subspace. By means of such projection, one can visually identify the structure of the dataset. As not all available projections reveal the data's structure in the same way, EPP defines an index aimed at measuring the "interestingness" of a projection, and then those projections that maximize that index are chosen.

As previously mentioned, EPP initially defines which indices represent interesting directions. When talking about projections, "interestingness" is usually linked to the fact that most projections give almost Gaussian distributions [45]. Consequently, in order to identify the most "interesting" features of the data, the directions generating projections as far from the Gaussian as possible should be found.

Once the most interesting projections are identified, the high-dimensional data are then projected onto a lower dimensional (2D or 3D) subspace, which makes it possible to visually examine the structure of the dataset. From the wide range of EPP projection methods that have been proposed until now, some neural implementations have been selected for HUEPs in present paper, as they have been successfully applied to a wide variety of fields and datasets.

2.1.1. Principal Component Analysis. Principal Component Analysis (PCA) is a statistical model that has been widely applied in last decade and still is applied at present time [46]. It was introduced in [47] and describes the variation in a high-dimensional dataset in terms of a set of uncorrelated variables (each one of these variables is a linear combination of the original ones). From a geometrical perspective, it consists of a rotation of the axes of the original coordinate system that generates a new set of orthogonal axes. In the case of PCA the new axes are ordered in terms of the amount of variance of the original data they account for. As a result, the first axes (those accounting for the highest variance) are the ones selected to obtain the new visualization of data. It should be noted that even if we are able to visualize the data with a few variables, it does not follow that an interpretation will ensue, as it depends on the original dataset. As previously proposed [48, 49], PCA can be performed by means of neural networks.

2.1.2. Maximum Likelihood Hebbian Learning. Among all the neural alternatives of performing EPP, Maximum Likelihood Hebbian Learning [50] is one based on the Negative Feedback Network. It associates an input vector (x) with an output vector (y) computed as

$$y_i = \sum_{j=1}^{\mathbf{N}} W_{ij} x_j, \quad \forall i \tag{3}$$

where W_{ij} is the weight linking input j to output i.

At the training stage, when the output of the neural network is calculated, the activation (e_j) is fed back through the same weights and subtracted from the input:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \quad \forall j \tag{4}$$

Finally, weights are updated according to the specific learning rule:

$$\Delta W_{ij} = \eta . y_i . \text{sign}\left(e_j\right) \left|e_j\right|^{p-1} \tag{5}$$

where η is the learning rate and p is a parameter related to the energy function.

2.1.3. Cooperative Maximum Likelihood Hebbian Learning. The Cooperative MLHL (CMLHL) model was proposed [51] as an extension of MLHL by adding lateral connections between neurons in the output layer of the network (see Equation (5)). CMLHL can be defined through (4)-(7), where an N-dimensional input vector (x) is processed to obtain an M-dimensional output vector (y).

(1) Feed-forward step:

$$y_i = \sum_{j=1}^{\mathbf{N}} W_{ij} x_j, \quad \forall i \tag{6}$$

(2) Lateral activation passing:

$$y_i (t+1) = \left[y_i (t) + \tau \left(b - Ay\right)\right]^{+} \tag{7}$$

(3) Feedback step:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \quad \forall j \tag{8}$$

(4) Weight change:

$$\Delta W_{ij} = \eta . y_i . \text{sign}\left(e_j\right) \left|e_j\right|^{p-1} \tag{9}$$

where τ is a parameter to model the "strength" of the lateral connections, b is the bias parameter, A is a symmetric matrix used to modify the response to the data, η is the learning rate, and p is a parameter related to the energy function.

2.2. Clustering. EPP has been described in the section above as a method for solving the difficult problem of identifying structure in high-dimensional data. Although for many datasets these dimension reduction methods effectively work to reveal groups, it has been previously highlighted that they are not specifically designed for preserving the clusters and neither the directions of maximum variation of data nor the departure from normality, what may ensure that the reduced space keeps the original structure of groups unaltered [7]. This is one of the main reasons for proposing HUEPs as an advanced visualization technique that provides with EPP visualizations while at the same time keeps information about clustering in the original dataset.

As previously stated, cluster analysis can be defined as the process of organizing data into groups that in some way have similar (or close) members. Data similarity or proximity is measured by a distance function defined on pairs of patterns. Up to now, many different distance measures have been used [52, 53].

On the other hand, all the different approaches to data clustering [2] are classified in two main types of methods: partitional or hierarchical. On the one hand, partitional methods are based on the idea of identifying the partition that optimizes (usually locally) a given clustering criterion. On the other hand, hierarchical methods generate a set of nested partitions that are iteratively merged according to a certain criterion. In present paper, one partitional and one hierarchical method have been applied and are described in following subsections.

2.2.1. K-Means. K-means [54] is a well-known partitional clustering method aimed at grouping data into a given number of clusters. In order to apply it, two parameters must be tuned: the given number of clusters (k) and the initial position of centroids. The latter can be chosen by the user or calculated in a preprocessing step. Once initial values are assigned to these parameters, each data in the dataset is assigned to the nearest cluster centroid, attaining the initial allocation of data in clusters. Then, the centroids are iteratively recalculated and a subsequent reallocation of data is made. This step is repeated until no further changes are made to the centroids, when the cluster assigned to each data is generated as the output.

This method heavily relies on its initial parameters; hence, a usual measure of the "goodness" of the grouping is the sum of the proximity Sums of Squared Error (SSE) that it attempts to minimize:

$$SSE = \sum_{j=1}^{k} \sum_{x \in G_j} \frac{p\left(x_i, c_j\right)}{n} \tag{10}$$

where c_j are the cluster centroids, $p()$ is the proximity function, n is the number of rows, and k is the number of groups.

Similarity or proximity is a key concept for the definition of a cluster. As a result, a measure of the similarity must be carefully chosen as it is crucial to most clustering methods. Among all the available measures of similarity for data whose features are all continuous, some of the most widely used ones are as follows:

(i) Squared Euclidean distance (sqEuclidean). Each centroid is calculated as the mean of the points in that cluster.

(ii) Cityblock: sum of absolute differences. Each centroid is calculated as the component-wise median of the points in that cluster.

(iii) Cosine: one minus the cosine of the included angle between points (treated as vectors). Each centroid is calculated as the mean of the points in that cluster, after normalizing those points to unit Euclidean length.

(iv) Correlation: one minus the sample correlation between points (treated as sequences of values). Each centroid is calculated as the component-wise mean of the points in that cluster. Previously, those points are centred and normalized to zero mean and unit standard deviation.

As a result of the clustering, a scalar is provided for each input vector, being the number of the cluster to which the vector has been assigned.

2.2.2. Hierarchical Methods. Differentiating from partitional clustering methods, hierarchical ones can be divided into two types:

(1) Agglomerative: they begin with each data in a different cluster, and clusters are successively merged together until a stopping criterion is met or until a single cluster is obtained.

(2) Divisive: they begin with all data assigned to the only cluster, that is split (and its descendants) until a stopping criterion is satisfied or every data is assigned to a different cluster.

In the present study, due to the successful results in initial experiments, agglomerative clustering has been selected in order to be compared to the partitional approach (k-means). In the case of agglomerative clustering, there is a variety of linking methods that can be applied. In present study, the following ones have been tested:

(i) Single: shortest distance.

(ii) Complete: furthest distance.

(iii) Ward: inner squared distance (minimum variance algorithm), appropriate for Euclidean distances only.

(iv) Median: weighted centre of mass distance (WPGMC: Weighted Pair Group Method with Centroid Averaging), appropriate for Euclidean distances only.

(v) Average: unweighted average distance (UPGMA: Unweighted Pair Group Method with Arithmetic Averaging).

(vi) Centroid: centroid distance (UPGMC: Unweighted Pair Group Method with Centroid Averaging), appropriate for Euclidean distances only.

(vii) Weighted: weighted average distance (WPGMA: Weighted Pair Group Method with Arithmetic Averaging).

3. Case Study: Internationalization of Spanish SMEs

In order to validate the proposed HUEPs, they are applied to an interesting problem that has not yet been addressed by means of EPP or clustering methods. Hence, HUEPs are generated to analyse the internationalization strategy of companies, what involves a high number of features.

The dataset analysed in the present study is based on a sample of all Spanish MNEs registered with the Foreign Trade Institute (ICEX) and from the Web site http://www.oficinascomerciales.es, both managed by the Spanish Ministry of Industry, Tourism, and Trade. In order to analyse a representative sample of companies with sufficient autonomy, we restricted the sample to keep only those large and independent enough to conduct and decide their own internationalization strategy. Thus, following a well-established cutoff point in international business literature, used for example by Eurostat (http://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Enterprise_size), we dropped from the sample those with less than 250 employees. We also dropped those companies with a foreign majority owner controlling more than half of the capital.

It is also important to note the huge impact of the financial crisis on the Spanish economy, which forced many multinational enterprises to sell or postpone international operations in order to focus on the problems of the home market. To avoid distortions in the results due to this exogenous effect, we took the year 2007 as our base year. Overall, the sample consists of 164 companies investing in 119 countries worldwide. Unfortunately, Afghanistan, Andorra, Puerto Rico, and São Tomé and Príncipe are not included in the sample due to lack of data. In addition, Serbia, Montenegro, and Kosovo are included as a group because at the time of the study they constituted a single country.

For the above-mentioned companies and countries, the following data about each one of the cases of international presence were collected (further details about the different features can be found in [32]):

(i) Company sector: 5 binary features stating the economy sector the company belongs to (manufacturing, food, construction, regulated, and others).

(ii) Company product diversification: 3 binary features (nondiversified, related or unrelated diversification).

(iii) Other company characteristics: assets, number of employees, return on assets (ROA), ROA growth, age, number of countries where the company operates, and leverage and whether or not the company is included in a stock market.

(iv) Host country characteristics: GDP, GDP growth, total inward Foreign Direct Investment, population, unemployment, level of corruption, and Economic Freedom Index.

(v) Geographic and psychic distance stimuli between home and host countries: the data for each psychic distance stimulus is calculated by Dow & Karunaratna [28] based on a principal component analysis of a single factor. The calculations are based on critical factors widely used in the literature to explain cross-national differences at the macro level. Thus, the education distance stimulus is based on differences on literacy rate and enrolment in second and third-level education building on data from the United Nations. The industrial development stimulus takes into account differences in ten dimensions such as in energy consumption, vehicle ownership, employment in agriculture, and number of telephones and televisions. The language stimulus is based on the differences between the dominant languages and the bilateral influence of each country's major language in the other country. The democracy stimulus includes differences in the type of political systems in terms of political rights, civil liberties and POLCON and POLITY IV indices which account for the political constraints of the government of the country based on the existence and alignment of other independent political agents who can keep reducing the government discretional power. The political ideology stimulus is based on the ideological leanings of the chief executive's political party and the largest political party in the government. Finally, the religion stimulus is calculated based on the differences between the dominant religions and the bilateral influence of each country's dominant religion in the other country.

As a result, a dataset containing 1456 samples and 33 features was obtained and is analysed by means of HUEPs as it is presented in the following section.

4. Results and Discussion

Data from the aforementioned real-world case study are shown on low-dimensional spaces, on which they can be visually compared. In this section, the main results (HUEPs) are presented; for comparison purposes, combinations of the three EPP methods (PCA, MLHL, and CMLHL) with two clustering methods (hierarchical clustering and k-means) are shown. Additionally, Psychic-Democracy information is added through the glyph metaphor. As it is a continuous variable ranging from 0 to 2, it has been discretized in quartiles and data are shown accordingly (see the legend in Figure 2).

Combinations of different values were tested during experimentation for each one of the parameters of the applied models. After that, the best results were selected and are

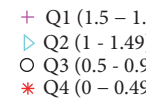


Figure 2: Legend for the glyph metaphor when using Psychic-democracy distance.

presented in Section 4 for the sake of brevity. In order to obtain such results, the different parameters were tuned with the following values:

(i) PCA: number of output dimensions: 2 and 3.

(ii) MLHL: number of output dimensions: 2 and 3, number of iterations: 3000, learning rate: 0.08009, p: 0.54.

(iii) CMLHL: number of output dimensions: 2 and 3, number of iterations: 3000, learning rate: 0.000175, p: 1.96, τ: 0.034.

(iv) k-means: k-means++ algorithm for cluster centre initialization, squared Euclidean distance and values of k equal to 3 and 6.

(v) Agglomerative clustering: cosine distance, single linkage method, and a cutoff value adjusted to obtain the same number of clusters as in the case of k-means (3 and 6).

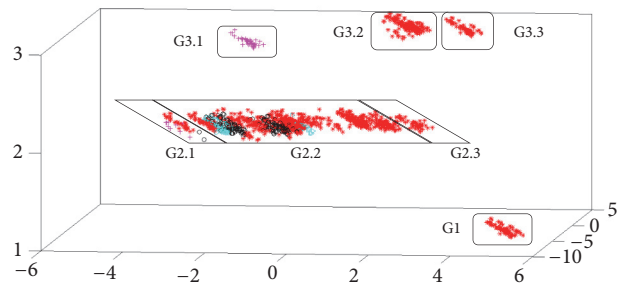
4.1. HUEP: EPP + Partitional Clustering. Firstly, HUEPs generated by the combination of the three EPP methods together with partitional clustering (k-means) are presented and their most relevant characteristics are discussed.

From a general point of view, visualizations in Figure 3 reveal a certain structure in the analysed dataset. The results from Figure 3(a) clearly depict groups at three different levels of the vertical axis (that is, the output of the k-means clustering method when the k parameter equals to 3). The first one (labelled as G1) is made of subsidiaries located in the United States. The second one includes three subgroups made of countries sharing specific characteristics. Subgroup G2.1 includes subsidiaries located in countries with economic and political problems such as Venezuela and Bangladesh. Subgroup G2.2 includes subsidiaries in emerging and growing economies, with a more stable environment compared to G2.1, such as Argentina, Brazil, Chile, Colombia, Hungary, Morocco, Mexico, Russia, Thailand, Turkey, Poland, Philippines, Slovakia, and Slovenia. This subgroup also includes some European countries with relatively advanced economies such as Belgium, Ireland, and Portugal. Finally, subgroup G2.3 includes small European countries with advanced economies and stable democracies such as the Netherlands and Norway. The third level includes also three subgroups with particular characteristics. The subgroup G3.1 includes subsidiaries located in China. The subgroup G3.2 includes Japan and some of the largest Western economies such as France, Italy, and Germany. Finally, subgroup G3.3 includes another developed European economy, the UK. Overall, Figure 3(a) offers a very clear determination of a cluster of a country with a very low level of democracy (China), at the extreme left side of the visualization, compared to democratic societies which appear on the right side. However, the HUEP also clearly distinguishes between advanced societies with a similar pluralistic political system to Spain, such as other geographically closer Western Europe economies of a similar size (France, Italy, and Germany), as opposed to another democratic country but with a different political organization based more on a bipartisan system. Smaller economies are located in the second, intermediate level of the vertical axis, but clearly differentiated according to their level of economic and political development, with those less developed economies at the extreme left side of the graph, stable and growing emerging countries in the middle and more advanced countries at the extreme right side.

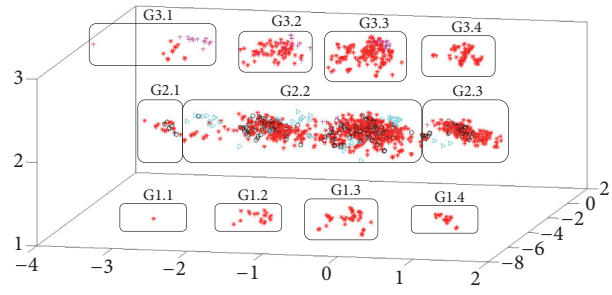
The results from Figure 3(b) exhibit a very similar pattern to those of Figure 3(a). Subgroups G1.1, G1.2, G1.3, and G1.4 gather all of the subsidiaries located in the United States, similar to the group G1 in Figure 3(a). The second level is again a mixed combination of emerging economies from all over the world together with democratic societies of a smaller size of Spain, very similar to what happened in Figure 3(a). In this case, however, it is worth noting that the subgroups of this level are much more heterogeneous and it is not easy to differentiate them according to their level of development as in the previous visualization. Both emerging and advanced countries appear in all subgroups. Finally, the main difference between both figures is that Group 3 is also less clear in Figure 3(b) than in Figure 3(a). While in Figure 3(a) China was clearly identified as an independent subgroup and all the subsidiaries located in this country were included in a single subgroup, in this case they appear simultaneously in subgroups G3.1, G3.2, and G3.3. Besides, subsidiaries, located in the common law based in UK, do not appear in a slightly separated subgroup, but mixed in all G3 subgroups.

As the best results are obtained by CMLHL, HUEP generated by this EPP method together with k-means is individually shown in Figure 4.

The results from Figure 4 are consistent with the previous ones but offer an insightful nuance. First, G1 is consistent with Figures 3(a) and 3(b) and includes all subsidiaries located in the United States. Next, G2 can be split into two subgroups, the first one (G2.1) is made of subsidiaries in Serbia, Montenegro, and Kosovo. While these three countries used to be a single one, Montenegro held an independence referendum in 2006 and Kosovo declared its unilateral independence in 2008. This particular method, unlike the previous ones, shows the ability to distinguish these historic events taking place around the time the sample was collected. The second one, G2.2, is made of the same mix of emerging economies and democratic advanced countries smaller in size than Spain. Finally, G3 is split into three subgroups. The first one (G3.1) in the left extreme of the graph shows subsidiaries located in China. The second one (G3.2) includes Western European countries close to Spain in terms of geography, size, and democratic systems (France, Italy, and Germany), and the third one (G3.3) includes the subsidiaries located in the UK. According to the Psychic-Democracy information that is also depicted, HUEP generated from CMLHL projection



(a) HUEP: PCA + partitional clustering + Psychic - Democracy information



(b) HUEP: MLHL + partitional clustering + Psychic - Democracy information

FIGURE 3: HUEP samples on partitional clustering (k=3).

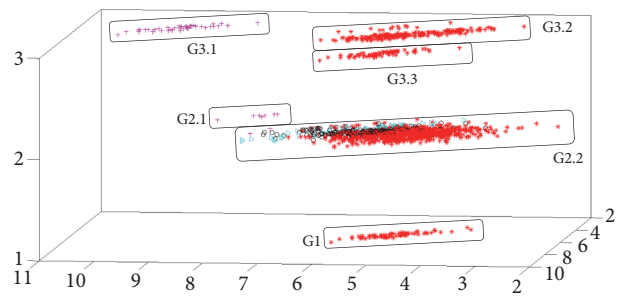


FIGURE 4: HUEP: CMLHL + partitional clustering (k=3) + Psychic - Democracy information.

reveals the structure of the dataset in a better way; in Figure 4 subgroups 2.1 and 3.1 only gather countries with highest scores (Q1 - pink crosses). Data with intermediate values of Psychic-Democracy information are all gathered in subgroup 2.2 but in an ordered way, starting from Q2 (blue triangles) close to the data with highest values (Q1) on the left side of the visualization. Then, data from Q3 are visualized (black circles) and data with lowest values (red stars) can be seen on the right side.

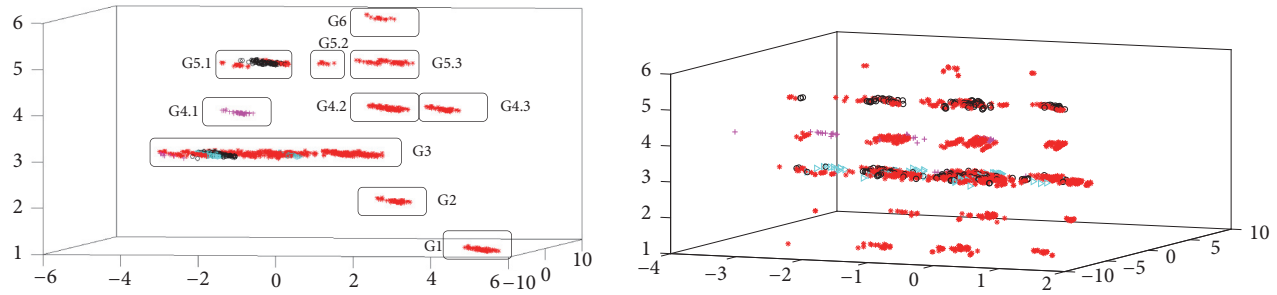
To check the effect of increasing the target number of clusters to be identified (k parameter) by k-means, some other experiments were run. The results for the three EPP methods when the k parameter equals to 6 are shown below in Figures 5(a), 5(b), and 6.

In general terms, it can be said that the results from Figure 5(a) are comparable to those from Figure 3(a). At the lower level, on the right extreme of the graph, the visualization identifies two separate large economies. G1 includes subsidiaries in the US and G2 those in Germany. G3, that is more on the left side of the graph, includes subsidiaries in various emerging economies such as Argentina, Chile, Morocco, Poland, and Turkey. G4 is similar to G3 in Figure 2(a). G4.1 identifies subsidiaries located in China. G4.2 includes subsidiaries in France and Italy and G4.3 in UK. G5.1 includes subsidiaries in three large emerging countries: Brazil, Mexico, and Russia. G5.2 identifies one country in particular, South Korea. G5.3 includes subsidiaries in advanced economies such as Australia, Canada, and the Netherlands. Finally, at the top of the graph, G6 identifies subsidiaries in Japan.

Overall, this HUEP in Figure 5(a) shows consistent results with those from Figure 3(a) as it displays countries according to their level of economic and political development from left to right, and it also identifies specific countries that are relevant and with a particular idiosyncrasy. As in the case of Figure 3(a), US, China, and UK are highlighted given their differences with the continental European system of Spain. However, in this case, Germany, South Korea, and Japan are also identified in specific subgroups. The former may be due to its federal political organization in which the constituent states (*Länder*) retain a measure of sovereignty. The latter two are two stable and advanced countries with a well-functioning democratic, parliament-based, political system. However, their large cultural distance from Spain and political tensions with China and North Korea may explain why this visualization separates them from the rest.

No clear structure is revealed in Figure 5(b): many different subgroups are generated with a heterogeneous mixture of countries. As a result, and for the sake of brevity, results in this figure are not described.

The results from Figure 6 show a very similar pattern to those from Figure 5(a). G1 includes subsidiaries in the US and G2 in Germany. G3.2 includes emerging economies similar to the previous G3 subgroup in Figure 5(a). However, in this case, the visualization separates Serbia in G3.1, due to the previously mentioned events happening in Montenegro and Kosovo. Also similar to G4 in Figure 5, here the subgroup G4.1 located in the left extreme of the graph includes subsidiaries in China, whereas G4.2 includes those in France and Italy and G4.3 those in the UK. Finally, this visualization identifies Japan in G6, but contrary to the previous visualization, South Korea is included in a larger and more heterogeneous group with other countries in G5. This group includes all the countries that in Figure 5(a) were part of subgroups G5.1 (Brazil, Mexico, and Russia) and G5.3 (Australia, Canada, and the Netherlands). Overall, while this visualization offers the advantage of identifying Serbia separately as Figure 4, it shows a less clear picture in G5 compared to Figure 5(a).



(a) HUEP: PCA + partitional clustering + Psychic - Democracy information

(b) HUEP: MLHL + partitional clustering + Psychic - Democracy information

FIGURE 5: HUEP samples on partitional clustering (k=6).

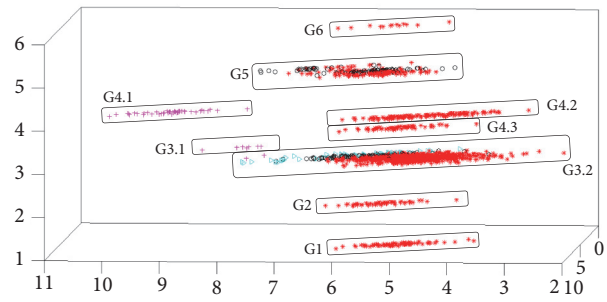


FIGURE 6: HUEP: CMLHL + partitional clustering (k=6) + Psychic - Democracy information.

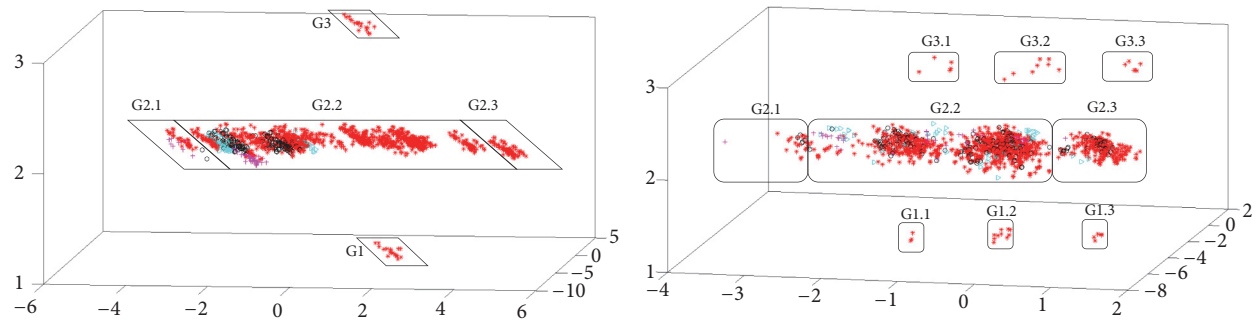
When analysing Psychic-Democracy information depicted in Figure 6, it can be said that once again, groups are coherently organized according to such criteria. Only 2 subgroups (out of 9) contain data from more than one quartile. Furthermore, there is a global and decreasing ordering from left (data in Q1, depicted as pink crosses) to right (data in Q4, depicted as red stars).

4.2. HUEP: EPP + Hierarchical Clustering. In order to check the validity of proposed HUEPs to combine results from different clustering methods, results from hierarchical clustering (combined with the 3 different EPP methods, namely, PCA, MLHL, and CMLHL) are shown in this subsection.

Figure 7(a) shows some interesting differences compared to previous visualizations. While also organized in three levels (3 output clusters) as Figures 3(a) and 3(b), the countries uniquely identified in separate groups are different. In the lower level, G1 identifies Australia and in the upper level G3 identifies Ireland. In the middle level, three subgroups are identified. In this case, consistent with Figures 3(a) and 3(b), countries on the left show lower levels of economic and political development. Thus, G2.1 includes Venezuela and Bangladesh, while G3 identifies the US. G2.2 is a very heterogeneous group including all other countries in the world. In this case, the HUEP underlines the particular situation of Ireland, a location where the laws of the country offer very favourable conditions given the low corporate tax, noticeably lower than in the rest of Europe. As a result, many MNEs have located their subsidiaries, often leading to controversial debates and loss of legitimacy. For example, Zara's owner Inditex has been accused of tax evasion (https://www.independent.ie/business/irish/zara-owner-used-ireland-to-slash-its-tax-bill-meps-claim-35279873.html), reporting millions of euros in turnover but having no employees on the payroll (https://fashionunited.uk/news/business/inditex-accused-of-dodging-585-million-euros-in-taxes/2016120822765). The case of Australia might be due to the fact that it is a country perceived as distant both in terms of geography and culture, which represents an obstacle to FDI, and pertaining to the Commonwealth and therefore based on a common law system with relevant similarities with the UK.

Figure 7(b) is also structured in three levels and shows very consistent results with the previous one. However, in this case, G1 and G3 are split into three subgroups. Similar to Figure 7(a), G1 includes subsidiaries in Australia and G3 includes subsidiaries in Ireland. However, in this visualization it is possible to observe differences based on the specific sector of the firms. Subgroups on the left (G1.1 and G3.1) include companies in the infrastructure sector such as ACS, Ferrovial or Indra, and other highly regulated sectors such as airlines (Iberia). Subgroups in the middle (G1.2 and G3.2) include large companies in manufacturing such as Inditex and Mango. Finally, subgroups on the right side of the graph include smaller (albeit also MNEs) companies such as Teka, Tamisa, or Valdepesa. While this visualization is more precise about the sectors of these two particular countries, the subgroups in the middle level of the vertical axis (G2.1, G2.2, and G2.3) are heterogeneous and it is not easy to identify groups based on their level of economic or political development compared to G2 in Figure 7(a).

The results of Figure 8 are quite similar to those in Figure 7(a). G1 includes subsidiaries in Australia and G3 includes those in Ireland. However, G2.1 includes two countries that have been repeatedly identified in independent subgroups in previous visualization (China and Serbia), although in this case they form a group together due to the perspective. Finally, in G2.3 the UK is identified as a slightly separate group compared to the larger G2.2 which includes the majority of countries of the world. It is worth highlighting that the ordering (from left to right) of Psychic-Democracy



(a) HUEP: PCA + hierarchical clustering + Psychic - Democracy information

(b) HUEP: MLHL + hierarchical clustering + Psychic - Democracy information

Figure 7: HUEP samples on hierarchical clustering.

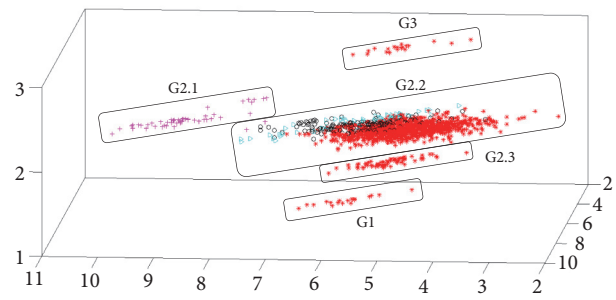


Figure 8: HUEP: CMLHL + hierarchical clustering + Psychic - Democracy information.

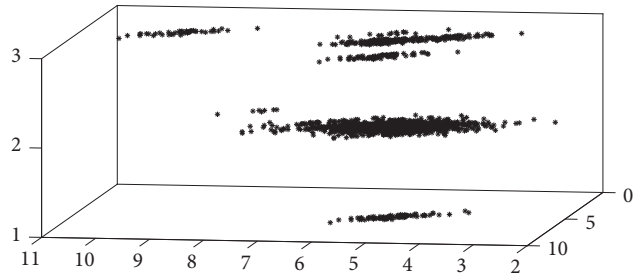


Figure 9: CMLHL + partitional clustering.

information is preserved in Figure 8, with a more precise definition than in Figure 4.

From previous Sections 4.1 and 4.2 it can be concluded that HUEPs successfully combine the output from different EPP (PCA, MLHL, and CMLHL) and clustering (partitional and hierarchical) methods.

4.3. Comparison to Alternative Visualizations. Up to the authors' knowledge, there is not any validation method to test HUEPs with quantitative metrics. As a consequence, the obtained results are visually compared with some other visualizations of the same dataset. For a fair comparison, 3D scatterplots have also been generated.

Initially, HUEPs are compared to a combination of EPP together with partitional clustering, without using the glyph metaphor with any additional information as it has been used in Figures 3–8 (Psychic-Democracy).

In Figure 9 the same structure that has already been described in the case of Figure 4 is revealed. The same data are located in the same groups, but obviously adding further details through the glyph metaphor makes HUEPs more informative. Thanks to the different colours and shapes, it is easy to get an idea of the global ordering that has been previously mentioned (for Figures 4 and 8) and to know which countries are located in some of the groups.

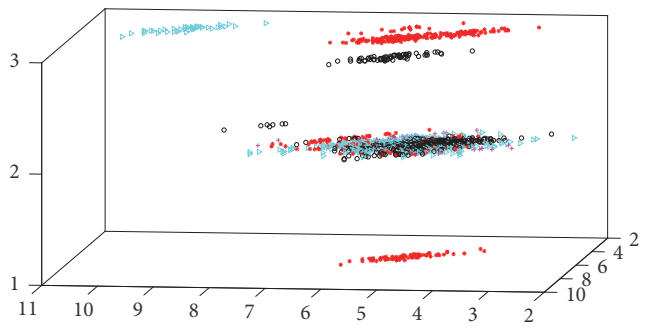
4.3.1. Alternative Distance Information. In order to check the results of the proposed HUEPs when visualizing some other distance criteria (different from Psychic-Democracy) through the glyph metaphor, Figures 10–12 are provided. As these also are continuous variables, they have been discretized in quartiles, as in the case of Figures 3–8.

Yet, in order to provide a comprehensive analysis, we also conduct the analysis of all psychic dimensions altogether. To do so we rely on the operationalization suggested by [26] as this method has been proven superior to the simple average of dimensions since it also takes into account the differences in variance of the dimensions. Algebraically, this method can be expressed as

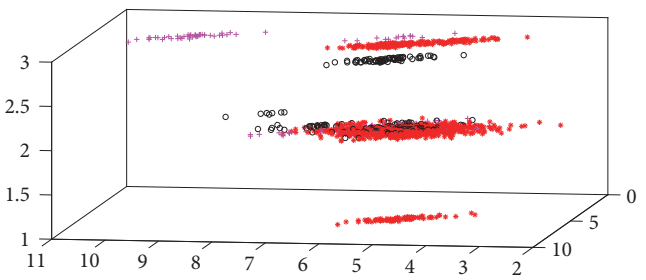
$$KS_j = \frac{\sum_{i=1}^{6} \left(\left(I_{ij} - I_{iu} \right)^2 / V_i \right)}{6} \tag{11}$$

where I_{ij} is country j's score on the ith cultural dimension, I_{iu} is the score for Spain on this dimension, and V_i is the variance of the score on the dimension.

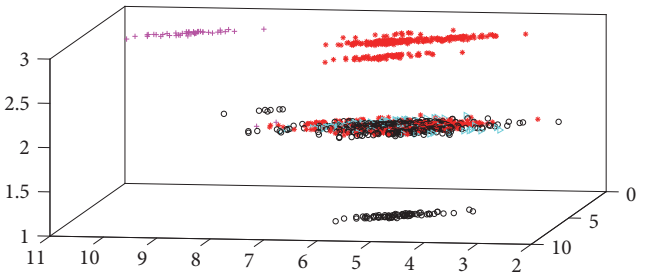
While in the previous visualization we focused on the democracy distance, we also controlled the visualizations of other psychic distance stimuli and also that of the overall psychic distance aggregated into a single construct using the Kogut & Singh's formula. For the sake of parsimony, we show here only those with a clearer visualization of the different groups, in particular the one using education distance, the one using Religion distance, and the one using the aggregated



(a) HUEP: CMLHL + partitional clustering (k=3) + Psychic - Education

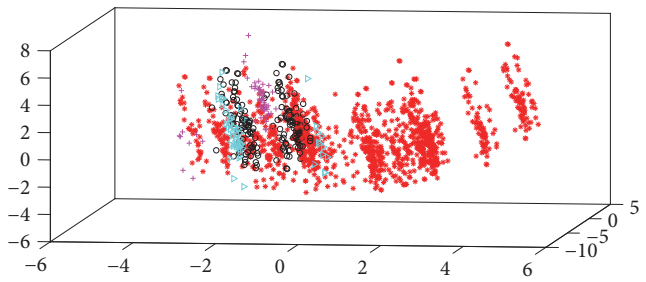


(b) HUEP: CMLHL + partitional clustering (k=3) + Psychic - Religion

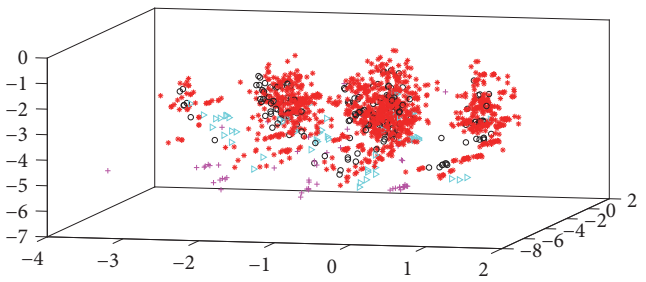


(c) HUEP: CMLHL + partitional clustering (k=3) + Kogut & Singh information

FIGURE 10: Comparison of HUEPs when visualizing different distance criteria through the glyph metaphor.



(a) PCA 3D visualization + Psychic - Democracy information



(b) MLHL 3D visualization + Psychic - Democracy information

FIGURE 11: 3D plots from EPP methods.

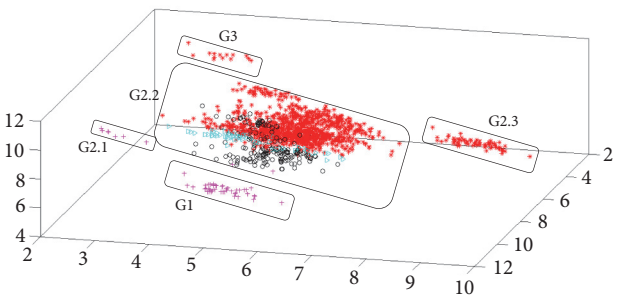


FIGURE 12: CMLHL 3D visualization + Psychic - Democracy information.

Kogut & Singh's formula. However, as it can be seen in Figures 10(a), 10(b), and 10(c), the visualization is less clear as it is observed in the more fuzzy combination of colours in some of the groups. Although different criteria can be visualized by the HUEPs, not all of them are equally informative for a given projection. In the case of the criteria visualized in Figure 10 it can be seen that some of the data from the same quartile are gathered in the same groups but some others are not. Furthermore, a global ordering is not revealed as in the case of Psychic-Democracy (Figures 4 and 8).

4.3.2. 3D EPP Projections. Finally, the comprehensive comparison of visualizations also comprises simpler 3D plots where only the output (3 first components) of the EPP methods is depicted, together with the glyph metaphor.

When compared with previous HUEPs that combine the outputs of corresponding EPP methods (PCA or MLHL), Figure 11 does not reveal the structure of the whole dataset in a sparse and clear way, although some subgroups could be identified.

In the case of Figure 12, the 3D CMLHL visualization reveals more clearly defined groups than those from Figure 11. In this case, the majority of countries are included in a very heterogeneous group in G2.2. However, the method identifies the subsidiaries located in China in group G1, the subsidiaries located in Serbia in the subgroup G2.1, and the subsidiaries located in the UK in the subgroup G2.2, countries showing some specific features as already described. Finally, G3 includes the subsidiaries in South Africa, a country that was never represented in its own separate group. As in the case of Australia, that was singled out in some previous visualizations, this is a country that is both geographically and culturally distant to Spain and with a political system based on the UK's common law system, as a former colony and part of the Commonwealth. While this visualization identifies specific countries such as China, Serbia, and South Africa, the very heterogeneous nature of countries included in G1 makes the visualization less clear than previous ones such as those of Figures 4 and 8. When compared to the corresponding HUEPs, it can be said that adding the clustering information makes the visualization more precise, as data are split in a larger number of more separated groups, which let us gain deeper knowledge of the case of study.

5. Conclusions and Future Work

From the results presented in Section 4, it can be concluded that HUEPs are a useful technique to visually analyse internationalization data in order to better understand it. More specifically, the presented visualizations provide insightful information about the geographical distribution of Spanish subsidiaries. They also allow for the identification of specific countries exhibiting specific political and legal characteristics or going through particular historic events (e.g., China, the UK, US, Serbia, etc.). This type of data represents a valuable source of information for managers in enterprises who can learn from vicarious experience (i.e., the knowledge that companies can obtain from the actions of other firms sharing a common characteristic, such as nationality) [32]. By observing the behavior of other companies, firms can imitate best practices and avoid previous mistakes [55]. Besides, the data is also very relevant for policy-makers interested in attracting larger volumes of foreign investors, as these investments can provide key technology or managerial talent missing in the country and also positive spillovers in the form of a boost for the competitiveness of other related industries in the economy [56].

When considering the different EPP methods that have been applied, it can be said that CMLHL provides the more sparse projections, what is consistent with previous work. On the other hand, both clustering methods generate meaningful outputs and it is worth mentioning that HUEPs greatly accommodate to a varying number of clusters (higher than 1). According to the glyph metaphor comparison, adding Democracy (Psychic) distance let us better understand the nature of the analysed dataset by HUEPs. Thanks to the more precise definition and higher number of groups in the visualizations, HUEPs contribute to overcome some of the drawbacks of scatter plots: overplotting and overlapping. All in all, it has been proven that HUEPs are a valid proposal to combine the outputs from different EPP and clustering methods. Additionally, the 3D scatterplots can be enriched with information from different sources (distance criteria in the present case study).

In future work, HUEPs will be applied to some other multidimensional datasets, comprising companies from other countries apart from Spain and thus comparing the internationalization strategies of companies from different countries.

Data Availability

Previously reported ICEX data were used to support this study and are available at http://www.oficinascomerciales.es. These prior datasets are cited at relevant places within the text.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work was conducted during Álvaro Herrero's research stay at KEDGE Business School in Bordeaux (France). Some results of this ongoing research, from the same dataset, have been presented in the 13th International Conference on Soft Computing Models in Industrial and Environmental Applications, as a paper entitled "Visualizing Industrial Development Distance to Better Understand Internationalization of Spanish Companies".

References

[1] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Transactions on Computers*, vol. 23, pp. 881–890, 1974.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[3] W. Desarbo, K. Jedidi, K. Cool, and D. Schendel, "Simultaneous multidimensional unfolding and cluster analysis: An investigation of strategic groups," *Marketing Letters*, vol. 2, no. 2, pp. 129–146, 1991.

[4] M. Vichi and H. A. L. Kiers, "Factorial k-means analysis for two-way data," *Computational Statistics & Data Analysis*, vol. 37, no. 1, pp. 49–64, 2001.

[5] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, New York, NY, USA, 2nd edition, 2002.

[6] T. Tarpey, "Linear transformations and the k-means clustering algorithm: applications to clustering curves," *The American Statistician*, vol. 61, no. 1, pp. 34–40, 2007.

[7] G. Menardi and N. Torelli, "Reducing data dimension for cluster detection," *Journal of Statistical Computation and Simulation*, vol. 83, no. 11, pp. 2047–2063, 2013.

[8] T. Liu, C. K. L. Lekamalage, G. Huang, and Z. Lin, "Extreme learning machine for joint embedding and clustering," *Neurocomputing*, vol. 277, pp. 78–88, 2018.

[9] S. Jun, S.-S. Park, and D.-S. Jang, "Document clustering method using dimension reduction and support vector clustering to overcome sparseness," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3204–3212, 2014.

[10] K. K. Bharti and P. K. Singh, "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3105–3114, 2015.

[11] J. Guo, X. Zhao, X. Yuan, Y. Li, and Y. Peng, "Discriminative unsupervised 2D dimensionality reduction with graph embedding," *Multimedia Tools and Applications*, vol. 77, no. 3, pp. 3189–3207, 2018.

[12] C.-L. Liu, W.-H. Hsaio, and T.-H. Chang, "Locality sensitive *K*-means clustering," *Journal of Information Science and Engineering*, vol. 34, no. 1, pp. 289–305, 2018.

[13] R. Sánchez, Á. Herrero, and E. Corchado, "Visualization and clustering for SNMP intrusion detection," *Cybernetics and Systems*, vol. 44, no. 6-7, pp. 505–532, 2013.

[14] Á. Arroyo and Á. Arroyo, "A hybrid intelligent system for the analysis of atmospheric pollution: a case study in two european regions," *Logic Journal of the IGPL*, vol. 25, no. 6, pp. 915–937, 2017.

[15] R. J. Bolton and W. J. Krzanowski, "Projection pursuit clustering for exploratory data analysis," *Journal of Computational and Graphical Statistics*, vol. 12, no. 1, pp. 121–142, 2003.

[16] Y. G. Yatracos, "Detecting clusters in the data from variance decompositions of its projections," *Journal of Classification*, vol. 30, no. 1, pp. 30–55, 2013.

[17] R. Redondo, "Neural visualization for the analysis of energy and water consumptions in the automotive industry," in *Proceedings of the International Joint Conference SOCO18-CISIS18-ICEUTE18*, Springer International Publishing, Cham, Switzerland, 2019.

[18] S. L. Marie-Sainte, "Detection and visualization of non-linear structures in large datasets using exploratory projection pursuit laboratory (EPP-Lab) software," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 1, pp. 2–18, 2017.

[19] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Systems with Applications*, vol. 92, pp. 507–520, 2018.

[20] S. Zaheer, M. S. Schomaker, and L. Nachum, "Distance without direction: Restoring credibility to a much-loved construct," *Journal of International Business Studies*, vol. 43, no. 1, pp. 18–27, 2012.

[21] A. Jiménez, D. Benito-Osorio, J. Puck, and P. Klopf, "The multifaceted role of experience dealing with policy risk: The impact of intensity and diversity of experiences," *International Business Review*, vol. 27, no. 1, pp. 102–112, 2018.

[22] P. Ghemawat, "Distance still matters. The hard reality of global expansion.," *Harvard Business Review*, vol. 79, no. 8, pp. 137–142, 2001.

[23] H. Berry, M. F. Guillén, and N. Zhou, "An institutional approach to cross-national distance," *Journal of International Business Studies*, vol. 41, no. 9, pp. 1460–1480, 2010.

[24] S. L. Merker, *Culture's Consequences: International Differences in Work-Related Values*, vol. 27, Sage Publications, Beverly Hills, Ca, USA, 1980.

[25] G. Hofstede, G. J. Hofstede, and M. Minkov, *Cultures and Organizations: Software of the Mind*, McGraw-Hill, New York, NY, USA, 2010.

[26] B. Kogut and H. Singh, "The effect of national culture on the choice of entry mode," *Journal of International Business Studies*, vol. 19, no. 3, pp. 411–432, 1988.

[27] R. L. Tung and A. Verbeke, "Beyond hofstede and GLOBE: improving the quality of cross-cultural research," *Journal of International Business Studies*, vol. 41, no. 8, pp. 1259–1274, 2010.

[28] D. Dow and A. Karunaratna, "Developing a multidimensional instrument to measure psychic distance stimuli," *Journal of International Business Studies*, vol. 37, no. 5, pp. 578–602, 2006.

[29] L. Håkanson and B. Ambos, "The antecedents of psychic distance," *Journal of International Management*, vol. 16, no. 3, pp. 195–210, 2010.

[30] P. A. Brewer, "Operationalizing psychic distance: A revised approach," *Journal of International Marketing*, vol. 15, no. 1, pp. 44–66, 2007.

[31] J. Johanson and J.-E. Vahlne, "The internationalization process of the firm: a model of knowledge development and increasing foreign market commitments," *Journal of International Business Studies*, vol. 8, no. 1, pp. 23–32, 1977.

[32] A. Jiménez and D. de la Fuente, "Learning from others: the impact of vicarious experience on the psychic distance and FDI relationship," *Management International Review*, vol. 56, no. 5, pp. 633–664, 2016.

[33] O. Shenkar, "Cultural distance revisited: towards a more rigorous conceptualization and measurement of cultural differences," *Journal of International Business Studies*, vol. 32, no. 3, pp. 519–535, 2001.

[34] B. L. Kirkman, K. B. Lowe, and C. B. Gibson, "A quarter century of culture's consequences: A review of empirical research incorporating Hofstede's cultural values framework," *Journal of International Business Studies*, vol. 37, no. 3, pp. 285–320, 2006.

[35] A. Delios and W. J. Henisz, "Political hazards, experience, and sequential entry strategies: The international expansion of Japanese firms, 1980-1998," *Strategic Management Journal*, vol. 24, no. 11, pp. 1153–1164, 2003.

[36] G. L. Holburn and B. A. Zelner, "Political capabilities, policy risk, and international investment strategy: Evidence from the global electric power generation industry," *Strategic Management Journal*, vol. 31, no. 12, pp. 1290–1315, 2010.

[37] A. Jiménez, "Does political risk affect the scope of the expansion abroad? Evidence from Spanish MNEs," *International Business Review*, vol. 19, no. 6, pp. 619–633, 2010.

[38] T. Lawton, T. Rajwani, and J. Doh, "The antecedents of political capabilities: A study of ownership, cross-border activity and organization at legacy airlines in a deregulatory context," *International Business Review*, vol. 22, no. 1, pp. 228–242, 2013.

[39] A. Jiménez, I. Luis-Rico, and D. Benito-Osorio, "The influence of political risk on the scope of internationalization of regulated companies: insights from a Spanish sample," *Journal of World Business*, vol. 49, no. 3, pp. 301–311, 2014.

[40] G. Conti, *Security Data Visualization: Graphical Techniques for Network Analysis*, No Starch Press, 2007.

[41] L. Wilkinson, *The Grammar of Graphics. Statistics and Computing*, Springer, New York, 2nd edition, 2005.

[42] D. A. Keim, "Information visualization and visual data mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.

[43] N. Elmqvist, P. Dragicevic, and J.-D. Fekete, "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1141–1148, 2008.

[44] C. Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann, Boston, Mass, USA, 3rd edition, 2013.

[45] D. Williams and D. Freedman, "Asymptotics of graphical projection pursuit," *The Annals of Statistics*, vol. 12, no. 3, pp. 793–815, 1984.

[46] F. Segovia, J. M. Górriz, J. Ramírez, and F. J. Martinez-Murcia, "Using deep neural networks along with dimensionality reduction techniques to assist the diagnosis of neurodegenerative disorders," *Logic Journal of the IGPL. Interest Group in Pure and Applied Logics*, vol. 26, no. 6, pp. 618–628, 2018.

[47] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 11, pp. 559–572, 1901.

[48] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, no. 6, pp. 927–935, 1992.

[49] C. Fyfe, "A neural network for PCA and beyond," *Neural Processing Letters*, vol. 6, no. 1-2, Article ID 1009606706736, pp. 33–41, 1997.

[50] E. Corchado, D. MacDonald, and C. Fyfe, "Maximum and minimum likelihood Hebbian learning for exploratory projection pursuit," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 203–225, 2004.

[51] E. Corchado and C. Fyfe, "Connectionist techniques for the identification and suppression of interfering underlying factors," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 8, pp. 1447–1466, 2003.

[52] B. Andreopoulos, A. An, X. Wang, and M. Schroeder, "A roadmap of clustering algorithms: finding a match for a biomedical application," *Briefings in Bioinformatics*, vol. 10, no. 3, pp. 297–314, 2009.

[53] W. Zhuang, Y. Ye, Y. Chen, and T. Li, "Ensemble clustering for internet security applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1784–1796, 2012.

[54] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, p. 14, University of California Press, Berkeley, Calif, USA, 1967.

[55] A. Terlaak and Y. Gong, "Vicarious learning and inferential accuracy in adoption processes," *Academy of Management Review (AMR)*, vol. 33, no. 4, pp. 846–868, 2008.

[56] K. E. Meyer and E. Sinani, "When and where does foreign direct investment generate positive spillovers? A meta-analysis," *Journal of International Business Studies*, vol. 40, no. 7, pp. 1075–1094, 2009.

Hindawi
Complexity
Volume 2019, Article ID 1306039, 22 pages
https://doi.org/10.1155/2019/1306039

Review Article

A Systematic Review of Deep Learning Approaches to Educational Data Mining

Antonio Hernández-Blanco,[1] **Boris Herrera-Flores**,[2] **David Tomás**,[3] **and Borja Navarro-Colorado**[3]

[1]*Technical University of the North, Ecuador*
[2]*Central University of Ecuador, Ecuador*
[3]*University of Alicante, Spain*

Correspondence should be addressed to Antonio Hernández-Blanco; antoniojhb@gmail.com

Received 28 December 2018; Revised 28 March 2019; Accepted 10 April 2019; Published 12 May 2019

Academic Editor: Roberto Natella

Educational Data Mining (EDM) is a research field that focuses on the application of data mining, machine learning, and statistical methods to detect patterns in large collections of educational data. Different machine learning techniques have been applied in this field over the years, but it has been recently that Deep Learning has gained increasing attention in the educational domain. Deep Learning is a machine learning method based on neural network architectures with multiple layers of processing units, which has been successfully applied to a broad set of problems in the areas of image recognition and natural language processing. This paper surveys the research carried out in Deep Learning techniques applied to EDM, from its origins to the present day. The main goals of this study are to identify the EDM tasks that have benefited from Deep Learning and those that are pending to be explored, to describe the main datasets used, to provide an overview of the key concepts, main architectures, and configurations of Deep Learning and its applications to EDM, and to discuss current state-of-the-art and future directions on this area of research.

1. Introduction

The research field of *Educational Data Mining* (EDM) focuses on the application of techniques and methods of data mining in educational environments. EDM is concerned with developing, researching, and applying machine learning, data mining, and statistical methods to detect patterns in large collections of educational data that would otherwise be impossible to analyze [1].

EDM leverages e-learning platforms such as *Learning Management Systems* (LMS), *Intelligent Tutoring Systems* (ITS), and, in the last years, Massive Open Online Courses (MOOC), to obtain rich and multimodal information from student's learning activities in educational settings. For instance, these platforms record when the students access a learning object, how many times they accessed it, whether the answer provided to an exercise is correct or not, or the amount of time spent reading a text or watching a video.

All this information can be analyzed to address different educational issues, such as generating recommendations, developing adaptative systems, and providing automatic grading for the students' assignments. Different machine learning techniques have been applied over time to analyze this data, but it has been in recent years that the use of Deep Learning techniques has emerged in the field of EDM.

The topic of *Deep Learning* (DL) has gained increasing attention in the industry and research areas in the last decade, revolutionizing the field of machine learning by obtaining state-of-the-art results in perception tasks such as image and speech recognition [2]. Major companies such as Google, Facebook, Microsoft, Amazon, and Apple are heavily investing in the development of software and hardware innovations in this field, trying to leverage DL potential in the production of smart products.

DL is based on neural network architectures with multiple layers of processing units that apply linear and nonlinear

transformations to the input data. These architectures can be applied to all type of data: image, audio, text, numerical, or some combination of them. Many research fields have benefited from applying these technologies, and EDM is not an exception.

In the last few years there has been a proliferation of research in the EDM field using DL architectures. This article presents a review on the literature of DL techniques applied to EDM, from its first appearance in 2015 to the present day. The primary contributions of this article are as follows:

(i) Summarize the main EDM tasks and classify the existing works that have applied DL on each of these tasks.

(ii) Identify the tasks that have gained major attention and those that are still unexplored.

(iii) Describe and categorize the main public and private datasets employed to train and test DL models in EDM tasks.

(iv) Introduce key DL concepts and technologies, describing the techniques and configurations most widely used in EDM and its specific tasks.

(v) Discuss future directions for research in DL applied to EDM based on the information gathered in this study.

The rest of this article is organized as follows: Section 2 presents and compares previous surveys in the field of EDM; Section 3 describes the process carried out to retrieve the papers reviewed in this study, including a quantitative analysis of the papers gathered; Section 4 describes the main tasks in EDM, identifies the existing literature in each task, and describes the main datasets employed in the field; Section 5 presents the key concepts of DL, the main architectures, configurations, and frameworks, summarizing the characteristics (in terms of DL technologies) of the work done in EDM; Section 6 presents a discussion about the information compiled during this review work; finally, conclusions are presented in Section 7.

2. Review of Previous Surveys

The application of data mining techniques to educational environments has been an active research field in the last few decades, gaining much popularity in recent times thanks to the availability of online datasets and learning systems. Different surveys have been published about EDM so far, and this section summarizes these works and presents the key differences between the current proposal and the previous reviews in this field.

The first EDM survey identified in the literature was developed in 2007 by Romero and Ventura [3], which was further improved in 2010 [4] and 2013 [5]. In the later, the authors analyzed more than 300 studies carried out before 2010, identifying eleven categories or tasks in EDM: analysis and visualization of data, providing feedback for supporting instructors, recommendations for students, predicting student's performance, student modeling detecting undesirable student behaviors, grouping students, social network analysis, developing concept maps, constructing coursewares, and planning and scheduling. The survey presented methods and techniques employed in the EDM field in each of these categories.

In 2009, a new EDM survey was presented by Baker and Yacef [6]. This study discussed trends and shifts in research conducted by this community, comparing its current state with the early years of EDM. In this case, the authors identified four applications/tasks in this field: improving student models, improving domain models, studying the pedagogical support provided by learning software, and scientific research into learning and learners. The most-cited papers in EDM between 1995 and 2005 were listed, discussing their influence on the EDM community.

Peña-Ayala proposed in 2014 a thorough survey by applying data mining techniques to more than 240 papers in EDM [7]. The execution of statistical and clustering processes identified a set of educational functionalities, a pattern of EDM approaches, and two patterns of value-instances to depict EDM approaches based on descriptive and predictive models. Unlike previous literature reviews, this work mainly focused on computational techniques rather than EDM applications.

More recently, two new studies have been added to this list of surveys. The first one was carried out by Bakhshinategh et al. in 2018 [8]. This work studied various tasks and applications existing in the field of EDM and categorized them based on their purposes. Based on the eleven categories proposed by [4], they suggested a hierarchy of thirteen categories grouped into five main tasks: Student Modeling, Decision Support Systems, Adaptive Systems, Evaluation, and Scientific Inquiry. In Section 4.1, this taxonomy of tasks is used as the basis to classify the current studies in DL for EDM.

Finally, the most recent review devoted to EDM has been developed by Aldowah et al. [9] in 2019. This study constrained the research to works applied in the context of higher education. The analysis presented was based on four dimensions: computer supported learning analytics, computer supported predictive analytics, computer supported behavioral analytics, and computer supported visualization. Based on the results of previous studies, the authors found that specific EDM techniques could offer the best means of solving certain learning problems, offering student-focused strategies and tools for educational institutions.

In these review papers there are two aspects that have not been studied in a systematic way, and that the present work intends to analyze: the existing datasets and the use of DL techniques in EDM. Firstly, in order to empirically compare different approaches, it is necessary to know the underlying datasets employed in the experiments. In this paper, a section is devoted to review and summarize these resources (see Section 4.2). Secondly, although previous proposals have taken into account (shallow) neural networks approaches in the literature, none of them is specifically focused on DL techniques. In this paper, Section 5 provides an introduction to the foundations of DL (main architectures, training process, hyperparameters, and frameworks), characterizing these techniques in the EDM domain and relating them to the papers reviewed.

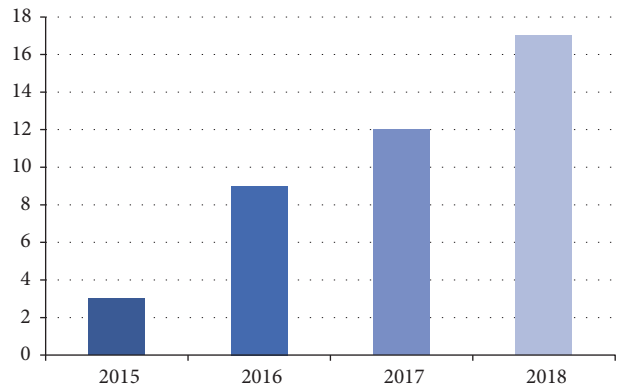


Figure 1: Number of papers published per year.

3. Methodology

This section describes the methodology followed to carry out this review and the process of gathering, analyzing and extracting the existing works on DL applications to EDM.

In order to perform a systematic review, the following scientific repositories were accessed: ACM Digital Library (https://dl.acm.org/), Google Scholar (https://scholar.google.es/), and IEEE Xplore (https://ieeexplore.ieee.org/). These sources were queried with the following search string: `"deep learning" AND "educational data mining"`. As a result, a large set of papers was retrieved and a manual review process was applied to filter out duplicates and papers on unrelated topics. The bibliography cited in the papers that initially passed the filter was also reviewed. This allowed expanding the number of relevant papers retrieved. The final set contained 41 papers. Figure 1 summarizes the number of publications per year. The first papers applying DL to EDM were published just four years ago, in 2015, and there is clearly an increase in the number of publications over the years until 2018.

Table 1 summarizes the number of papers published in each publication venue. Most of them have been published in conferences (80%). The *International Conference on Educational Data Mining* accumulates the maximum number of publications (considering the last three editions), with a total of 16. Not surprisingly, this is the congress of reference in the EDM field.

Finally, Figure 2 shows a choropleth map of the world showing the density of researchers per country involved in the area of DL applied to EDM, based on their affiliation. Authors are weighted by the number of contributors to the paper. For instance, in a paper with n authors, each one will contribute to their country with a weight of $1/n$. The map shows that United States is the more active country in this field, followed (at a great distance) by India, Canada and China. Other countries where researchers have contributed to this field are New Zealand, Singapore, Japan, Argentina, Australia, and Serbia.

4. Educational Data Mining

The first part of this section shows taxonomy of the tasks addressed by EDM systems. The works reviewed are briefly described and classified using this taxonomy in order to differentiate the tasks that have been faced by DL approaches from those that are still unexplored. The second part of the section describes the main datasets used in the field, also grouped by the task addressed.

4.1. Tasks. In the last years, different surveys have focus in different aspects of EDM systems. A recent study is described in [8]. An interesting aspect of this work is the development of a novel taxonomy of tasks in EDM. This taxonomy is used in this section as the basis to classify the papers gathered in the field of DL applied to EDM. The taxonomy comprises thirteen tasks:

(i) Predicting student performance: the objective is to estimate a value or variable describing the students' performance or the achievement of learning outcomes.

(ii) Detecting undesirable student behaviors: the focus here is on detecting undesirable student behavior, such as low motivation, erroneous actions, cheating, or dropping out.

(iii) Profiling and grouping students: the purpose is to profile students based on different variables, such as knowledge background, or to use this information to group students for various purposes.

(iv) Social network analysis: the aim is to obtain a model of students in the form of a graph, showing different possible relationships among them.

(v) Providing reports: the purpose is to find and highlight the information related to course activities which may be of use to educators and administrators, providing them with feedback.

(vi) Creating alerts for stakeholders: the objective is to predict student characteristics and detect unwanted behavior, serving as an online tool for informing stakeholders or creating alerts in real time.

(vii) Planning and scheduling: the aim is to help stakeholders in the task of planning and scheduling.

(viii) Creating courseware: the purpose is to help educators to automatically create and development course materials using students' usage information.

(ix) Developing concept maps: the objective is to develop concept maps of various aspects to help educators define the process of education.

(x) Generating recommendation: the objective is to make recommendations to any stakeholders, although the main focus is usually on helping students.

(xi) Adaptive systems: this task is related to the use of intelligent systems in computer based learning, where the system has to adapt to the user's behavior.

Table 1: Number of papers over publication venue.

Type	Publication venue	Number
Conference	International Conference on Educational Data Mining (2016, 2017, 2018)	16
	Third ACM Conference on Learning @ Scale (2016, 2017)	5
	Artificial Intelligence in Education	2
	IEEE International Conference on Data Mining Workshop (ICDMW 2015)	1
	International Symposium on Educational Technology (ISET)	1
	Seventh International Learning Analytics and Knowledge Conference	1
	Annual Conference on Neural Information Processing Systems (NIPS)	1
	Conference on Empirical Methods in Natural Language Processing (2016)	1
	26th Conference on User Modeling, Adaptation and Personalization	1
	2nd International Conference on Crowd Science and Engineering	1
	Neural Information Processing Systems, Workshop on Machine Learning for Education	1
	2nd International Conference on Innovation in Artificial Intelligence	1
	20th ACM International Conference on Multimodal Interaction	1
Journal	CoRR	4
	International Journal of Applied Engineering Research	1
	Journal of Educational Data Mining	1
	Journal of Engineering and Applied Sciences	1
	Journal of Educational Computing Research	1

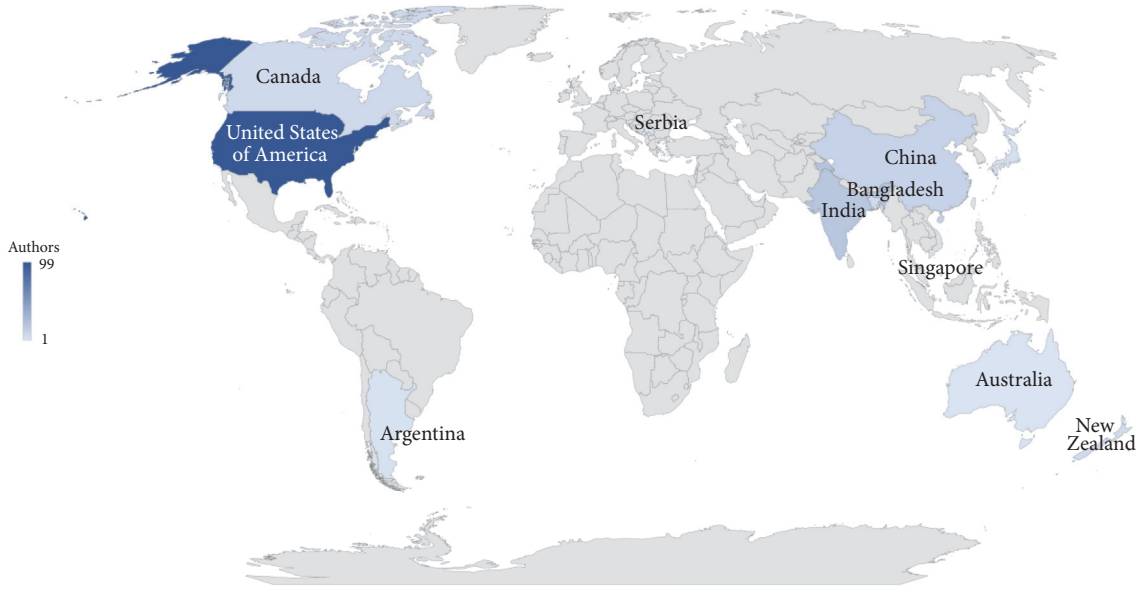


Figure 2: Choropleth map showing the density of researchers per country in the papers reviewed based on their affiliation.

(xii) Evaluation: the goal is to provide an automatic evaluation tool to help educators.

(xiii) Scientific inquiry: mostly targeted on researchers as the end users, but developed or tested theories can be used afterwards in other applications with different stakeholders.

All the works analyzed in this review fall into four of these thirteen categories: predicting student performance, detecting undesirable student behavior, generation recommendations, and evaluation. The other nine categories remain empty. Table 2 summarizes these four tasks in EDM (first column), the references to the works in the field (second column), the datasets employed (third column), and the types of datasets (fourth column). This last column specifies if the dataset has been created specifically for the experiments carried out ("Specific") or if it is a general dataset used in other works ("General"). The following subsections present

Table 2: Summary of EDM tasks, approaches, datasets, and types of datasets. "Specific" means that the dataset has been created for a specific study, and "General" means that it has been used in different publications.

Task	Reference	Dataset	Type
Predicting student performance, achievement of learning outcomes or characteristics	Lin and Chi, 2017 [11]	ITS Pyrenees	Specific
	Zhang et al., 2017 [49]	ASSISment and OLI datasets	General
	Kim et al., 2018 [26]	Udacity	Specific
	Lalwani and Agrawal, 2017 [14]	Funtoot dataset	Specific
	Okubo et al., 2017 [24]	Information Science Course dataset	Specific
	Guo et al., 2015 [23]	High schools dataset	Specific
	Sharada et al., 2018 [22]	ASSIStment 2018	General
	Wang et al., 2017 [12]	Code course dataset	Specific
	Tang et al., 2016 [21]	Kaggle Automated Essay Scoring	General
	Bendangnuksung and P., 2018 [20]	Kaggle Students' Academic Performance dataset	General
	Mao et al., 2018 [15]	ITS Pyrenees and ITS Cordillera	Specific
	Wilson et al., 2016 [50]	ASSISTment 2009-2010, KDD Cup 2010 and ITS Knewton	General
	Wilson et al., 2016 [16]	ASSISTment 2009-2010 dataset, KDD Cup 2010 dataset and ITS Knewton	General
	Khajah et al., 2016 [17]	Assistment 2009-2010 dataset, virtual student dataset, and data from Spanish and Engineering courses	General and Specific
	Xiong et al., 2016 [18]	ASSISTments 2009-2010 dataset	General
	Wang et al., 2017 [53]	KDD Cup 2015 dataset	General
	Kim et al., 2018 [27]	Udacity	Specific
	Montero et al., 2018 [13]	ASSISTment 2009-2010 dataset, KDD Cup 2010 dataset and ITS Woot Math	General and specific
	Piech et al., 2015 [10]	Virtual student dataset and Assistments 2009-2010 dataset	General
	Singh et al. 2018 [54]	Kaggle Automated Essay Scoring	General
	Alam et al., 2018 [25]	Kaggle Students' Academic Performance dataset	Specific
	Yeung and Yeung, 2018 [19]	ASSISTment 2009, ASSISTment 2015, ASSISTment Challenge, Statics2011, Simulated-5	Specific
Detecting undesirable student behaviors	Aung et al., 2018 [36]	YouTube videos of school classrooms	Specific
	Sharma et al., 2016 [34]	StyleX dataset (multimedia)	Specific
	Teruel and Alemany, 2018 [29]	ASSISTment 2009-2010 dataset and KDD Cup 2015	General
	Fei and Yeung, 2015 [28]	-	-
	Whitehill et al., 2017 [31]	HarvardX MOOCs	General
	Wang et al., 2017 [30]	Code course dataset	Specific
	Min et al., 2016 [33]	Game-based virtual learning environment Crystal Island	Specific
	Tato et al., 2017 [37]	French corpus	Specific
	Yang et al., 2018 [35]	Videos collected in unconstrained environments	Specific
	Xing and Du, 2018 [32]	Canvas project management MOOC	Specific
Generating recommendations	Wong, 2018 [39]	Student transcript records	Specific
	Abhinav et al., 2018 [38]	Learner's profile data	Specific
Evaluation	Akram et al., 2018 [44]	problem-solving dataset from game-based learning environment	Specific
	Zhang et al., 2016 [42]	Short answers from ITS Cordillera	Specific
	Taghipour and Ng, 2016 [41]	Kaggle Automated Essay Scoring	General
	Zhao et al., 2017 [40]	ASSISTment 2009-2010 and Kaggle Automated Essay Scoring	General
	Alvarado et al., 2018 [43]	Short-answer question dataset from biology course	Specific
	Choi et al., 2017 [45]	PODS dataset	Specific
	Sales et al., 2018 [46]	2015 ASSISTments Skill Builder Data	General

each task and the works related in more detail. The details about the DL implementation on each paper are described in Section 5.

4.1.1. Predicting Student Performance. One of the challenges that has gained more attention in this area is *knowledge tracing*. In this subtask the goal is to predict student's future performance based on their past activity. Piech et al. [10] were the first to introduce DL techniques to address this task, largely outperforming previous approaches based on traditional machine learning techniques. These remarkable achievement leads to other researchers to question the validity of the results. A series of works were published afterwards that were for [11–13] or against [14–19] the claims in this paper. The studies that disagree with Piech et al. tried to replicate the results of the experiments and compare them with traditional machine learning techniques in a more fair scenario, arguing that the differences between DL and previous models were not so evident. Also in the task of knowledge tracing, but away from the controversy initiated by Piech et al., the work in [20] proposed also a DL classifier to predict whether students will fail or pass an assignment.

The work by [21] leveraged a DL model to explore two different contexts within the educational domain: writing samples from students and clickstream activity within a MOOC. The use of a single model and architecture highlighted the flexibility and broad applicability of DL to large, sequential student data.

The work by [22] applied DL to a dataset obtained from a web based mathematics tutor to model student knowledge retention, i.e., the ability of the students to retain the acquired knowledge. The proposal significantly outperformed the baseline method proposed. This approach was later employed to personalize retention tests.

In [23], the authors presented a DL classifier for predicting students' performance, which took advantage of a relatively large real world students' dataset of unlabeled data. The system automatically learned multiple levels of representation and the experimental results showed the effectiveness of the method. In this line, [24] proposed a method for predicting final grades of students applying DL to the log data stored in an educational system. The log data represented the learning activities of students who used the LMS, the e-portfolio system, and the e-book system. The results showed that DL outperformed the traditional machine learning baseline proposed. Reference [25] proposed a model to categorize students into high, medium and low, to determine their learning capabilities and help them to improve their study techniques. A DL model was implemented to provide predictions based on the top features identified. Finally, [26, 27] recast the student performance prediction problem as a sequential event prediction problem and proposed a DL algorithm, called GritNet. The results showed that their proposal outperformed the baseline chosen, obtaining substantially gain in the few weeks when accurate predictions are most challenging.

4.1.2. Detecting Undesirable Student Behaviors. The works focused in the task of detecting undesirable students' behavior have faced three different subtasks: *predicting dropping out in MOOC platforms, addressing the problem of students engagement in their learning, and evaluating social functions.*

In the subtask of dropout prediction in MOOCs, [28] treated this task from a sequence labeling perspective, applying temporal models to solve the problem. Using DL techniques, they obtained significantly better performance than traditional machine learning methods for all three definitions of dropout: participation in the final week, last week of engagement, and participation in the next week. References [29, 30] defined dropout as a binary classification problem. Reference [30] combined different DL architectures in a bottom-up manner, selecting three attributes from the dataset as an input. The results showed that the proposed model could achieve comparable performance to approaches relying on feature engineering performed by experts. Reference [29] optimized a joint embedding function to represent both students and course elements into a single shared space. The results indicated that coembeddings were able to capture the latent causes involved in dropout, outperforming other disjoint and not embedded representations. Reference [31] questioned the fact that dropout prediction focuses on exploring different feature representations and classification architectures, comparing the accuracy of a standard dropout prediction architecture with clickstream features, classified by logistic regression, across a variety of different training settings in order to better understand the trade-off between accuracy and practical deployability of the classifier. Finally, [32] focused on personalize student intervention to compute the dropout probability of individual students each week. A DL model was used to build dropout models and further produce individual student dropout probabilities. Instructors could use this information to personalize and prioritize intervention for academically at-risk students. The results supported the benefits of DL for prediction and personalized intervention design on a MOOC course data.

Regarding the study of how engaged are students in their learning, in [33] the students were observed through a live feed that included the student's facial video, the student's gaze superimposed in real time over a video capture of the screen, and the student's voice as recorded through a headset microphone. To these end, a DL-based dialogue act classifier that utilizes these three data sources was implemented. Empirical results suggested that DL models that utilize game trace logs and facial action units achieved the highest predictive accuracy. In [34] the assumption was that if educational videos are not engaging, then students tend to lose interest in the course content. The authors combined audio and visual information to predict the liveliness in a video using DL. The results demonstrated significant improvement compared to traditional state-of-the-art methods. The work by [35] was focused on the movements of gaze and pose to determine the engagement intensity while watching online educational courses videos. The authors developed a DL framework that accepted multiple input features (statistical characteristics, facial descriptors, and action features) and evaluated how different modalities performed using this framework. Experimental results demonstrated the effectiveness of the method proposed. Another work addressing students engagement

was developed by [36]. They identified disengaged or distressed students, helping teachers to better recognize whether they are paying attention to the right thing or the right student in the classroom. A DL-based prototype system was developed for automated eye gaze following, which estimated for each person in the classroom where they were looking at. The proposed method could estimate the gaze target location of each person in the image with accuracy substantially better than chance and higher than other traditional baseline methods.

Finally, the work by [37] proposed a DL model to evaluate sociomoral reasoning maturity, a key social ability necessary for adaptive social functioning. This model was used in a serious game to evaluate students, outperforming traditional machine learning approaches in this context.

4.1.3. Generating Recommendation. There are two works addressing the recommendation of learning items to assist students. Both studies focused on generating personalized searches based on their preferences and curriculum planning.

Reference [38] proposed a hybrid recommendation system (called *LeCoRe*) that recommended learning opportunities to students based on their (implicit or explicit) preferences, allowing connecting them by similar interests on the platform. LeCoRe combined both content-based and collaborative filtering techniques in its phases. The learner training step applied traditional collaborative filtering algorithms and content-based DL algorithms separately. The authors concluded that the proposed framework was able to successfully model the learner's preferences. In another work, [39] focused on the less investigated problem of curriculum planning for students, providing a novel approach to this domain based on two components: a DL approach to sequential recommendations and a recommender to provide a personalized pathway to completion using sequence, constraint, and contextual parameters.

4.1.4. Evaluation. Different approaches have faced the challenge of providing evaluation tools to help teachers in the grading process. These approaches can be broadly classified in two subtasks: *automated essay scoring* (AES) and *automatic short answer grading* (ASAG).

AES systems are used to evaluate and score written student essays based on a given prompt. Reference [40] proposed a DL-based automated grading model. For each possible score in the rubric, student responses graded with the same score were collected and used as the grading criteria. The DL model learned to predict a score by computing the relevance between the students response and the grading criteria collected. In [41] the authors followed a DL approach to identify the best feature representation to learn the relation between an essay and its assigned score. Results showed an improvement with respect to other approaches requiring feature engineering.

ASAG systems automatically classify students answers as correct or not, based on a previous set of correct answers. Reference [42] studied answer-based, questions, and student models features, both individually and combined, integrating them in different machine learning models. DL obtained the best performance in their experiments. In [43], the authors compared several features for the classification of short open-ended answers, such as n-gram models, entity mentions and entity embeddings. The authors obtained inconclusive results regarding the benefits of using embeddings with respect to traditional n-grams.

Other specific subtasks related to evaluation are also faced in the DL for EDM literature. Reference [44] introduced a temporal analytics framework for stealth assessment that analyzed students' problem-solving strategies in a game-based learning environment. The authors used a DL model on a dataset of problem-solving behaviors, outperforming baseline approaches with respect to stealth assessment predictive accuracy. Reference [45] explored how a DL-based text analysis tool could help assess how students think about different moral aspects. The model was not compared in this case with traditional machine learning approaches. Finally, [46] proposed a DL method to help estimate whether students achieved skill mastery in a set of experiments using A/B tests. This proposal was not compared with traditional machine learning methods.

4.2. Datasets. All these EDM related tasks need different types of educational datasets, both for training and for evaluating the machine learning systems. Some of these datasets are related to how students learn (for example, the success of students developing different types of exercises) and others to how student interact with digital learning platforms (e.g., clickstream or eye-tracking data in MOOCs). This section presents an overview of the main datasets used for EDM in the reviewed papers, as well as other datasets developed for specific studies. These datasets will be related to the tasks identified in the previous section. This information is summarized in the last two columns of Table 2.

4.2.1. Predicting Student Performance. In order to predict student performance it is necessary a dataset of exercises with answers gathered from real students during a period of time. This is exactly the aim of ASSISTment (https://sites.google.com/site/assistmentsdata/) [47, 48]. This dataset is used in many papers to predict student performance [10, 13, 16, 18, 19, 22, 29, 46, 49, 50]. It is composed of a series of mathematics exercises offered to middle-school students through the ASSISTment platform (https://www.assistments.org/), including information such as assignment and user identification, whether the answer is correct on the first attempt or not (a binary flag indicating if the student completed the exercise correctly), the number of student attempts on a problem, answer type, etc. (The full list of features is available here: https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data.) The platform is currently up and running, with new and updated datasets released occasionally (see https://sites.google.com/site/assistmentsdata/home and https://sites.google.com/view/edm-longitudinal-workshop/home).

This dataset is often used jointly with others. For instance, [10] combined ASSISTments 2009-2010 with another two datasets: a sample of anonymized student usage interactions on Khan Academy (https://www.khanacademy.org/) (1.4 million exercises completed by 47,495 students across 69 different exercises) and a dataset of 2,000 virtual students performing the same sequence of 50 exercises drawn from 5 skills. Reference [13] also combined ASSISTments 2009-2010 dataset, in this case with KDD Cup 2010, and with a dataset collected by the Woot Math system (https://www.wootmath.com/). The KDD Cup 2010 dataset comes from an EDM Challenge in 2010 (http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp) and comprises 100 skills from 574 students. These data was extracted from the Cognitive Algebra Tutor system during 2005 and 2006 [51]. The dataset collected by the Woot Math system, a startup that develops adaptive learning environments for mathematics, consists of exercises and the correctness or not of the answers (binary outcome). Reference [18] used also ASSISTments 2009-2010, together with ASSISTments 2014-2015 and KDD Cup 2010. References [16, 50] combined also these datasets and used in addition data collected from the Knewton adaptive learning platform (https://www.knewton.com/). The work by [49] also combined ASSISTments 2009-2010, in this case with the OLI Engineering Statics dataset (https://pslcdatashop.web.cmu.edu/Project?id=48), which included college-level engineering statics. Reference [17] presented a large dataset combining different resources: the ASSISTments 2009-2010 dataset, a synthetic dataset developed by [10], a dataset of 578,726 trials from 182 middle-school students practicing Spanish exercises (translations and simple skills such as verb conjugation), and a dataset from a college-level engineering statics course comprising 189,297 trials of 1,223 exercises from 333 students [52] (https://pslcdatashop.web.cmu.edu/).

Besides this popular dataset, there are others that have been compiled for specific analysis or experiments. All of them are extracted from educational platforms or Intelligent Tutoring Systems (ITS). Regarding educational platforms, [26, 27] compiled several datasets with information about 30,000 students in Udacity (https://www.udacity.com). This data represents users taking a specific action such as watching a video, reading a text page, taking a quiz, or receiving a grade on a project at a certain time stamp. Another work that leverages educational platforms is [20], which used the Students' Academic Performance Dataset from Kaggle (https://www.kaggle.com/aljarah/xAPI-Edu-Data). This dataset consists of 500 students records collected from a learning management system (Kalboard 360) with 16 different features such as gender, nationality, place of birth, topic, visited resource, discussion group, parent answering survey, parent satisfaction, and student absent days. This resource was also used by [25].

In addition to educational platforms, different works used ITS to collect their datasets. Such is the case of [11]. They extracted information from a ITS called Pyrenees. In this case, the dataset contained information about the degree of success of 524 students answering several tests about probability. All the students received the same 12 training problems in the same order. Pyrenees was also used in [15] (68,740 data points from 475 students) together with other dataset collected from a natural language physics ITS, named Cordillera, that teaches students introductory college physics (44,323 data points from 169 students). Another ITS used in these works is Funtoot (https://www.funtoot.com/). Reference [14] used this system to develop a dataset that comprised information about knowledge tracing in online courses, such as the scope of the question (e.g., subject, topic and complexity), start time, total attempts allowed based on the student's performance, time taken, and attempts taken.

Finally, other studies used their own platforms to gather the data. Reference [23] collected real world data from 100 junior high schools. This data was a multilevel representation of student related information: demographic data (e.g., gender, age, health status, and family status), past studies, school assessment data (e.g., school type and school ranking), study data (e.g., middle-term exam, final-term exam, and average), and personal data (e.g., personality, attention and psychology related data). Reference [24] presented a specific dataset for predicting final grades of students, including information about reports, quiz answers, and logbooks of lectures of 108 students attending an Information Science course.

To sum up, either in isolation or in combination with others, the main dataset used for predicting student performance is 2009-2010 ASSITments. Other popular datasets are KDD Cup 2010 and the datasets available at DataShop repository. The rest are specific datasets used in individual studies, which extract data (mainly exercises with real answers) from educational platforms or ITS such as Khan Academy, Woot Math, Udacity, Knewton, Funtoot, and Cordillera.

4.2.2. Detecting Undesirable Student Behaviors. As shown in the previous section, the most salient task for detecting undesirable student behaviors is the study of student dropout in MOOC platforms. There is a set of general purpose datasets that have been developed to address this task.

The main dataset is the KDD Cup 2015 competition (https://biendata.com/competition/kddcup2015/). The challenge proposed in this competition was to predict student dropout on XuetangX, one of the largest MOOC platforms in China. The dataset contains, among others, information about which student enrolls in which course and activity records of the students from 39 courses. Unfortunately, it seems that the data is no longer available. This dataset was used in [29, 53]. The largest dataset for the analysis of student dropout was presented in [31]. This corpus comprises 40 MOOCs from HarvardX with information about number of registered participants and number of participants who certified. It includes additional information such as clickstream data about answers to quiz questions, play/pause/rewind events on lecture videos, and reading and writing to the discussion form. Reference [32] presented a specific dataset for student dropout analysis created from a project management MOOC course hosted by Canvas. It included information about clicks (pages, sources visited, etc.), data from the discussion forum, and quiz scores for every student.

References [12, 30] used a corpus of programming exercises (http://code.org/research) that contains 1,263,360 code submissions about multiple concepts such as loops, if-else statements and nested statements. It is important to note that this dataset is focused on the knowledge of the student (exercises and answers) rather than their behavior in the MOOC platform.

Besides these datasets focused on student dropout, other works have developed datasets for more specific tasks in the context of detecting undesirable student behavior. Related to multimodal interactions, [33] developed a dataset of students interactions within a game-based virtual learning environment called Crystal Island. Both game actions and parallel sensor data were captured to collect cognitive and affective features. This dataset includes information about student interactions in the virtual environment, but not about the student's body of knowledge. Reference [34] also developed a multimedia corpus for the analysis of liveliness of educational videos. The dataset comprises 450 one-minute video snippets featuring 50 different instructors, 10 major topics in engineering, and various accents of spoken English, all of them annotated for liveliness by multiple annotators. Reference [35] presented also a multimedia dataset for engagement prediction. It includes more than 200 videos of 5 minutes long approximately, about 78 subjects (25 female and 53 male) that have been collected in unconstrained environments including office, hotels, and open ground.

In order to detect PODS (privilege, oppression, diversity, and social justice) issues in learning environments, [45] created a domain-specific corpus of short written responses from students on PODS topic in a School of Social Work. From this corpus, authors extracted a specific PODS vocabulary. Finally, for the specific analysis of sociomoral reasoning maturity, [37] developed a corpus of 691 texts in French manually coded by experts, stating the level of maturity in a range from 5 (highest) to 1 (lowest).

4.2.3. Generating Recommendations. Two datasets from the papers reviewed fall in the category of generating recommendation sequences for learning. The first one was described in [38] and presents a dataset of learner's profile information and the courses they have enrolled or completed. The dataset consists of 5,000 unique learners and 49,202 unique course contents, resulting in a total of 2,140,476 enrollments. The second dataset addresses the curriculum planning problem. Reference [39] developed a corpus with 10 years of university student transcript records including 2.1 million transcript results, 30 degrees, 14 majors, 400 courses and 72,000 graduation records. In their research, the authors used a subset containing only undergraduate Engineering and IT students information.

4.2.4. Evaluation. As mentioned in Section 4.1.4, the task of evaluation comprises two main subtasks: automated essay scoring and automatic short answer grading. The essay scoring subtask requires real essays, written by students and graded by teachers, in order to develop systems that are able to score text essays automatically. For this purpose, the Kaggle platform has been used to obtain datasets for automated essay scoring. In fact, there were a specific competition for this task called ASAP (https://www.kaggle.com/c/asap-aes) whose dataset has been used in different works [21, 40, 54]. It consists of essays written in English by students (from Grade 7 to Grade 10), including a score for each one. The essays length is between 150 and 550 words. Reference [21] combined the Kaggle ASAP dataset with clickstream data from a BerkeleyX MOOC from Spring 2013. This is an interesting dataset since it combines content-based resources that show student knowledge with data about student behavior in an online educational platform.

The subtask of automatic short answer grading requires datasets of questions and answers from real students. Reference [42] gathered a corpus from the ITS Cordillera (already mentioned above as a resource for predicting students performance). This dataset includes 16,228 short answers selected from a total of 27,868 dialogues about physics. 61.66% of the corpus is labeled as "correct" while the rest is labeled as "incorrect". Reference [43] presented a corpus of short answer question responses from students, but in this case the topic of the course was human biology. Specifically, the authors used six questions in which students were expected to explain or describe the knowledge obtained during the course in their own words. The answers were manually evaluated by experts with labels like "correct", "incorrect", "incomplete", or "don't-know", among others. Finally, [44] presented a dataset of 244 middle-school students' problem-solving behaviors collected from interactions within a game-based learning environment. The topic of these problems was computational thinking.

5. Deep Learning

DL is undoubtedly the most trending research area in the field of artificial intelligence nowadays. DL is a subfield of machine learning that uses neural network architectures to model high-level abstractions in data. These architectures consist of multiple layers with processing units (*neurons*) that apply linear and nonlinear transformations to the input data. Different DL architectures have been developed and successfully applied to different supervised and unsupervised tasks in the broad fields of natural language processing and computer vision [55].

DL algorithms learn multiple levels of data representations, where higher-level features are derived from lower level features to form a hierarchy. For instance, in an image classification task, the DL model can take pixel values in the input layer and assign labels to the objects in the image in the output layer. Between these layers there are a set of transformation (*hidden*) layers that construct successive higher-order features that are less sensitive to conditions such as lighting and the position of the objects.

The "deep" in DL refers to the multiple transformation layers and levels of representation that lie between the network inputs and outputs. There is no de facto standard in the number of layers that makes a neural network "deep", but

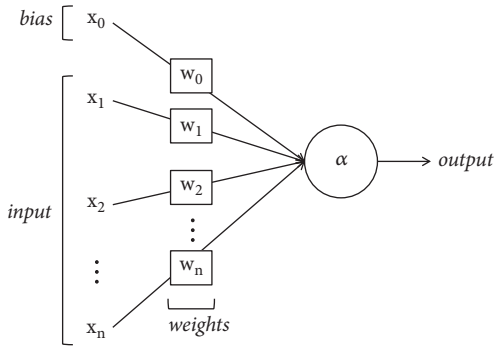


Figure 3: Simple artificial neuron.

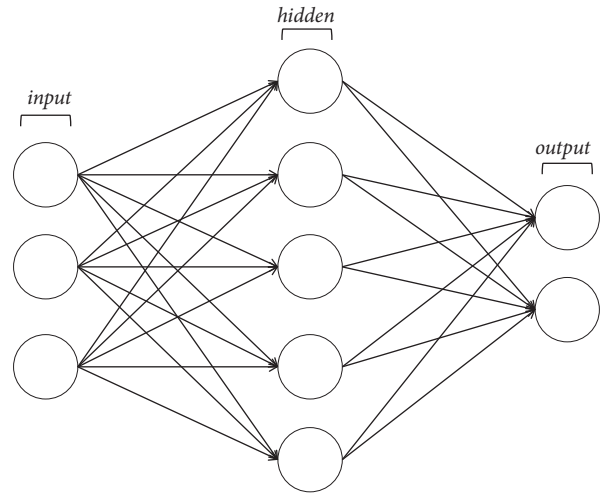


Figure 4: Basic structure of a neural network. Each circular node represents a neuron. Arrows represent connections from the output of one neuron to the input of another.

most research in the field considers that there must be more than two intermediate transformation layers [56].

Many concepts of DL were developed thirty years ago, and some of them long before. However, the most important achievements of DL have taken place in the last ten years. Although there are many factors to explain the raise of DL, it is agreed that the two main causes are the availability of massive amounts of data and the advances in computing power thanks to the use of *Graphic Processing Units* (GPU). In the first case, *big data* facilitates DL algorithms to generalize well. In the second case, GPUs allow massive parallel computing to train bigger and deeper models. Another key factor in the development of DL has been the emergence of software frameworks like TensorFlow, Theano, Keras, and PyTorch, which have allowed researches to focus in the structure of the models rather than in low-level implementation details (see Section 5.5).

Another reason for DL success is that it avoids the need for the feature engineering process. In traditional machine learning, *feature engineering* is the process of selecting the most representative features necessary for the algorithms to work, discarding noninformative attributes. This process is difficult and time-consuming since the correct choice of features is fundamental to the performance of the system [57]. DL performs *feature learning* to automatically discover the representations needed for the task at hand [58].

The following sections describe the foundations of neural networks, training process, main architectures, hyperparameter tuning, and frameworks for developing DL models. Besides providing a general introduction, all these topics will be characterized within the EDM domain, relating them to the papers reviewed.

5.1. Neural Networks. *Neural networks* are computational models based on large sets of simple artificial neurons that try to mimic the behavior observed in the axons of the neurons in human brains. Each node in the network is a neuron, which is the basic processing unit of a neural network.

The form of a simple neuron is depicted in Figure 3. The components of the neuron are input data (x_1, x_2, ..., x_n), which can be the output of another neuron in the network; *bias* (x_0), a constant value that is added to the input of the activation function of the neuron; the weights of each input (w_0, w_1, w_2, ..., w_n), identifying the relevance of the neurons in the model; and the output produced (α). The output of the neuron is computed following this equation:

$$\alpha = f\left(\sum_{i=0}^{n} w_i \cdot x_i\right), \tag{1}$$

where f is the *activation function* of the neuron. This function provides flexibility to neural networks, allowing to estimate complex nonlinear relations in the data and providing a normalization effect on the neuron output (e.g., bounding the resulting value between 0 and 1). The most widely used activation functions are *sigmoid*, *tanh* (hyperbolic tangent), and *ReLU* (Rectified Linear Unit). Each neuron is connected to many others and the links between them can increment or inhibit the activation state of the adjacent neurons.

Figure 4 shows the basic structure of a neural network. The first layer is the *input layer*, which is used to provide input data or features to the network. The *output layer* provides the predictions of the model. Depending on the problem, the activation function used in this layer differs: for binary classification, where output values are either 0 or 1, the sigmoid function is used; for multiclass classification, *softmax* (a generalization of sigmoid to multiple classes) apply; for a regression problem where there are no predefined categories, a linear function can be used.

The ReLU activation function is commonly used in hidden layers. The hidden layers can compute complex functions by cascading simpler functions. The type of hidden layers defines the different neural network architectures, such as CNN, RNN, or LSTM (see Section 5.3). The number of hidden layers determines the depth of the network. In general, networks with more hidden layers can learn more complex functions. In DL architectures, usually dozens or even hundreds of hidden layers are used, which can automatically learn as the model is trained with data.

5.2. Training Process. Any machine learning algorithm tries to assign inputs (e.g., an image) to target outputs (e.g., the "cat" label) by observing many input and output examples. As mentioned before, DL does this mapping between inputs and objective outputs (i.e., what the network is expected to produce) using artificial neural networks composed of a large number of layers forming a hierarchy.

The network learns something simple in the initial layer of the hierarchy and then sends this information to the next layer. This layer then takes this simple information, combines it with something more complex, and sends it to a third layer. This process continues, each layer building something more complex from the input received from the previous layer. The specification of what each layer is doing to the input received is stored in the weights of the layer. In order for the network to learn, it is necessary to find the weights of each layer that provides the best mapping between the input examples and the corresponding objective outputs.

Training the neural network means finding the right parameters setting (weights) for each processing unit in the network. The problem is that DL networks may potentially have millions of these parameters and finding the correct values for all of them can be a really difficult task. For example, VGG16 [59], a popular neural network architecture applied to image classification, has 138 million parameters. Initially, the weights of each neuron can be assigned randomly, or follow some initialization strategy, including unsupervised pretraining [60].

In order to control the quality of the output of the neural network, it is necessary to measure how close is the obtained output from the expected output. This task is carried out by the *loss function* of the network. This function takes the predictions of the model and the objective values and calculates how far the predicted outputs are from the objective values. The result of this function indicates how well is working the model for the specified examples. A common loss function is the *Mean Squared Error* (MSE), which measures the average of squared errors made by the neural network over all the input instances.

The goal of the training process is to find the weights that minimize the loss function. The error calculated by this function is fed back through the network, usually by means of *backpropagation*. This information is used to adjust the weights of each connection in the network in order to reduce the error. This process can be carried out by applying a general method for nonlinear optimization called *gradient descent*, in which the network computes the derivative of the loss function with respect to the weights, changing them such that the error decreases. The amount by which the weights are changed is determined by a parameter called *learning rate* (see Section 5.4).

After a number of training cycles (known as *epochs*) repeating this process, the model will usually converge to a state where the error is small and the network is considered to have learned the target function.

5.3. Architectures. Depending on the type of input (images, text, audio, etc.) there are different neural network architectures that are better suited to process that information. The number of architectures and algorithms that are used in DL is wide and varied. In this section, the most popular architectures, their common tasks, and their use in EDM will be described. Table 3 summarizes the works in EDM studied in this article (first column), the architectures implemented (second column), the baseline methods employed (third column), the evaluation measures used to compare DL approaches and baseline methods (fourth), and the performance achieved by DL methods in that comparison (fifth).

The architectures include MLP (Multilayer Perceptron), LSTM (Long Short-Term Memory), WE (Word Embeddings), CNN (Convolutional Neural Networks) and variants (VGG16 and AlexNet), FNN (Feedforward Neural Networks), RNN (Recurrent Neural Networks), autoencoder, BLSTM (Bidirectional LSTM), and MN (Memory Networks).

The baseline methods are SVD (Singular Value Decomposition), Slope One, K-NN (K-Nearest Neighbors), Majority class, RF (Random Forest), SVM (Support Vector Machine), N-grams, Random guess, LinReg (Linear Regression), DT (Decision Tree), NB (Naïve Bayes), LogReg (Logistic Regression), HMM (Hidden Markov Model), IOHMM (Input Output HMM), BKT (Bayesian Knowledge Tracing), IBKT (Intervention BKT), PFA (Principal Factor Analysis), Majority voting, CRF (Computational Random Fields), LSA (Latent Semantic Analysis), LDA (Latent Dirichlet Allocation), SVR (Support Vector Regression), BLRR (Bayesian Linear Ridge Regression), AdaBoost, GTB (Gradient Tree Boosting), GNB (Gaussian Naïve Bayes), IRT (Item Response Theory), TIRT (Temporal IRT), and HIRT (Hierarchical IRT).

Finally, evaluation measures include MAE (Mean Absolute Error), RMSE (Root Mean Square Error), Accuracy, Precision, Recall, F-measure, AUC (Area Under the Curve), Krippendorff's alpha, Log Loss (Logarithmic Loss), R^2, Gini, MPCE (Mean per Class Error), and QWK (Quadratic Weighted Kappa). The last column of this table indicates whether, in the experiments carried out in the paper, the DL approach outperformed baseline methods (">"), underperformed ("<"), or obtained similar results, with higher performance in some of the evaluations and lower performance in others ("="). The symbol "–" represents approaches that do not compare DL with traditional machine learning techniques. Instead, they present comparisons of different DL architectures [19, 29, 35, 45, 50], comparisons of different hyperparameters for the same DL architecture [31, 46], or proposals not evaluated yet [39].

5.3.1. Feedforward Neural Networks. FNNs represent the first generation of neural networks. Nodes in these networks do not form cycles, i.e., the information propagates always forward in a single direction, from the input nodes to the output nodes [61]. The main representatives of this type of networks are *perceptron* and *Multilayer Perceptron* (MLP).

Perceptrons are the simplest kind of neural network [62]. They consist of a single layer of output nodes, where inputs are sent directly to the output via a series of weights. Each node calculates the sum of the products of the weights and the inputs. If the result is above a threshold, the neuron activates; otherwise it takes the deactivated value. Single-layer

Table 3: Deep Learning approaches in the EDM field: architectures employed, baseline methods, and evaluation measures. The column Performance indicates whether the approaches outperformed baseline methods (>), underperformed (<), or obtained similar results (=).

Reference	Architecture	Baseline	Evaluation	Performance
Abhinav et al., 2018 [38]	MLP	SVD, Slope One, K-NN	MAE, RMSE	>
Akram et al., 2018 [44]	LSTM	Majority class, RF, SVM	Accuracy, Precision, Recall, F1	>
Alam et al., 2018 [25]	MLP	DT, RF, SVM, KNN	Accuracy	>
Alvarado et al., 2018 [43]	WE	N-grams	Precision, Recall, F-measure	=
Aung et al., 2018 [36]	CNN, VGG16, AlexNet	Random guess, LinReg	AUC	>
Bendangnuksung and P., 2018 [20]	FNN	DT, NB, MLP	Accuracy	>
Choi et al., 2017 [45]	WE	-	Krippendorff's alpha	–
Fei and Yeung, 2015 [28]	RNN, LSTM	SVM, LogReg, IOHMM	AUC	>
Guo et al., 2015 [23]	Autoencoder	NB, SVM, MLP	Accuracy	>
Khajah et al., 2016 [17]	LSTM	BKT	AUC	=
Kim et al., 2018 [27]	BLSTM	LogReg	AUC	>
Kim et al., 2018 [26]	BLSTM	LogReg	AUC	>
Lalwani and Agrawal, 2017 [14]	LSTM	PFA, BKT	AUC	=
Lin and Chi, 2017 [11]	RNN, LSTM	Majority voting, BKT	Accuracy, Precision, Recall, F-measure	>
Mao et al., 2018 [15]	LSTM	BKT, IBKT	RMSE, Accuracy, Recall, F-measure, AUC	=
Min et al., 2016 [33]	LSTM	CRF	Accuracy	=
Montero et al., 2018 [13]	LSTM	BKT	AUC	>
Okubo et al., 2017 [24]	LSTM	LinReg	Accuracy	>
Piech et al., 2015 [10]	RNN, LSTM	BKT	AUC	>
Sales et al., 2018 [46]	LSTM	-	-	–
Sharada et al., 2018 [22]	MLP	RF, LogReg	Log Loss, RMSE, R^2, AUC, Gini, MPCE	<
Sharma et al., 2016 [34]	CNN, AlexNet, VGG16, LSTM	SVM, HMM	Accuracy	>
Taghipour and Ng, 2016 [41]	LSTM	SVR, BLRR	QWK	>
Tang et al., 2016 [21]	LSTM	Majority class	Accuracy	=
Tato et al., 2017 [37]	CNN	SVM, NB, LSA, LDA, MLP	Accuracy, F-measure	>
Teruel and Alemany, 2018 [29]	LSTM	LSTM	AUC, RMSE, R^2	–
Wang et al., 2017 [30]	CNN, RNN	SVM, LogReg, DT, AdaBoost, GTB, RF, GNB	Precision, Recall, F-measure, AUC	=
Wang et al., 2017 [12]	LSTM	LogReg	Accuracy	>
Wang et al., 2017 [53]	LSTM	LogReg	Recall, Precission, F-measure	=
Whitehill et al., 2017 [31]	FNN	-	AUC	–
Wilson et al., 2016 [50]	RNN	IRT, TIRT, HIRT	Accuracy, AUC	<

Table 3: Continued.

Reference	Architecture	Baseline	Evaluation	Performance
Wilson et al., 2016 [16]	LSTM	-	-	–
Wong, 2018 [39]	LSTM	-	-	–
Xiong et al., 2016 [18]	LSTM	PFA, BKT	AUC, R^2	>
Xing and Du, 2018 [32]	RNN	DT, KNN, SVM	Accuracy, AUC	>
Yang et al., 2018 [35]	LSTM	-	MSE	–
Yeung and Yeung, 2018 [19]	LSTM	-	AUC	–
Zhang et al., 2016 [42]	DBN	NB, LogReg, DT, Perceptron, SVM	Accuracy, AUC, Precission, Recall, F-measure	>
Zhang et al., 2017 [49]	LSTM, Autoencoder	LSTM	AUC, R^2	>
Zhao et al., 2017 [40]	MN	FNN, SVR, BLRR, LSTM	QWK	>

perceptrons are only capable of learning linearly separable patterns. Networks without hidden layers are quiet limited in the patterns they can learn, and introducing more layers of linear units does not overcome this limitation. It is therefore necessary to introduce multiple layers of nonlinear hidden units. MLP consists of multiple layers of neurons, where each neuron in one layer has directed connections to the neurons of the following layer. In many applications, the sigmoid function is used as the activation function in these neurons.

FNNs are applicable to many areas where classical machine learning techniques have been applied, although major success have been achieved in computer vision [63] and speech recognition applications [64]. FNNs are primarily used for supervised learning tasks where the input data is neither sequential nor time-dependent, offering good results when the number of layers, neurons and training data is large enough. One of the main problems of this architecture is the possibility of ending up in a local minima of the loss function, getting a suboptimal solution to the problem at hand.

In the area of EDM, FNNs have been used for predicting students performance [20, 22] and for recommending learning opportunities to students based on their preferences [38].

Another type of FNN is *autoencoders* [65]. This architecture is similar to MLP, but in this case the output layer has the same number of neurons as the input layer. The goal is to reconstruct its own inputs instead of predicting a target value. This is an example of unsupervised learning, since no labeled data is required. Autoencoders (and its variants *stacked*, *sparse* and *denoising*) are typically used to learn compact representations of data [66]. Another application of this architecture is pretraining a deep network: a stacked autoencoder is trained in an unsupervised way and weights are obtained. Then this weight can be used for the deep network (with the same configuration in terms of hidden layers, number of neuros per layer, etc.) as a better choice rather than using randomly initialized weights [67]. Focusing in EDM, the work by [23] used a sparse autoencoder in the task of predicting students performance. They pretrained hidden layers of features using an unsupervised sparse autoencoder from unlabeled data, and then used supervised training to fine-tune the parameters of the network.

5.3.2. Convolutional Neural Networks. CNNs are multilayer neural networks particularly useful in image-processing applications [68]. In this architecture, the first layers recognize simple features in images (e.g., edges) and the last layers combine these initial features into higher-level abstractions (e.g., recognizing faces). CNNs are similar to FNNs in different aspects: they are composed of neurons where bias and weights have to be learned, each neuron has some inputs, performs a dot product, and applies an activation function, and there is a loss function in the last (fully connected) layer that measures the difference between the predicted and the expected value.

In general, a CNN is formed by an structure that contains three different types of layers: a convolutional layer that extracts features from the input (usually an image); a reduction (*pooling*) layer, which reduces the dimensionality of the extracted features through down-sampling while retaining the most important information (usually *max pooling* is applied [69]); and a fully connected classification layer, which provides the final result at the end of the network. The use of deep layers of convolution, pooling and classification, has facilitated the emergence of new applications of CNN. In addition to image processing [70], this type of networks has been applied to video recognition [71], game playing [72], and different natural language processing tasks [73].

The main advantage of CNNs is their accuracy in pattern recognition tasks, such as image recognition, requiring considerably fewer parameters than FNNs. On the negative side, they have disadvantages such as the high computation cost, the need for large amounts of training data, and the work required to properly initialize the network according to the problem addressed.

In the field of EDM, CNNs have been used in detecting undesirable student behaviors using VGG16 [59] and AlexNet [70] architectures for video analysis [36], using also VGG16 and AlexNet architectures for audio and video analysis [34],

performing text classification [37], and predicting student dropout [30].

5.3.3. Recurrent Neural Networks. A distinctive feature of FNNs is that they do not provide persistence mechanisms. RNNs address this problem by implementing a feedback loop that allows for information to persist [74]. Instead of completely feedforward connections, RNNs may have connections that feed back previous or the same layer. This feedback allows RNNs to keep a memory of past inputs.

RNNs can be thought as networks with multiple copies of themselves, in which each one passes a message to its successor. This structure makes them convenient for dealing with sequences and lists, and thus one of their common uses is modeling text. RNNs have been successfully applied to a variety of problems such as speech recognition [75], language modeling [76], and machine translation [77]. One of the main disadvantages of RNNs is the issue of vanishing gradients, where the magnitude of the gradients (values used to update the neural network weights) gets exponentially smaller (vanish) as the network back propagates, resulting in a very slow learning of the weights in the lower layers of the RNN. This makes the training process difficult in several ways: this architecture cannot be stacked into very deep models and cannot keep track of long-term dependencies. Another issue of RNNs is that they require a high performance hardware to train and run the models.

In the context of EDM, this type of networks has been used in the task of anticipate students dropout [28, 30, 32], and in the task of predicting students performance for learning gain predictions [11] and proficiency estimation [50].

There are different RNN architectures (see LSTM in the next section). The key difference is the feedback mechanisms within the network, which can manifest in a hidden layer, in the output layer or in a combination of them. RNNs can be trained with standard backpropagation or by using a variant called backpropagation through time (BPTT) [78].

5.3.4. Long Short-Term Memory Networks. LSTMs are a special type of RNN that has grown in popularity in recent years [79]. This architecture introduces the concept of memory cell, which allows to learn dependencies in the long term. The memory cell retains its value for a period of time as a function of its inputs and contains three gates that control information flow into and out of the cell: the *input gate* defines when new information can flow into the memory; the *forget gate* controls when the information stored is forgotten, allowing the cell to store new data; the *output gate* decides when the information stored in the cell is used in the output.

Each gate in the memory cell is also controlled by weights. The training algorithm (e.g., BPTT) optimizes these weights based on the resulting network output error. Recently, a simplification of LSTM called *Gated Recurrent Unit* (GRU) has been introduced [80]. This recurrent unit has fewer parameters than LSTMs, since it has two gates instead of three, lacking an output gate.

As a type of recurrent network, LSTMs are especially suitable for problems dealing with sequences. Several tasks can be added to the list of tasks previously mentioned for RNNs: text generation [81], question answering [82] and action recognition in video sequences [83], among others. In conjunction with CNNs, LSTMs have been used to produce image [84] and video [85] captioning: the CNN implements the image/video processing whereas the LSTM converts CNN output into natural language. One of the main advantages of LSTMs, compared to RNNs, is the extension of the memory that allows this architecture to remember their inputs over a long period of time. Unlike LSTMs, a RNN may leave out important information from the beginning while trying to process a paragraph of text to do predictions. LSTMs also overcome the issue of the vanishing gradient described above for RNNs. Finally, compared to this architecture, LSTMs reduce the amount of training data required to build the models.

In the set of works studied in this article, LSTM has been the most widely used architecture. In fact, it has been applied to all the EDM tasks covered by DL approaches: predicting students performance [21, 24, 53]; detecting undesirable student behaviors by predicting students dropout [28], predicting dialogue acts [33], modeling student behavior in learning platforms [29], and predicting engagement intensity [35]; generating recommendations [39]; and evaluation by doing stealth assessment [44], improving casual estimates from A/B tests [46], and automating essay scoring [41].

As already mentioned in Section 4.1.1, there is a controversy between a set of studies, falling in the task of predicting students performance, which have focused on knowledge tracing, i.e., modeling the knowledge of students as they interact with coursework. The controversy arose after the publication of *Deep Knowledge Tracing* (DKT) [10], an LSTM-based model which significantly outperformed previous approaches that used BKT and PFA. A series of works were published afterwards that were for [11–13, 19] or against [14–18] the claims in this paper. All these studies used the LSTM implementation of DKT, although some of them introduced their own variants.

5.3.5. Other Architectures. Apart from the architectures already described, other network structures have been employed in the literature reviewed on DL applied to EDM. One of these architectures is *Deep Belief Networks* (DBN), used in the task of evaluation [42]. This type of neural network has been used for image recognition, information retrieval and natural language understanding, among other tasks. The DBN is a multilayer network where each pair of connected layers is a *Restricted Boltzmann Machine* (RBM) [86]. Training in DBN occurs in two steps: unsupervised pretraining and subsequent supervised fine-tuning. In the unsupervised phase, each RBM is trained to reconstruct its input using the previous hidden layer output [87].

Memory networks (MN) have also been used in the task of evaluation [40]. MN are a new class of models designed to address the problem of learning long-term dependencies in sequential data, including a long-term memory component that can be read and written to provide an explicit memory representation for each token in the sequence [88].

Finally, *bidirectional LSTM* (BLSTM) is employed in the work developed for the task of predicting student performance [26, 27]. The difference with conventional LSTMs is that these networks only preserve information from the past, whereas BLSTMs run inputs in two ways: one from past to future and other from future to past, preserving information from the future in the backward run [89].

5.4. Hyperparameters Tuning. DL models include *hyperparameters*, which are variables set before optimizing the parameters (weights and bias) of the models. Hyperparameters can be set by hand, selected by a search algorithm (e.g., grid search or random search), or optimized applying a model-based method [90].

This section describes hyperparameters typically found when building neural networks. They have been classified in two types: those related to the training process and those related to the model itself. Although not all the studies analyzed in this article provide details about the hyperparameters used, references are provided when available.

5.4.1. Training. The hyperparameters described here that affect the training process are learning rate, batch size, momentum, weight update, and stopping criteria.

Learning Rate. The *learning rate* controls how much the weights of the network are adjusted with respect to the loss gradient. The lower the value is, the slower the algorithm traverses the downward slope. This helps to avoid missing local minima, but on the downside it takes a long time to converge and arrive at the best accuracy of the model.

The learning rate employed in the works studied ranges from a minimum of 0.0001 [34, 36] to a maximum of 0.1 [31], with other values such as 0.00025 [23] and 0.01 [19, 29, 35, 41].

Batch Size. The *batch size* defines the number of training instances that are propagated through the neural network. For instance, a set of 1000 training samples could be split in 10 batches of 100 samples. Using a batch size lower than the number of all samples has some benefits, such as requiring less memory (the network is trained using fewer samples in each propagation) and training faster (weights are updated after each propagation). The disadvantage of using a batch instead of all samples is that the smaller the batch size, the less accurate the estimate of the gradient.

Batch sizes used in the works reviewed include 10 [31, 38], 32 [19, 27, 33, 41], 48 [25], 100 [10, 11, 18], 500 [37], and 512 [23].

Momentum. *Momentum* is a popular extension of backpropagation that helps to prevent the network from falling into local minima. This technique adds a fraction of the previous weight update to the current weight. When the gradient keeps pointing in the same direction, this increases the size of the steps taken towards the minimum. When the gradient keeps changing direction, momentum will smooth out the variations.

Only three papers in EDM explicitly stated the use of momentum, all of them with a value of 0.9 [23, 35, 36].

Weight Update. DL models usually employ *stochastic gradient descent* (SGD) in the training phase. Although this is an easy to implement approach, it is difficult to tune and parallelize, making it challenging to debug and scale up DL networks. There are more sophisticated optimization methods such as *limited memory Broyden–Fletcher–Goldfarb–Shanno* (L-BFGS) and *conjugate gradient* (CG) that can speed up the process of training DL algorithms [91].

Most of the papers reviewed used SGD in the training phase [10, 18–20, 22, 27, 31–33, 36, 40, 41, 49, 50]. Other works used *Adam* [25, 38], an efficient gradient descent algorithm [92]. Finally, as an alternative to backpropagation in the training process, some studies used BPTT to train RNNs [28, 29, 34].

Stopping Criteria. There are different ways to determine the number of epochs employed to train the algorithms. If training and validation errors are high, the system is probably underfitting (it can neither model the training data nor generalize to new data), and the number of epochs can be increased. *Early stopping* is a form or regularization used to avoid overfitting. It updates the network so as to make it better fit the training data with each iteration, improving also the model performance on the validation dataset. At a certain point, improving the model fit to the training data increases generalization errors. Early stopping rules provide a guide to identify how many iterations can be run before overfitting.

Most of the works studied established a fixed number of epochs to train the algorithms: 22 [22], 50 [20, 38, 41, 49], 60 [35], 100 [11], 150 [21], and 250 [37]. In [25] the authors employed 50,000 epochs, but considering a very limited number of input features. Reference [36] used a validation set for early stopping, whereas [33] defined a strategy consisting in stopping the training if there is no improvement in the last 15 epochs (with a maximum of 100 epochs).

5.4.2. Model. The hyperparameters listed here, related to the model architecture, are depth and width of the network, initial weights, and dropout.

Depth and Width. These hyperparameters refer to the number of hidden layers (depth) and the number of hidden units (width) in the network. There is no analytical approach to setting these two parameters and choosing the best configuration for a task is sometimes a matter of trial and error. Whereas shallow neural networks (with a single hidden layer) can in theory approximate any function (according to the *universal approximation theorem* [93]) many empirical results in different tasks and domains demonstrate that adding more hidden layers improves the performance of the network. A possible explanation for this phenomenon is that the number of units in a shallow network grows exponentially with task complexity, requiring much more neurons than a deep network to achieve the same performance [2].

Since these are two key elements of a network architecture, most of the papers reviewed provide information about the depth and width of their implementation. Regarding the number of layers, most of the implementation ranges from

1 to 6 layers: 1 hidden layer [10, 13, 14, 17–19, 24, 32, 49, 50, 53], 2 hidden layers [11, 15, 20, 21, 34, 44], 3 hidden layers [22], 4 hidden layers [23, 26, 27, 37, 40, 41], 5 hidden layers [25, 31], and 6 hidden layers [30, 38]. In [35] the authors set 2 hidden layers for each modality feature (e.g., eye gaze and head pose), adding up to 8 hidden layers. The work by [36] defined 16 (since it employs the VGG16 architecture). Reference [33] implemented an LSTM with 64 layers (obtaining better results than with 32 layers). Finally, [29] experimented with different configurations of layers: 20, 50, 100, and 200.

With respect to the number of units per hidden layer, the most common value in the papers reviewed is 200 [10, 11, 14, 15, 17–19, 49], followed by 100 [22, 40, 50], 64 [33, 35], 128 [21, 27], and 256 [26, 34]. Other configurations include 5 [31], 15 [44], 20 [28], 40 [37], and 300 [20]. Some works tested different ranges of width values in their implementation: 10 to 200 [13], 50 to 300 [41], and 64 to 512 [36].

Initial Weights. The initial values assigned to the weights of the network play an important role in finding the global minima of the cost function in a deep neural network [94]. One way to do this initialization is assigning random values, although this method can potentially lead to two issues: vanishing gradient (the weight update is minor and the optimization of the loss function is slow) and exploding gradient (oscillating around the minima). There are more sophisticated approaches such as using unsupervised stacked RBMs to choose these weights.

The most common initialization procedure in the papers reviewed is to randomly select the initial weights: Gaussian distribution with zero mean and small variance [19], uniform weights in the range $[-0.1, 0.1]$ [20, 28, 44], and uniform weights in the range $[-0.05, 0.05]$ [13]. A sparse autoencoder was used for pretraining in [23]. Transfer learning [95] was used in [36] to initialize CNNs with weights pretrained on ImageNet. Finally, [31] used *Net2Net*, a technique to accelerate transfer learning from a previous network to a new one [96].

Dropout. *Dropout* is a regularization technique used in neural networks to prevent overfitting. The core of this approach is to randomly select neurons that will be ignored ("dropped out") during the training process. Their contribution to the activation of neurons in the next layer is temporally removed on the forward pass and weight updates are not applied to the neuron on the backward pass [97]. As neurons are randomly dropped out during training, other neurons have to handle the representation required to make predictions for the missing units. The result is that the neural network is less sensitive to specific weights of neurons achieving better generalization.

Some of the works studied reported dropout values in their network configurations. The most repeated values are 0.2 [11, 27, 34] and 0.5 [19, 23, 41], followed by 0.3 [29, 36]. Other values reported are 0.25 [50], 0.4 [49], 0.6 [13], and 0.7 [33]. There are three works that used dropout in their networks but did not reported the specific value of this hyperparameter [10, 18, 38].

5.5. Frameworks. Nowadays there are a large set of frameworks available for fast prototyping DL models that can efficiently run in parallel taking advantage of GPU infrastructures. In this way, researchers can focus on the architecture of the model and overlook low-level details. This section introduces the frameworks used in the DL for EDM literature, including some additional popular frameworks that have not yet been used in this domain. Note that not all the papers reviewed provide implementation details.

Keras (https://keras.io/) is the most popular framework in the articles reviewed. It was used in their implementation by [11, 14, 17, 25, 31, 38, 41, 44]. Keras provides a Python interface to facilitate the rapid prototyping of different deep neural networks, such as CNNs and RNNs, which can be executed on top of other more complex frameworks such as TensorFlow and Theano (see below). The code produced using Keras runs seamlessly on both CPUs and GPUs.

TensorFlow (https://www.tensorflow.org/) is the second most popular framework in this list. It is available for both desktop and mobile applications, and supports developing DL models using languages such as Python, C++ and R. The framework includes TensorBoard, a tool to visualize data modeling and network performance. It is supported by Google and by a large community of developers that provide numerous documentation, tutorials and guides. The works in EDM using this framework are [13, 18–20, 25, 29, 49].

Third in the list is Theano (http://deeplearning.net/software/theano/). It was the most widely used library for DL before the arrival of other competitors such as Tensorflow, Caffe, and PyTorch. It is a low-level library supporting both CPU and GPU computation. After launching the release of version 1.0, it was announced that the development and support for this tool would be discontinued. The works by [23, 30, 50] used this framework.

Caffe (http://caffe.berkeleyvision.org/) is a library written in C++ that includes a Python interface. It is specialized in the development of CNNs for image-processing tasks. One of the biggest benefits of using this library is the ability to access out-of-the-box pretrained networks from the Caffe Model Zoo (http://caffe.berkeleyvision.org/model_zoo.html). It was used by [36] for automatic eye gaze following in the classroom.

Torch (http://torch.ch/) is a relatively old machine learning library, since it was first released fifteen years ago. The primary programming language is Lua, although there is an implementation in C. It contains both DL and other traditional machine learning algorithms, supporting CUDA for parallel computation. It was used by [10, 12] to develop their DL models using Lua for the task of knowledge tracing. There is an open-source machine learning library for Python based on Torch, called PyTorch (https://pytorch.org/), which has gained increasing attention from the DL community since its release in 2016. This library was used in the work by [35].

Other relevant frameworks for DL, not used in any of the presented works, are Caffe2 (https://caffe2.ai/), Deeplearning4j (https://deeplearning4j.org/), MXNet (urlhttps://mxnet.apache.org/), Microsoft Cognitive Toolkit (https://www.microsoft.com/en-us/cognitive-toolkit/), and Chainer (https://chainer.org/).

Some works described in this article use *word embeddings* to reduce the dimensionality of the input space. Word embeddings are used in the area of natural language processing to map words (or phrases) to vectors of real numbers. This mapping can be done using neural network approaches [98]. They aim to identify semantic similarities between words based on their cooccurrence with other words in large samples of texts. The frameworks chosen for this task in the EDM field are word2vec [29, 45] and Glove (https://nlp.stanford.edu/projects/glove/) [40, 43]. Other popular frameworks to work with word embeddings are fastText (https://fasttext.cc/), although none of the works described here used it in their implementation.

6. Discussion

The first question to analyze in this section is the current status of EDM tasks with respect to the use of DL models. Based on the taxonomy of EDM applications defined by [8], the papers reviewed on the present study were categorized according to the problem addressed. This classification revealed that only 4 of the 13 tasks defined in that taxonomy have been faced using DL approaches: predicting students performance, detecting undesirable student behaviors, generating recommendations, and automatic evaluation. The other 9 tasks remain as an opportunity for researchers in the field to explore the application of DL techniques.

In the task of predicting student performance, a large sample of the papers analyzed were devoted to compare the performance of BKT (probabilistic) and DKT (deep learning) models, resulting in an interesting discussion between traditional and deep learning approaches (see Section 5.3.4). While DKT usually obtained better performance, BKT offered better interpretation of its predictions. Since DL is a very active research topic, it is expected that advances in DL will provide in a future theoretical understanding and interpretability of the models generated, and these findings will benefit all the fields where DL is applied, including EDM.

The prediction of dropping out in MOOC platforms is the subtask that has gained more attention in detecting undesirable student behaviors. Most of these studies focused on predicting student's dropout at a given point in time. These studies performed video analysis to identify the loss of interest in the contents of the course, extracting features such as the student's gaze. Including multimodal features to train DL models, such as behavioral traits (e.g., asking for help in the classroom or cheating in tests), could benefit future approaches to this task.

The third task studied, generating recommendations, was the target of two papers that focused on generating personalized searches based on the students' preferences and curriculum planning. An open challenge for future research is the recommendation of learning resources in an informal setting. The problem in this case would be the impossibility to manually structure the large amount of data that comes from sources such as expert communities and educational blogs.

Finally, in the evaluation task different frameworks were built to help teachers in the grading process, primarily focused on automatic essay scoring and short answer grading. The use of game-based environments and A/B testing has demonstrated its benefits as an automatic evaluation tools, and either would be an interesting line of research for future works.

The second relevant aspect of this work is the study of existing datasets used by DL models in educational contexts. As shown in Section 4.2, several datasets have been developed for predicting student performance and student behaviors in online platforms. Although only some of them are available (e.g., ASSISTment and KDD cup 2010 for predicting student performance, and KDD Cup 2015 for predicting student dropout), there are many online learning platforms, ITSs and MOOCs that can provide large amounts of information to train EDM systems.

Based on the literature reviewed, it seems necessary to develop specific datasets for two tasks: educational recommender systems based on data mining and automatic essay scoring. The main problem for the first task is that there is not a single "correct" sequence of learning items to recommend to a student, and this recommendation largely depends on the background knowledge, abilities, and goals of the learner. For this reason, it is necessary not only datasets with coherent sequences of learning (such as the sequences that can be found in a MOOC), but also to know which sequences are appropriate for each student profile. The second task, automatic essay scoring, is a hard challenge that requires a deep linguistic analysis to achieve automatic evaluations of texts. Although there are successful machine learning based natural language processing tools, automatic essay scoring requires a fine and deep semantic analysis in order to identify the topic of the essay, the main idea, arguments for and against, and, in general, the reasoning process carried out by the student. Unfortunately, there is no dataset available today that comprises this type of complex linguistic information that would benefit DL approaches in this task.

Finally, the last point studied in this review is the different DL models and configurations used in the EDM literature. Regarding DL architectures, LSTMs have been the most used approach, both in terms of frequency of use (59% of the papers used it) and variety of tasks covered, since it was applied in the four EDM tasks addressed by the works analyzed. In principle, this could be considered a good starting point to develop a system in any of the tasks covered. In the case of other architectures, Vanilla RNNs were used for predicting students performance and detecting undesirable student behaviors, FNNs were constrained to predicting students performance and CNNs to detect undesirable student behaviors. Other proposals considered the use of MLP, DBN, MN, and autoencoders.

The main hyperparameters of DL models were also reviewed in the previous section. Given the empirical nature of the development process of DL models, there is no one-size-fits-all solution to set the best configuration for a specific architecture, and the hyperparameters chosen will depend on the input data available and the task at hand. Among those analyzed, learning rate, batch size, and the stopping criteria (number of epochs) are considered to be critical to model performance. In theory, larger batch sizes imply

more stable gradients, facilitating higher learning rates. A larger batch sizes is also more computationally efficient, as the number of samples processed in each iteration increases. Nevertheless, a general advice with deep neural networks is to take many small steps (smaller batch sizes and learning rates) instead of fewer larger ones, although this is a design trade-off that requires experimentation. The third hyperparameter mentioned, the number of epochs, must also be properly adjusted to avoid the problem of overfitting. Another aspect to take into account is the size of the network. Adding more layers (depth) and neurons (width) can lead to more powerful models, but these architectures are also easier to overfit. A model with a large number of parameters requires also a large number of samples to achieve generalization. In this respect, more training data means almost always better DL models.

Manually choosing these hyperparameters is time-consuming and error-prone. As the models change, previous choices may no longer be the best ones. To avoid this drawback, there are a number of techniques to automatically pick the best hyperparameters (such as grid search). The summary provided in Section 5.4 can give a hint of the starting point and suitable ranges of values for these hyperparameters in the development of new architectures. In this regard, the most commonly used configuration values were: 0.0001 and 0.01 learning rate; 32 and 100 batch size; 0.9 momentum; SGD weighting update; 50 epochs stopping criteria; 1 or 2 hidden layers depth; 100 or 200 hidden units per layer width; random weight initialization; and 0.2 dropout.

It should be noted that the limited number of hidden layers in most of these works, with 79% of the implementation using 5 or less hidden layers. Indeed, according to [56], 54% of the works reviewed could be considered "shallow" neural networks, since they only include 1 or 2 hidden layers in their architectures. This suggests that there is room for applying more complex and deep architectures in the field of EDM. Popular techniques and architectures, such as transfer learning, reinforcement learning, Generative Adversarial Networks, and unified frameworks, are almost unexplored in the EDM domain.

With respect to the performance of DL techniques in these works, leaving aside the papers that do not offer a comparison between DL and traditional machine learning techniques, 67% of the works reported that DL outperformed the existing baselines, 27% showed inconclusive results (DL performed better only in some of the experiments), and only 6% reported a lower performance of DL techniques. These figures are not surprising given the successful results of DL techniques in many different domains. Nevertheless, these results are not exempt from controversy. Out of the field of EDM, there are detractors who claim that the inner mechanisms of the DL models generated are so complex that researchers often cannot explain why a model produces a particular output from a set of inputs. This controversy has also arisen in EDM, with the aforementioned arguments for and against DKT and BKT.

Taking into account the current DL techniques applied to EDM, there are many open paths to explore new approaches to this field, such as the use or transfer learning for initialization of the neural networks (only used in [36]), the use of reinforcement learning [99], a promising learning technique that reduces the need for training data, and the application of architectures such as MN, DBN, and *generative adversarial networks* (GAN), in tasks where language or image generation are required [100].

7. Conclusions

This study has reviewed the emergence of DL applications to EDM, a trend that started in 2015 with 3 papers published, increasing its presence every year so far with 17 papers published in 2018. After a systematic search, 41 works were retrieved in this area. It is worth mentioning the presence of these approaches in relevant EDM forums such as the annual International Conference in Educational Data Mining, with 7 papers published in the last edition (for a total of 16 in the last three years).

Based on the taxonomy of EDM applications defined by [8], only 4 of the 13 tasks proposed in that study have been addressed by DL techniques. This reveals that there are many open opportunities for the use of DL in unexplored EDM tasks, moreover taking into account the promising results obtained by these models in the works reviewed (67% of them reported that DL outperformed the "traditional" machine learning baselines in all their experiments).

The study carried out also included a revision of the main datasets used in the EDM tasks covered. As in other research areas, some of them are publicly available for the scientific community, which allows for reproducibility of the experiments, whereas others were developed *ad hoc* for specific studies. In the EDM field, an additional problem that exists to make the datasets freely available is the existence of sensitive information concerning (underage) students. This problem could be overcome with proper anonymization of the data.

A thorough study of DL techniques were also provided in this work, starting with an introduction to the field, an analysis of the types of DL architectures used in every task, a review of the most common hyperparameter configurations, and a list of the existing frameworks to help in the development of DL models. Since defining a DL architecture relays mostly in an empirical process, the information provided in this study can serve as a basis for starting future developments of DL applications in EDM.

Given the increasing adoption of DL techniques in EDM, this work can provide a valuable reference and a starting point for researches in both DL and EDM fields that want to leverage the potential of these techniques in the educational domain.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, *Handbook of educational data mining*, CRC Press, 2010.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Mass, USA, 2016.

[3] C. Romero and S. Ventura, "Educational data mining: a survey from 1995 to 2005," *Expert Systems with Applications*, vol. 33, no. 1, pp. 135–146, 2007.

[4] C. Romero and S. Ventura, "Educational data mining: A review of the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 6, pp. 601–618, 2010.

[5] C. Romero and S. Ventura, "Data mining in education," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 3, no. 1, pp. 12–27, 2013.

[6] R. S. Baker and Y. Yacef, "The state of educational data mining in 2009: A review and future visions," *JEDM-Journal of Educational Data Mining*, vol. 1, no. 1, pp. 3–17, 2009.

[7] A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1432–1462, 2014.

[8] B. Bakhshinategh, O. R. Zaiane, S. ElAtia, and D. Ipperciel, "Educational data mining applications and tasks: A survey of the last 10 years," *Education and Information Technologies*, vol. 23, no. 1, pp. 537–553, 2018.

[9] H. Aldowah, H. Al-Samarraie, and W. M. Fauzy, "Educational data mining and learning analytics for 21st century higher education: A review and synthesis," *Telematics and Informatics*, vol. 37, pp. 13–49, 2019.

[10] C. Piech, J. Bassen, J. Huang et al., "Deep knowledge tracing," in *Annual Conference on Neural Information Processing Systems (NIPS)*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., pp. 505–513, Curran Associates, Inc., 2015.

[11] C. Lin and M. Chi, "A comparisons of bkt, rnn and lstm for learning gain prediction," in *Artificial Intelligence in Education*, vol. 10331 of *Lecture Notes in Computer Science*, pp. 536–539, Springer International Publishing, 2017.

[12] L. Wang, A. Sy, L. Liu, and C. Piech, "Deep Knowledge Tracing On Programming Exercises," in *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale, (L@S '17)*, pp. 201–204, ACM, New York, NY, USA, April 2017.

[13] S. Montero, A. Arora, S. Kelly, B. Milne, and M. Mozer, "Does deep knowledge tracing model interactions among skills?" in *Proceedings of the 11th International Conference on Educational Data Mining*, 2018.

[14] A. Lalwani and S. Agrawal, "Few hundred parameters outperform few hundred thousand?" in *Proceedings of the 10th International Conference on Educational Data Mining*, 2017.

[15] Y. Mao, C. Lin, and M. Chi, "Deep learning vs. bayesian knowledge tracing: Student models for interventions," *Journal of Educational Data Mining*, vol. 10, no. 2, pp. 28–54, 2018.

[16] K. H. Wilson, X. Xiong, M. Khajah et al., "Estimating student proficiency: Deep learning is not the panacea," in *Neural Information Processing Systems, Workshop on Machine Learning for Education*, pp. 1–8, 2016.

[17] M. Khajah, R. V. Lindsey, and M. Mozer, "How deep is knowledge tracing?" in *Proceedings of the 9th International Conference on Educational Data Mining*, 2016.

[18] X. Xiong, S. Zhao, E. V. Inwegen, and J. Beck, "Going deeper with deep knowledge tracing," in *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 545–550, 2016.

[19] C. Yeung and D. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," 2018, https://arxiv.org/abs/1806.02180.

[20] P. P. Bendangnuksung, "Students' performance prediction using deep neural network," *International Journal of Applied Engineering Research*, vol. 13, no. 2, pp. 1171–1176, 2018.

[21] S. Tang, J. C. Peterson, and Z. A. Pardos, "Deep neural networks and how they apply to sequential education data," in *Proceedings of the Third ACM Conference on Learning @ Scale (L@S '16)*, pp. 321–324, ACM, New York, NY, USA, April 2016.

[22] N. Sharada, M. Shashi, and X. Xiong, "Modeling student knowledge retention using deep learning and random forests," *Journal of Engineering and Applied Sciences*, vol. 13, no. 6, pp. 1347–1353, 2018.

[23] B. Guo, R. Zhang, G. Xu, C. Shi, and L. Yang, "Predicting students performance in educational data mining," in *Proceedings of the International Symposium on Educational Technology, (ISET '15)*, pp. 125–128, China, July 2015.

[24] F. Okubo, T. Yamashita, A. Shimada, and H. Ogata, "A neural network approach for students' performance prediction," in *Proceedings of the the Seventh International Learning Analytics & Knowledge Conference (LAK '17)*, pp. 598-599, ACM., New York, NY, USA, March 2017.

[25] M. M. Alam, M. K. Islam, K. Mohiuddin, M. S. Kaonain, A. K. Das, and M. H. Ali, "A reduced feature based neural network approach to classify the category of students," in *Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence, (ICIAI '18)*, pp. 28–32, China, March 2018.

[26] B. Kim, E. Vizitei, and V. Ganapathi, "Gritnet 2: Real-time student performance prediction with domain adaptation," 2018, https://arxiv.org/abs/1809.06686.

[27] B. Kim, E. Vizitei, and V. Ganapathi, "Gritnet: Student performance prediction with deep learning," 2018, https://arxiv.org/abs/1804.07405.

[28] M. Fei and D.-Y. Yeung, "Temporal models for predicting student dropout in massive open online courses," in *Proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW '15)*, pp. 256–263, IEEE Computer Society, Washington, DC., USA, November 2015.

[29] M. Teruel and L. A. Alemany, "Co-embeddings for student modeling in virtual learning environments," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, (UMAP '18)*, pp. 73–80, ACM, New York, NY, USA, July 2018.

[30] W. Wang, H. Yu, and C. Miao, "Deep model for dropout prediction in MOOCs," in *Proceedings of the 2Nd International Conference on Crowd Science and Engineering, (ICCSE'17)*, pp. 26–32, New York, NY, USA, July 2017.

[31] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley, "Delving deeper into MOOC student dropout prediction," 2017, https://arxiv.org/abs/1702.06404.

[32] W. Xing and D. Du, "Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention," *Journal of Educational Computing Research*, pp. 1–24, 2018.

[33] W. Min, J. B. Wiggins, L. Pezzullo et al., "Predicting dialogue acts for intelligent virtual agents with multimodal student interaction data," in *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 454–459, 2016.

[34] A. Sharma, A. Biswas, A. Gandhi, S. Patil, and O. Deshmukh, "LIVELINET: A multimodal deep recurrent neural network to predict liveliness in educational videos," in *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 215–222, 2016.

[35] J. Yang, K. Wang, X. Peng, and Y. Qiao, "Deep recurrent multi-instance learning with spatio-temporal features for engagement

intensity prediction," in *Proceedings of the 20th ACM International Conference on Multimodal Interaction, (ICMI '18)*, pp. 594–598, Boulder, CO, USA, October 2018.

[36] A. M. Aung, A. Ramakrishnan, and J. Whitehill, "Who are they looking at? automatic eye gaze following for classroom observation video analysis," in *Proceedings of the 11th International Conference on Educational Data Mining*, pp. 166–170, Xi'an, China, May 2018.

[37] A. A. N. Tato, R. Nkambou, and A. Dufresne, "Convolutional neural network for automatic detection of sociomoral reasoning level," in *Proceedings of the 10th International Conference on Educational Data Mining*, 2017.

[38] K. Abhinav, V. Subramanian, A. Dubey, P. Bhat, and A. D. Venkat, "Lecore: A framework for modeling learner's preference," in *Proceedings of the 11th International Conference on Educational Data Mining*, 2018.

[39] C. Wong, "Sequence based course recommender for personalized curriculum planning," in *Artificial Intelligence in Education*, vol. 10948 of *Lecture Notes in Computer Science*, pp. 531–534, Springer International Publishing, 2018.

[40] S. Zhao, Y. Zhang, X. Xiong, A. Botelho, and N. Heffernan, "A memory-augmented neural model for automated grading," in *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, pp. 189–192, ACM, Cambridge, Massachusetts, USA, April 2017.

[41] K. Taghipour and H. T. Ng, "A Neural Approach to Automated Essay Scoring," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1882–1891, Association for Computational Linguistics, Austin, Texas, November 2016.

[42] Y. Zhang, R. Shah, and M. Chi, "Deep learning + student modeling + clustering: a recipe for effective automatic short answer grading," in *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 562–567, 2016.

[43] J. G. Alvarado, H. A. Ghavidel, A. Zouaq, J. Jovanovic, and J. McDonald, "A comparison of features for the automatic labeling of student answers to open-ended questions," in *Proceedings of the 11th International Conference on Educational Data Mining*, 2018.

[44] B. Akram, W. Min, E. N. Wiebe, B. W. Mott, K. Boyer, and J. C. Lester, "Improving stealth assessment in game-based learning with lstm-based analytics," in *Proceedings of the 11th International Conference on Educational Data Mining*, 2018.

[45] H. Choi, Z. Wang, C. Brooks, K. Collins-Thompson, B. G. Reed, and D. Fitch, "Social work in the classroom? A tool to evaluate topical relevance in student writing," in *Proceedings of the 10th International Conference on Educational Data Mining*, 2017.

[46] A. Sales, A. Botelho, T. Patikorn, and N. T. Heffernan, "Using big data to sharpen design-based inference in A/B tests," in *Proceedings of the 11th International Conference on Educational Data Mining*, 2018.

[47] M. Feng, N. Heffernan, and K. Koedinger, "Addressing the assessment challenge in an online system that tutors as it assesses," *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, vol. 19, no. 3, pp. 243–266, 2009.

[48] N. T. Heffernan and C. L. Heffernan, "The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching," *International Journal of Artificial Intelligence in Education*, vol. 24, no. 4, pp. 470–497, 2014.

[49] L. Zhang, X. Xiong, S. Zhao, A. Botelho, and N. T. Heffernan, "Incorporating rich features into deep knowledge tracing," in *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, pp. 169–172, ACM, New York, NY, USA, April 2017.

[50] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham, "Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation," 2016, https://arxiv.org/abs/1604.02336.

[51] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. J. Gordon, and K. R. Koedinger, "Algebra I 2005-2006. Challenge data set from KDD Cup 2010 Educational Data Mining Challenge," 2010, http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

[52] K. Koedinger, R. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper, "A data repository for the EDM community: The PSLC DataShop," in *Handbook of Educational Data Mining*, C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, Eds., Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, pp. 43–55, CRC Press, 2010.

[53] L. Wang, A. Sy, L. Liu, and C. Piech, "Learning to represent student knowledge on programming exercises using deep learning," in *Proceedings of the 10th International Conference on Educational Data Mining*, pp. 201–204, Cambridge, Mass, USA, April 2017.

[54] H. Singh, S. K. Saini, R. Chaudhry, and P. Dogga, "Modeling hint-taking behavior and knowledge state of students with multi-task learning," in *Proceedings of the 11th International Conference on Educational Data Mining*, 2018.

[55] W. G. Hatcher and W. Yu, "A survey of deep learning: platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018.

[56] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[57] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[58] G. Zhong, L. Wang, X. Ling, and J. Dong, "An overview on data representation learning: From traditional feature learning to recent deep learning," *The Journal of Finance and Data Science*, vol. 2, no. 4, pp. 265–278, 2016.

[59] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[60] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, vol. 7700 of *Lecture Notes in Computer Science*, pp. 437–478, Springer, Berlin, Germany, 2nd edition, 2012.

[61] T. L. Fine, *Feedforward Neural Network Methodology*, Springer, Berlin, Germany, 1999.

[62] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[63] R. Beale and T. Jackson, *Neural Computing - An Introduction*, CRC Press, Boca Raton, Fla, USA, 1990.

[64] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Perseus Publishing, 1991.

[65] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, 2014.

[66] S. Chandar, S. Lauly, H. Larochelle et al., "An autoencoder approach to learning bilingual word representations," in *Advances in Neural formation Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 1853–1861, Curran Associates, Inc., 2014.

[67] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[68] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[69] J. Nagi, F. Ducatelle, G. A. Di Caro et al., "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Proceedings of the IEEE International Conference on Signal and Image Processing Applications (ICSIPA'11)*, pp. 342–347, Malaysia, November 2011.

[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 60, pp. 1097–1105, Curran Associates, Inc., 2012.

[71] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[72] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[73] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[74] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 79, no. 8, pp. 2554–2558, 1982.

[75] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 6645–6649, May 2013.

[76] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 655–665, Association for Computational Linguistics, June 2014.

[77] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP '14)*, pp. 1724–1734, Qatar, October 2014.

[78] M. C. Mozer, "A focused backpropagation algorithm for temporal pattern recognition," in *Backpropagation*, vol. 3, pp. 137–169, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1995.

[79] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[80] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, vol. 2014, pp. 103–111, Association for Computational Linguistics, Doha, Qatar, October 2014.

[81] A. Graves, "Generating sequences with recurrent neural networks," 2013, https://arxiv.org/abs/1308.0850.

[82] D. Wang and E. Nyberg, "A long short-term memory model for answer sentence selection in question answering," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 707–712, Association for Computational Linguistics, China, July 2015.

[83] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[84] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," 2014, https://arxiv.org/abs/1412.6632.

[85] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence - Video to text," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 4534–4542, December 2015.

[86] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp. 194–281, MIT Press, Cambridge, MA, USA, 1986.

[87] G. E. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, article 5947, 2009.

[88] A. Bordes, S. Chopra, and J. Weston, "Memory networks," 2014, https://arxiv.org/abs/1410.3916.

[89] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[90] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 2546–2554, USA, 2011.

[91] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*, pp. 265–272, Omnipress, USA, 2011.

[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[93] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[94] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning, ICML'13*, pp. 1139–1147, 2013.

[95] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[96] T. Chen, I. J. Goodfellow, and J. Shlens, "Net2net: Accelerating learning via knowledge transfer," 2015, https://arxiv.org/abs/1511.05641.

[97] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[98] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*, pp. 3111–3119, Curran Associates Inc., USA, 2013.

[99] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[100] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Advances in Neural formation Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672–2680, Curran Associates, Inc., 2014.

Hindawi
Complexity
Volume 2019, Article ID 4057849, 14 pages
https://doi.org/10.1155/2019/4057849

Research Article

MI-Based Robust Waveform Design in Radar and Jammer Games

Bin Wang, Xu Chen, Fengming Xin, and Xin Song

Northeastern University at Qinhuangdao, China

Correspondence should be addressed to Bin Wang; wangbinneu@qq.com

Received 24 October 2018; Revised 2 February 2019; Accepted 28 February 2019; Published 26 March 2019

Guest Editor: Jose Garcia-Rodriguez

Due to the uncertainties of the radar target prior information in the actual scene, the waveform designed based on the radar target prior information cannot meet the needs of parameter estimation. To improve the performance of parameter estimation, a novel transmitted waveform design method under the hierarchical game model of radar and jammer, which maximizes the mutual information (MI) between the radar target echo and the random target spectrum response, is proposed. In the hierarchical game model of radar and jammer, the radar is in a leading position while the jammer is in a following position. The strategy of the jammer is optimized based on the radar transmitted waveform of previous moment, then the radar selects its own strategy based on the strategy of the jammer. It is generally assumed that the radar and the jammer have intercepted the real target spectrum and then the optimal jamming and the optimal transmitted waveform spectrum are obtained. However, the exact characteristic of the real target spectrum is hard to capture accurately in actual scenes. To simulate this, the real target spectrum is considered to be within an uncertainty range which is confined by known upper and lower bounds. Then, the minimax robust jamming and the maximin robust transmitted waveform are designed successively based on the MI criteria, which optimizes the performance under the most unfavorable condition of the radar and the jammer, respectively. Simulation results demonstrate that the robust transmitted waveform design method guarantees the parameter estimation performance effectively and provides useful guidance for waveform energy allocation.

1. Introduction

Cognitive radar (CR) is a new radar system concept proposed in recent years. This system is inspired by the bat echolocation, which improves the system performance of the radar through using the feedback structure from the receiver to the transmitter to optimize the transmitted waveform based on the recognition of the target and the scene [1]. The transmitted waveform of the traditional radar is independent of the environment and each transmission repeats the same waveform. Therefore the research of the traditional radar is devoted to optimizing the receiver design through radar signal processing [2]. Different from the traditional radar, CR transmitter can adjust the transmitted waveform adaptively to achieve optimal matching with the environment according to the acquired information [3]. During the past decades, many adaptive waveform design methods for radar target parameter estimation have been developed by a lot of experts and scholars. MI is a useful information metric in information theory [4], which has been widely adopted in cognitive radar and other engineering fields [5, 6]. One important method for radar waveform design is to use information theory. Researchers such as Vaidyanathan applied information theory to the radar of the MIMO system [7]. Many radar experts are devoted to improving the parameter estimation performance by boosting MI [8–10]. The innovative study in [9] optimizes the transmitted waveform through maximizing the conditional MI between the radar target echo and the random target spectrum response. Kwon et al. studied multitarget detection at low SNR, using the maximum eigenvalue of the sample covariance matrix and the correlation coefficient between the transmitted signal and the echo signal to obtain a modified full correlation detector from the perspective of average mutual information [11]. Under certain assumptions, such as fully known noise PSD and white noise, better estimation performance can be obtained by maximizing MI, as shown in [12].

However, the designed optimal waveforms under the environment of complex target model are not well known and do not consider the complex battlefield game environment, while in practice precise estimation of the real target spectrum is impossible and the game between radar and jammer is real existence in many cases. The mismatch of prior information of the real target and the ignorance of battlefield game environment might reduce the waveform performance transmitted by the radar transmitter.

In this paper, a novel transmitted waveform design technique based on MI under the environment of complex battlefield game and complicated target model is presented. Minimax robust jamming and maximin robust waveform design methods are proposed successively to reduce the impact of insufficient prior information on the performance of the designed waveform. Our main contribution is that the imprecise estimation of target spectrum [12] is considered in the optimal jamming and the optimal transmitted waveform design strategies. In addition, we also establish a hierarchical game model of radar and jammer, which regards radar as the leader and jammer as the follower. The minimax robust jamming and maximin robust transmitted waveform techniques under the established hierarchical game model above based on the MI are developed successively. In summary, first of all, given that the real target spectrum is known, the optimal jamming and optimal transmitted waveform design methods for random target based on MI are proposed successively. Secondly, by considering the uncertainty of the target spectrum, the MI-based minimax robust jamming and maximin robust transmitted waveform techniques are proposed successively. In this paper, we consider the single target model and multitarget model, respectively; then the minimax robust jamming and maximin robust transmitted waveform techniques under the two different target models above based on MI are proposed, respectively. The minimax robust jamming and maximin robust transmitted waveform design methods optimize the performance under the most unfavorable condition of the jammer and the radar transmitter, respectively. Their behaviour with regard to the uncertainty of target spectrum is also analyzed. The MI-based robust jamming and robust waveform provide useful guidance for waveform energy allocation strategy. These two waveform design techniques are easy to implement in an intelligent jammer and a cognitive radar and will have important applications in electronic warfare. In the actual situation, the minimax and maximin robust waveform design methods proposed in this paper can provide the most favorable strategies for intelligent jammer and radar, respectively, in complex target environments. And the waveform design method proposed in this paper can be well applied to the battlefield game environment which regards radar as the leader and the intelligent jammer as the follower. In electronic warfare, radar and jammer can design their own waveform, respectively, according to the strategy of their opponent, which can improve their own performance and weaken the performance of their opponent. As the method proposed in this paper assumes that the radar is more powerful than the jammer, the radar finally won this electronic warfare, and the transmitted waveform designed by radar maximizes the estimation performance of radar.

2. Signal Model and Problem Formulation

For the jammer, to impair the estimation performance, the minimization of MI means that the radar target echo contains little information on the target, which will result in poor performance of the radar transmitter. But for the radar transmitter, the maximization of the MI will improve the estimation performance.

2.1. Model of Random Target and Optimal Jamming Design Based on MI. In this subsection, to minimize the estimation performance of a general radar system, the optimal jamming design method based on MI is proposed. The model of the random target is given in Figure 1 [9, 13], where Figure 1(a) illustrates that the duration of the random target is finite. In this model, $a(t)$ denotes a window function with duration T_h and $g(t)$ represents a generalized stationary random process. Thus, the product $h(t) = a(t)g(t)$ is a generalized stationary random process within $[0, T_h]$. The random target model is shown in radar signal processing system in Figure 1(b), where $x(t)$ denotes the transmitted waveform signal and $h(t)$ represents the signal model of random target. The spectrum response of $x(t)$ can be denoted by $X(f)$, and similarly $H(f)$ is the spectrum response of $h(t)$. $r(t)$ denotes the signal model of receiver filter and $n(t)$ is a noise process with the power spectrum density (PSD) $S_{nn}(f)$. Likewise, $c(t)$ represents a jamming component which is a Gaussian random process and the PSD of $c(t)$ can be denoted by $J(f)$.

The energy spectrum variance (ESV) of $h(t)$ is represented as follows [9, 13].

$$\sigma_H{}^2(f) = \mathrm{E}\left[\left|H(f) - \mu_H(f)\right|^2\right] \tag{1}$$

In the expression of (1), E[·] represents the expectation of an input entity, and $\mu_H(f)$ denotes the mean value of $H(f)$ which is assumed to be 0.

The signal model depicted in Figure 1(b) can be applied for MI-based jamming and transmitted waveform design. Therefore, the expression of MI for the model of random target can be denoted by [13]

$$MI = T_y \int_{BW} \ln\left[1 + \frac{\sigma_H{}^2(f)\left|X(f)\right|^2}{T_y\left(J(f)\left|X(f)\right|^2 + S_{nn}(f)\right)}\right] df \tag{2}$$

where T_y represents the duration of the radar target echo $y(t)$. The energy constraint of the jamming is set to be E_X. Therefore, for the jammer to impair the performance of the radar transmitter, the optimization problem can be expressed as follows.

$$\min_{J(f)} \quad \mathrm{MI}\left(J(f)\right) \tag{3}$$

$$\text{s.t.} \quad \int_{BW} J(f)\, df \le E_X \tag{4}$$

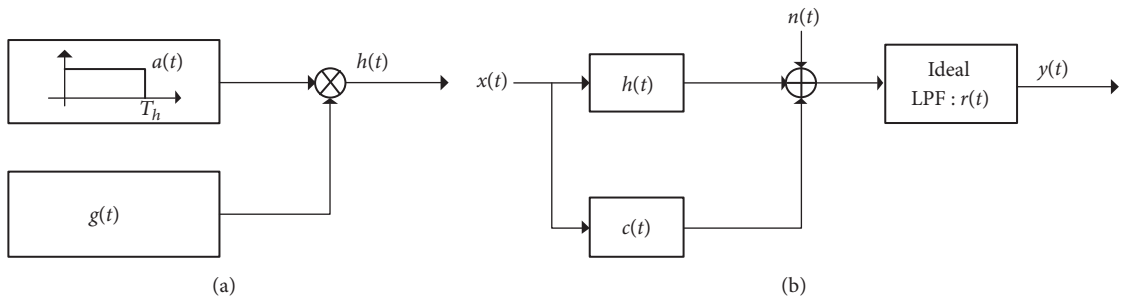


Figure 1: Random target model for waveform design based on MI. (a) Signal model for random target with finite duration T_h. (b) Signal model for waveform design based on MI.

In the expression of (4), BW is the bandwidth that the spectrum response of transmitted waveform and jamming are virtually limited to. The minimization of MI means that the radar target echo contains little information on the real target, which will result in poor performance for the radar transmitter. The equation of MI is expressed by the transmitted waveform of previous moment, the noise PSD, the target ESV, and the jamming PSD.

The optimal jamming solution obtained by Lagrange multiplier method that minimizes the MI (2) under the energy constraint (4) can be denoted by [14]

$$J(f) = \max\left[0, B(f)\left(A + D(f)\right)\right] \tag{5}$$

where

$$B(f) = -\frac{{\sigma_H}^2(f)\left|X(f)\right|^2}{2T_y \cdot S_{nn}(f) + {\sigma_H}^2(f)\left|X(f)\right|^2}, \tag{6}$$

$$D(f) = \frac{T_y {S_{nn}}^2(f) + {\sigma_H}^2(f)\left|X(f)\right|^2 S_{nn}(f)}{{\sigma_H}^2(f)\left|X(f)\right|^4}, \tag{7}$$

and A is a constant that can be derived from the energy constraint of the jamming.

$$\int_{BW} \max\left[0, B(f)\left(A + D(f)\right)\right] df \le E_X \tag{8}$$

The results show that the optimal jamming solution based on MI can be obtained by water-filling operation which assumes that the target spectrum $H(f)$ and transmitted waveform of previous moment $X(f)$ are greater than zero at each sampling frequency.

2.2. MI-Based Transmitted Waveform Design. According to the jamming solved above, for radar transmitter to improve the parameter estimation performance, MI is also adopted as the criterion. The energy constraint of the transmitted waveform is also set to be E_X. In (9), the MI is expressed by the jamming designed by jammer, the noise PSD, the target ESV, and the transmitted waveform spectrum. The designed optimal transmitted waveform should satisfy the following.

$$\max_{|X(f)|^2} \quad \mathrm{MI}\left(\left|X(f)\right|^2\right) \tag{9}$$

$$\text{s.t.} \quad \int_{BW} \left|X(f)\right|^2 df \le E_X \tag{10}$$

The maximization of MI means that the radar target echo contains much target information, which will result in rich parameter estimation performance for the radar.

The optimal transmitted waveform obtained by Lagrange multiplier method which maximizes the MI (9) under the energy constraint (10) can be written as

$$\left|\overline{X}(f)\right|^2 = \max\left[0, \overline{B}(f)\left(\overline{A} - \overline{D}(f)\right)\right] \tag{11}$$

where

$$\overline{B}(f) = \frac{{\sigma_H}^2(f)}{2T_y \cdot J(f) + {\sigma_H}^2(f)}, \tag{12}$$

$$\overline{D}(f) = \frac{T_y S_{nn}(f)}{{\sigma_H}^2(f)}, \tag{13}$$

and $\overline{A}$ is a constant that can be derived from the energy constraint of the transmitted waveform.

$$\int_{BW} \max\left[0, \overline{B}(f)\left(\overline{A} - \overline{D}(f)\right)\right] df \le E_X \tag{14}$$

The optimal transmitted waveform based on MI is obtained by performing a water-filling operation when the jamming spectrum, the target ESV, and the noise PSD are known. By using the designed transmitted waveform, the available energy can be used more efficiently in battlefield game environment, which will achieve better performance of the transmitted waveform. Simulation results show that the optimal jamming and optimal transmitted waveform actually lead to opposite energy allocation strategies.

Note that in the designed jamming and transmitted waveform above, the real target spectrum response is assumed to be fully known, while in practice the real target spectrum is difficult to capture. When the target model is blurred, the designed jamming and transmitted waveform based on target

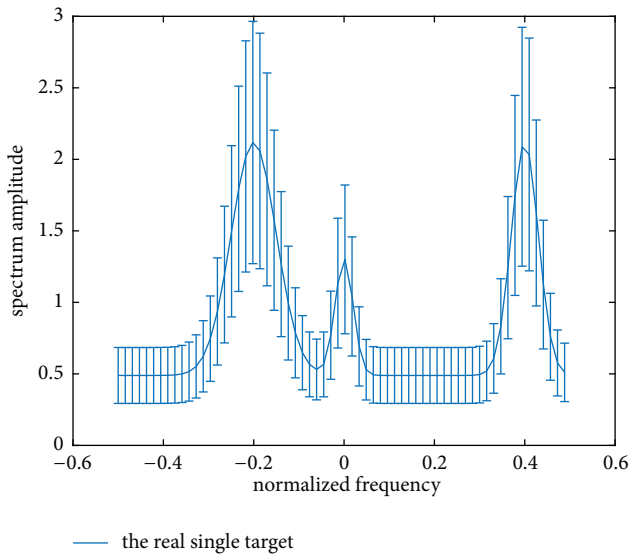


Figure 2: Model of the uncertainty range of single target spectrum.

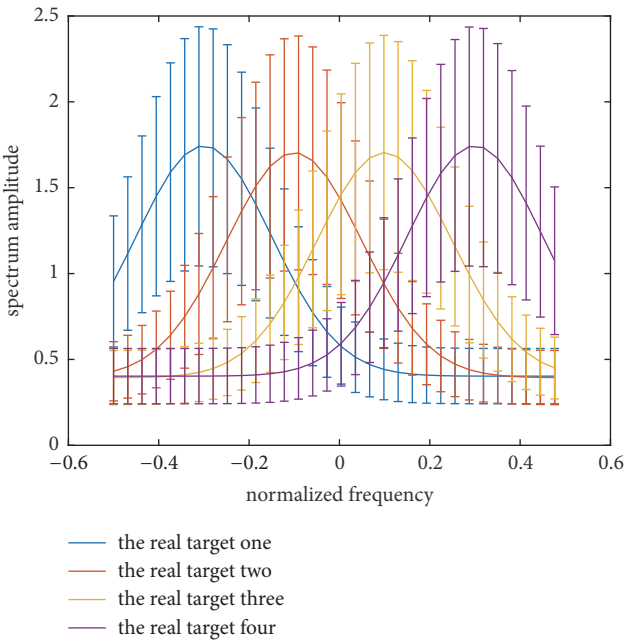


Figure 3: Model of the uncertainty range of multitarget spectrum.

prior information will not guarantee the performance of the radar or the jammer effectively, so it is critical to minimize the loss of the performance. Therefore, the robust jamming technique and robust transmitted waveform technique are considered next.

3. Maximin Robust Waveform Design

Taking the target spectrum uncertainty into account, the band model in [15] is adopted. For single target, assume that the real target spectrum exists in an uncertainty range ε where both the upper and the lower bound are known, that is,

$$H(f) \in \varepsilon = \{l_k \le H(f_k) \le u_k, k = 1, 2 \ldots, K\} \quad (15)$$

where f_k denotes the sampling frequency. The model of uncertainty single target is shown as Figure 2.

For multiple targets, assume that each real target spectrum of the multiple targets exists within an uncertainty range ε_i which is confined also by known upper and lower bounds, that is,

$$H_i(f) \in \varepsilon_i = \{l_{ik} \le H_i(f_k) \le u_{ik}, k = 1, 2 \ldots, K\} \quad (16)$$

where $i = 1, 2, 3, 4 \ldots$, which is used to distinguish different targets. The uncertainty class ε_i for each target is different. The model of uncertainty multiple targets is shown as Figure 3.

In practice, uncertainty target model is widely adopted in robust waveform design because the uncertainty range can be captured through spectrum estimation [15]. The larger the difference between the upper and the lower bound, the greater the uncertainty of the target spectrum. Moreover, pay attention to the fact that the difference in amplitude between the upper and the lower bound of the blurred target spectrum could be different at each sampling frequency.

Now the game model of radar and jammer is built: assume that the radar is in a leading position and the jammer is in a following position. The information of the leader and follower is not equal. Firstly, the strategy of the jammer is optimized according to the radar transmitted waveform of the previous moment. Then the radar selects its own strategy according to the jammer's strategy.

For each particular target spectrum, there exist an optimal jamming and an optimal transmitted waveform, respectively, for jammer and radar. However, the real target spectrum may vary in this uncertainty range, so the minimax robust technique for jammer and the maximin robust technique for radar are good approaches which guarantee the performance under the most unfavorable case. In this section, the minimax robust jamming technique and the maximin robust transmitted waveform technique for MI are proposed successively.

3.1. Minimax Robust Jamming Technique Based on MI. Let $\xi(J(f), {\sigma_H}^2(f))$ denote the optimization criterion MI. The expression of MI can be expressed by the jamming spectrum $J(f)$ and the target ESV ${\sigma_H}^2(f)$ or target spectrum $H(f)$. Note that the expressions of ${\sigma_H}^2(f)$ for single target and multiple targets are different, which will be given in Sections 3.1.1 and 3.1.2, respectively. The minimax robust jamming design method should satisfy the following [12, 15, 16].

$$\min_{J(f)} \left\{ \max_{|H(f)| \in \varepsilon} \xi\left(J(f), {\sigma_H}^2(f)\right) \Big|_{\int_{BW} J(f)df \le E_X} \right\} \quad (17)$$

Based on the theory of robust signal processing in [16], the solution of this minimax optimization problem can be

denoted as follows.

$$\begin{aligned}&\xi\left(J^{\min\max}(f),{\sigma_H}^2(f)\right)\Big|_{\int_{BW}J^{\min\max}(f)df\le E_X}\\&\le\xi\left(J^{\min\max}(f),{\sigma_{H_{worst}}}^2(f)\right)\Big|_{\int_{BW}J^{\min\max}(f)df\le E_X}\\&\le\xi\left(J(f),{\sigma_{H_{worst}}}^2(f)\right)\Big|_{\int_{BW}J(f)df\le E_X}\end{aligned}\tag{18}$$

From the right side of the inequality above, the minimax optimal jamming is optimal for the jammer when ${\sigma_H}^2(f) = {\sigma_{H_{worst}}}^2(f)$. It minimizes the performance of radar transmitter. If other jamming spectrum is adopted, the performance of the jammer will be degraded. Meanwhile, the left side of the inequality indicates that ${\sigma_{H_{worst}}}^2(f)$ is the most unfavorable target ESV for the minimax optimal jamming. If the minimax optimal jamming spectrum $J^{\min\max}(f)$ is adopted, for all target ESV in the uncertainty range ε or ε_i, the MI performance will be better than the unfavorable case, at least as good as the case of ${\sigma_H}^2(f) = {\sigma_{H_{worst}}}^2(f)$. Therefore the minimax optimal jamming for the most unfavorable target ESV within the uncertainty range is optimal. By ensuring the performance under the most unfavorable condition, the performance for all target spectra within the uncertainty range will not be worse than this case.

3.1.1. Minimax Robust Jamming Technique for Single Target Based on MI. For the jammer, the upper bound of the uncertainty range is taken as the most unfavorable target spectrum. Therefore the minimax robust jamming technique for single target based on MI should satisfy the following.

$$\min_{J(f)}\left\{\max_{|H(f)|\in\varepsilon}\mathrm{MI}\left(J(f),{\sigma_H}^2(f)\right)\Big|_{\int_{BW}J(f)df\le E_X}\right\}\tag{19}$$

Theorem 1. *The solution to the minimax optimum problem described in (19) is*

$$\widetilde{J}^{\min\max}(f)=\max\left[0,\widetilde{B}(f)\left(\widetilde{A}+\widetilde{D}(f)\right)\right]\tag{20}$$

where

$$\widetilde{B}(f)=-\frac{{\sigma_U}^2(f)\left|X(f)\right|^2}{2T_y\cdot S_{nn}(f)+{\sigma_U}^2(f)\left|X(f)\right|^2}\tag{21}$$

and

$$\widetilde{D}(f)=\frac{T_y{S_{nn}}^2(f)+{\sigma_U}^2(f)\left|X(f)\right|^2S_{nn}(f)}{{\sigma_U}^2(f)\left|X(f)\right|^4}.\tag{22}$$

${\sigma_U}^2(f) = |U(f)|^2$ denotes the unfavorable target ESV for jammer, where $U(f) = \{u_k, k = 1, 2, ..., K\}$ represents the upper bound of the target uncertainty range, and $\widetilde{A}$ is a constant which can be derived by the following.

$$\int_{BW}\max\left[0,\widetilde{B}(f)\left(\widetilde{A}+\widetilde{D}(f)\right)\right]df\le E_X\tag{23}$$

3.1.2. Minimax Robust Jamming Technique for Multiple Targets Based on MI. The minimax robust jamming technique for multiple targets based on MI should satisfy the following.

$$\min_{J(f)}\left\{\max_{|H_i(f)|\in\varepsilon_i}\mathrm{MI}\left(J(f),{\sigma_H}^2(f)\right)\Big|_{\int_{BW}J(f)df\le E_X}\right\}\tag{24}$$

Theorem 2. *The solution to the minimax optimum problem described in (24) is*

$$\widetilde{\widetilde{J}}^{\min\max}(f)=\max\left[0,\widetilde{\widetilde{B}}(f)\left(\widetilde{\widetilde{A}}+\widetilde{\widetilde{D}}(f)\right)\right]\tag{25}$$

where

$$\widetilde{\widetilde{B}}(f)=-\frac{{\sigma_U}^2(f)\left|X(f)\right|^2}{2T_y\cdot S_{nn}(f)+{\sigma_U}^2(f)\left|X(f)\right|^2}\tag{26}$$

and

$$\widetilde{\widetilde{D}}(f)=\frac{T_y{S_{nn}}^2(f)+{\sigma_U}^2(f)\left|X(f)\right|^2S_{nn}(f)}{{\sigma_U}^2(f)\left|X(f)\right|^4}.\tag{27}$$

$|U_i(f)| = \{u_{ik}, k = 1, 2, ..., K\}$ represents the upper bound of $i-th$ target uncertainty range, where ${\sigma_U}^2(f) = \sum_{i=1}^{M}P_i|U_i(f)|^2 - |\sum_{i=1}^{M}P_iU_i(f)|^2$ [13] in (26) and (27), M denotes the number of targets, P_i denotes the occurrence probability of $i-th$ target, and $\widetilde{\widetilde{A}}$ is a constant which can be derived by the following.

$$\int_{BW}\max\left[0,\widetilde{\widetilde{B}}(f)\left(\widetilde{\widetilde{A}}+\widetilde{\widetilde{D}}(f)\right)\right]df\le E_X\tag{28}$$

Note that the optimization problem in (24) which minimizes the MI for the multitarget model is similar to the problem described in (19). The difference is that the expression of ${\sigma_H}^2(f)$ is varied from ${\sigma_H}^2(f) = |H(f)|^2$ to ${\sigma_H}^2(f) = \sum_{i=1}^{M}P_i|H_i(f)|^2 - |\sum_{i=1}^{M}P_iH_i(f)|^2$.

Proof of Theorems 1 and 2. To prove the conclusion above, the optimal problem should satisfy the following.

$$\begin{aligned}&\xi\left(J^{\min\max}(f),{\sigma_H}^2(f)\right)\Big|_{\int_{BW}J^{\min\max}(f)df\le E_X}\\&\le\xi\left(J^{\min\max}(f),{\sigma_{H_{worst}}}^2(f)\right)\Big|_{\int_{BW}J^{\min\max}(f)df\le E_X}\\&\le\xi\left(J(f),{\sigma_{H_{worst}}}^2(f)\right)\Big|_{\int_{BW}J(f)df\le E_X}\end{aligned}\tag{29}$$

Firstly, we prove the right side of inequality (29). The expression of MI can be denoted as follows.

$$\begin{aligned}&MI(J(f))\\&=T_y\int_{BW}\ln\left[1+\frac{{\sigma_H}^2(f)\left|X(f)\right|^2}{T_y\left(J(f)\left|X(f)\right|^2+S_{nn}(f)\right)}\right]df\end{aligned}\tag{30}$$

The expression of ${\sigma_H}^2(f)$ in (30) is varied from single target to multiple targets, assuming that the most unfavorable target spectrum can be captured, which is $H_{worst}(f) =$

$|U(f)|$. Therefore the most unfavorable target ESV is ${\sigma_U}^2(f)$. Similarly, ${\sigma_U}^2(f)$ is the upper bound of ${\sigma_H}^2(f)$. The optimal problem is equivalent to designing the optimal jamming that minimizes the MI when the real target spectrum is $U(f)$.

We determine an objective function by using the Lagrangian multiplier technique.

$$\begin{aligned} &L\left(J\left(f\right),\lambda\right) \\ &= T_y \int_{BW} \ln\left[1 + \frac{\left|X\left(f\right)\right|^2 {\sigma_U}^2\left(f\right)}{T_y\left(S_{nn}\left(f\right) + \left|X\left(f\right)\right|^2 J\left(f\right)\right)}\right] df \\ &\quad + \lambda\left[E_X - \int_{BW} J\left(f\right) df\right] \end{aligned} \tag{31}$$

This is equivalent to minimizing $L(J(f))$ with respect to $J(f)$; the expression of (31) can be converted into

$$\begin{aligned} &L\left(J\left(f\right),\lambda\right) \\ &= T_y \int_{BW} \ln\left[1 + \frac{\left|X\left(f\right)\right|^2 {\sigma_U}^2\left(f\right)}{T_y\left(S_{nn}\left(f\right) + \left|X\left(f\right)\right|^2 J\left(f\right)\right)}\right] df \\ &\quad - \lambda \int_{BW} J\left(f\right) df \end{aligned} \tag{32}$$

where $L(J(f))$ can be denoted as follows.

$$\begin{aligned} L\left(J\left(f\right)\right) &= T_y \\ &\quad \cdot \ln\left[1 + \frac{\left|X\left(f\right)\right|^2 {\sigma_U}^2\left(f\right)}{T_y\left(S_{nn}\left(f\right) + \left|X\left(f\right)\right|^2 J\left(f\right)\right)}\right] \\ &\quad - \lambda J\left(f\right) \end{aligned} \tag{33}$$

The second order derivation of $L(J(f))$ with regard to $J(f)$ is greater than zero. Therefore, deriving $L(J(f))$ to $J(f)$ and setting it to zero yield the optimal jamming $J^{\min\max}(f)$, that is,

$$\begin{aligned} &J^{\min\max}\left(f\right) \\ &\quad = \max\left[0, -\widetilde{R}\left(f\right) + \sqrt{\widetilde{R}^2\left(f\right) - \widetilde{S}\left(f\right)\left(\widetilde{A} + \widetilde{D}\left(f\right)\right)}\right] \end{aligned} \tag{34}$$

where $\widetilde{A}$ is a constant which can be derived by

$$\begin{aligned} &\int_{BW} \max\left[0, -\widetilde{R}\left(f\right)\right. \\ &\quad \left. + \sqrt{\widetilde{R}^2\left(f\right) - \widetilde{S}\left(f\right)\left(\widetilde{A} + \widetilde{D}\left(f\right)\right)}\right] df \le E_X \end{aligned} \tag{35}$$

where

$$\widetilde{R}\left(f\right) = \frac{S_{nn}\left(f\right)}{\left|X\left(f\right)\right|^2} + \frac{{\sigma_U}^2\left(f\right)}{2T_y} \tag{36}$$

$$\widetilde{S}\left(f\right) = \frac{{\sigma_U}^2\left(f\right)}{T_y} \tag{37}$$

$$\widetilde{D}\left(f\right) = \frac{T_y {S_{nn}}^2\left(f\right) + {\sigma_U}^2\left(f\right)\left|X\left(f\right)\right|^2 S_{nn}\left(f\right)}{{\sigma_U}^2\left(f\right)\left|X\left(f\right)\right|^4} \tag{38}$$

respectively.

We define the following

$$\widetilde{Q}\left(f\right) = -\widetilde{R}\left(f\right) + \sqrt{\widetilde{R}^2\left(f\right) - \widetilde{S}\left(f\right)\left(\widetilde{A} + \widetilde{D}\left(f\right)\right)} \tag{39}$$

and use the first order Taylor approximation to (39) to yield

$$Q\left(f\right) = \widetilde{B}\left(f\right)\left(\widetilde{A} + \widetilde{D}\left(f\right)\right) \tag{40}$$

where

$$\widetilde{B}\left(f\right) = -\frac{{\sigma_U}^2\left(f\right)\left|X\left(f\right)\right|^2}{2T_y \cdot S_{nn}\left(f\right) + {\sigma_U}^2\left(f\right)\left|X\left(f\right)\right|^2}. \tag{41}$$

Thus the designed jamming can be denoted as follows.

$$J^{\min\max}\left(f\right) = \max\left[0, \widetilde{B}\left(f\right)\left(\widetilde{A} + \widetilde{D}\left(f\right)\right)\right] \tag{42}$$

Therefore we obtain the following.

$$\begin{aligned} &\xi\left(J^{\min\max}\left(f\right), {\sigma_{H_{worst}}}^2\left(f\right)\right)\Big|_{\int_{BW} J^{\min\max}(f)df \le E_X} \\ &\quad \le \xi\left(J\left(f\right), {\sigma_{H_{worst}}}^2\left(f\right)\right)\Big|_{\int_{BW} J(f)df \le E_X} \end{aligned} \tag{43}$$

Then we continue to prove that $H_{worst}(f) = |U(f)|$ is the most unfavorable target spectrum which means that ${\sigma_{Hworst}}^2(f) = {\sigma_U}^2(f)$ is the most unfavorable target ESV. Substituting the designed jamming into the expression of MI in (30) for any $H(f) \in \varepsilon$ or $H_i(f) \in \varepsilon_i$, the integral is approximated by summation, which is

$$\begin{aligned} &\xi\left(J^{\min\max}\left(f\right), {\sigma_H}^2\left(f\right)\right)\Big|_{\int_{BW} J^{\min\max}(f)df \le E_X} = T_y \cdot \sum_{k=1}^{K} \Delta f \\ &\quad \cdot \ln\left[1 + \frac{\left|X\left(f_k\right)\right|^2 {\sigma_H}^2\left(f_k\right)}{T_y\left(S_{nn}\left(f_k\right) + \left|X\left(f_k\right)\right|^2 J^{\min\max}\left(f_k\right)\right)}\right] \\ &\quad = T_y \cdot \sum_{k=1}^{K} \Delta f \cdot \ln\left[1\right. \\ &\quad \left. + \frac{\left|X\left(f_k\right)\right|^2 {\sigma_H}^2\left(f_k\right)}{T_y\left(S_{nn}\left(f_k\right) + \left|X\left(f_k\right)\right|^2 \cdot \max\left(0, Q\left(f_k\right)\right)\right)}\right] = T_y \\ &\quad \cdot \sum_{k=1}^{K} \Delta f \cdot \ln\left[1\right. \\ &\quad \left. + \frac{\left|X\left(f_k\right)\right|^2 {\sigma_H}^2\left(f_k\right)}{T_y \cdot \max\left(S_{nn}\left(f_k\right), \left|X\left(f_k\right)\right|^2 \cdot Q\left(f_k\right) + S_{nn}\left(f_k\right)\right)}\right] \\ &\quad \le T_y \cdot \sum_{k=1}^{K} \Delta f \cdot \ln\left[1\right. \\ &\quad \left. + \frac{\left|X\left(f_k\right)\right|^2 {\sigma_U}^2\left(f_k\right)}{T_y \cdot \max\left(S_{nn}\left(f_k\right), \left|X\left(f_k\right)\right|^2 \cdot Q\left(f_k\right) + S_{nn}\left(f_k\right)\right)}\right] \\ &\quad = \xi\left(J^{\min\max}\left(f\right), {\sigma_{H_{worst}}}^2\left(f\right)\right)\Big|_{\int_{BW} J^{\min\max}(f)df \le E_X} \end{aligned} \tag{44}$$

where Δf denotes the sampling frequency interval. Therefore, the most unfavorable target spectrum which maximizes the MI is $H_{worst}(f) = |U(f)|$, and similarly the most unfavorable target ESV is ${\sigma_{Hworst}}^2(f) = {\sigma_U}^2(f)$; the proof is complete. □

3.2. Robust Transmitted Waveform Design Based on MI. According to the minimax robust jamming solved above, which is the strategy of the jammer, in order to improve the estimation performance, it is time for the radar to select its own strategy.

Let $\xi(|X(f)|^2, {\sigma_H}^2(f), J^{\min\max}(f))$ represent the optimization criterion MI. The MI criteria can be expressed by the transmitted waveform spectrum $X(f)$, minimax robust jamming $J^{\min\max}(f)$, and the target ESV ${\sigma_H}^2(f)$ or target spectrum $H(f)$. Note that the expressions of ${\sigma_H}^2(f)$ for single target and multiple targets are different, which will be given in Sections 3.2.1 and 3.2.2, respectively. The maximin robust transmitted waveform design method should satisfy the following [15, 16].

$$\max_{|X(f)|^2}\left\{\min_{|H(f)|\in\varepsilon}\xi\left(\left|X(f)\right|^2, {\sigma_H}^2(f), J^{\min\max}(f)\right)\Big|_{\int_{BW}|X(f)|^2df\le E_X}\right\} \tag{45}$$

Based on the theory of robust signal processing in [16], the solution of this maximin optimization problem can be denoted as follows.

$$\begin{aligned}&\xi\left(\left|X^{\max\min}(f)\right|^2, {\sigma_H}^2(f), J^{\min\max}(f)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2df\le E_X}\\ &\ge \xi\left(\left|X^{\max\min}(f)\right|^2, {\sigma_{H_{worst}}}^2(f), J^{\min\max}(f)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2df\le E_X} \ge \xi\left(\left|X(f)\right|^2, {\sigma_{H_{worst}}}^2(f), J^{\min\max}(f)\right)\Big|_{\int_{BW}|X(f)|^2df\le E_X}\end{aligned} \tag{46}$$

From the right side of the inequality above, the maximin optimal transmitted waveform is optimal for the radar transmitter when ${\sigma_H}^2(f) = {\sigma_{H_{worst}}}^2(f)$. It maximizes the performance of radar transmitter. If another waveform spectrum is adopted, the performance of the radar will be degraded. Meanwhile, the left side of the inequality indicates that ${\sigma_{H_{worst}}}^2(f)$ is the most unfavorable target ESV for the maximin optimal transmitted waveform. If the maximin optimal transmitted waveform spectrum $|X^{\max\min}(f)|^2$ is adopted, for all target ESV in the uncertainty range ε or ε_i, the MI performance will be better than the unfavorable case, at least as good as the case of ${\sigma_H}^2(f) = {\sigma_{H_{worst}}}^2(f)$. Therefore the maximin optimal transmitted waveform for the most unfavorable target ESV within the uncertainty range is optimal. By ensuring the performance under the most unfavorable condition, the performance for all target spectra within the uncertainty range will not be worse than this case.

3.2.1. Robust Transmitted Waveform Design for Single Target Based on MI. For the radar, the lower bound of the uncertainty range is taken as the most unfavorable target spectrum. Therefore the maximin robust transmitted waveform technique for single target based on MI should satisfy the following.

$$\max_{|X(f)|^2}\left\{\min_{|H(f)|\in\varepsilon}\xi\left(\left|X(f)\right|^2, {\sigma_H}^2(f), J^{\min\max}(f)\right)\Big|_{\int_{BW}|X(f)|^2df\le E_X}\right\} \tag{47}$$

Theorem 3. *The solution to the maximin optimum problem described in (47) is*

$$\left|\widehat{X}^{\max\min}(f)\right|^2 = \max\left[0, \widehat{B}(f)\left(\widehat{A} - \widehat{D}(f)\right)\right] \tag{48}$$

where

$$\widehat{B}(f) = \frac{{\sigma_L}^2(f)}{2T_y \cdot J^{\min\max}(f) + {\sigma_L}^2(f)} \tag{49}$$

and

$$\widehat{D}(f) = \frac{T_y S_{nn}(f)}{{\sigma_L}^2(f)}. \tag{50}$$

${\sigma_L}^2(f) = |L(f)|^2$ *denotes the unfavorable target ESV for radar, where* $|L(f)| = \{l_k, k = 1, 2, ..., K\}$ *represents the lower bound of the single target spectrum uncertainty range, and* $\widehat{A}$ *is a constant which can be derived by the following.*

$$\int_{BW}\max\left[0, \widehat{B}(f)\left(\widehat{A} - \widehat{D}(f)\right)\right]df \le E_X \tag{51}$$

3.2.2. Robust Transmitted Waveform Design for Multiple Targets Based on MI. The maximin robust transmitted waveform technique for multiple targets based on MI should satisfy the following.

$$\max_{|X(f)|^2}\left\{\min_{|H_i(f)|\in\varepsilon_i}\xi\left(\left|X\left(f\right)\right|^2,{\sigma_H}^2\left(f\right),J^{\min\max}\left(f\right)\right)\Big|_{\int_{BW}|X(f)|^2df\le E_X}\right\} \tag{52}$$

Theorem 4. *The solution to the maximin optimum problem described in (52) is*

$$\left|\widehat{\widehat{X}}^{\max\min}\left(f\right)\right|^2=\max\left[0,\widehat{\widehat{B}}\left(f\right)\left(\widehat{\widehat{A}}-\widehat{\widehat{D}}\left(f\right)\right)\right] \tag{53}$$

where

$$\widehat{\widehat{B}}\left(f\right)=\frac{{\sigma_L}^2\left(f\right)}{2T_y\cdot J^{\min\max}\left(f\right)+{\sigma_L}^2\left(f\right)} \tag{54}$$

and

$$\widehat{\widehat{D}}\left(f\right)=\frac{T_yS_{nn}\left(f\right)}{{\sigma_L}^2\left(f\right)}. \tag{55}$$

$|L_i(f)|=\{l_{ik},k=1,2,...,K\}$ *denotes the lower bound of* $i-th$ *target spectrum uncertainty range, where* ${\sigma_L}^2(f)=\sum_{i=1}^{M}P_i|L_i(f)|^2-|\sum_{i=1}^{M}P_iL_i(f)|^2$ *in (54) and (55), and* $\widehat{\widehat{A}}$ *is a constant which can be derived by the following.*

$$\int_{BW}\max\left[0,\overline{B}\left(f\right)\left(A-\overline{D}\left(f\right)\right)\right]df\le E_X \tag{56}$$

Note that the optimization problem in (52) which maximizes the MI for the multitarget model is similar to the problem described in (47). The difference is that the expression of ${\sigma_H}^2(f)$ *is varied from* ${\sigma_H}^2(f)=|H(f)|^2$ *to* ${\sigma_H}^2(f)=\sum_{i=1}^{M}P_i|H_i(f)|^2-|\sum_{i=1}^{M}P_iH_i(f)|^2$.

Proof of Theorems 3 and 4. To prove the conclusion above, the optimal problem should satisfy the following.

$$\begin{aligned}&\xi\left(\left|X^{\max\min}\left(f\right)\right|^2,{\sigma_H}^2\left(f\right),\right.\\&\left.J^{\min\max}\left(f\right)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2df\le E_X}\\&\ge\xi\left(\left|X^{\max\min}\left(f\right)\right|^2,{\sigma_{H_{worst}}}^2\left(f\right),\\&J^{\min\max}\left(f\right)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2df\le E_X}\ge\xi\left(\left|X\left(f\right)\right|^2,\\&{\sigma_{H_{worst}}}^2\left(f\right),J^{\min\max}\left(f\right)\right)\Big|_{\int_{BW}|X(f)|^2df\le E_X}\end{aligned} \tag{57}$$

Firstly, we prove the right side of inequality (57). The expression of MI can be denoted as follows.

$$\begin{aligned}MI\left(\left|X\left(f\right)\right|^2\right)&=T_y\int_{BW}\ln\left[1\right.\\&\left.+\frac{{\sigma_H}^2\left(f\right)\left|X\left(f\right)\right|^2}{T_y\left(J^{\min\max}\left(f\right)\left|X\left(f\right)\right|^2+S_{nn}\left(f\right)\right)}\right]df\end{aligned} \tag{58}$$

The expression of ${\sigma_H}^2(f)$ in (58) is still different from single target to multiple targets, supposing that the most unfavorable target spectrum can be captured, which is $H_{worst}(f)=|L(f)|$. Therefore the most unfavorable target ESV is ${\sigma_L}^2(f)$. Similarly, ${\sigma_L}^2(f)$ is the lower bound of ${\sigma_H}^2(f)$. The optimal problem is equivalent to designing the optimal transmitted waveform that minimizes the MI when the real target spectrum is $L(f)$.

We determine an objective function by using the Lagrangian multiplier technique.

$$\begin{aligned}L\left(\left|X\left(f\right)\right|^2,\lambda\right)&=T_y\int_{BW}\ln\left[1\right.\\&\left.+\frac{\left|X\left(f\right)\right|^2{\sigma_L}^2\left(f\right)}{T_y\left(S_{nn}\left(f\right)+\left|X\left(f\right)\right|^2J^{\min\max}\left(f\right)\right)}\right]df\\&+\lambda\left[E_X-\int_{BW}\left|X\left(f\right)\right|^2df\right]\end{aligned} \tag{59}$$

This is equivalent to maximizing $L(|X(f)|^2)$ with respect to $|X(f)|^2$; the expression of (59) can be converted into

$$\begin{aligned}L\left(\left|X\left(f\right)\right|^2,\lambda\right)&=T_y\int_{BW}\ln\left[1\right.\\&\left.+\frac{\left|X\left(f\right)\right|^2{\sigma_L}^2\left(f\right)}{T_y\left(S_{nn}\left(f\right)+\left|X\left(f\right)\right|^2J^{\min\max}\left(f\right)\right)}\right]df\\&-\lambda\int_{BW}\left|X\left(f\right)\right|^2df\end{aligned} \tag{60}$$

where $L(|X(f)|^2)$ can be denoted as follows.

$$L\left(\left|X\left(f\right)\right|^2\right) = T_y \cdot \ln\left[1 + \frac{\left|X\left(f\right)\right|^2 {\sigma_L}^2\left(f\right)}{T_y\left(S_{nn}\left(f\right) + \left|X\left(f\right)\right|^2 J^{\min\max}\left(f\right)\right)}\right] - \lambda\left|X\left(f\right)\right|^2 \tag{61}$$

The second order derivation of $L(|X(f)|^2)$ with regard to $|X(f)|^2$ is greater than zero. Therefore, deriving $L(|X(f)|^2)$ to $|X(f)|^2$ and setting it to zero yield the optimal transmitted waveform $|X^{\max\min}(f)|^2$, that is,

$$\left|X^{\max\min}\left(f\right)\right|^2 = \max\left[0, -\widehat{R}\left(f\right) + \sqrt{\widehat{R}^2\left(f\right) + \widehat{S}\left(f\right)\left(\widehat{A} - \widehat{D}\left(f\right)\right)}\right]. \tag{62}$$

$\widehat{A}$ is a constant which can be derived by

$$\int_{BW} \max\left[0, -\widehat{R}\left(f\right) + \sqrt{\widehat{R}^2\left(f\right) + \widehat{S}\left(f\right)\left(\widehat{A} - \widehat{D}\left(f\right)\right)}\right] df \le E_X \tag{63}$$

where

$$\widehat{R}\left(f\right) = \frac{S_{nn}\left(f\right)\left(2T_y \cdot J^{\min\max}\left(f\right) + {\sigma_L}^2\left(f\right)\right)}{2J^{\min\max}\left(f\right)\left(T_y \cdot J^{\min\max}\left(f\right) + {\sigma_L}^2\left(f\right)\right)} \tag{64}$$

$$\widehat{S}\left(f\right) = \frac{S_{nn}\left(f\right){\sigma_L}^2\left(f\right)}{J^{\min\max}\left(f\right)\left(T_y \cdot J^{\min\max}\left(f\right) + {\sigma_L}^2\left(f\right)\right)} \tag{65}$$

$$\widehat{D}\left(f\right) = \frac{S_{nn}\left(f\right)}{{\sigma_L}^2\left(f\right)T_y} \tag{66}$$

respectively.

We define the following

$$\widehat{N}\left(f\right) = -\widehat{R}\left(f\right) + \sqrt{\widehat{R}^2\left(f\right) + \widehat{S}\left(f\right)\left(\widehat{A} - \widehat{D}\left(f\right)\right)} \tag{67}$$

and use the first order Taylor approximation to (67) to yield

$$N\left(f\right) = \widehat{B}\left(f\right)\left(\widehat{A} - \widehat{D}\left(f\right)\right) \tag{68}$$

where

$$\widehat{B}\left(f\right) = \frac{{\sigma_L}^2\left(f\right)}{2T_y \cdot J^{\min\max}\left(f\right) + {\sigma_L}^2\left(f\right)}. \tag{69}$$

Thus the designed transmitted waveform can be denoted as follows.

$$\left|X^{\max\min}\left(f\right)\right|^2 = \max\left[0, \widehat{B}\left(f\right)\left(\widehat{A} - \widehat{D}\left(f\right)\right)\right] \tag{70}$$

Therefore we obtain the following.

$$\xi\left(\left|X^{\max\min}\left(f\right)\right|^2, {\sigma_{H_{worst}}}^2\left(f\right), J^{\min\max}\left(f\right)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2 df \le E_X} \ge \xi\left(\left|X\left(f\right)\right|^2, {\sigma_{H_{worst}}}^2\left(f\right), J^{\min\max}\left(f\right)\right)\Big|_{\int_{BW}|X(f)|^2 df \le E_X} \tag{71}$$

Then we continue to prove that $H_{worst}(f) = |L(f)|$ is the most unfavorable target spectrum which means that ${\sigma_{Hworst}}^2(f) = {\sigma_L}^2(f)$ is the most unfavorable target ESV. Substituting the designed transmitted waveform into the expression of MI in (58) for any $H(f) \in \varepsilon$ or $H_i(f) \in \varepsilon_i$, the integral is approximated by summation, which is as follows.

$$\begin{aligned}
&\xi\left(\left|X^{\max\min}\left(f\right)\right|^2, {\sigma_H}^2\left(f\right), J^{\min\max}\left(f\right)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2 df \le E_X} = T_y \cdot \sum_{k=1}^{K}\Delta f \\
&\cdot \ln\left[1 + \frac{\left|X^{\max\min}\left(f_k\right)\right|^2 {\sigma_H}^2\left(f_k\right)}{T_y\left(S_{nn}\left(f_k\right) + \left|X^{\max\min}\left(f_k\right)\right|^2 J^{\min\max}\left(f_k\right)\right)}\right]\Bigg|_{\int_{BW}|X^{\max\min}(f_k)|^2 df \le E_X} = T_y \cdot \sum_{k=1}^{K}\Delta f \cdot \ln\left[1\right. \\
&\left. + \frac{\max\left(0, N\left(f_k\right)\right) \cdot {\sigma_H}^2\left(f_k\right)}{T_y\left(S_{nn}\left(f_k\right) + \max\left(0, N\left(f_k\right)\right) \cdot J^{\min\max}\left(f_k\right)\right)}\right] = T_y \cdot \sum_{k=1}^{K}\Delta f \cdot \ln\left[1\right. \\
&\left. + \frac{\max\left(0, N\left(f_k\right) \cdot {\sigma_H}^2\left(f_k\right)\right)}{T_y \cdot \max\left(S_{nn}\left(f_k\right), N\left(f_k\right) \cdot J^{\min\max}\left(f_k\right) + S_{nn}\left(f_k\right)\right)}\right] \ge T_y \cdot \sum_{k=1}^{K}\Delta f \cdot \ln\left[1\right. \\
&\left. + \frac{\max\left(0, N\left(f_k\right) \cdot {\sigma_L}^2\left(f_k\right)\right)}{T_y \cdot \max\left(S_{nn}\left(f_k\right), N\left(f_k\right) \cdot J^{\min\max}\left(f_k\right) + S_{nn}\left(f_k\right)\right)}\right] = \xi\left(\left|X^{\max\min}\left(f\right)\right|^2, {\sigma_{H_{worst}}}^2\left(f\right), J^{\min\max}\left(f\right)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2 df \le E_X}
\end{aligned} \tag{72}$$

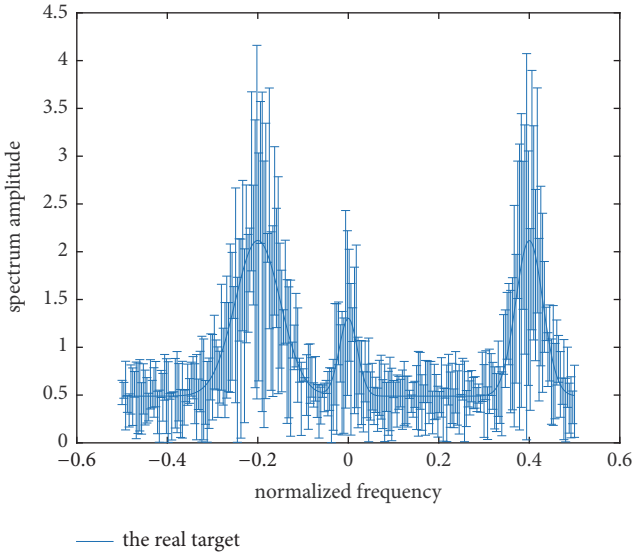


Figure 4: Bounded single target spectrum samples.

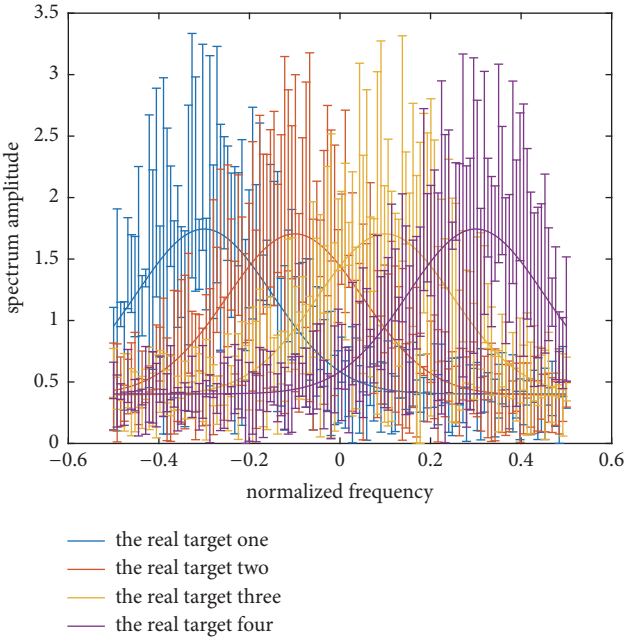


Figure 5: Bounded multitarget spectrum samples.

Therefore, the most unfavorable target spectrum which minimizes the MI is $H_{worst}(f) = |L(f)|$, and similarly the most unfavorable target ESV is ${\sigma_{Hworst}}^2(f) = {\sigma_L}^2(f)$; this guarantees that

$$\begin{aligned}&\xi\left(\left|X^{\max\min}(f)\right|^2, {\sigma_H}^2(f), J^{\min\max}(f)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2 df \le E_X}\\ &\ge \xi\left(\left|X^{\max\min}(f)\right|^2, {\sigma_{H_{worst}}}^2(f), J^{\min\max}(f)\right)\Big|_{\int_{BW}|X^{\max\min}(f)|^2 df \le E_X},\end{aligned} \tag{73}$$

which is the left side of (57). Thus, the proof of Theorems 3 and 4 is complete. □

For the minimax robust jamming, the most unfavorable case is the upper bound of the target uncertainty range, and the lower bound of the uncertainty range is the most unfavorable case for the maximin robust transmitted waveform. The MI performance of the jammer and the radar for other target ESV within this uncertainty range will be better than the performance of these two unfavorable cases. Therefore, considering the uncertainty range of the target spectrum can optimize the system performance of the jammer and the radar. Through considering the hierarchical game model of radar and jammer, the maximin robust transmitted waveform is designed based on the minimax robust jamming. Although the jamming designed by the jammer can greatly impair the performance of the radar system, the radar is in a leading position and the jamming is intercepted by the radar system. Therefore, the transmitted waveform designed by the radar transmitter can finally guarantee the performance of the radar system. Furthermore, the robust jamming and robust transmitted waveform techniques based on MI provide useful guidance for waveform energy allocation.

4. Simulation and Results

To demonstrate the validity of the MI-based robust jamming and robust transmitted waveform techniques proposed in this paper, a lot of simulation analyses are performed. The uncertainty ranges of the single target and multitarget spectrum are presented in Figures 4 and 5, respectively, where the real single target and multitarget spectrum are denoted by the solid line. The main energy of the real single target is allocated near the normalized frequency -0.2, 0, and 0.4. For each target of the nominal multiple targets, the main energy is allocated near the normalized frequency 0.2, 0.4, 0.6, and 0.8, respectively, with the occurrence probability 0.1, 0.2, 0.3, and 0.4. The upper and the lower bound at each sampling frequency are represented by the deviation bounds. The amplitude of the upper bound is the real amplitude that added a random value, and similarly the lower bound is the real amplitude that subtracted a random value.

In Figures 6 and 7, the total energy of the transmitted waveform of previous moment is 1W, and the main energy is allocated near the normalized frequency of 0.3; the target spectrum response $H(f)$ in Figure 6 is the same as the solid line in Figure 4. As ${\sigma_H}^2(f)$ and $H(f)$ have similar shapes, ${\sigma_H}^2(f)$ is not presented in Figure 6. The real ESV of multiple targets is illustrated in Figure 8. The duration of the target echo $y(t)$ is supposed to be $T_y = 1.5s$. The energy constraint of the noise signal is 1W. The optimal jamming under the real target spectrum and the robust jamming under the most unfavorable target spectrum are also illustrated in Figures 6 and 7. Both the two jamming energy constraints are assumed to be 1W. In Figure 6, both the two jamming techniques distribute the finite energy in frequency bands where both the real target spectrum response and the transmitted waveform of previous moment are relatively strong. But in Figure 7,

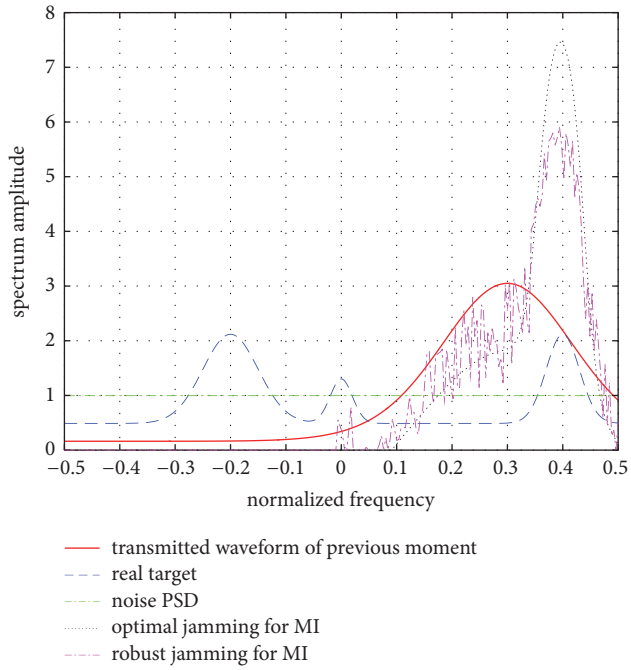


Figure 6: Jamming results for single target.

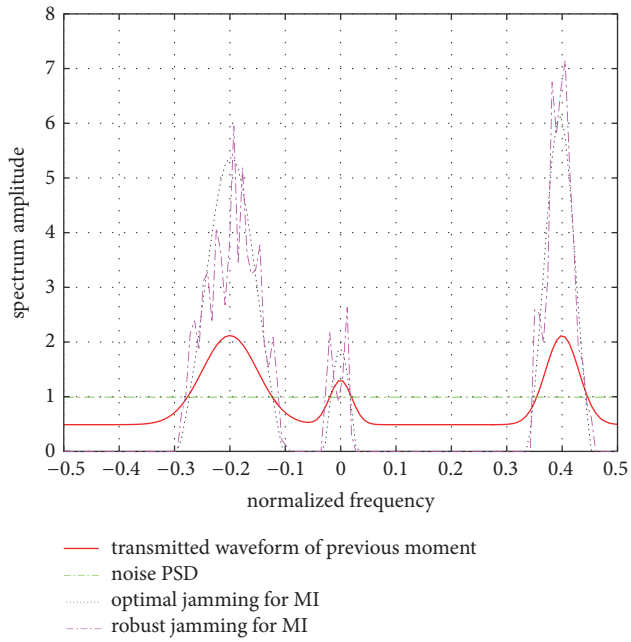


Figure 7: Jamming results for multiple targets.

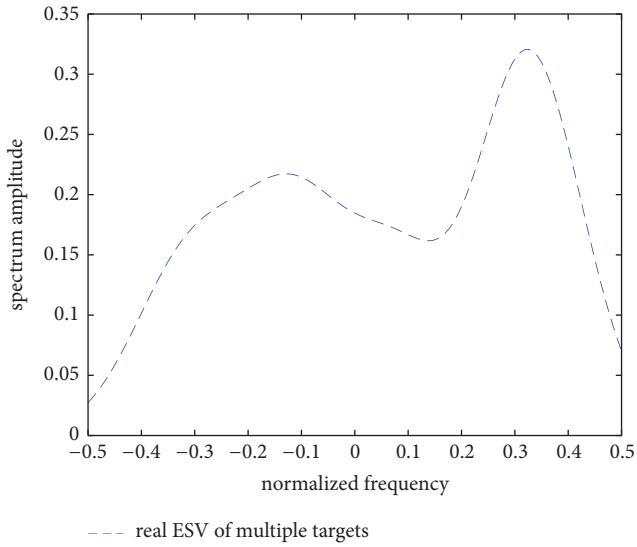


Figure 8: Nominal ESV for multiple targets.

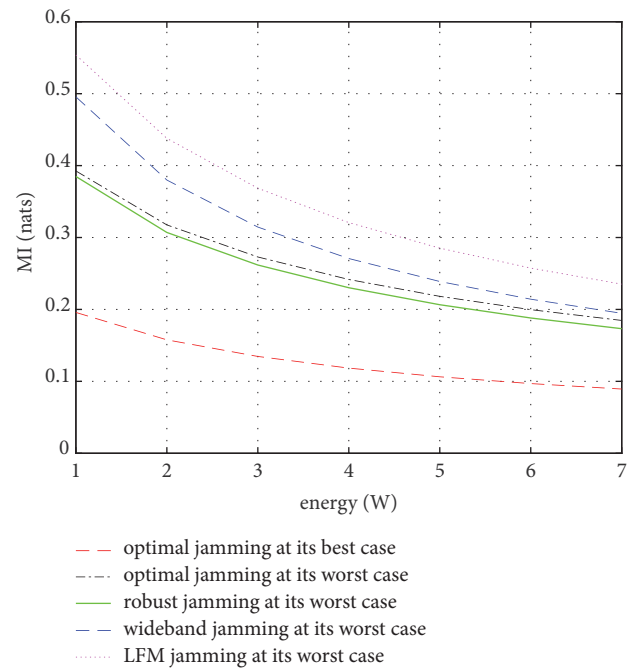


Figure 9: MI performance of the jamming results for single target.

both the two jamming techniques distribute the finite energy only in frequency bands where the transmitted waveform of previous moment is relatively strong, because the value of the nominal ESV of multiple targets is too small compared to the value of the waveform.

Assume that the total energy of the jamming varies from 1 to 7 W, the MI units (MIs) corresponding to the optimal jamming for real target spectrum, the optimal jamming for the most unfavorable target spectrum, the robust jamming for the most unfavorable target spectrum, the wide-band jamming for the most unfavorable target spectrum, and the LFM (linear frequency modulation) jamming for the most unfavorable target spectrum are compared in Figures 9 and 10 for single target and multiple targets, respectively. Simulation results show that the MI of the optimal jamming for real target spectrum is the smallest, which reaches its best performance for the jammer; the reason is that the real target spectrum is adopted and the optimal jamming is designed based on the real target spectrum. When using the most unfavorable target spectrum for the jammer, that is, the upper bound of the uncertainty range, we can get the MI corresponding to the

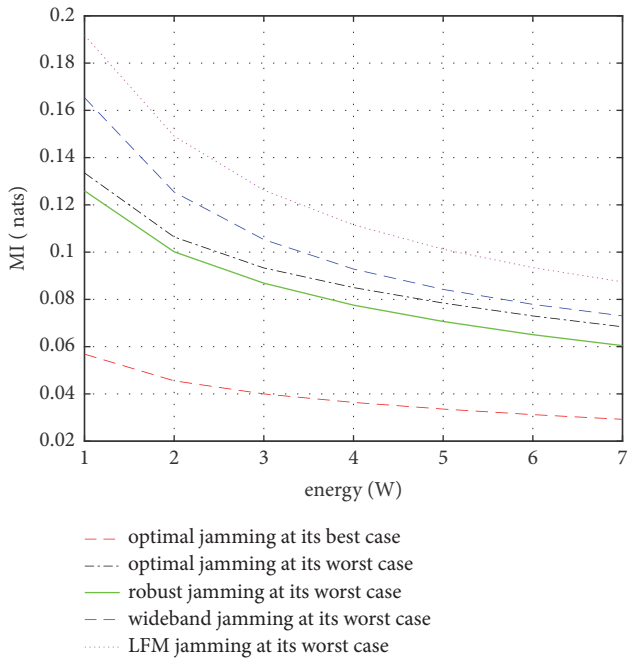


Figure 10: MI performance of the jamming results for multiple targets.

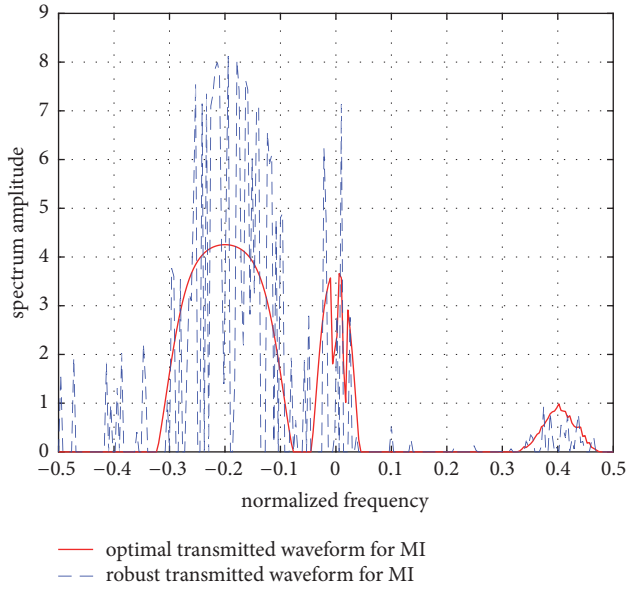


Figure 11: Waveform results for single target.

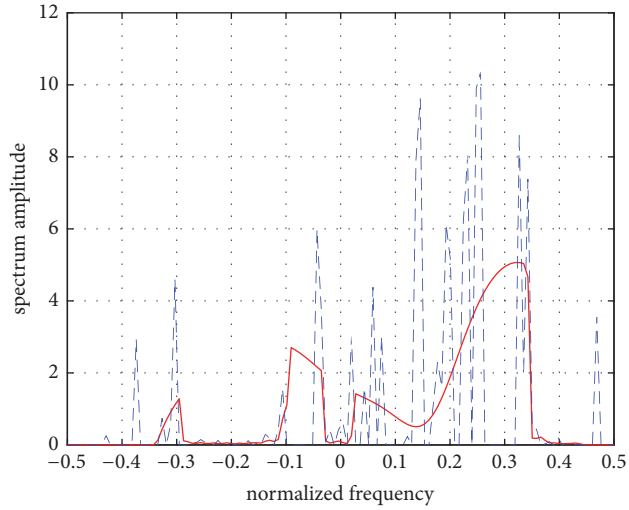


Figure 12: Waveform results for multiple targets.

optimal jamming for the most unfavorable target spectrum. As expected, the MI corresponding to the robust jamming for the most unfavorable target spectrum is between the above two. That is because the prior knowledge of the real target for the designed robust jamming is less. However, it is better than the MI corresponding to the optimal jamming for the most unfavorable target spectrum because the minimax robust technique improves the most unfavorable performance of the jammer. The wide-band jamming indicates that the jamming spectrum is a straight line over the entire frequency band, and the LFM jamming means that the instantaneous frequency of the jamming signal changes linearly with time. Both of these two jamming techniques do not contain the information about the target, noise, and transmitted waveform of previous moment. Therefore the MIs corresponding to the wide-band jamming and the LFM jamming for the most unfavorable target spectrum are larger than the above-mentioned three cases.

According to the prior information of the robust jamming shown in Figures 6 and 7, which denote the strategies of the jammer, the optimal waveform for real target spectrum and the robust waveform for the worst case are shown in Figure 11 for single target and Figure 12 for multiple targets. Both the waveform energy constraints are assumed to be 1W and distribute the finite energy in frequency bands where the target spectrum response is strong and the jamming spectrum response is weak.

Assume that the total energy of the transmitted waveform varies from 1 to 7 W, the MIs corresponding to the optimal transmitted waveform for real target spectrum, the optimal transmitted waveform for the most unfavorable target spectrum, the robust transmitted waveform for the most unfavorable target spectrum, the wide-band transmitted waveform for the most unfavorable target spectrum, and the LFM transmitted waveform for the most unfavorable target spectrum are compared in Figure 13 for single target and Figure 14 for multiple targets. Simulation results show that the MI of the optimal transmitted waveform for real target spectrum is the largest, which reaches its best performance for the radar; the reason is that the real target spectrum is adopted and the optimal transmitted waveform is designed based on the real target spectrum. When using the most unfavorable target spectrum for the radar, that is, the lower bound of the uncertainty range, we can get the MI corresponding to the optimal transmitted waveform for the most unfavorable target spectrum. As expected, the MI corresponding to the

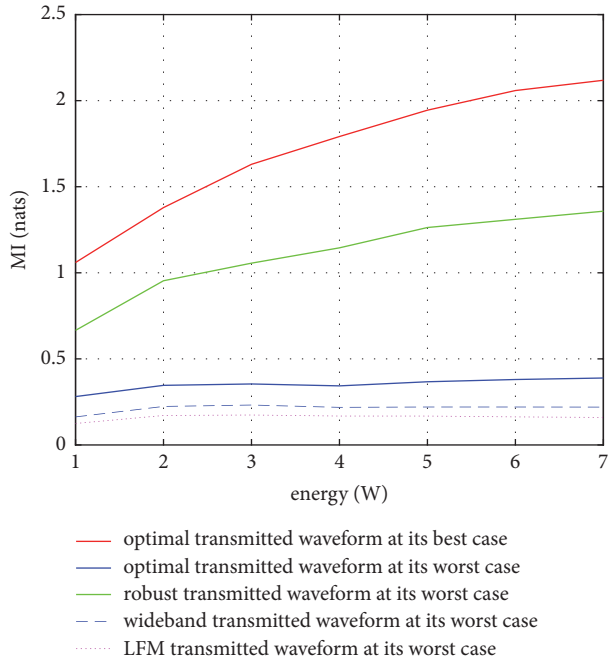


Figure 13: MI performance of the transmitted waveform results for single target.

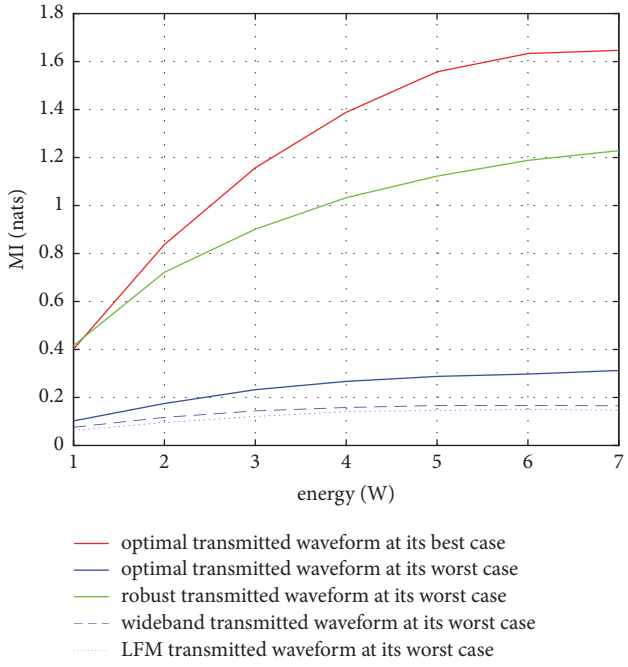


Figure 14: MI performance of the transmitted waveform results for multiple targets.

robust transmitted waveform for the most unfavorable target spectrum is between the above two. That is because the prior knowledge of the real target for the designed robust transmitted waveform is less. However, it is better than the MI corresponding to the optimal transmitted waveform for the most unfavorable target spectrum because the maximin robust technique improves the most unfavorable performance of the radar. The wide-band transmitted waveform indicates that the transmitted waveform spectrum is a straight line over the entire frequency band, and the LFM transmitted waveform means that the instantaneous frequency of the transmitted waveform signal changes linearly with time. Both of these two transmitted waveforms do not contain the information about the target, noise, and jamming, which is similar to the wide-band jamming and the LFM jamming. Therefore the MIs corresponding to the wide-band transmitted waveform and the LFM transmitted waveform for the most unfavorable target spectrum are smaller than the above-mentioned three cases.

5. Conclusion

In this paper, through assuming that the real target spectrum is known, the MI-based jamming and radar transmitted waveform techniques are proposed firstly. The designed jamming and transmitted waveform under the hierarchical game model of radar and jammer are proper for restricted energy condition. Then, the uncertainty range of the target spectrum is taken into account. The target model has been adopted, which assumes that the real target spectrum exists in an uncertainty range defined by the known upper and lower bounds. According to the uncertainty range above, the minimax robust jamming and maximin robust transmitted waveform have been designed successively. Although the jamming designed by the jammer can greatly impair the performance of the radar system, the radar is in a leading position and the jamming is intercepted by the radar system. Simulation results show that the transmitted waveform designed by the radar transmitter can finally guarantee the estimation performance of the radar system and provide useful guidance for waveform energy allocation. Although the data used in this paper does not come from the real world experimentation, the data has a certain representativeness, and it can provide reference for future hardware or actual device implementation.

Data Availability

All data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of Hebei Province (No. F2018501051) and the National Natural Science Foundation of China (Nos. 61403067, 61473066, and 61601109).

References

[1] S. Haykin, "Cognitive radar: A way of the future," *IEEE Signal Processing Magazine*, vol. 23, no. 1, pp. 30–40, 2006.

[2] J. D. Zhang, D. Zhu, and G. Zhang, "Adaptive compressed sensing radar oriented toward cognitive detection in dynamic sparse target scene," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1718–1729, 2012.

[3] L. E. Brennan and L. S. Reed, "Theory of adaptive radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 9, no. 2, pp. 237–252, 1973.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Hoboken, NJ, USA, 1991.

[5] S.-H. Lee, S.-M. Lee, G.-Y. Sohn, and J.-Y. Kim, "Fuzzy entropy design for non convex fuzzy set and application to mutual information," *Journal of Central South University of Technology*, vol. 18, no. 1, pp. 184–189, 2011.

[6] C. C. Tan, S. A. Shanmugam, and K. A. L. Mann, "Medical image registration by maximizing mutual information based on combination of intensity and gradient information," in *Proceedings of the International Conference on Biomedical Engineering (ICoBE)*, pp. 368–372, Penang Island, Malaysia, 2012.

[7] C.-Y. Chen and P. P. Vaidyanathan, "MIMO radar waveform optimization with prior information of the extended target and clutter," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3533–3544, 2009.

[8] M. R. Bell, *Information Theory and Radar: Mutual Information and the Design and Analysis of Radar Waveforms and Systems*, California Institute of Technology, Pasadena, Calif, USA, 1988.

[9] M. R. Bell, "Information theory and radar waveform design," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1578–1597, 1993.

[10] W. Zhang and L. Yang, "Communications-inspired sensing: a case study waveform design," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 792–803, 2010.

[11] Y. Kwon, R. M. Narayanan, and M. Rangaswamy, "A multi-target detector using mutual information for noise radar systems in low SNR regimes," in *Proceedings of the 2010 5th International Waveform Diversity and Design Conference, WDD 2010*, pp. 105–109, Canada, August 2010.

[12] L. Wang, H. Wang, K.-K. Wong, and P. V. Brennan, "Minimax robust jamming techniques based on signal-to-interference-plus-noise ratio and mutual information criteria," *IET Communications*, vol. 8, no. 10, pp. 1859–1867, 2014.

[13] R. A. Romero, J. Bae, and N. A. Goodman, "Theory and application of SNR and mutual information matched illumination waveforms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 912–927, 2011.

[14] L.-L. Wang, H.-Q. Wang, Y.-Q. Cheng, and Y.-L. Qin, "A novel SINR and mutual information based radar jamming technique," *Journal of Central South University*, vol. 20, no. 12, pp. 3471–3480, 2013.

[15] Y. Yang and R. S. Blum, "Minimax robust MIMO radar waveform design," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 1, pp. 147–155, 2007.

[16] S. A. Kassam and H. V. Poor, "Robust techniques for signal processing: a survey," *Proceedings of the IEEE*, vol. 73, no. 3, pp. 433–481, 1985.

Hindawi
Complexity
Volume 2019, Article ID 5245373, 10 pages
https://doi.org/10.1155/2019/5245373

Research Article

Activity Feature Solving Based on TF-IDF for Activity Recognition in Smart Homes

Jinghuan Guo,[1,2] Yong Mu,[1] Mudi Xiong,[1] Yaqing Liu,[1,2] and Jingxuan Gu[3]

[1]*School of Information Science & Technology, Dalian Maritime University, Dalian 116026, China*
[2]*Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong 643000, China*
[3]*School of Mathematical Sciences, Dalian University of Technology, Dalian 116026, China*

Correspondence should be addressed to Yaqing Liu; liuyaqing@dlmu.edu.cn

Received 21 December 2018; Revised 13 February 2019; Accepted 3 March 2019; Published 24 March 2019

Guest Editor: Jose Garcia-Rodriguez

Smart homes based on the Internet of Things have been rapidly developed. To improve the safety, comfort, and convenience of residents' lives with minimal cost, daily activity recognition aims to know resident's daily activity in non-invasive manner. The performance of daily activity recognition heavily depends on solving strategy of activity feature. However, the current common employed solving strategy based on statistical information of individual activity does not support well the activity recognition. To improve the common employed solving strategy, an activity feature solving strategy based on TF-IDF is proposed in this paper. The proposed strategy exploits statistical information related to both individual activity and the whole of activities. Two distinct datasets have been commissioned, to mitigate against any possible effect of coupling between dataset and sensor configuration. Finally, a number of machine learning (ML) techniques and deep learning technique have been evaluated to assess their performance for residents activity recognition.

1. Introduction

The world's population is aging, leading to uneven population composition. It is estimated that, by 2050, there are more than 20% of the population who will exceed 64 years and the number of people over the age of 80 in the world will reach nearly 379 million, about 5.5 times (69 million) in 2000 [1, 2]. This increase in population aging is expected to lead to an increase in age-related diseases, which in turn will provide an additional burden on health care [2]. As a population ages, the potential support ratio tends to fall. PSR is the number of people aged 15–64 per one older person aged 65 or older. This ratio describes the burden placed on the working population (unemployment and children are not considered in this measure) by the non-working elderly population. Between 1950 and 2009, the potential ratio reduced from 12 to 9 potential workers per person aged 65 or over [1].

In recent years, smart homes based on the Internet of Things have been rapidly developed in order to improve the safety, comfort, and convenience of residents' lives with minimal cost. They are mainly used in intelligent video surveillance, patient monitoring systems, and human-computer interaction, virtual reality, smart security, athlete-assisted training and so on. Obviously, the fundamental of smart home is the recognition of user activity.

The main purpose of Ambient Assisted Living (AAL) is to support the independent living and subsequently alleviate a portion of the problems associated with ageing. It is widely seen as an effective approach to solving some of the problems associated with supporting population ageing [3, 4]. With the continued development of smart homes technologies, individuals, such as the elderly and disabled, can improve their quality of life and can live independently at home.

Activity recognition (AR) is one of the important ways of AAL. AR is a complex process and can be generally classified into two categories in terms of the type of sensor that is used for activity monitoring. The first is called as vision-based activity recognition. The methods in this category utilize computer vision techniques, including feature extraction, structural modeling, movement segmentation,

action extraction, and movement tracking to analyze visual observations for pattern recognition. The second is called as sensor-based activity recognition. Sensor data generated by sensor-based monitoring is primarily a time series of state changes and/or various parameter values typically processed by data fusion, probabilistic or statistical analysis methods, and formal knowledge techniques for activity recognition. Sensor-based activity recognition can be divided into two categories. The first is based on wearable sensor activity monitoring, which is more concerned in mobile computing. The second is dense sensing, which is more suitable for applications that support smart environments.

In this paper, we focus on sensor-based activity recognition. A key step of sensor-based activity recognition is activity feature solving. However, the current common employed solving strategy based on statistical information of individual activity does not support well the activity recognition. To improve the common employed solving strategy, an activity feature solving strategy based on TF-IDF is proposed in this paper. The proposed strategy exploits more statistical information related to both individual activity and the whole of activities.

The rest of paper is organized as follows. Section 2 describes related work. Section 3 describes process of activity recognition. The proposed feature solving strategy is explained in Section 4. Section 5 describes the implementation of the experiment and method evaluation. Section 6 concludes the paper.

2. Related Work

At present, a number of methods of identifying activity have been developed. According to the sensor type, it can be divided into video sensor based, wearable sensor based, and embedded sensor based. For video sensors, Ashish Khare et al. proposed a video sensor-based behavioral recognition approach that integrates local binary patterns [5]. Lin et al. proposed a new network-based transmission (NTB) algorithm for human activity recognition in video [6]. However, user's privacy is a huge challenge, and many users are reluctant to place sensors in sensitive places such as bedrooms and bathrooms. And video sensors are also affected by factors such as day and night, the environment, and so on. For wearable sensors, Kevin Bouchard et al. use passive RFID-based activity recognition systems to detect anomalies in cognitive impairment [7]. Yang et al. proposed a simple method for identifying human activities based on simple object information involved in RFID usage activities [8]. Andrey et al. proposed an accelerometer-based convolutional network for activity recognition [9]. However, it is inconvenient for users to carry, most users are not willing to carry the sensor on their body, and the acquisition of activity sometimes depends on factors such as the location carried by the sensor. The embedded sensor solves the problem brought by video sensor and wearable sensor. The embedded sensor has the advantages of effectively protecting the user's personal privacy, being free from the influence of the surrounding environment and not requiring the user to carry [10–12].

Activity recognition in smart homes can be divided into knowledge-driven and data-driven [13–15]. For knowledge-driven approach, knowledge is generated from field experts. In [16], Chen et al. proposed a real-time continuous activity recognition of multi-sensor data streams in knowledge-driven smart homes. In addition, ontology is often integrated into knowledge-driven methods. In [17], Latfi et al. present an ontology-based model of the TSH for elderly activity recognition. Salguero et al. propose that the ontology automatically generates the features of the ADL classifier for behavior recognition [18]. The ontology-based approach is clear and easy to understand. Knowledge-driven is therefore called a top-down approach, but it is poor in dealing with uncertainty and time information.

In contrast, data-driven approaches collect data from a large number of sensor streams, organize the data to form information, then integrate and refine related information, and use machine learning technology to train and fit to form an automated decision model based on the data [19]. In [20], a framework for acquiring and developing different layers of context models in a smart environment is proposed. Tapia et al. propose a real-time algorithm to automatically identify physical activities [21]. Data driven is also known as a bottom-up approach. Strong ability to deal with uncertainty and time information. Therefore, this paper uses data-driven activity recognition.

Data-driven approaches are generally divided into a generation method and a discrimination method. In the generation mode, Patterson et al. propose multiple different HMM models for activity recognition [22]. In order to improve the HMM model for identifying complex activities, a multi-layer hidden Markov model (HHMM) is proposed in [23]. Vail et al. propose a new, effective feature selection algorithm for m-estimates-based CRF to identify the most important features for behavior recognition [24]. Although it works better with uncertain or incomplete data, it requires a lot of data to learn to optimize the model. With the development of neural networks, deep learning is gradually applied to activity recognition. Li et al. proposed a BP neural network for representing and identifying human activities from observed sensor sequences [25]. Deep Belief Network (DBN) model is proposed for successful human activity recognition [26]. Guan proposes an ensemble of deep long-term short-term memory (LSTM) networks for behavior recognition [27]. In [28], Chen et al. use LSTM recurrent neural networks to analyze sensor readings from accelerometers and gyroscopes to identify human activity and provide position-aware methods to improve recognition accuracy.

For activity feature selection and solving, start time and duration of activity instance are commonly used temporal features. Individual sensors, set of frequent sensors and sequence of frequent sensors are common used space features [29]. For space features, the common solving strategy of feature includes frequency, density, etc., that space features are activated. Because current solving strategy only takes into account statistical information of individual activity, it does not support well the activity recognition.

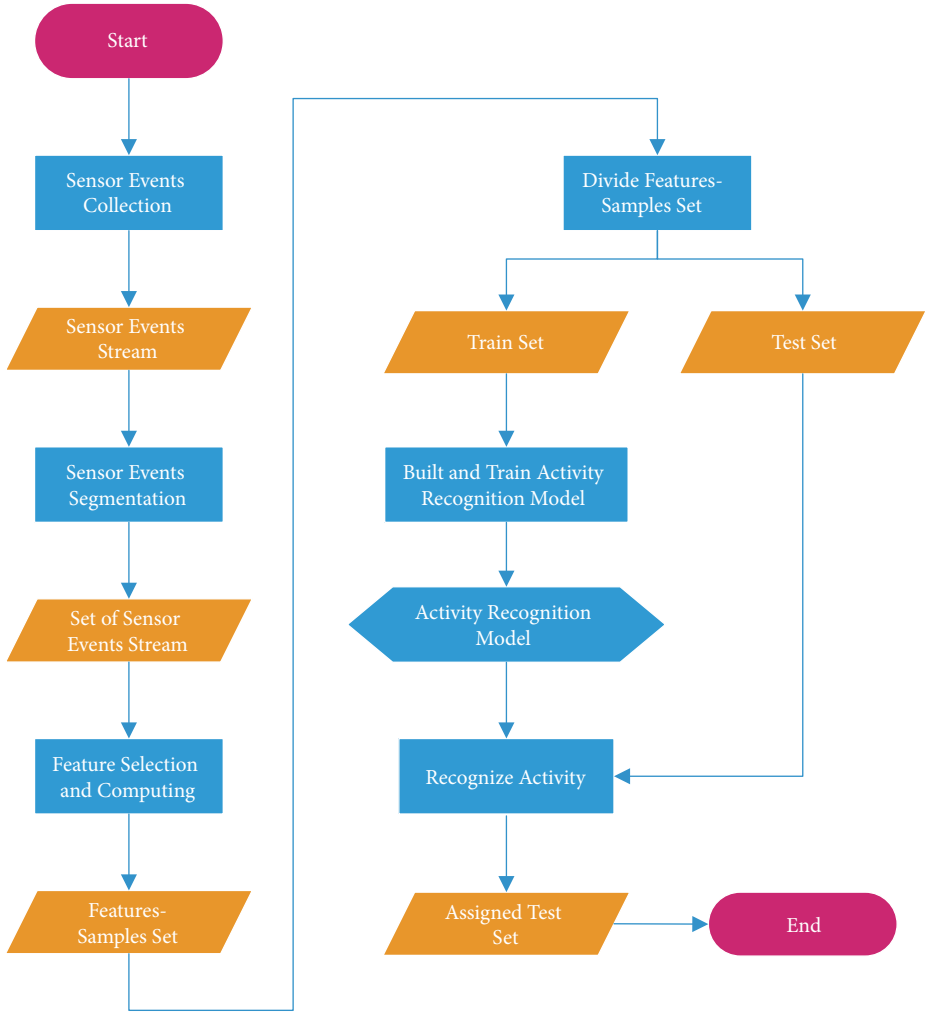


Figure 1: Process of activity recognition.

3. Process of Activity Recognition

As shown in Figure 1, activity recognition process includes four stages.

In the first stage, raw sensors events are collected in form of stream when a daily activity is occurring. In Figure 2, raw sensors events of a sample of activity "Sleep" are presented. When a daily activity instance starts, some sensors will be activated orderly in time series until the daily activity instance ends. When some sensor is activated, the activated date, the activated time, the name, and the value of the sensor are stored. For example, the first activated sensor is "M021" with value "ON" at time "00:06:32.834414" in 2011-06-15 for activity "Sleep" in Figure 2.

In the second stage, sensor events sequence is separated into a number of sub sequences. Each subsequence corresponds to an entire activity instance.

In the third stage, features of daily activity are selected and solved. Generally, features are divided into temporal features and space features. Start time and duration of an activity instance are common temporal features. Sensors are common space features. Temporal features and space features are used to characterize daily activity instances. After features are selected, features can be solved according to some strategy.

In the last stage, activity recognition model is built. Then, training data is provided to train recognition model. Trained recognition model is employed to assign an activity label to each of test activity instances.

4. Activity Feature Selection and Solving

4.1. Activity Feature Selection. As mentioned above, our work focuses on activity feature selection and solving. The task of feature selection is to determine feature set. It is common to previous work that both temporal features and space features are involved in our work [10]. Temporal features include start time and duration of an activity instance. Space features are divided into two categories by formula of feature solving. The first category of space features is named Start-End Frequency (SEF) features. Each of SEF features corresponds to a sensor. The second category of space features is named as TF-IDF

2011-06-15	00:06:32. 834414	M021	ON	Sleep="begin"
2011-06-15	00:06:33. 988964	M021	OFF	
2011-06-15	00:12:32. 670631	BATV012		9540
2011-06-15	00:15:01. 957718	LS013 6		
2011-06-15	00:25:01. 892474	LS013 7		
2011-06-15	01:05:01. 622637	BATV013		9460
2011-06-15	01:10:17. 369388	BATV001		9500
2011-06-15	01:25:47. 117808	BATV105		3100
2011-06-15	01:36:27. 665051	BATV021		9520
2011-06-15	01:37:01. 761435	BATV102		3160
2011-06-15	01:58:18. 094543	BATV022		9480
2011-06-15	02:04:42. 634918	BATV019		9440
2011-06-15	02:22:22. 244805	BATV010		9480
2011-06-15	02:40:03. 347644	BATV006		9440
2011-06-15	02:52:19. 076981	BATV002		9500
2011-06-15	03:11:25. 513881	BATV015		9520
2011-06-15	03:37:46. 585185	M021	ON	
2011-06-15	03:37:47. 706265	M021	OFF	
2011-06-15	03:38:11. 211961	M021	ON	
2011-06-15	03:38:11. 306285	MA020	ON	
2011-06-15	03:38:12. 265799	M021	OFF	
2011-06-15	03:38:12. 381306	MA020	OFF	
2011-06-15	03:38:13. 535141	M021	ON	
2011-06-15	03:38:13.877151	MA020	ON	
2011-06-15	03:38:14. 924765	MA020	OFF	
2011-06-15	03:38:15. 885063	MA020	ON	
2011-06-15	03:38:17. 022055	MA020	OFF	
2011-06-15	03:38:17. 132255	M021	OFF	
2011-06-15	03:38:17. 750829	M021	ON	
2011-06-15	03:38:17. 814393	MA020	ON	
2011-06-15	03:38:22. 584179	M021	OFF	
2011-06-15	03:38:23. 203947	M021	ON	
2011-06-15	03:38:23. 271939	MA020	OFF	
2011-06-15	03:38:24. 259673	M021	OFF	
2011-06-15	03:38:28. 094897	MA020	ON	
2011-06-15	03:38:28. 21206	M021	ON	
2011-06-15	03:38:29. 213955	MA020	OFF	
2011-06-15	03:38:29. 32819	M021	OFF	Sleep="end"

Figure 2: Activated sensor events stream on activity "Sleep".

Input: $\{ai_1, ai_2, \ldots, ai_m\}$, a set of activity instances
$\{f_{11}, f_{12}, \ldots, f_{1n}\}$, set of SEF features
$S = \{s_1, s_2, \ldots, s_n\}$
Output: FV

1. $FV \longleftarrow \{(f_{11}, v_{11}, v_{12}, \ldots, v_{1m}), (f_{12}, v_{21}, v_{22}, \ldots, v_{2m}), \ldots, (f_{1n}, v_{n1}, v_{n2}, \ldots, v_{nm})\}$;
2. Assign 0 to v_{kj}, where k >=1 and k<=n, j>=1 and j<=m
3. $j \longleftarrow 0$;
4. **while**(j<=m)
5. Extract the first activated sensor ss;
6. Extract the last activated sensor es;
7. $k \longleftarrow 1$; $j++$;
8. **while**(k<=n)
9. **if** ss is same to f_{1k} **then**
10. v_{ki}++;
11. **end if**
12. **if** es is same to f_{1k} **then**
13. v_{ki}++;
14. **end if**
15. k++;
16. **end while**
17. **end while**
18. **return** FV

Algorithm 1: *solveSEFFeature*.

features. Each of TF-IDF features also corresponds to a sensor.

Formally, let $S = \{s_1, s_2, \ldots, s_n\}$ be the set of sensors which are deployed in a smart home. Feature set is defined $F = \{st, du\} \cup \{f_{11}, f_{12}, \ldots, f_{1n}\} \cup \{f_{21}, f_{22}, \ldots, f_{2n}\}$. st and du denote start time and duration of an activity instance, respectively. $\{f_{11}, f_{12}, \ldots, f_{1n}\}$ is set of SEF features. $\{f_{21}, f_{22}, \ldots, f_{2n}\}$ is set of TF-IDF features.

4.2. Activity Feature Solving

4.2.1. Temporal Activity Feature Solving. For an activity instance, start time and duration are extracted as the values of features st and du. In Figure 2, the values of st and du of the activity instance "Sleep" are "00:06:32" and 12717 seconds, respectively.

4.2.2. SEF Activity Feature Solving. SEF activity feature solving process is presented in Algorithm 1. For an activity instance ai and a sensor s_k (k>=1 and k<=n), the corresponding SEF feature value f_{1k} is assigned to 2 if both the first sensor and the last sensor are s. The corresponding SEF feature value f_{1k} is assigned to 1 if the first sensor or the last sensor is s. The corresponding SEF feature value f_{1k} is assigned to 0 if neither of the first sensor and the last sensor are s. For activity

Input: $\{ai_1, ai_2, \ldots, ai_m\}$, a set of activity instances
$\{f_{21}, f_{22}, \ldots, f_{2n}\}$, set of *FF* features
$S = \{s_1, s_2, \ldots, s_n\}$
Output: *FV*

```
FV ⟵ {(f21, v11, v12, ..., v1m), (f22, v21, v22, ..., v2m), ..., (f2n, vn1, vn2, ..., vnm)};
j ⟵ 0;
while(j<=m)
    Collect all sensors Sj which are activated when aij is active;
    Calculate the TF-IDF value fs of s ∈ Sj using TF-IDF(s, ai), formula (1) and formula (2);
    k ⟵ 1; j++;
    while(k<=n)
        for each s in Sj
            if s is same to f2k then
                vki ⟵ fs;
            end if
        end for
        k++;
    end while
end while
return FV
```

Algorithm 2: *solveTF-IDFFeature*.

Table 1: Statistical information concerning datasets "tulum2009" and "cairo."

	Sensors	Activity Categories	Activity Instances	Residents	Measurement Time
tulum2009	20(2 categories)	27	1396	2	84 days
cairo	32(2 categories)	16	580	2	57 days

instance "Sleep" in Figure 2, the value of SEF feature f is assigned to 2 when f is corresponding to sensor "M021".

4.2.3. TF-IDF Activity Feature Solving

(1) TF-IDF. Considering a set of terms $T = \{t_1, t_2, \ldots, t_m\}$ and a set of documents $D = \{d_1, d_2, \ldots, d_n\}$, Term Frequency-Inverse Document Frequency (TF-IDF) is a common weighting formula which is employed to evaluate how important a term $t \in T$ is to a document $d \in D$ in the field of information retrieval [30]. Formally, TF-IDF is defined as $\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(d, t, D)$. $\text{TF}(t, d) = \text{times}(t, d)/(\text{times}(t_1, d) + \text{times}(t_2, d) + \cdots + \text{times}(t_m, d))$, where $\text{times}(t, d)$ is how many times the term t appears in the document d. $\text{IDF}(t, d, D) = \lg(|D|/(1 + |\{d \mid d \in D \text{ and times } (t, d) > 0\}|))$.

In this paper, TF-IDF is employed to evaluate how important a sensor is to an activity instance. Considering a set of sensors $S = \{s_1, s_2, \ldots, s_n\}$ and a set of activity instances $AI = \{ai_1, ai_2, \ldots, ai_m\}$, $\text{TF-IDF}(s, ai)$ is defined as $\text{TF-IDF}(s, ai) = \text{TF}(s, ai) \times \text{IDF}(ai, t, AI)$.

Ranges of different TF-IDF feature values vary considerably. To normalize TF-IDF feature values, two optimization functions are introduced into TF-IDF feature solving. The first function is sigmoid function which is shown in Formula (1). It can map TF-IDF feature value to the interval of $[0, 1]$. The second function is Tanh function which is shown in Formula (2). It can map TF-IDF feature value to the interval of [-1, 1]. TF-IDF activity features solving process is presented in Algorithm 2.

$$sigmod\,(TF - IDF\,(s, ai)) = \frac{1}{1 + e^{-(TF-IDF(s,ai))}} \quad (1)$$

$$\tanh\,(TF - IDF\,(s, ai)) = \frac{e^{TF-IDF(s,ai)} - e^{-(TF-IDF(s,ai))}}{e^{TF-IDF(s,ai)} + e^{-(TF-IDF(s,ai))}} \quad (2)$$

5. Evaluation

5.1. Data Availability. In this study, we employ two public datasets, "tulum2009" and "cairo" in [31], to illustrate the applicability of the proposed approach. These datasets have been published by the Washington State University [31]. Statistical information concerning the two data sets are described in Table 1. Values listed under column "Sensors" correspond to the number of sensors involved and their corresponding categories. Similarly, values listed under column "Activity Categories" correspond to the number of activity classes involved while those listed under column "Activity Instances" correspond to the number of involved activity instances. Values listed under column "Residents" correspond to the number of residents involved. Lastly, values listed under "Measurement Time" correspond to durations over which data were collected duration that data is collected.

For the "tulum2009" dataset, the following identifier categories were considered.

Table 2: Involved activities concerning on "tulum2009".

Atom Activity	Interactive Activities		
	Two Activities	Three Activities	Four Activities
C_B	C_B&E_B	C_B&E_B&C_L	E_B&C_L&W_D&L_H
C_L	C_L&L_H	E_B&C_L&W_D	S&W_T&S&W_T
E_H	C_L&E_B	S&W_T&S	
G_M	W_T&S	C_L&E_B&W_D	
L_H	L_H&E_B	S&E_H&W_T	
E_B	E_B&W_T	E_B&C_L&L_H	
S	S&E_H		
W_D	S&W_D		
W_T	W_S&E_B		
	W_T&L_H		

Table 3: Involved activities concerning on "cairo".

Atom Activity	Interative Activity
B_T_T	W_I_O&W
B	W&W
S	S&S
W	W&L_H
W_I_O	W_I_O&S
D	
Lau	
L_H	
Lch	
N_W	
T_M	

(1) Identifiers with names starting with "M" indicate infrared motion sensors—M001–M018.

(2) Identifiers with names starting with "T" indicate temperature sensors—T001–T002.

Involved atom activities include "*Cook_Breakfast*"("*C_B*"), "*Cook_Lunch*"("*C_L*"), "*Enter_Home*"("*E_H*"), "*Group_Meeting*"("*G_M*"), "*Leave_Home*"("*L_H*"), "*Eat_Breakfast*"("*E_B*"), "*Snack*"("*S*"), "*Wash_Dishes*"("*W_D*"), "*Watch_TV*"("*W_T*"). Involved atom activities and interactive activities are presented in Table 2.

Similarly, for the "cairo," dataset, the following identifier categories were considered.

(1) Identifiers with names starting with "M" indicate infrared motion sensors—M001-M027.

(2) Identifiers with names starting with "T" indicate temperature sensors—T001–T005.

Involved activities include "*Bed to toilet*" ("*B_T_T*"), "*Breakfast*" ("*B*"), "*sleep*"("*S*"), "*wake*" ("*W*"), "*work in office*" ("*W_I_O*"), "*Dinner*" ("*D*"), "*Laundry*" ("*Lau*"), "*Leave home*" ("*L_H*"), "*Lunch*" ("*Lch*"), "*Night wandering*"("*N_W*"), "*take medicine*"("*T_M*"). Involved atom activities and interactive activities are presented in Table 3.

5.2. Experimental Preparation. In this study, the proposed approach was compared against frequency based feature solving approach. Frequency based activity features solving is commonly employed in previous research [Liu17]. Frequency based activity features solving process is presented as follows. For an activity instance ai and a sensor s_k (k>=1 and k<=n), the corresponding feature value f_{2k} is assigned by the frequency that f_{2k} is activated. For activity instance "Sleep" in Figure 1, the values of features are (BATV001, 1), (BATV002, 1), (BATV006, 1), (BATV010, 1), (BATV012, 1), (BATV013, 1), (BATV015, 1), (BATV019, 1), (BATV021, 1), (BATV022, 1), (BATV102, 1), (BATV105, 1), (LS013, 2), (M021,14), (MA020, 10) when the values of features are greater than zero.

These approaches are evaluated by their corresponding performance of activity recognition through Support Vector Machine (SVM), Sequential minimal optimization (SMO), and Random Forest (RF). The used toolset employed was Weka 3.9. In addition, we experiment on the same datasets using a state-of-the-art deep learning technique Long Short-Term Memory (LSTM), which is appropriate for time series data. The used LSTM consists of an input layer, two hidden layers and an output layer. In the dataset cario, the numbers of neurons in the input, hide, and output layers are set to 20, 40, 40, and 21, respectively. In the dataset tulum2009, the numbers of neurons in the input, hidden, and output layers are set to 20, 40, 40, and 37, respectively. Epoch is set to 1, 5, 10, and 15, respectively. 10-fold cross validation was performed. Evaluation metrics considered included accuracy, precision, and F-measure.

5.3. Results

5.3.1. The Whole Results. Recognition accuracies concerning the tulum2009 dataset are depicted in Table 4. The accuracy using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh far exceeded that the one using features FF when employing SVM and SMO. The accuracies are almost equal when employing RF. Recognition accuracies concerning the cairo dataset are depicted in Table 5. The accuracies using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh far exceeded the one using features FF when employing SVM. The accuracies using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh

Table 4: Recognition accuracies using three classifiers concerning dataset tulum2009.

Feature	SVM	SMO	RF
Frequency Feature (FF)	*67.2%*	*79%*	**90.8%**
TF-IDF	**87%**	85%	**90.8%**
TF-IDF+Sigmod	**87%**	**88.4%**	*90.5%*
TF-IDF+Tanh	85.9%	87.2%	90.7%

Table 5: Recognition accuracies using three classifiers concerning dataset cairo.

Feature	SVM	SMO	RF
FF	*41%*	*83.1%*	*87.4%*
TF-IDF	**83.8%**	**86.4%**	**88.8%**
TF-IDF+Sigmod	83.6%	83.6%	87.8%
TF-IDF+Tanh	80.3%	83.7%	88.6%

Table 6: Recognition precisions using three classifiers concerning dataset tulum2009.

Feature	SVM	SMO	RF
FF	*64.3%*	*71.5%*	88%
TF-IDF	**82.7%**	78.5%	**88.1%**
TF-IDF+Sigmod	81.9%	**85.3%**	*87.5%*
TF-IDF+Tanh	80.7%	82.1%	87.9%

Table 7: Recognition precisions using three classifiers concerning dataset cairo.

Feature	SVM	SMO	RF
FF	*52.7%*	*79.8%*	*85%*
TF-IDF	**82%**	**85%**	**86.3%**
TF-IDF+Sigmod	80.5%	83.2%	85.4%
TF-IDF+Tanh	73.3%	**85%**	86.1%

still a little exceeded those using features FF when employing SMO and RF.

Recognition precisions concerning the tulum2009 dataset are depicted in Table 6. The precisions using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh less or more exceeded that the one using features FF when employing all of three classifiers. Recognition precisions concerning the cairo dataset are depicted in Table 7. The precisions using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh exceeded the one using features FF when employing all of three classifiers.

Recognition F-Measures concerning the tulum2009 dataset are depicted in Table 8. The F-Measures using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh less or more exceeded that the one using features FF when employing all of three classifiers. Recognition F-Measures concerning the cairo dataset are depicted in Table 9. The F-Measures using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh exceeded the one using features FF when employing all of three classifiers.

Recognition results using LSTM concerning the tulum2009 dataset are depicted in Table 10. The best accuracy 76.01%, the best precision 80.13%, and the best F-Measure 77.99% are achieved when 5 is assigned to Epoch. The accuracies and F-Measures using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh exceeded the best counterpart using LSTM when employing SVM, SMO, and RF. The precisions using features TF-IDF+Sigmod or TF-IDF+Tanh exceeded the best one using LSTM when employing SMO. Only the best precision using LSTM a little exceeded the one using feature TF-IDF when employing SMO.

Table 8: Recognition F-Measures using three classifiers concerning dataset tulum2009.

Feature	SVM	SMO	RF
FF	*65.7%*	*75.1%*	**89.4%**
TF-IDF	**84.8%**	81.6%	**89.4%**
TF-IDF+Sigmod	84.4%	**86.8%**	*88%*
TF-IDF+Tanh	83.2%	84.6%	89.3%

Table 9: Recognition F-Measures using three classifiers concerning dataset cairo.

Feature	SVM	SMO	RF
FF	*45.8%*	*81.4%*	*86.2%*
TF-IDF	**82.9%**	85.7%	**87.5%**
TF-IDF+Sigmod	82%	83.4%	86.6%
TF-IDF+Tanh	76.6%	**85.8%**	87.3%

Table 10: Recognition results using LSTM concerning dataset tulum2009.

Epoch	Accuracy	Precision	F-Measure
1	69.41%	*75.58%*	*72.32%*
5	**76.01%**	**80.13%**	**77.99%**
10	75.73%	78.48%	77.03%
15	*69.08%*	79.71%	73.16%

Table 11: Recognition results using LSTM concerning dataset cairo.

Epoch	Accuracy	Precision	F-Measure
1	*37.58%*	*49.05%*	*42.32%*
5	51.03%	61.1%	55.51%
10	58.1%	**66.18%**	**61.82%**
15	**58.79%**	63.4%	60.93%

Recognition results using LSTM concerning the cairo dataset are depicted in Table 11. The best precision 66.18% and the best F-Measure 66.82% are achieved when 10 is assigned to Epoch. The best accuracy 58.79% is achieved when 15 is assigned to Epoch. The accuracies, precisions, and F-Measures using features TF-IDF, TF-IDF+Sigmod, or TF-IDF+Tanh exceeded the best counterpart using LSTM when employing SVM, SMO, and RF. By the results, LSTM is not enough effective to activity recognition. The main reason is that sparse training data and relatively more neural network nodes lead to overfitting of the training set.

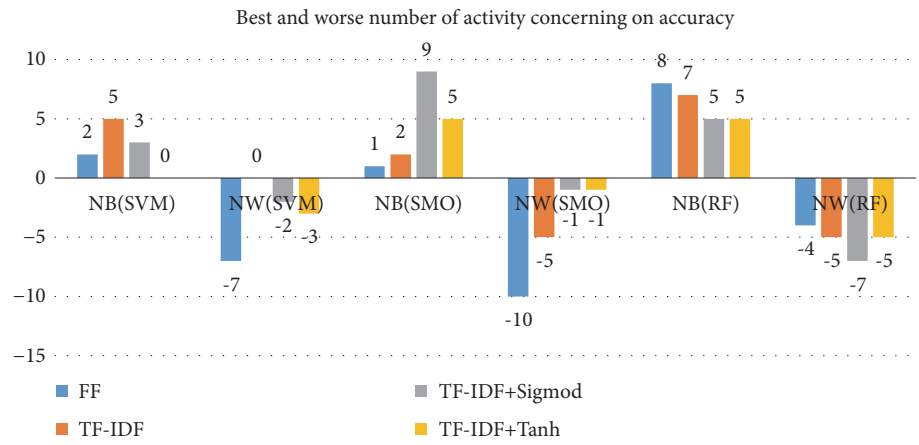


Figure 3: Best and worst number of activity concerning on accuracy concerning on "tulum2009".

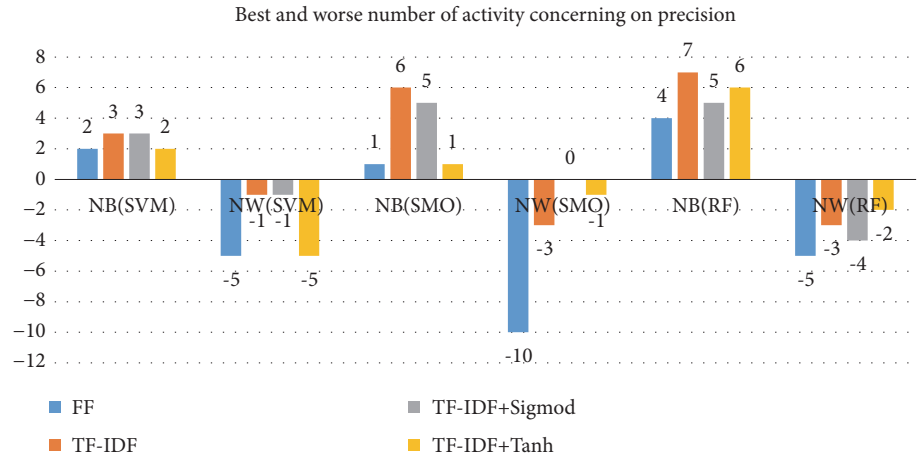


Figure 4: Best and worst number of activity concerning on precision concerning on "tulum2009".

5.3.2. The Results of Individual Activity. Best and worst recognition number of activities are counted for both accuracy and precision. Let $AC = \{ac_1, ac_2, \ldots, ac_k\}$ be set of activity categories. Let FS={"FF", "TF-IDF", "TF-IDF+Sigmod", "TF-IDF+Tanh"} be set of feature categories. For $ac \in AC$ and $f \in FS$, $ifbest_{acc}(ac, f)$ and $ifworese_{acc}(ac, f)$ denote whether ac get best and worst accuracies using f feature solving. $ifbest_{pre}(ac, f)$ and $ifworese_{pre}(ac, f)$ denote whether ac get best and worst precisions using f feature solving. For accuracy and precision, the number of best activities recognition is defined as $NBacc_f = \{ac \mid ac \in AC \text{ and } ifbest_{acc}(ac, f) == true\}$ and $NBpre_f = \{ac \mid ac \in AC \text{ and } ifbest_{pre}(ac, f) == true\}$. The number of worst activities recognition is defined as $NWacc_f = \{ac \mid ac \in AC \text{ and } ifworst_{acc}(ac, f) == true\}$ and $NWpre_f = \{ac \mid ac \in AC \text{ and } ifworst_{pre}(ac, f) == true\}$.

$NBacc_f$, $NBpre_f$, $NWacc_f$, and $NWpre_f$ of individual activity are shown in Figures 3–6 concerning two datasets. For the dataset tulum2009, FF is worst in two of three classifiers concerning on $NWacc_f$. FF is worst in all of three classifiers concerning on $NWpre_f$. FF is best only in RF concerning on $NBacc_f$. FF is not best in any of three classifiers concerning on $NBpre_f$. TF-IDF, TF-IDF+Sigmod, and TF-IDF+Tanh are close in all of three classifiers concerning on $NBacc_f$ and $NWacc_f$. TF-IDF is best in all of three classifiers concerning on $NBpre_f$. TF-IDF+Sigmod is best in two of three classifiers concerning on $WBpre_f$.

For the dataset cario, FF is worst in all of three classifiers concerning on both $NWacc_f$ and $NWpre_f$. FF is not best in any of three classifiers concerning on both $NBacc_f$ and $NBpre_f$. TF-IDF is best in two of three classifiers concerning on $NBacc_f$. TF-IDF and TF-IDF+Sigmod are best in two of three classifiers concerning on $NBpre_f$. TF-IDF and TF-IDF+Tanh are best in two of three classifiers concerning on $WBpre_f$.

In accordance with results obtained in this study, the following points must be noted. Strategies based on TF-IDF feature outperform strategy based on FF feature in accuracy, precision, and F-Measure regardless of whole or individual of activities.

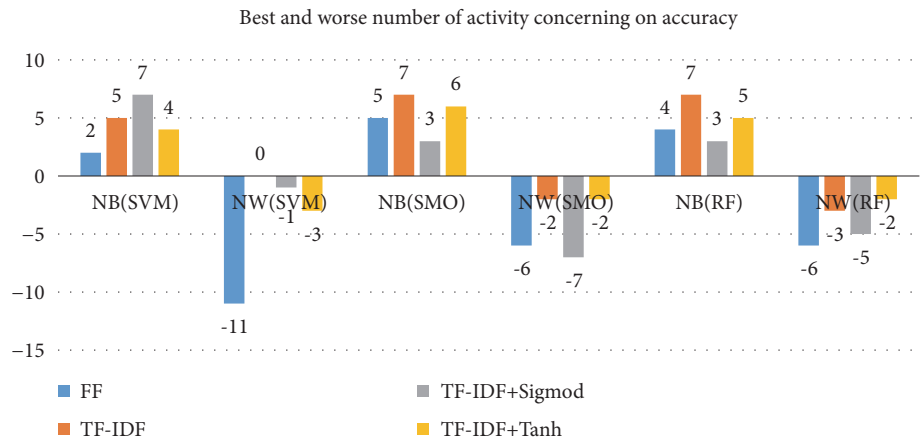


Figure 5: Best and worst number of activity concerning on accuracy concerning on "cario".

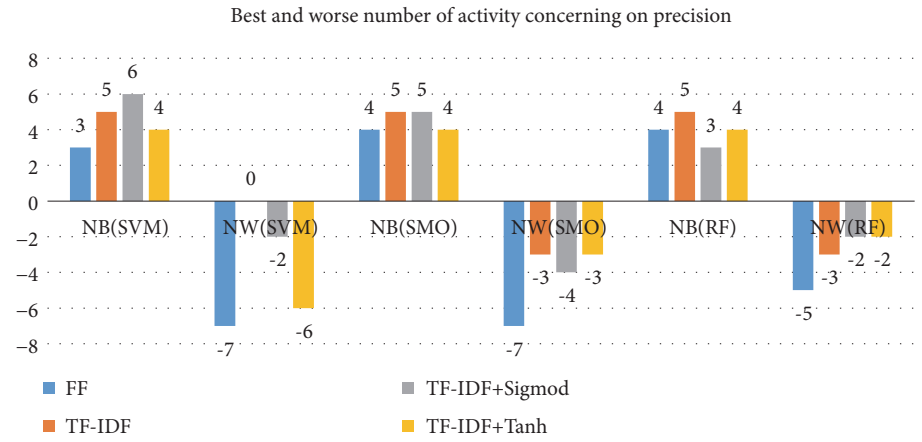


Figure 6: Best and worst number of activity concerning on precision concerning on "cario".

6. Conclusion

This paper presents the strategies based on TF-IDF as a means of activity features solving with regard to activity recognition applications. The proposed strategies were evaluated using three classifiers on two distinct datasets, and results obtained in this study demonstrate the ability of strategy based on TF-IDF to dramatically improve the performance of activity recognition systems.

Data Availability

The authors employed two public datasets "tulum2009" and "cairo" in to illustrate the applicability of the proposed approach. These datasets have been published by the Washington State University [31]. The url is http://casas.wsu.edu/datasets/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (no. 3132018194) and the Open Project Program of Artificial Intelligence Key Laboratory of Sichuan Province (no. 2018RYJ09) and CERNET Innovation Project (no. NGII20181203).

References

[1] United Nations, *World Population Ageing 2009*, Population Studies Series, United Nations, New York, NY, USA, 2010.

[2] E. De Luca d'Alessandro, S. Bonacci, and G. Giraldi, "Aging populations: the health and quality of life of the elderly," *La Clinica Terapeutica*, vol. 162, no. 1, pp. e13–e18, 2011.

[3] M. Chan, D. Estève, C. Escriba, and E. Campo, "A review of smart homes—present state and future challenges," *Computer Methods and Programs in Biomedicine*, vol. 91, no. 1, pp. 55–81, 2008.

[4] B. De Ruyter and E. Pelgrim, "Ambient assisted-living research in carelab," *Interactions*, vol. 14, no. 4, pp. 30–33, 2010.

[5] S. Nigam and A. Khare, "Integration of moment invariants and uniform local binary patterns for human activity recognition in video sequences," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 1–30, 2015.

[6] W. Lin, Y. Chen, J. Wu et al., "A new network-based algorithm for human group activity recognition in videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 826–841, 2014.

[7] D. Fortin-Simard, J.-S. Bilodeau, K. Bouchard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Exploiting passive RFID technology for activity recognition in smart homes," *IEEE Intelligent Systems*, vol. 30, no. 4, pp. 7–15, 2015.

[8] J. Yang, J. Lee, and J. Choi, "Activity recognition based on RFID object usage for smart mobile devices," *Journal of Computer Science and Technology*, vol. 26, no. 2, pp. 239–246, 2011.

[9] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2017.

[10] S. K. Guo, Y. Q. Liu, R. Chen et al., "Using an improved SMOTE algorithm to deal imbalanced activity classes in smart homes," *Neural Processing Letters*, pp. 1–24, 2018.

[11] W. Deng, H. Zhao, X. Yang, J. Xiong, M. Sun, and B. Li, "Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment," *Applied Soft Computing*, vol. 59, pp. 288–302, 2017.

[12] W. Deng, J. Xu, and H. Zhao, "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," *IEEE Access*, vol. 6, pp. 20632–20640, 2018.

[13] Y. Liu, D. Ouyang, Y. Liu, and R. Chen, "A novel approach based on time cluster for activity recognition of daily living in smart homes," *Symmetry*, vol. 9, no. 10, 2017.

[14] Y. Liu, X. Yi, R. Chen, Z. Zhai, and J. Gu, "Feature extraction based on information gain and sequential pattern for English question classification," *IET Software*, vol. 12, no. 6, pp. 520–526, 2018.

[15] W. Deng, R. Yao, H. Zhao, X. Yang, and G. Li, "A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm," *Soft Computing*, pp. 1–18, 2017.

[16] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961–974, 2012.

[17] F. Latfi, B. Lefebvre, and C. Descheneaux, "Ontology-based management of the telehealth smart home, dedicated to elderly in loss of cognitive autonomy," in *Proceedings of the CEUR workshop*, 2007.

[18] A. G. Salguero and M. Espinilla, "Ontology-based feature generation to improve accuracy of activity recognition in smart environments," *Computers & Electrical Engineering*, vol. 68, pp. 1–13, 2018.

[19] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proceedings of the 2nd International Conference on Pervasive Computing*, Lecture Notes in Computer Science, Springer, Berlin, Germany, 2004.

[20] O. Brdiczka, J. L. Crowley, and P. Reignier, "Learning situation models in a smart home," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 56–63, 2009.

[21] E. M. Tapia, S. S. Intille, W. Haskell et al., "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *Proceedings of the 11th IEEE International Symposium on Wearable Computers, ISWC '07*, 2007.

[22] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *Proceedings of the 9th IEEE International Symposium on Wearable Computers*, IEEE, Osaka, Japan, 2005.

[23] T. L. van Kasteren, G. Englebienne, and B. J. Kröse, "Hierarchical activity recognition using automatically clustered actions," in *Proceedings of the 2nd International Joint Conference on Ambient Intelligence*, vol. 7040 of *Lecture Notes in Computer Science*, pp. 82–91, Springer Berlin Heidelberg, Berlin, Germany.

[24] D. L. Vail, M. M. Veloso, and J. D. Lafferty, "Conditional random fields for activity recognition," in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '07)*, ACM, Hawaii, HI, USA, May 2007.

[25] H. Fang and L. He, "BP neural network for human activity recognition in smart home," in *Proceedings of the International Conference on Computer Science and Service System (CSSS '12)*, pp. 1034–1037, Nanjing, China, 2012.

[26] M. M. Hassan, S. Huda, M. Z. Uddin, A. Almogren, and M. Alrubaian, "Human activity recognition from body sensor data using deep learning," *Journal of Medical Systems*, vol. 42, no. 99, 2018.

[27] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017.

[28] W.-H. Chen, C. A. B. Baca, and C.-H. Tou, "LSTM-RNNs combined with scene information for human activity recognition," in *Proceedings of the 19th IEEE International Conference on e-Health Networking, Applications and Services, Healthcom '17*, pp. 1–6, China, October 2017.

[29] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and Mobile Computing*, vol. 10, pp. 138–154, 2014.

[30] N. Kondylidis, M. Tzelepi, and A. Tefas, "Exploiting tf-idf in deep convolutional neural networks for content based image retrieval," *Multimedia Tools and Applications*, vol. 77, no. 23, pp. 30729–30748, 2018.

[31] WSU CASAS Datasets, 2016, http://ailab.wsu.edu/casas/datasets.html.

Hindawi
Complexity
Volume 2019, Article ID 1403829, 12 pages
https://doi.org/10.1155/2019/1403829

Research Article

Improved Permutation Entropy for Measuring Complexity of Time Series under Noisy Condition

Zhe Chen,[1] Yaan Li ,[1] Hongtao Liang,[2] and Jing Yu[1]

[1]*School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China*
[2]*School of Physics and Information Technology, Shaanxi Normal University, Xi'an 710119, China*

Correspondence should be addressed to Yaan Li; liyaan@nwpu.edu.cn

Received 23 November 2018; Revised 8 February 2019; Accepted 25 February 2019; Published 11 March 2019

Guest Editor: Jose Garcia-Rodriguez

Measuring complexity of observed time series plays an important role for understanding the characteristics of the system under study. Permutation entropy (PE) is a powerful tool for complexity analysis, but it has some limitations. For example, the amplitude information is discarded; the equalities (i.e., equal values in the analysed signal) are not properly dealt with; and the performance under noisy condition remains to be improved. In this paper, the improved permutation entropy (IPE) is proposed. The presented method combines some advantages of previous modifications of PE. Its effectiveness is validated through both synthetic and experimental analyses. Compared with PE, IPE is capable of detecting spiky features and correctly differentiating heart rate variability (HRV) signals. Moreover, it performs better under noisy condition. Ship classification experiment results demonstrate that IPE achieves 28.66% higher recognition rate than PE at 0dB. Hence, IPE could be used as an alternative of PE for analysing time series under noisy condition.

1. Introduction

Measuring complexity of observed time series allows a better understanding of the characteristics of the system under study [1]. There is a lack of consensus on the definition of complexity [2–5]. Entropy is one of the most powerful metrics to evaluate the complexity of a signal [6]. By this definition, complexity is associated with disorder degree (randomness) and unpredictability. Many entropy approaches have been proposed in recent years, such as permutation entropy (PE) [7], approximate entropy [8], sample entropy [9], and fuzzy entropy [10], each of which has its own strengths and weaknesses. Compared with other entropy algorithms, PE is famous for its uniqueness of being computationally fast and conceptually simple. Furthermore, it is applicable for any types of signals, be they deterministic, chaotic, stochastic, stationary, or nonstationary [11]. We will therefore concentrate on PE in what follows.

PE was firstly introduced by Bandt and Pompe in 2002. Since then, it has been extensively applied in various fields. Without being exhaustive, applications such as fault diagnosis [12, 13], biomedical signal processing [6, 14–16], and stock market analysis [17, 18] can be enumerated. Despite considerable success, there are still defects in PE, which have motivated researchers to present modifications for the original algorithm. Firstly, PE is single-scale-based. Signals generated by complex systems usually show structures on multiple temporal scales. As a result, the single-scale-based PE cannot describe such time series comprehensively [19]. To remedy this, Zunino et al. proposed to calculate PE as a function of the time delay, offering a way to unveil the presence of structures on multiple temporal scales [20]. Moreover, Costa et al. proposed a coarse-graining technique [19, 21, 22]. Based on Costa's work, Aziz introduced the multiscale permutation entropy (MPE) [23] by combining the coarse-graining procedure with PE. The method is able to provide more precise descriptions for complex signals. Secondly, when a signal is mapped to permutation patterns (or ordinal patterns) using Bandt and Pompe's approach, information regarding the amplitudes is not taken into consideration. To this end, weighted-permutation entropy (WPE) [2] and amplitude-aware permutation entropy (AAPE) [24] have been developed, respectively. By assigning weights to distinct patterns, the modified methods greatly improve the ability

to detect abrupt changes in magnitude. Thirdly, the PE estimation is liable to be affected by the equal values in time series [25–27]. In the case that the sequence under study has a continuous distribution, equal values are very rare and can be simply ignored. Unfortunately, real-world data are digitalized; thus equalities are inevitable to exist. The situation could be more serious if the amplitude resolution is low. Bandt and Pompe suggested ranking the equal values according to their temporal order or breaking them by adding random perturbations [7]. However, as pointed out in a recent study [26], Bandt's method for processing equal values might lead to erroneous conclusion. To address this issue, Bian et al. have proposed modified permutation entropy (mPE) as an alternative [27], which assigns the same symbol to equal values. Although mPE can significantly improve the performance of distinguishing the heart rate variability (HRV) signals under different conditions, it also brings additional problems. For example, mPE does not assign the maximum entropy value to the white Gaussian noise (WGN), which disagrees with the fact that WGN is completely random. Lastly, PE is susceptible to noise. In order to improve PE's ability under noisy condition, researchers have suggested to apply symbolic dynamics to symbolize the time series prior to entropy estimation [28, 29]. For example, Porta et al. proposed an integrated approach based on uniform quantization (IAUQ), which has shown great ability to differentiate normal subjects and heart failure patients.

Although previous works solve some problems of PE, these methods are still deficient in some aspects: (I) mPE still overlooks the amplitude information; (II) the presence of equal values will also do harm to the WPE and AAPE algorithm; and (III) the fluctuations of signals are not taken into account by the IAUQ. In the present study, the improved permutation entropy (IPE) is proposed. The presented method not only considers the amplitude information and fluctuations of signals but also tackles the limitation of equal values. Besides, it can be directly combined with coarse-graining technique for multiscale analysis. As it will be shown below, IPE is capable of detecting spiky features and correctly differentiating HRV signals (time series with a lot of equal values). Moreover, compared with PE and its modifications, IPE performs better under noisy condition. The experimental results further validate the effectiveness of the proposed method.

The remainder of this paper is organized as follows: detailed description of the IPE algorithm is provided in Section 2; the effect of different parameters is studied in Section 3; synthetic and experimental data are analysed in Section 4; the paper is concluded in Section 5.

2. Methods

In this section, the proposed IPE algorithm is described in detail. To gain insight into the advantages of IPE, differences between the PE, IAUQ, and IPE are compared. For the purpose of multiscale analysis, a multiscale version of IPE is also introduced.

2.1. Permutation Entropy. For a time series $x = \{x_i\}_{i=1}^{N}$, with given embedding dimension m and time delay τ, the embedding vectors are represented as

$$X(j) = \left[x_j, x_{j+\tau}, \cdots x_{j+(m-1)\tau}\right], \tag{1}$$

where $j = 1, 2, \cdots N-(m-1)\tau$. Then each of the $N-(m-1)\tau$ subvectors is arranged in ascending order:

$$\begin{aligned} x\left(j+\left(k_1-1\right)\tau\right) \le x\left(j+\left(k_2-1\right)\tau\right) \le \cdots \\ \le x\left(j+\left(k_m-1\right)\tau\right). \end{aligned} \tag{2}$$

If $x(j+(k_{l1}-1)\tau) = x(j+(k_{l2}-1)\tau)$, Bandt and Pompe suggested ranking the equal values according to their temporal order [7]; that is, $x(j+(k_{l1}-1)\tau) < x(j+(k_{l2}-1)\tau)$, when $k_{l1} < k_{l2}$. Next, an ordinal pattern, which is one of $m!$ possibilities, can be assigned to each $X(j)$ and be denoted as

$$\pi_l(j) = \left(k_1 k_2 \cdots k_m\right), \tag{3}$$

where $1 \le l \le m!$. Compute the probability distribution p_l of each ordinal pattern; the normalized permutation entropy is finally defined as

$$H_{PE}(m,\tau) = \frac{\left[-\sum_{l=1}^{h} p_l * \ln\left(p_l\right)\right]}{\ln(m!)}, \tag{4}$$

where $h \le m!$ and $\ln(m!)$ represents the maximum value of H_{PE}.

It is known that there exist some drawbacks of the above definition of PE [2, 24–27]. Take Figure 1 as an example; in spite of the great amplitude differences between vectors, $[1, 1.1, 3]$, $[1, 2.9, 3]$, $[1, 1.1, 1.2]$, and $[2.8, 2.9, 3]$ are mapped to the same pattern (012). This is due to the fact that only the order relation is retained in PE [2, 24]. Moreover, because of the mechanism of PE for dealing with equal values, $[1, 1, 1]$ and $[2, 2, 2]$are also symbolized as (012). For time series having a number of repeated values, probability distribution of some ordinal patterns may be overestimated, leading to a biased PE evaluation [26]. Last but not least, small difference of the amplitude values between sample points may be caused by noise. However, PE will assign different patterns for vectors $[1.01, 1, 1.01]$ and $[1, 1.01, 1.01]$, which might also result in inaccurate entropy estimation, especially under noisy condition. All above-mentioned limitations of PE motivate us to propose the IPE algorithm.

2.2. Improved Permutation Entropy. The IPE algorithm is mainly composed of two major parts: (I) definition of pattern and (II) entropy estimation. Consider the reconstruction vectors in (1). The first column of the embedding vectors, that is, $X(:, 1)$, is firstly symbolized based on uniform quantization (UQ). As shown in (5), $x_{\min}$ and $x_{\max}$ stand for the minimum and maximum value of the observed time series x, respectively. L denotes the discretization level and $\Delta = (x_{\max} - x_{\min})/L$. For an input data μ, the UQ procedure produces an integer ranging from 0 to $L-1$.

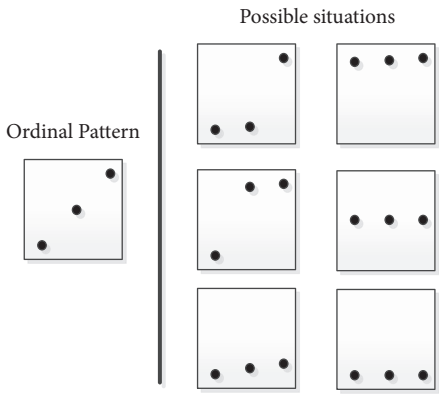


Figure 1: An example of some m-dimensional subvectors that are symbolized to the same ordinal pattern ($m = 3$ is used in this example).

$$UQ(\mu) = \begin{cases} 0 & x_{\min} \le x < \Delta \\ 1 & \Delta \le x < 2\Delta \\ \vdots & \vdots \\ L-1 & (L-1)\Delta \le x \le x_{\max} \end{cases}, \quad (5)$$

Let $S(:,1)$ denote the symbolization result of $X(:,1)$. Then, for the kth column of embedding vectors $X(:,k)$, $2 \le k \le m$, $S(:,k)$ is calculated by the following equation:

$$S(j,k) = S(j,1) + \left\lfloor \frac{(X(j,k) - X(j,1))}{\Delta} \right\rfloor, \quad 1 \le j \le N - (m-1)\tau,\ 2 \le k \le m. \quad (6)$$

Finally, S is defined as the pattern matrix. Each row of S corresponds to a pattern π_l, $1 \le l \le L^{\wedge}m$. Compute the probability distribution p_l of each pattern π_l; the normalized IPE is written as

$$H_{IPE}(m,\tau,L) = \frac{\left[-\sum_{l=1}^{h} p_l * \ln(p_l)\right]}{\ln(L^{\wedge}m)}, \quad (7)$$

where $h \le L^{\wedge}m$ and $\ln(L^{\wedge}m)$ is the maximum value of H_{IPE}, which is only reached when the patterns have a uniform distribution.

It is worth noting that the main difference between the IAUQ and IPE is the definition of pattern. Unlike IAUQ, only the first element of the embedding vector is symbolized by UQ in the IPE algorithm. The patterns of other elements are calculated by (6), which takes the fluctuations of signals into consideration. Take a vector $[1.9, 1, 2.1, 4]$ as an example; let $L = 3$ and $m = 4$; the vector will be symbolized as $[0, 0, 1, 2]$ and $[0, 0, 0, 2]$ by IAUQ and IPE, respectively.

There are 4 major differences between the PE and IPE algorithm. Firstly, amplitude information and fluctuations of signals are considered in IPE. Unlike PE that assigns the same pattern (012) to all vectors in Figure 1, for $L = 3$ and $m = 3$, IPE maps vectors $[1, 1.1, 3]$, $[1, 2.9, 3]$, $[1, 1.1, 1.2]$, and $[2.8, 2.9, 3]$ to (002), (022), (000), and(222), respectively. Secondly, the same symbol is assigned to equal values in IPE. For example, $[1, 1, 1]$ and $[2, 2, 2]$ are separately symbolized as (000) and (111). The way that IPE processes repeated values will not cause overestimating of permutation patterns; thus a more precise complexity measure can be obtained for time series with numerous equal values. Thirdly, IPE is more robust to noise interference. Vectors $[1.01, 1, 1.01]$ and $[1, 1.01, 1.01]$ are both transformed to (000), meaning that the presence of proper noise will not influence the IPE estimation. Finally, there are $L^{\wedge}m$ possible patterns in IPE, while that in PE is $m!$.

2.3. Multiscale Improved Permutation Entropy. Within the multiscale improved permutation entropy (MIPE) algorithm, only the coarse-graining procedure is required to proceed prior to IPE estimation. Given a scale factor s, the input sequence $x = \{x_i\}_{i=1}^{N}$ is decomposed by the coarse-graining technique, yielding a new subsequence of length N/s, which can be written as

$$y_i^s = \frac{1}{s} \sum_{i=(j-1)s+1}^{js} x(i), \quad (8)$$

where $1 \le j \le N/s$. The obtained new time series is then served as the input to the IPE algorithm for multiscale analysis. It is important to note that IPE can also be combined with other multiscale analysis techniques (e.g., [20]). In the present study, we select the prevalent coarse-graining technique for subsequent multiscale analysis.

3. Selection of Parameters

There are some parameters that need to be predetermined for computing the IPE and MIPE algorithm, such as embedding dimension m, time delay τ, discretization level L, scale factor s, and data-length N. The time lag is analogous to downsampling to some extent, and $\tau = 1$ is usually taken for structural preservation [30, 31]. Without specification, $\tau = 1$ is chosen for subsequent study. In the following, the selection of other parameters is investigated through two synthetic signals whose characteristics are known: (I) WGN and (II) 1/f noise.

3.1. Selection of Embedding Dimension. In this subsection, we examined how IPE estimations varied as a function of the embedding dimension m. There were 30 independent realizations generated for each synthetic signal with a data-length of 50000. Discretization level $L = 4$ was used in this experiment. The average IPE values with their standard deviation (SD) error bars over a varying m are provided in Figure 2. As can be seen, the IPE has very low SD, implying that it offers consistent entropy estimation. There is a slight decrease in entropy values for both synthetic signals as m increases. The entropy loss phenomenon at large embedding dimensions agrees with the inference in [32], which states that the trajectory of higher-dimensional embedding vectors is more predictable than those with lower dimension. Hence, lower entropy (complexity) could be expected at higher

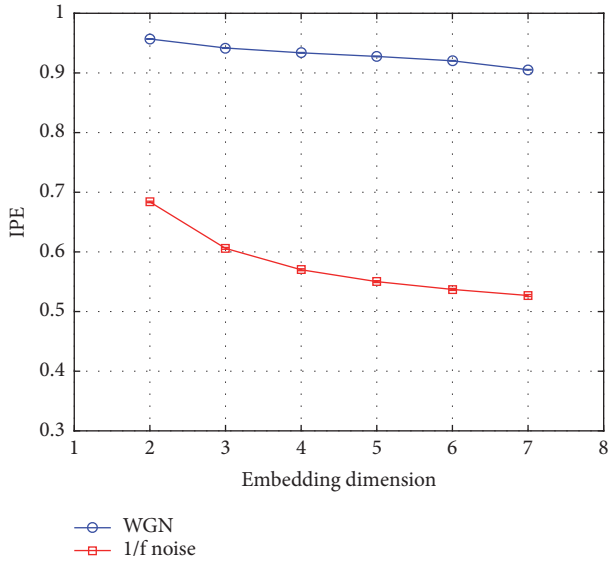


Figure 2: Error bar plot of IPE over a varying embedding dimension.

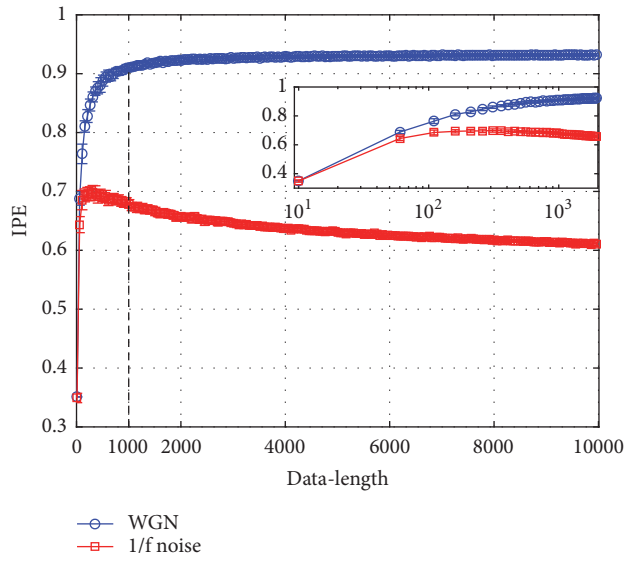


Figure 3: Error bar plot of IPE over a varying data-length.

embedding dimensions. For practical purposes, Bandt and Pompe suggested setting $3 \leq m \leq 7$ for computing PE [7]. Since m has only little influence on IPE evaluation within that range, without loss of generality, we selected $m = 4$ for subsequent study.

3.2. Selection of Data-Length. The effect of data-length on IPE evaluation is depicted in Figure 3. The results were obtained by averaging 30 independent trials. The data-length of synthetic signals was varied from 10 to 10000 with a step length of 50. Discretization level $L = 4$ was chosen for computing the IPE algorithm. As shown in the picture, with increasing sample points, IPE curves of both synthetic signals firstly increase and then gradually converge to a constant value. The result implies that the IPE method provides unreliable entropy estimation for very short time series. For example, the WGN and the 1/f noise become indistinguishable by IPE when $N \leq 100$. According to [26, 33], $N >> m!$ must be satisfied to achieve a reliable PE measurement, where $m!$ denotes the number of potential permutation patterns in PE. Analogously, N should fulfill $N >> L^{\wedge}m$ in the IPE algorithm, where $L^{\wedge}m$ is the possible pattern in IPE. Because $m = 4$ and $L = 4$ were used in Figure 3, there are 256 possible patterns. As can be seen from Figure 3, IPE curves of both synthetic signals start to converge when $N = 1000$. Since $1000 \approx 4 * 256$, we can roughly deduce that $N \geq 4(L^{\wedge}m)$ should be satisfied for reliable IPE estimation.

3.3. Selection of Discretization Level. The discretization level L is a very important parameter that affects the performance of the IPE approach. The relation between IPE values and the discretization level is shown in Figure 4. The mean IPE values with their SD error bars over a varied L were plotted by averaging 30 independent trials. Data-length $N = 50000$ was selected for generating synthetic signals. As the discretization level increases, mean IPE values of the WGN firstly grow and then gradually converge, while that of the 1/f noise keeps progressively increasing. It is also seen that IPE does not reach its maximum for the completely random WGN. This is due to the fact that the definition of pattern in the IPE algorithm is partly based on UQ, in which only the dominant features of dynamics are faithfully preserved [29, 34]. A larger L means more abundant information of the observed time series is retained, but it becomes more sensitive to noise and calls for more sample points to provide reliable results. The situation is just the reverse when L is small. Hence, the selection of L involves a trade-off between accurate entropy estimation and high noise immunity. Although the IPE approach does not assign the maximum entropy value for the WGN, it obtains large enough entropy measurements (>0.93) when $L \geq 4$, which is generally identical to the fact. Without loss of generality, $L = 4$ is selected for subsequent study.

3.4. Selection of Scale Factor. According to (8), an increasing scale factor will result in the data-length of subsequence rapidly decreasing. As mentioned in Section 3.2, $N \geq 4(L^{\wedge}m)$ must be satisfied in the IPE algorithm. Since the data-length of subsequence is equal to N/s, we can therefore deduce that $s \leq N/4(L^{\wedge}m)$ must be fulfilled in the MIPE method.

4. Results and Discussion

After having established the basic properties of the IPE algorithm, in this section, its advantages are fully demonstrated through synthetic and experimental analysis. For comparison purpose, other entropic approaches, such as PE, WPE, AAPE, mPE, and IAUQ (Shannon entropy is used to implement IAUQ) are also utilized for analysis.

4.1. Spiky Data Analysis. A signal having abrupt changes in magnitude was firstly tested. As shown in Figure 5(a), the synthetic signal consists of an impulse and additive WGN. Sliding windows of 500 samples with 400 points overlapped

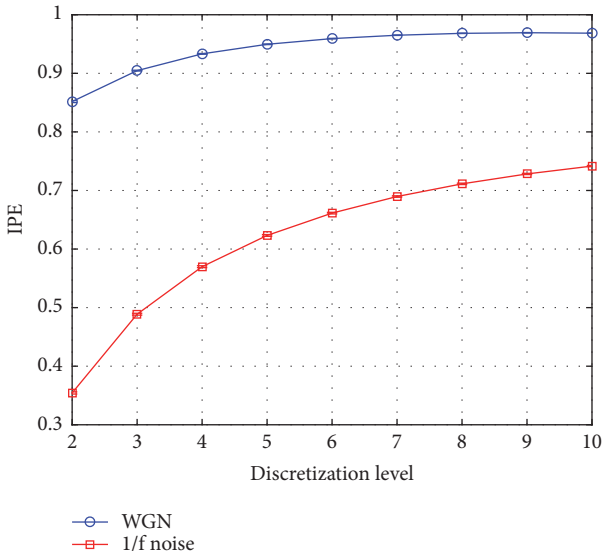


Figure 4: Error bar plot of IPE over a varying discretization level.

were used for entropy calculation. It is important to point out that the parameters in PE, WPE, AAPE, mPE, and IAUQ were set the same as those in IPE, and the adjusting coefficient was selected as 0.5 for AAPE. Without specification, the same parameters will be used for subsequent study. It is shown in Figure 5(b) that both PE and mPE remain constant across all windows, while a remarkable drop of entropy values can be noticed in the impulse region for all IPE, IAUQ, WPE, and AAPE methods. The results are due to the fact that PE and mPE overlook the amplitude information, while that is fully considered in IPE, IAUQ, WPE, and AAPE approaches [2, 7, 24, 27]. The result suggests the strong ability of IPE in detecting spiky features.

4.2. Heart Rate Variability Signal Analysis. Typically, the HRV signals derived from electrocardiogram have numerous equal values because of the limited sampling frequency. We therefore used such time series to examine how the repeated values would affect the entropy values of various entropic methods. The HRV signals analysed in this paper originate from the MIT-BIH Fantasia database, which has been widely used in scientific works [25, 27, 35]. Herein, we analysed a collection of 10 heart-beat time series including 5 young and 5 elderly subjects, each of which has 4096 sample points. By averaging all the subjects, Table 1 gives the percentage of equal values in embedding vectors with different embedding dimensions. It is found that the percentage of equal values approximately grows by 10% when the embedding dimension increases by 1, indicating that the equal values are almost randomly distributed in the time series. Figure 6 provides the entropy analysis results of the HRV signals. Unfortunately, the young and elderly subjects are unclassifiable by the PE, WPE, and AAPE methods. This occurs because the presence of numerous randomly distributed equal values leads to a stochastic distribution of ordinal patterns. Consequently, the entropy values of all these methods are close to 1, being indistinguishable. On the other hand, IPE, IAUQ, and

Table 1: Percentage of equal values found in embedding vectors with diverse embedding dimensions.

Embedding dimension	m=3	m=4	m=5	m=6
Percentage (%)	17.91	30.27	41.83	51.72

mPE map the equal values to the same symbol, so they have more sufficient potential patterns to correspond with diverse embedding vectors. The subjects are therefore well differentiated by these methods. The result in Figure 6 proves the advantage of IPE in processing signals with lots of equal values.

4.3. Analysis of Autoregressive Models. We next utilized the entropic approaches to investigate the autoregressive (AR) processes over a range of coarse-grained scales. The synthetic AR time series were generated by

$$AR_p(t) = \sum_{i=1}^{p} \alpha_i AR(t-i) + n(t), \tag{9}$$

where $n(t)$ is the WGN with zero mean and unit variance, p denotes the order of AR processes, and α_i stands for the correlation coefficients. Parameters for generating AR processes with diverse orders are listed in Table 2. For each order, 30 independent realizations with 10000 samples were produced.

The entropy analysis results of the synthetic AR time series are shown in Figure 7, where averaged entropy values with their SD error bars over varying scale factors (1~20) are plotted. Figures 7(b)–7(e) are the results of PE, WPE, AAPE, and mPE, respectively. Very similar entropy curves are obtained for these methods except for the differences in absolute entropy values. As can be seen, AR_6 and AR_7 are not well distinguished by them. By contrast, both IPE and IAUQ differentiate all synthetic AR time series well. Particularly, for all scales, the mean IPE values are ranked in a descending order as the order of the AR time series increases, which is more consistent with the fact. As shown in (9), as the order of AR process grows, there is an increasing correlation among sample points. In terms of predictability, a higher-order AR time series is more predictable than that with a lower order and should have been assigned lower entropy. Comparing Figure 7(a) with Figure 7(f), it is seen that the result of IPE ranges from 0.2 to 0.9, while that of IAUQ ranges from 0.5 to 1. The difference is due to the diverse definition of pattern in two methods. The results in Figure 7 suggest that IPE is powerful for distinguishing signals with different predictability.

4.4. Analysis of Ship-Radiated Noise. We finally tested the effectiveness of IPE under noisy condition. To this end, three types of real ship-radiated noise were analysed. Due to the effect of ocean ambient noise, the ship sounds are usually recorded in noisy conditions. The experimental data were taken from ShipsEar [36], which is an open database of underwater recordings of ship sounds. The sounds of three types of marine vessels were measured at a sampling rate of 52734 Hz. Three types of ships belong to the passenger ship,

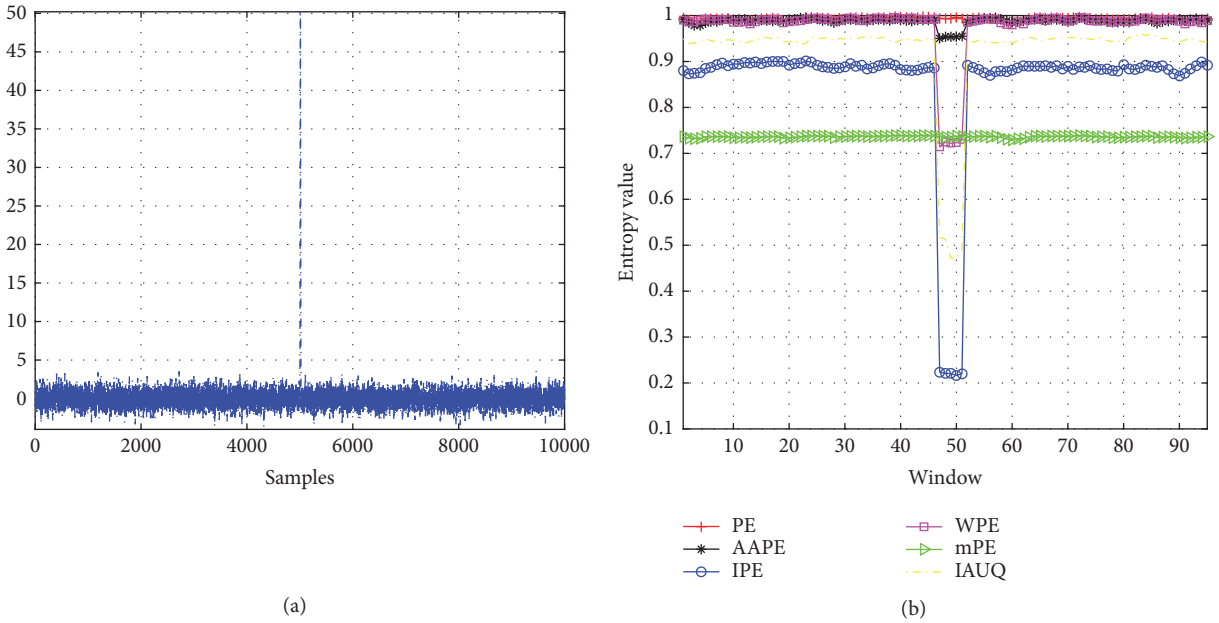


Figure 5: Entropy analysis for time series having spiky features. (a) Waveform of the synthetic signal. (b) Entropy estimation of the synthetic signal.

Table 2: The correlation coefficients for generating AR processes.

	α_1	α_2	α_3	α_4	α_5	α_6	α_7
AR_1	1/2	-	-	-	-	-	-
AR_2	1/2	1/4	-	-	-	-	-
AR_3	1/2	1/4	1/8	-	-	-	-
AR_4	1/2	1/4	1/8	1/16	-	-	-
AR_5	1/2	1/4	1/8	1/16	1/32	-	-
AR_6	1/2	1/4	1/8	1/16	1/32	1/64	-
AR_7	1/2	1/4	1/8	1/16	1/32	1/64	1/128

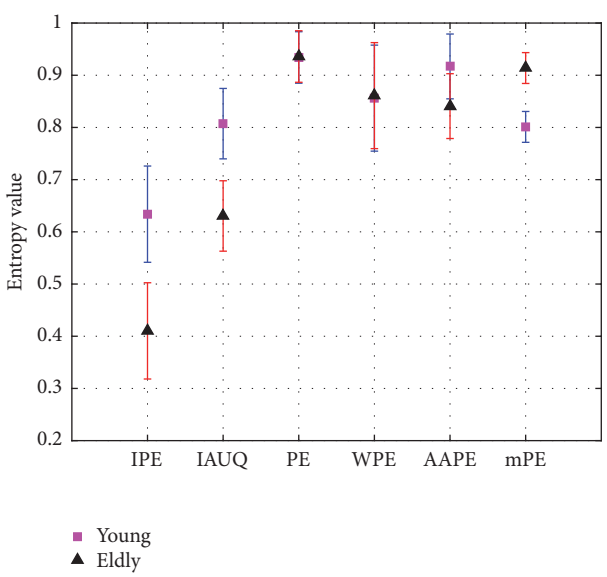


Figure 6: Entropy analysis of HRV time series.

ocean line, and motorboat, respectively. Classifying these ships from their radiating noise can be helpful for monitoring the maritime traffic. For more detailed descriptions of the data, please refer to [36]. Figure 8 shows the recorded time series of three ships.

For each type of ship-radiated noise, the data was equally cut into 50 pieces, each of which contains 52734 sample points. Figure 9 provides the feature extraction results using various entropic methods. The averaged entropy values with their SD error bars over varying scale factors (1~20) were plotted. Again, PE and mPE achieve very similar entropy curves except for the difference in absolute entropy values. This occurs owing to the high sampling rate in the experiment. The high sampling rate results in very few equal values existing in the ship signals; the definitions of pattern in both approaches become similar in such situation; mPE is thus approximate to PE. Because both WPE and AAPE consider the amplitude information of signals through assigning weights to different patterns, similar entropy curves can also be found in these two algorithms. Visually, three types of ships are more distinguishable when utilizing IPE, IAUQ,

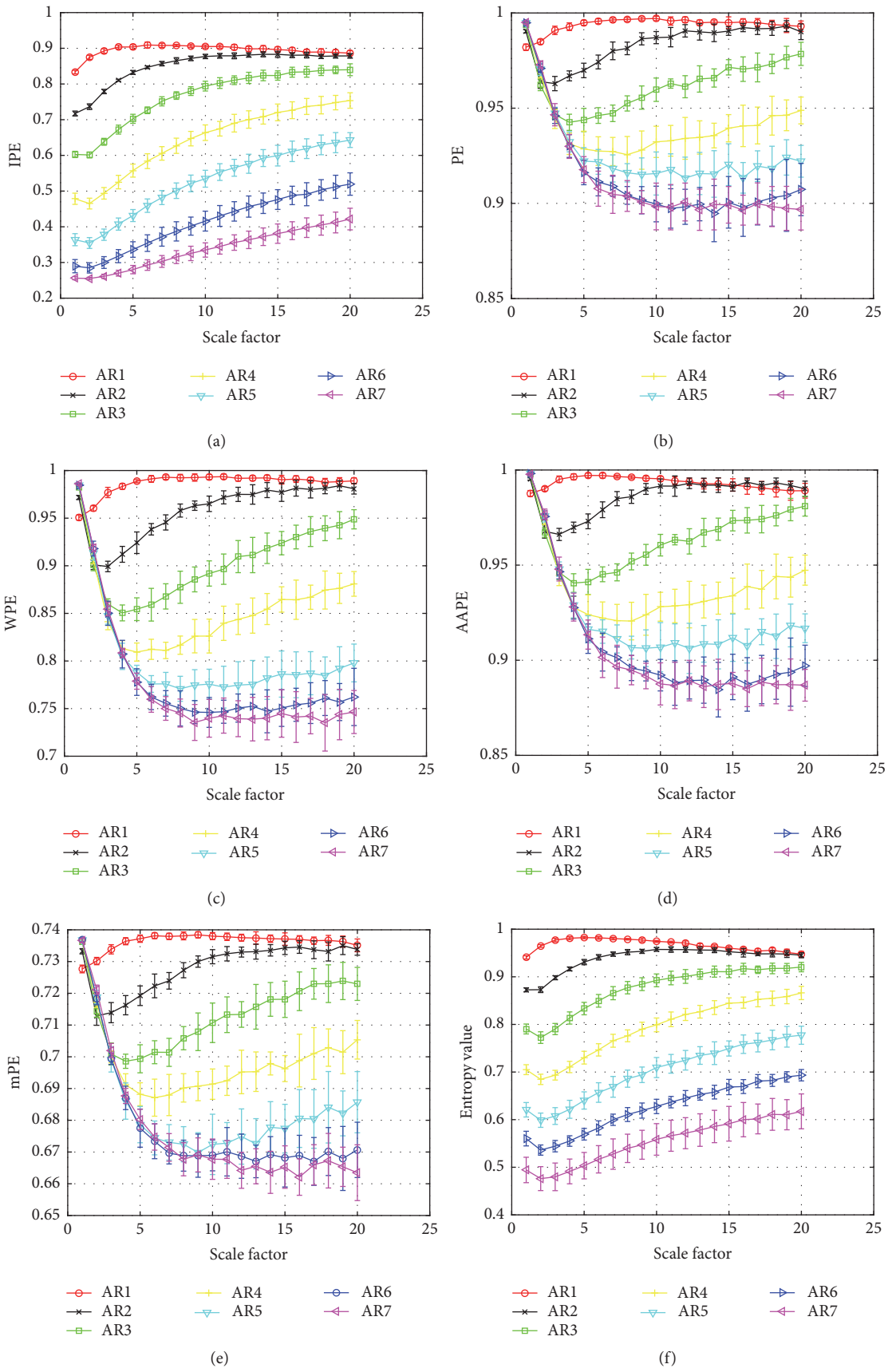


Figure 7: Entropy analysis of AR time series. (a) Result of IPE. (b) Result of PE. (c) Result of WPE. (d) Result of AAPE. (e) Result of mPE. (f) Result of IAUQ.

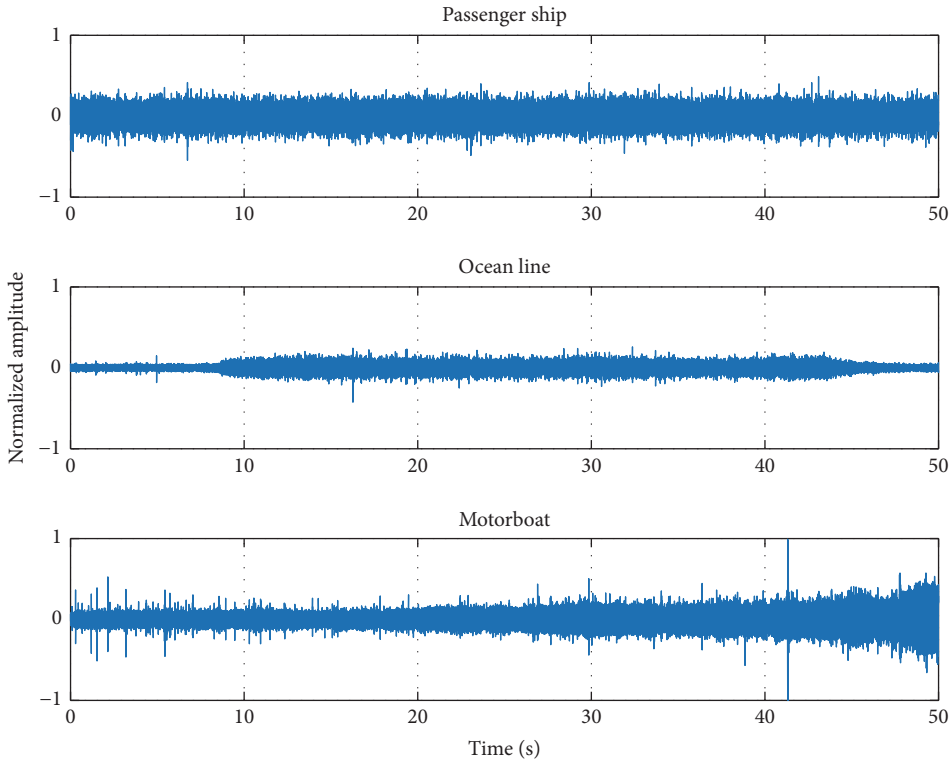


Figure 8: Recorded time series of three types of ship-radiated noise.

WPE, and AAPE. This may be due to the fact that all these methods take amplitude information into consideration.

Lower signal to noise ratio (SNR) condition was generated by adding WGN to the ship-radiated noise. Figure 10 gives the entropy analysis results under 5dB condition. Except for IPE, adding WGN seriously affects the entropy estimation of other algorithms, especially at lower scale factors. For example, compared with corresponding results in Figure 9, these methods assign much higher entropy values for all three types of ships when $s = 1$. As can be seen, there is little difference between Figures 9(a) and 10(a), meaning that IPE is more robust to noise. It is also found that the difference between IPE and IAUQ is obvious if comparing Figure 10(a) with Figure 10(f). Since IPE also considers the fluctuations of signals, it performs better under noisy condition.

The extracted entropy features under different SNR conditions were further processed by using the probability neural network (PNN) [37], which is a powerful tool for classification. For each type of vessels, 20 noise-free pieces were used for training and the other 30 pieces were for testing. Regarding situations with different SNR, all 50 pieces were set as test datasets. Table 3 shows the detailed classification results, which agree well with the entropy analysis results in Figures 9 and 10. All the entropic methods perfectly classify three types of ships in noise-free or high SNR (10dB) condition. With a decreasing SNR (5dB), classification performance of PE, AAPE, and mPE sharply declines, while that of IPE, IAUQ, and WPE remains unchanged. As the SNR further decreases (0dB), recognition rate of other entropic methods reduces to 53.33% or lower, while IPE still achieves an acceptable accuracy of 69.33%. This result validates the effectiveness of the IPE under noisy condition.

Table 3: Classification accuracy of three types of ships by PNN.

	0dB	5dB	10dB	Original
IPE	69.33%	100%	100%	100%
PE	40.67%	65.33%	100%	100%
WPE	53.33%	100%	100%	100%
AAPE	48.67%	66.67%	100%	100%
mPE	40%	65.33%	100%	100%
IAUQ	52.67%	100%	100%	100%

5. Conclusions

The IPE algorithm is proposed in this paper. The parameter selection of IPE is investigated, and the effectiveness of the proposed method is tested through synthetic and experimental analysis. It shows some advantages as listed below. Firstly, IPE takes amplitude information of time series into account and thus is more powerful for detecting spiky features than PE and mPE. Secondly, IPE assigns the same symbol to equalities and has sufficient potential patterns to correspond with different embedding vectors. Compared with PE, WPE, and AAPE, it shows better performance for processing signals with numerous repeated values, such as the HRV signals. Moreover, it can better characterize signals with diverse predictability. Lastly, IPE achieves much higher

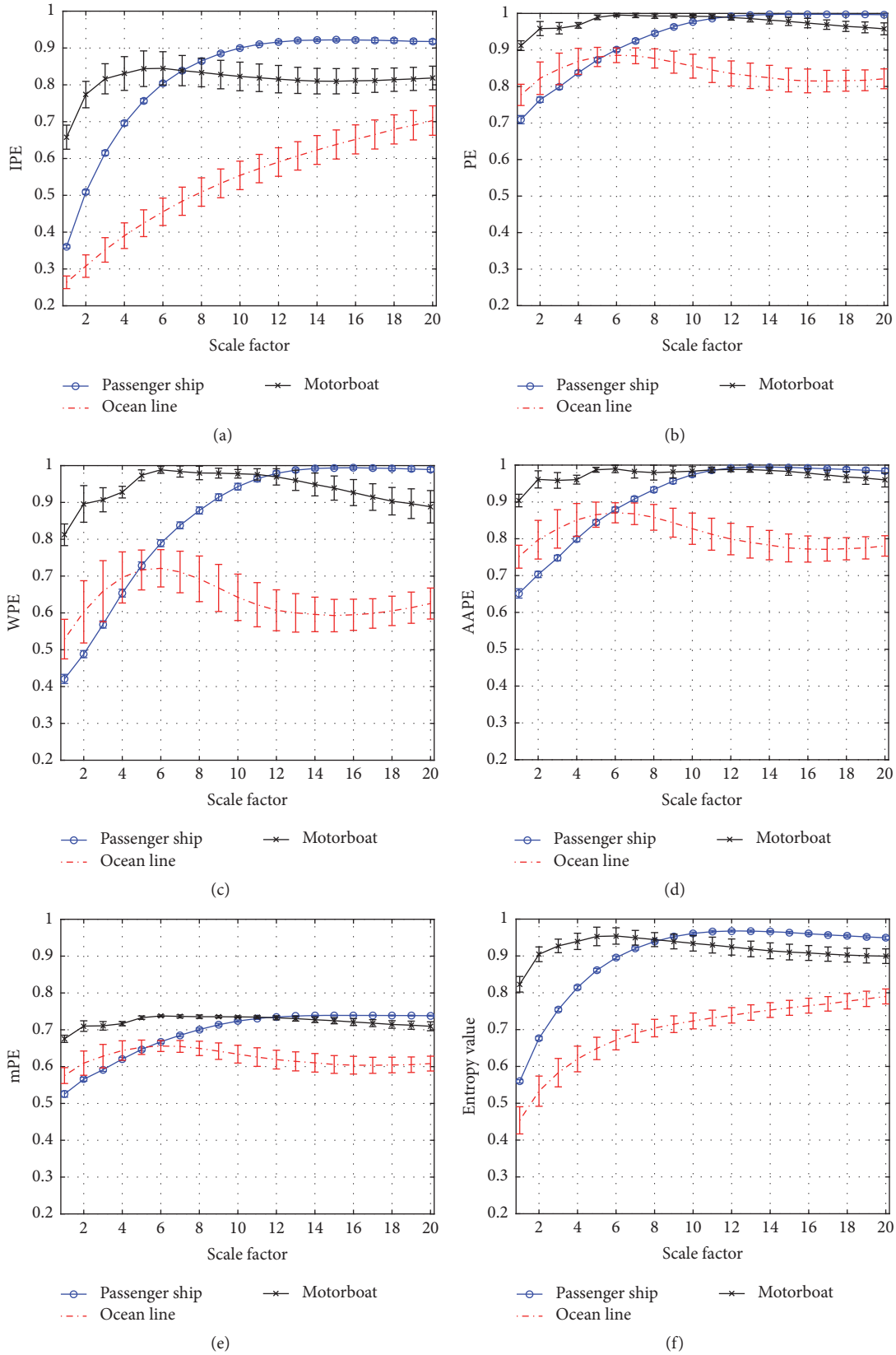


Figure 9: Entropy analysis of three types of ship-radiated noise. (a) Result of IPE. (b) Result of PE. (c) Result of WPE. (d) Result of AAPE. (e) Result of mPE. (f) Result of IAUQ.

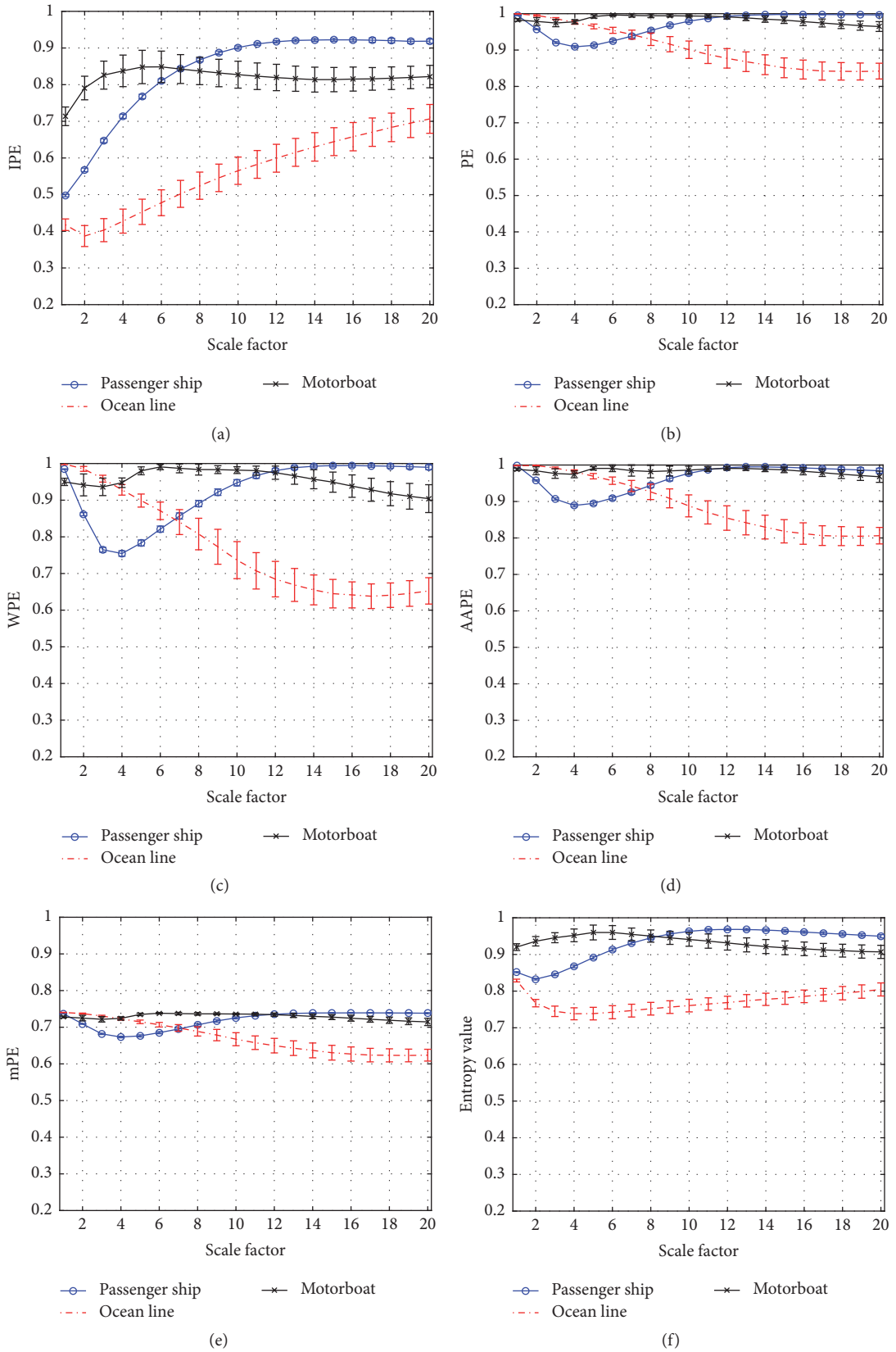


Figure 10: Entropy analysis of three types of ship-radiated noise in 5dB condition. (a) Result of IPE. (b) Result of PE. (c) Result of WPE. (d) Result of AAPE. (e) Result of mPE. (f) Result of IAUQ.

recognition rate for classifying ships under noisy conditions than PE and its modifications, implying that it is applicable for analysing signals under noisy condition. In the future work, IPE could be applied to various engineering applications such as fault diagnosis, acoustic signal processing, and stock market analysis.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (no. 51178157, no. 51409214, no. 11574250, and no. 51709228).

References

[1] M. Zanin, A. Rodriguez-Gonzalez, E. Menasalvas Ruiz, and D. Papo, "Assessing time series reversibility through permutation patterns," *Entropy*, vol. 20, no. 9, Article ID 665, 2018.

[2] B. Fadlallah, B. Chen, A. Keil, and J. Príncipe, "Weighted-permutation entropy: a complexity measure for time series incorporating amplitude information," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 87, no. 2, Article ID 022911, 2013.

[3] S. He, K. Sun, and H. Wang, "Modified multiscale permutation entropy algorithm and its application for multiscroll chaotic systems," *Complexity*, vol. 21, no. 5, pp. 52–58, 2016.

[4] R. López-Ruiz, H. L. Mancini, and X. Calbet, "A statistical measure of complexity," *Physics Letters A*, vol. 209, no. 5-6, pp. 321–326, 1995.

[5] O. A. Rosso, L. Zunino, D. G. Pérez et al., "Extracting features of Gaussian self-similar stochastic processes via the Bandt-Pompe approach," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 6, Article ID 061114, 2007.

[6] H. Azami and J. Escudero, "Improved multiscale permutation entropy for biomedical signal analysis: interpretation and application to electroencephalogram recordings," *Biomedical Signal Processing and Control*, vol. 23, pp. 28–41, 2016.

[7] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Physical Review Letters*, vol. 88, no. 17, Article ID 174102, 2002.

[8] S. M. Pincus and A. L. Goldberger, "Physiological time-series analysis: what does regularity quantify?" *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 266, no. 2, pp. 1643–1656, 1994.

[9] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.

[10] W. Chen, J. Zhuang, W. Yu, and Z. Wang, "Measuring complexity using FuzzyEn, ApEn, and SampEn," *Medical Engineering & Physics*, vol. 31, no. 1, pp. 61–68, 2009.

[11] A. Humeau-Heurtier, C.-W. Wu, and S.-D. Wu, "Refined Composite Multiscale Permutation Entropy to Overcome Multiscale Permutation Entropy Length Dependence," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2364–2367, 2015.

[12] Y. Li, G. Li, Y. Yang, X. Liang, and M. Xu, "A fault diagnosis scheme for planetary gearboxes using adaptive multi-scale morphology filter and modified hierarchical permutation entropy," *Mechanical Systems and Signal Processing*, vol. 105, pp. 319–337, 2018.

[13] S. Zhou, S. Qian, and W. Chang, "A novel bearing multi-fault diagnosis approach based on weighted permutation entropy and an improved SVM ensemble classifier," *Sensors*, vol. 18, no. 6, Article ID 1934, 2018.

[14] X. Li, S. Cui, and L. J. Voss, "Using permutation entropy to measure the electroencephalographic effects of sevoflurane," *Anesthesiology*, vol. 109, no. 3, pp. 448–456, 2008.

[15] X. Li, G. Ouyang, and D. A. Richards, "Predictability analysis of absence seizures with permutation entropy," *Epilepsy Research*, vol. 77, no. 1, pp. 70–74, 2007.

[16] X. Zhu, H. Xu, J. Zhao, and J. Tian, "Automated Epileptic Seizure Detection in Scalp EEG Based on Spatial-Temporal Complexity," *Complexity*, vol. 2017, Article ID 5674392, 8 pages, 2017.

[17] L. Zunino, M. Zanin, B. M. Tabak, D. G. Pérez, and O. A. Rosso, "Forbidden patterns, permutation entropy and stock market inefficiency," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 14, pp. 2854–2864, 2009.

[18] K. Xu and J. Wang, "Weighted fractional permutation entropy and fractional sample entropy for nonlinear Potts financial dynamics," *Physics Letters A*, vol. 381, no. 8, pp. 767–779, 2017.

[19] M. Costa, A. L. Goldberger, and C.-K. Peng, "Multiscale entropy analysis of complex physiologic time series," *Physical Review Letters*, vol. 89, no. 6, Article ID 068102, 2002.

[20] L. Zunino, M. C. Soriano, and O. A. Rosso, "Distinguishing chaotic and stochastic dynamics from time series by using a multiscale symbolic approach," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 86, no. 4, Article ID 046210, 2012.

[21] M. Costa, A. L. Goldberger, and C. Peng, "Multiscale entropy analysis of biological signals," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 71, no. 2, Article ID 021906, 2005.

[22] L. Faes, A. Porta, M. Javorka, and G. Nollo, "Efficient computation of multiscale entropy over short biomedical time series based on linear state-space models," *Complexity*, vol. 2017, Article ID 1768264, p. 13, 2017.

[23] W. Aziz and M. Arif, "Multiscale permutation entropy of physiological time series," in *Proceedings of the 9th International Multitopic Conference (INMIC '05)*, pp. 1–6, IEEE, Karachi, Pakistan, December 2005.

[24] H. Azami and J. Escudero, "Amplitude-aware permutation entropy: Illustration in spike detection and signal segmentation," *Computer Methods and Programs in Biomedicine*, vol. 128, pp. 40–51, 2016.

[25] D. Cuesta–Frau, M. Varela–Entrecanales, A. Molina–Picó, and B. Vargas, "Patterns with equal values in permutation entropy: Do they really matter for biosignal classification?" *Complexity*, vol. 2018, Article ID 1324696, pp. 1–15, 2018.

[26] L. Zunino, F. Olivares, F. Scholkmann, and O. A. Rosso, "Permutation entropy based time series analysis: Equalities in the input signal can lead to false conclusions," *Physics Letters A*, vol. 381, no. 22, pp. 1883–1892, 2017.

[27] C. Bian, C. Qin, Q. D. Y. Ma, and Q. Shen, "Modified permutation-entropy analysis of heartbeat dynamics," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 85, no. 2, Article ID 021906, pp. 439–441, 2012.

[28] U. Parlitz, S. Berg, S. Luther, A. Schirdewan, J. Kurths, and N. Wessel, "Classifying cardiac biosignals using ordinal pattern statistics and symbolic dynamics," *Computers in Biology and Medicine*, vol. 42, no. 3, pp. 319–327, 2012.

[29] A. Porta, L. Faes, M. Masé et al., "An integrated approach based on uniform quantization for the evaluation of complexity of short-term heart period variability: Application to 24h Holter recordings in healthy and heart failure humans," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 1, Article ID 015117, 2007.

[30] Z. Chen, Y. Li, H. Liang, and J. Yu, "Hierarchical cosine similarity entropy for feature extraction of ship-radiated noise," *Entropy*, vol. 20, no. 6, Article ID 425, 2018.

[31] F. Kaffashi, R. Foglyano, C. G. Wilson, and K. A. Loparo, "The effect of time delay on Approximate & Sample Entropy calculations," *Physica D: Nonlinear Phenomena*, vol. 237, no. 23, pp. 3069–3074, 2008.

[32] T. Chanwimalueang and D. P. Mandic, "Cosine similarity entropy: self-correlation-based complexity analysis of dynamical systems," *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies*, vol. 19, no. 12, Paper No. 652, 23 pages, 2017.

[33] M. Staniek and K. Lehnertz, "Parameter selection for permutation entropy measurements," *International Journal of Bifurcation and Chaos*, vol. 17, no. 10, pp. 3729–3733, 2007.

[34] C. S. Daw, C. E. A. Finney, and M. B. Kennel, "Symbolic approach for measuring temporal "irreversibility"," *Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 62, no. 2, pp. 1912–1921, 2000.

[35] N. Iyengar, C.-K. Peng, R. Morin, A. L. Goldberger, and L. A. Lipsitz, "Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics," *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 271, no. 4, pp. R1078–R1084, 1996.

[36] D. Santos-Domínguez, S. Torres-Guijarro, A. Cardenal-López, and A. Pena-Gimenez, "ShipsEar: An underwater vessel noise database," *Applied Acoustics*, vol. 113, pp. 64–69, 2016.

[37] D. F. Specht, "Probabilistic neural networks and the polynomial adaline as complementary techniques for classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 1, no. 1, pp. 111–121, 1990.

Hindawi
Complexity
Volume 2019, Article ID 5937274, 17 pages
https://doi.org/10.1155/2019/5937274

Research Article

Two-Phase Incremental Kernel PCA for Learning Massive or Online Datasets

Feng Zhao[1,2] **Islem Rekik**[3,4] **Seong-Whan Lee**[5] **Jing Liu**[6] **Junying Zhang**[7] **and Dinggang Shen**[5,8]

[1]*School of Computer Science and Technology, Shandong Technology and Business University, Yantai, China*
[2]*Shandong Co-Innovation Center of Future Intelligent Computing, Yantai, China*
[3]*BASIRA Lab, Faculty of Computer and Informatics, Istanbul Technical University, Istanbul, Turkey*
[4]*School of Science and Engineering, Computing, University of Dundee, UK*
[5]*Department of Brain and Cognitive Engineering, Korea University, Seoul 02841, Republic of Korea*
[6]*School of Electronic Engineering, Xian University of Posts and Telecommunications, Xi'an, China*
[7]*School of Computer Science and Engineering, Xidian University, Xi'an, China*
[8]*Department of Radiology and BRIC, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA*

Correspondence should be addressed to Dinggang Shen; dgshen@med.unc.edu

Received 2 October 2018; Revised 17 December 2018; Accepted 8 January 2019; Published 11 February 2019

Guest Editor: Jose Garcia-Rodriguez

As a powerful nonlinear feature extractor, kernel principal component analysis (KPCA) has been widely adopted in many machine learning applications. However, KPCA is usually performed in a batch mode, leading to some potential problems when handling massive or online datasets. To overcome this drawback of KPCA, in this paper, we propose a two-phase incremental KPCA (TP-IKPCA) algorithm which can incorporate data into KPCA in an incremental fashion. In the first phase, an incremental algorithm is developed to explicitly express the data in the kernel space. In the second phase, we extend an incremental principal component analysis (IPCA) to estimate the kernel principal components. Extensive experimental results on both synthesized and real datasets showed that the proposed TP-IKPCA produces similar principal components as conventional batch-based KPCA but is computationally faster than KPCA and its several incremental variants. Therefore, our algorithm can be applied to massive or online datasets where the batch method is not available.

1. Introduction

As a conventional linear subspace analysis method, principal component analysis (PCA) can only produce linear subspace feature extractors [1], which are unsuitable for highly complex and nonlinear data distributions. In contrast, as a nonlinear extension of PCA, kernel principal component analysis (KPCA) [2] can capture the higher-order statistical information contained in data, thus producing nonlinear subspaces for better feature extraction performance. This has propelled the use of KPCA in a wide range of applications such as pattern recognition, statistical analysis, image processing, and so on [3–8]. Basically, KPCA firstly projects all samples from the input space into a kernel space using nonlinear mapping and then extracts the principal components (PCs) in the kernel space. In practice, such nonlinear mapping is performed implicitly via the "kernel trick", where an appropriately chosen kernel function is used to evaluate the dot products of mapped samples without having to explicitly carry out the mapping. As a result, the extracted kernel principal component (KPC) of the mapped data is nonlinear with respect to the original input space.

Standard KPCA has some drawbacks which limit its practical applications when handling big or online datasets. *Firstly*, in the training stage, KPCA needs to store and compute the eigenvectors of a $N \times N$ kernel matrix, where N is the total number of samples. This computation results in a space complexity of $O(N^2)$ and a time complexity of $O(N^3)$, thus rendering the evaluation of KPCA on large-scale datasets very time-consuming. *Secondly*, in the testing

stage, the resulting kernel principal components have to be defined implicitly by the linear expression of the training data, and thus all the training data must be saved after training. For a massive dataset, this translates into high costs for storage resources and increases the computational burden during the utilization of kernel principal components (KPCs). Furthermore, KPCA is impractical for many real-world applications where online samples are progressively collected since it is used in a batch manner. This implies that each time new data arrive, KPCA has to be conducted from scratch.

To overcome these limitations, many promising methods have been proposed in the past few years. These methods can be grouped into two classes. The first class is the batch-based modeling method, which requires that all training data is available for estimating KPCs. Rosipal and Girolami proposed an EM algorithm for reducing the computational cost of KPCA [9]. However, the convergence behavior of the EM algorithm to KPCA cannot be guaranteed in theory. In [6], the kernel Hebbian algorithm (KHA) was proposed as an iterative variant of KPCA algorithm. By kernelizing the generalized Hebbian algorithm (GHA), KHA computes KPCA without storing the kernel matrix, such that large-scale datasets with high dimensionality can be processed. Nonetheless, KHA has a scalar gain parameter which is either held constant or decreased according to a predetermined annealing schedule, leading to slow convergence during the training stage. To improve the convergence of KHA, gain adaptation methods were developed by providing a separate gain for each eigenvector estimate [10]. An improved version of KPCA was proposed based on the eigenvalue decomposition of a symmetric matrix [11], where datasets are divided into multiple subsets, each of which is processed separately. One of the major drawbacks of this approach is that it requires storing the kernel matrix, which means that the space complexity could be extremely large for large-scale dataset. Another variant of conventional KPCA is greedy KPCA [12, 13], which was employed to approximate the KPCs by a prior filtering of the training data. However, prior filtering of the training data could be computationally expensive. Overall, compared with standard KPCA, these batch-based modeling methods can potentially reduce the time or space complexity to some degree. Unfortunately, such methods cannot handle online data.

The second class is incremental methods, which can compute KPCs incrementally to handle online data processing. Chin and Suter proposed an incremental version of KPCA [14, 15], which is called IKPCA-RS for the notational simplicity. In IKPCA-RS, singular value decomposition is used to update an eigenfeature space incrementally for incoming data. However, IKPCA-RS may lead to high time complexity especially when dealing with high-dimensional data. In [16, 17], an incremental KPCA was presented based on the empirical kernel map. It is more efficient in memory requirement than the standard KPCA. However, it is only an approximate method and only suitable for polynomial kernel function. Inspired by the incremental PCA algorithm proposed by Hall et al. [18], Kimura et al. presented an incremental KPCA algorithm [19] in which an incremental updating algorithm for eigenaxes is derived based on a set of linearly independent data. Subsequently, some modified versions are proposed by Ozawa and Takeuchi et al. [20, 21]. Furthermore, in order to incrementally deal with data streams which are given in a chunk of multiple samples at one time, other extensions of KPCA were also successively presented [22–24]. Hallgren and Northrop [25] proposed an incremental KPCA (INKPCA) by applying rank one updates to the eigendecomposition of kernel matrix. However, INKPCA needs to store the whole data when evaluated on a new sample. Notably, incremental methods have the capacity of integrating new data, initially unavailable, in some way that maintains nonincreasing memory. However, to the best of our knowledge, most of these methods operate in the kernel space where all the samples are *implicitly* represented. This has two key limitations. *First*, a number of incremental methods may suffer from high computational cost. *Second*, the others can only capture the approximate KPCs rather than the accurate ones, which may affect the accuracy of its subsequent process.

Before continuing, a note on mathematical notations is given as follows. We use lower case and upper case letters (e.g., i, j, l, N) to denote scalars, lower case letters with the subscript (e.g., k_{ij}, α_i) to denote an element from a matrix or a vector, lower case bold letters (e.g., $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}$) to denote vectors, and upper case bold letters (e.g., $\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{M}$) to denote matrices. We use $\boldsymbol{x}^T$ ($\boldsymbol{C}^{\mathrm{T}}$) to denote the transpose of a vector (matrix) and $\|\cdot\|$ to denote the L2-norm of a vector. Furthermore, we adopt $\{\boldsymbol{x_i}\}_{\mathrm{i=1}}^{\mathrm{N}}$ to denote a set, $[\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}]$ to denote a matrix with N column vectors and $[k_{ij}]_{1\le i\le M, 1\le j\le N}$ to denote a $M \times N$ matrix composed of the corresponding element k_{ij}. In this paper, $\boldsymbol{x_i}$ always denotes a column vector and the inner product between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is expressed as $\boldsymbol{x_i^T x_j}$. The lower case bold letter $\boldsymbol{\varphi}$ denotes a nonlinear mapping. The mapped sample $\boldsymbol{\varphi}(\boldsymbol{x})$ of the input sample $\boldsymbol{x}$ is a column vector.

To address these limitations, we propose a two-phase incremental KPCA (TP-IKPCA), where the mapped data is represented in an *explicit* form and KPCs are updated in an *explicit* space. The computational cost of the whole process is very low and the accuracy of KPCs can be theoretically guaranteed. An overview of TP-IKPCA is briefly illustrated in Figure 1. In this figure, $\{\boldsymbol{x_i}\}_{\mathrm{i=1}}^{\mathrm{N}}$ denotes the sample set in a d-dimensional input space and N denotes the total number of available samples. Let $\boldsymbol{\varphi}$ denote the nonlinear mapping which maps the sample set $\{\boldsymbol{x_i}\}_{i=1}^{N}$ into an h-dimensional implicit kernel space, resulting in the mapped sample set $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^{N}$. Here, h may be very large or even infinite, depending on the specific mapping. The TP-IKPCA includes two phases. *In the first phase*, we develop an incremental algorithm to capture standard orthogonal basis $\{\boldsymbol{\beta}_j\}_{j=1}^{r}$ of the subspace spanned by $\{\boldsymbol{\varphi}(\boldsymbol{x_i})\}_{i=1}^{N}$ and then *explicitly* obtain the projection vectors $\{\boldsymbol{y}_i\}_{i=1}^{N}$ of $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^{N}$ by

$$\boldsymbol{y}_i = \left[\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r\right]^T \boldsymbol{\varphi}\left(\boldsymbol{x}_i\right), \tag{1}$$

where r denotes the number of a standard orthogonal basis $\{\boldsymbol{\beta}_j\}_{j=1}^{r}$. *In the second phase*, the existing incremental method

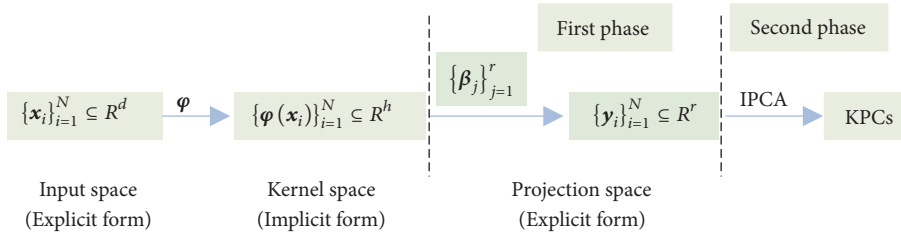


Figure 1: Overview of TP-IKPCA.

of PCA is employed to capture KPCs based on the explicit data $\{y_i\}_{i=1}^N$ in the projection space. In the following sections, we will detail how to incrementally express the implicit mapped data $\{\varphi(x_i)\}_{i=1}^N$ using an explicit form. We will also theoretically verify that performing PCA based on the implicitly mapped samples $\{\varphi(x_i)\}_{i=1}^N$ is equivalent to that of based on the explicit projection vectors $\{y_i\}_{i=1}^N$.

Here, we should clarify the relationship among some important quantities, including d, h, r (see Figure 1). In the case of KPCA or TP-IKPCA, the sample set $\{x_i\}_{i=1}^N$ in a d-dimensional input space is firstly mapped into an h-dimensional kernel space by a nonlinear mapping φ and then a linear PCA is performed based on the mapped set $\{\varphi(x_i)\}_{i=1}^N$. Usually, h may be very large or even infinite, depending on the specific mapping φ. So, we usually have $d \leq h$, which implies that the dimension of each mapped sample $\varphi(x_i)$ is larger than its original dimension d. In the case of TP-IKPCA, $\{\beta_j\}_{j=1}^r$ denote an orthonormal basis of the subspace spanned by $\{\varphi(x_i)\}_{i=1}^N$, and $\{y_i\}_{i=1}^N$ is the corresponding projection vectors of the mapped set $\{\varphi(x_i)\}_{i=1}^N$ on the basis $\{\beta_j\}_{j=1}^r$, which means that r is the dimension of the subspace spanned by $\{\varphi(x_i)\}_{i=1}^N$ and equals the dimension of each projected vector y_i $(i = 1, 2, \ldots, N)$. Generally, since the mapped data $\{\varphi(x_i)\}_{i=1}^N$ have strong linear correlation in the h-dimensional kernel space, we have $r \leq h$, which means that a few components generally suffice to capture the nonlinear distribution of the data. Furthermore, if the dimension r of the subspace spanned by $\{\varphi(x_i)\}_{i=1}^N$ is high, r may be larger than the dimension d of the input space. However, if the mapped data $\{\varphi(x_i)\}_{i=1}^N$ have strong linear correlation, which means r may be low and the dimension d of the input space is very high, we may get the contrast conclusion.

The main contributions of our work are fourfold: (1) Presenting an algorithm to express the mapped data in an explicit form. This will help for better understanding the distribution of the mapped data in the implicit kernel space. (2) Endowing KPCA with the capacity of handling dynamic dataset. (3) Compared to the standard KPCA, the computational complexity of TP-IKPCA is reduced from $O(N^3)$ to $O(r^3)$ and the storage complexity from $O(N^2)$ to $O(r^2)$, where N denotes the number of training samples and r is the number of bases of the subspace spanned by nonlinear mapped samples. Usually the assumption that $r \ll N$ is valid, which makes TP-IKPCA very convenient for processing large-scale datasets [26]. (4) In the testing stage, the feature extraction from one sample is faster than that of the batch KPCA, since TP-IKPCA only needs to calculate the kernel functions between the new sample and r selected training samples which forms the orthonormal basis.

The rest of the paper is organized as follows. Section 2 briefly introduces KPCA. In Section 3, we provide a theoretical analysis of the proposed TP-IKPCA method and elucidate the concrete steps for incrementally capturing KPCs based on the projection vectors in an explicit space. The effectiveness of TP-IKPCA is demonstrated in Section 4. Finally, the conclusions of our study are given in Section 5.

2. Kernel Principal Component Analysis (KPCA)

In this section, we briefly outline the standard procedure of KPCA. As mentioned above, in KPCA, the input sample set $\{x_i\}_{i=1}^N$ is mapped into a kernel space by a nonlinear mapping φ and then a linear PCA is performed based on $\{\varphi(x_i)\}_{i=1}^N$ in the kernel space.

To obtain the eigenvectors in the kernel space, the covariance matrix is defined as

$$C = \frac{1}{N}\sum_{i=1}^{N}\left(\varphi\left(x_i\right) - c\right)\left(\varphi\left(x_i\right) - c\right)^T, \tag{2}$$

where $c = (1/N)\sum_{i=1}^N \varphi(x_i)$. However, the eigendecomposition of C is hindered by the fact that the mapping function φ is implicit. To avoid the explicit calculation in the kernel space, a $N \times N$ kernel matrix K is defined, whose elements k_{ij} are determined by the virtue of the following kernel trick:

$$k_{ij} = \varphi\left(x_i\right)^T \varphi\left(x_i\right) = k\left(x_i, x_j\right) \quad i, j = 1, 2, \ldots, N. \tag{3}$$

where $k(\cdot, \cdot)$ is a kernel function that allows us to compute inner products in the kernel space [2].

Combining (2) and (3), Schölkopf et al. [2] derived the equivalent eigenvalue problem as follows:

$$K\alpha = N\lambda\alpha, \tag{4}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T$ denotes the column vector such that the orthogonal eigenvector v of the covariance matrix C satisfies

$$v = \sum_{i=1}^{N}\alpha_i\varphi\left(x_i\right). \tag{5}$$

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0$ ($k \leq N$) denote the first k nonzero eigenvalues of $\boldsymbol{K}$ and $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^k$ the corresponding complete set of eigenvectors (see (4)). We can obtain the corresponding eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$ of $\boldsymbol{C}$ using (5).

Considering $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$ need to be normalized, $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^k$ need to satisfy

$$\lambda_l \left(\boldsymbol{\alpha}^l\right)^T \left(\boldsymbol{\alpha}^l\right) = 1 \quad l = 1, 2, \ldots, k. \tag{6}$$

For a test sample $\boldsymbol{x}$, the projection of $\boldsymbol{\varphi}(\boldsymbol{x})$ on the l-th nonlinear principal component can be computed by

$$\begin{aligned} d_l(\boldsymbol{x}) &= \left(\boldsymbol{v}_l\right)^T \boldsymbol{\varphi}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i^l \left(\boldsymbol{\varphi}(\boldsymbol{x}_i)\right)^T \boldsymbol{\varphi}(\boldsymbol{x}) \\ &= \sum_{i=1}^{N} \alpha_i^l k(\boldsymbol{x}_i, \boldsymbol{x}). \end{aligned} \tag{7}$$

where α_i^l is the ith element of $\boldsymbol{\alpha}^l$; in other words, $\boldsymbol{\alpha}^l = [\alpha_1^l, \alpha_2^l, \ldots, \alpha_N^l]^{\boldsymbol{T}}$.

For the sake of simplicity, we assume that the mapped data $\boldsymbol{\varphi}(\boldsymbol{x}_i)$ ($i = 1, 2, \ldots, N$) is zero-centered (see (2)). The detailed description of the centering processing is given in [2].

Of note, for KPCA, the kernel matrix $\boldsymbol{K}$ needs to be predefined before performing eigendecompositions. Since the size of $\boldsymbol{K}$ scales with N^2, a large memory space is required for a massive dataset. Additionally, the eigendecomposition of $\boldsymbol{K}$ involves a time complexity of $O(N^3)$. This can severely handicap the computation of KPCA on large datasets. In online processing applications, the arrival of a new sample requires adding a new row and a new column in $\boldsymbol{K}$, and eigendecomposition has to be constantly reevaluated for an ever-growing kernel matrix to update the kernel subspaces. Hence, the batch KPCA is not convenient for such applications.

3. Explicit Representation of the Mapped Data

At present, there have been many incremental algorithms for PCA [27–34]. However, it is difficult to directly extend them to KPCA because all mapped samples $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ are expressed implicitly in the kernel space. Obviously, once $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ can be expressed using an explicit form, it will be straightforward to extend incremental PCA to KPCA. In fact, the geometrical structure of $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ can be captured by using a standard orthogonal basis of the subspace spanned by all the samples $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ [26]. Hence, we aim to explicitly express $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ using an indirect way. This motivation comes from the following property shown in Theorem 1.

Let $\{\boldsymbol{\beta}_j\}_{j=1}^r$ denote an orthonormal basis of the subspace spanned by $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$, and $\{\boldsymbol{y}_i\}_{i=1}^N$ is the corresponding projection vectors of $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ under $\{\boldsymbol{\beta}_j\}_{j=1}^r$ (see (1)). Theorem 1 is established as follows.

Theorem 1. *If linear PCA is performed based on $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ and $\{\boldsymbol{y}_i\}_{i=1}^N$, respectively, then their covariance matrices have the same nonzero eigenvalues, and those corresponding eigenvectors satisfy the following relationship:*

$$\boldsymbol{v}_l = \left[\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r\right] \boldsymbol{u}_l, \tag{8}$$

where $\boldsymbol{v}_l$ is the l-th eigenvector of the covariance matrix of $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ and $\boldsymbol{u}_l$ is the l-th eigenvector of the covariance matrix of $\{\boldsymbol{y}_i\}_{i=1}^N$. The proof is given in Appendix A.

Based on Theorem 1, linear PCA based on $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ can be converted into a linear PCA based on $\{\boldsymbol{y}_i\}_{i=1}^N$. So, if we can write $\{\boldsymbol{y}_i\}_{i=1}^N$ in an explicit form, then it will be easy to further extend KPCA using existing linear incremental algorithms. To incrementally obtain the orthonormal basis $\{\boldsymbol{\beta}_j\}_{j=1}^r$ and the projection vectors $\{\boldsymbol{y}_i\}_{i=1}^N$, we firstly introduce two correlative *lemmas*.

Lemma 2. *Let $\{\boldsymbol{\varphi}(\boldsymbol{x}_{bj})\}_{j=1}^r$ denote a basis of the subspace spanned by $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$, then the orthonormal basis $\{\boldsymbol{\beta}_j\}_{j=1}^r$ can be determined using*

$$\left[\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r\right] = \left[\boldsymbol{\varphi}(\boldsymbol{x}_{b1}), \boldsymbol{\varphi}(\boldsymbol{x}_{b2}), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{br})\right] \boldsymbol{D}, \tag{9}$$

where $\mathbf{D} = [\boldsymbol{\gamma}_1/\sqrt{\varepsilon_1}, \boldsymbol{\gamma}_2/\sqrt{\varepsilon_2}, \ldots, \boldsymbol{\gamma}_r/\sqrt{\varepsilon_r}]$. $\boldsymbol{\gamma}_j$ ($j = 1, 2, \ldots, r$) is the eigenvector of the kernel matrix $\boldsymbol{K}_{rr} = [\boldsymbol{\varphi}(\boldsymbol{x}_{b1}), \boldsymbol{\varphi}(\boldsymbol{x}_{b2}), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{br})]^{\boldsymbol{T}} [\boldsymbol{\varphi}(\boldsymbol{x}_{b1}), \boldsymbol{\varphi}(\boldsymbol{x}_{b2}), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{br})]$, scilicet $\boldsymbol{K}_{rr} = [k(\boldsymbol{x}_{bs}, \boldsymbol{x}_{bt})]_{1 \leq s,t \leq r}$. ε_j ($j = 1, 2, \ldots, r$) is the corresponding eigenvalue of $\boldsymbol{\gamma}_j$. The proof is given in Appendix B.

Based on Lemma 2, for any mapped sample $\boldsymbol{\varphi}(\boldsymbol{x}) \in \{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$, we can explicitly define its projection vector $\boldsymbol{y}$ under the orthonormal basis $\{\boldsymbol{\beta}_j\}_{j=1}^r$ as

$$\begin{aligned} \boldsymbol{y} &= \left[\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r\right]^T \boldsymbol{\varphi}(\boldsymbol{x}) \\ &= \boldsymbol{D}^T \left[\boldsymbol{\varphi}(\boldsymbol{x}_{b1}), \boldsymbol{\varphi}(\boldsymbol{x}_{b2}), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{br})\right]^T \boldsymbol{\varphi}(\boldsymbol{x}) \\ &= \boldsymbol{D}^T \boldsymbol{k}_{bx}, \end{aligned} \tag{10}$$

where $\boldsymbol{k}_{bx} = [k(\boldsymbol{x}_{b1}, \boldsymbol{x}), k(\boldsymbol{x}_{b2}, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_{br}, \boldsymbol{x})]^T$. Obviously, using the kernel function $k(\cdot, \cdot)$ can complete the computation of $\boldsymbol{y}$.

However, the orthogonalization process using Lemma 2 is a batch-based method. Subsequently, when samples are added one by one, its computational cost is still very expensive. So, inspired by the Gram-Schmidt orthogonalization process [35], we designed an online algorithm for incrementally finding the orthonormal basis and the projection vectors.

Lemma 3. *Let $\{\boldsymbol{\varphi}(\boldsymbol{x}_{bj})\}_{j=1}^r$ denote a basis of the subspace spanned by $\{\varphi(\boldsymbol{x}_i)\}_{i=1}^N$ and $\{\boldsymbol{\beta}_j\}_{j=1}^r$ is the orthonormal basis obtained by (9). Suppose $\boldsymbol{x}_{N+1}$ denotes a new sample we have just included into our dataset. We derive the following properties.*

(1) If $\delta = k_{N+1} - \boldsymbol{k}_{b(N+1)}^T \boldsymbol{D}\boldsymbol{D}^T \boldsymbol{k}_{b(N+1)} = 0$, then $\{\boldsymbol{\beta}_j\}_{j=1}^r$ is the orthonormal basis of the subspace spanned by $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^{N+1}$

and the projection vector $\boldsymbol{y}_{N+1}$ of $\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})$ can be computed using (10). Here, $k_{N+1} = k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1})$ and $\boldsymbol{k}_{b(N+1)} = [k(\boldsymbol{x}_{b1}, \boldsymbol{x}_{N+1}), k(\boldsymbol{x}_{b2}, \boldsymbol{x}_{N+1}), \ldots, k(\boldsymbol{x}_{br}, \boldsymbol{x}_{N+1})]^T$.

(2) If $\delta = k_{N+1} - \boldsymbol{k}_{b(N+1)}^T \boldsymbol{D}\boldsymbol{D}^T \boldsymbol{k}_{b(N+1)} \neq 0$, then the orthonormal basis $\{\boldsymbol{\beta}_j\}_{j=1}^{r+1}$ of the subspace spanned by $\{\boldsymbol{\varphi}(x_i)\}_{i=1}^{N+1}$ can be obtained by

$$\begin{aligned}&[\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r, \boldsymbol{\beta}_{r+1}] \\ &\quad = [\boldsymbol{\varphi}(\boldsymbol{x}_{b1}), \boldsymbol{\varphi}(\boldsymbol{x}_{b2}), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{br}), \boldsymbol{\varphi}(\boldsymbol{x}_{b(r+1)})] \\ &\quad \cdot \begin{bmatrix} \boldsymbol{D} & \dfrac{-\boldsymbol{D}\boldsymbol{D}^T \boldsymbol{k}_{b(N+1)}}{\sqrt{|\delta|}} \\ 0 & \dfrac{1}{\sqrt{|\delta|}} \end{bmatrix},\end{aligned} \tag{11}$$

where $\boldsymbol{x}_{b(r+1)} = \boldsymbol{x}_{N+1}$. The projection vector $\boldsymbol{y}_{N+1}$ of $\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})$ can be computed by

$$\boldsymbol{y}_{N+1} = \left[\boldsymbol{k}_{b(N+1)}^T \boldsymbol{D}, \ \sqrt{\delta}\right]^T. \tag{12}$$

Obviously, based on Lemma 3, it is straightforward to incrementally estimate the projection vector. Notably, the dimensionality of $\boldsymbol{y}_i$ $(i = 1, 2, \ldots, N)$ is smaller than that of $\boldsymbol{y}_{N+1}$ in the case of $\delta \neq 0$. In fact, based on the Gram-Schmidt orthogonalization process, let $\boldsymbol{y}_i'$ $(i = 1, 2, \ldots, N)$ denote the projection vector of $\boldsymbol{\varphi}(\boldsymbol{x}_i)$ on the orthonormal basis $\{\boldsymbol{\beta}_j\}_{j=1}^{r+1}$, then $\boldsymbol{y}_i' = [\boldsymbol{y}_i^T, 0]^T$. The proof of Lemma 3 is provided in Appendix C.

Combining both Lemmas 2 and 3, we summarize the online algorithm, which incrementally finds the orthonormal basis and the projection vectors as follows.

Algorithm 4. An online algorithm for incrementally finding the orthonormal basis and the projection vectors.

Step 0 (initialization). For the time N=1, we found a sample $\boldsymbol{x}_1$. We suppose $k(\boldsymbol{x}_1, \boldsymbol{x}_1) \neq 0$. Let the set $\boldsymbol{S} = \{x_1\}$, $\boldsymbol{D} = 1/\sqrt{k(\boldsymbol{x}_1, \boldsymbol{x}_1)}$, and $\boldsymbol{y}_1 = \sqrt{k(\boldsymbol{x}_1, \boldsymbol{x}_1)}$.

Step 1. Calculate δ for a new sample $\boldsymbol{x}_{N+1}$ according to

$$\delta = k_{N+1} - \boldsymbol{k}_{b(N+1)}^T \boldsymbol{D}\boldsymbol{D}^T \boldsymbol{k}_{b(N+1)}, \tag{13}$$

where $k_{N+1}, \boldsymbol{D}$ and $\boldsymbol{k}_{b(N+1)}$ have the same definition as in Lemma 3.

Step 2. If $\delta = 0$, then $\boldsymbol{y}_{N+1} = \boldsymbol{D}^T \boldsymbol{k}_{b(N+1)}$ and return to Step 1.

Step 3. If $\delta \neq 0$, then $\boldsymbol{S} = \boldsymbol{S} \cup \{\boldsymbol{x}_{N+1}\}$ and update $\boldsymbol{y}_{N+1}$ using (12). Finally, update $\boldsymbol{D}$ using (14) and return to Step 1.

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D} & \dfrac{-\boldsymbol{D}\boldsymbol{D}^T \boldsymbol{k}_{b(N+1)}}{\sqrt{|\delta|}} \\ 0 & \dfrac{1}{\sqrt{|\delta|}} \end{bmatrix}. \tag{14}$$

Obviously, if we map all the samples of $\boldsymbol{S}$ into the kernel space and get the dataset $\{\boldsymbol{\varphi}(\boldsymbol{x}_{bj}) \mid \boldsymbol{x}_{bj} \in \boldsymbol{S}\}$, then the mapped samples $\{\boldsymbol{\varphi}(\boldsymbol{x}_{bj}) \mid \boldsymbol{x}_{bj} \in \boldsymbol{S}\}$ are linearly independent. Furthermore, we can get an orthonormal basis based on $\{\boldsymbol{\varphi}(\boldsymbol{x}_{bj}) \mid \boldsymbol{x}_{bj} \in \boldsymbol{S}\}$ and $\boldsymbol{D}$ (see (9) or (11)). In fact, taking into account the actual calculation error, we usually use a very small threshold value θ to decide performing Step 2 or Step 3 in Algorithm 4. In other words, if $\delta < \theta$, we perform step 2; otherwise, we perform Step 3.

4. Incremental Learning of KPCA

In this section, we will outline the incremental learning method of KPCA based on the incremental version of PCA (IPCA) proposed by Hall et al. [18]. The key difference between our method and Hall et al.'s method is that the dimensionality of the projection vector $\boldsymbol{y}_i$ $(i = 1, 2, \ldots, N)$ in our case is not constant; hence we further adapt IPCA to our aim and address this limitation.

4.1. Description of IKPCA. Given a sample set $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ and its corresponding projection vector set $\{\boldsymbol{y}_i\}_{i=1}^N$ (see the Section 3), we assume we have already built a set of eigenvectors $\{\boldsymbol{u}_l\}_{l=1}^p$ and its corresponding matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p]$ with the $\{\boldsymbol{y}_i\}_{i=1}^N$ set as an input. Note that we have $p \leq r$ where r denotes the dimension of $\boldsymbol{y}_i$ $(i = 1, 2, \ldots, N)$. Let $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ denote the corresponding matrix of eigenvalues and $\overline{\boldsymbol{y}}$ the mean vector. Incremental building requires updating these eigenvectors when a new input sample $\boldsymbol{y}_{N+1}$ is obtained, which is the projection vector of $\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})$. Obviously, the dimensionality of $\boldsymbol{y}_{N+1}$ may be larger than that of $\boldsymbol{y}_i$ $(i = 1, 2, \ldots, N)$ (see (12)). When their dimensionalities are identical, we denote $r_{N+1} = r$, otherwise, $r_{N+1} \neq r$. Firstly, we update the mean:

$$\overline{\boldsymbol{y}} = \begin{cases} \dfrac{1}{N+1}\left(N\overline{\boldsymbol{y}} + \boldsymbol{y}_{N+1}\right) & r_{N+1} = r \\ \dfrac{1}{N+1}\left(N\begin{pmatrix}\overline{\boldsymbol{y}} \\ \boldsymbol{0}\end{pmatrix} + \boldsymbol{y}_{N+1}\right) & r_{N+1} \neq r, \end{cases} \tag{15}$$

where $(\overline{\boldsymbol{y}}, 0)^T$ means adding one zero to the original vector $\overline{\boldsymbol{y}}$. Then we update the set of eigenvectors $\{\boldsymbol{u}_l\}_{l=1}^p$ by adding a new vector $\boldsymbol{y}_{N+1}$ and applying a rotational transformation. In order to do this, we first compute the orthogonal residual vector:

$$\boldsymbol{h}_{N+1} = \begin{cases} (\boldsymbol{U}\boldsymbol{\eta}_{N+1} + \overline{\boldsymbol{y}}) - \boldsymbol{y}_{N+1} & r_{N+1} = r \\ \left(\begin{bmatrix}\boldsymbol{U} \\ \boldsymbol{0}\end{bmatrix}\boldsymbol{\eta}_{N+1} + \overline{\boldsymbol{y}}\right) - \boldsymbol{y}_{N+1} & r_{N+1} \neq r, \end{cases} \tag{16}$$

where $\boldsymbol{\eta}_{N+1}$ is computed by

$$\boldsymbol{\eta}_{N+1} = \begin{cases} \boldsymbol{U}^T\left(\boldsymbol{y}_{\boldsymbol{N+1}} - \overline{\boldsymbol{y}}\right) & r_{N+1} = r \\ \begin{bmatrix}\boldsymbol{U} \\ 0\end{bmatrix}^T \left(\boldsymbol{y}_{N+1} - \overline{\boldsymbol{y}}\right) & r_{N+1} \neq r. \end{cases} \tag{17}$$

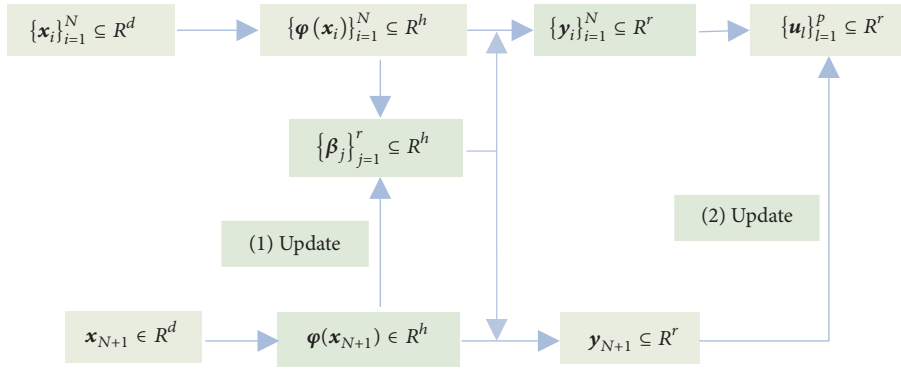


Figure 2: Flowchart of TP-IKPCA.

Subsequently, we normalize $\boldsymbol{h}_{N+1}$ to obtain $\widehat{\boldsymbol{h}}_{N+1} = \boldsymbol{h}_{N+1}/\|\boldsymbol{h}_{N+1}\|$ for $\|\boldsymbol{h}_{N+1}\| > 0$ and $\widehat{\boldsymbol{h}}_{N+1} = 0$ otherwise. The new matrix of eigenvectors $\boldsymbol{U}'$ is computed by

$$\boldsymbol{U}' = \begin{cases} \left[\boldsymbol{U}, \widehat{\boldsymbol{h}}_{N+1}\right] \boldsymbol{T} & r_{N+1} = r_N \\ \left[\begin{bmatrix} \boldsymbol{U} \\ 0 \end{bmatrix}, \widehat{\boldsymbol{h}}_{N+1}\right] \boldsymbol{T} & r_{N+1} \neq r_N, \end{cases} \tag{18}$$

where $\boldsymbol{T} \in R^{(p+1)\times(p+1)}$ is a rotation matrix with dimension p+1. $\boldsymbol{T}$ is the solution of the eigenproblem of the following form [18, 36]:

$$\boldsymbol{H}\boldsymbol{T} = \boldsymbol{T}\boldsymbol{\Lambda}'. \tag{19}$$

We compose $\boldsymbol{H} \in \mathrm{R}^{(p+1)\times(p+1)}$ as

$$\boldsymbol{H} = \frac{N}{N+1}\begin{bmatrix} \boldsymbol{\Lambda} & 0 \\ 0^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix} \boldsymbol{\eta}\boldsymbol{\eta}^T & \boldsymbol{\tau}\boldsymbol{\eta} \\ \boldsymbol{\tau}\boldsymbol{\eta}^T & \boldsymbol{\tau}^2 \end{bmatrix}. \tag{20}$$

where $\boldsymbol{\tau} = \widehat{\boldsymbol{h}}_{N+1}^T(\boldsymbol{y}_{N+1} - \overline{\boldsymbol{y}})$ and $\boldsymbol{\eta} = \boldsymbol{\eta}_{N+1}$.

Broadly, the procedure of our incremental method is similar to IPCA presented by Hall et al. [18]. Only under the condition $r_{N+1} \neq r_N$, we add one zero (zero vector) to the corresponding variant (matrix).

Once we determined the principal direction set $\{\boldsymbol{u}_l\}_{l=1}^p$, for a test sample $\boldsymbol{x}$, the projection of $\boldsymbol{\varphi}(\boldsymbol{x})$ onto the l-th nonlinear principal direction $\boldsymbol{v}_l$ can be obtained using the formulas in (8) and (10):

$$\begin{aligned} d_l(\boldsymbol{x}) &= (\boldsymbol{v}_l)^T \boldsymbol{\varphi}(\boldsymbol{x}) = (\boldsymbol{u}_l)^T (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r)^T \boldsymbol{\varphi}(\boldsymbol{x}) \\ &= (\boldsymbol{u}_l)^T \boldsymbol{D}^T \boldsymbol{k}_{bx} = (\boldsymbol{u}_l)^T \boldsymbol{y}. \end{aligned} \tag{21}$$

4.2. Framework of TP-IKPCA. Based on the analysis in Section 3 and in Section 4.1, we present the flowchart of TP-IKPCA in Figure 2. Here, $\{\boldsymbol{x}_i\}_{i=1}^N$ denotes the input sample set and $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ represents its corresponding mapped set by an implicit nonlinear mapping $\boldsymbol{\varphi}$. Firstly, an orthonormal basis $\{\boldsymbol{\beta}_j\}_{j=1}^r$ of the subspace spanned by $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ (see (9) or (11)) and the corresponding projection vectors set $\{\boldsymbol{y}_i\}_{i=1}^N$ (see (10)) are obtained. Subsequently, a set of eigenvectors $\{\boldsymbol{u}_l\}_{l=1}^p$ of $\{\boldsymbol{y}_i\}_{i=1}^N$ are built. Then, for a new coming sample $\boldsymbol{x}_{N+1}$, its mapped sample $\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})$ is used to update the orthonormal basis $\{\boldsymbol{\beta}_j\}_{j=1}^r$ and its projection vector $\boldsymbol{y}_{N+1}$ is computed using Algorithm 4. Finally, based on the IKPCA described in Section 4.1, the eigenvector set $\{\boldsymbol{u}_l\}_{l=1}^p$ is updated using $\boldsymbol{y}_{N+1}$. It can be seen from Figure 2 that TP-IKPCA includes two main steps. The first step is based on incremental learning algorithm that represents the mapped data using an explicit form (see Algorithm 4). The second step incrementally computes the principal components of $\{\boldsymbol{y}_i\}_{i=1}^{N+1}$ using IKPCA (see Section 4.1).

The dimensions d,h, and r of the input, kernel, and projection spaces, respectively, generally satisfy the following inequalities: $d \leq h$ and $r \leq h$ (see Section 1). Here, we need to focus on the meaning of p. In this paper, p denotes the number of the eigenvectors derived from the covariance matrix of the projection vector set $\{\boldsymbol{y}_i\}_{i=1}^N$. In other words, p is the number of the principal components of $\{\boldsymbol{y}_i\}_{i=1}^{N+1}$. We note that the number of the principal components, that is p, should be also smaller than the dimension r of the projection space. In summary, we have $p \leq r \leq h$.

4.3. Complexity Analysis of TP-IKPCA. Suppose that given the current sample set $\{\boldsymbol{x}_i\}_{i=1}^N$, the dimension of the subspace spanned by $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ is r. When a new sample $\boldsymbol{x}_{N+1}$ arrives, TP-IKPCA first conducts Algorithm 4 where the most time-consuming step is to compute $\boldsymbol{D}^T\boldsymbol{k}_{b(N+1)}$ appearing in (11)-(14). The time complexity for this step is $O(r^2)$ since $\boldsymbol{D}$ is an $r \times r$ orthogonal transformation matrix and $\boldsymbol{k}_{b(N+1)}$ is an N-dimensional vector. Then, TP-IKPCA incrementally computes the principal components using IKPCA (see Section 4.1). In this step, the computation of the eigendecomposition of matrix $\boldsymbol{H} \in R^{(p+1)\times(p+1)}$ in (20) is most time-consuming. We can define the upper bound for its time complexity as $O(p^3) \leq O(r^3)$ since we have $p \leq r$. Considering the above two steps, the overall time complexity of TP-IKPCA in worst case can be estimated as $O(r^3)$. In fact, $r \ll N$ is usually tenable [26, 37], which makes TP-IKPCA convenient for processing large-scale or online datasets. Meanwhile, TP-IKPCA requires storing an

$r \times r$ orthogonal transformation matrix $\boldsymbol{D}$ (see (9)), which only involves a space complexity of $O(r^2)$. Especially in the testing stage, obtaining the principal component of a new sample only needs to calculate the kernel functions between the new sample and the r old training samples that compose the orthonormal basis (see (21)), which results in improving the computing speed.

5. Experiments

We evaluated and compared the performance of TP-IKPCA on synthetic and real datasets with several typical KPCA-based approaches in terms of accuracy and time complexity. The comparison methods include (1) conventional batch mode KPCA, (2) incremental KPCA [14, 15] with reduced-set (IKPCA-RS), and (3) the recently developed incremental KPCA [25] based on rank one updates to the eigendecomposition of kernel matrix (INKPCA). The time complexity can be captured by two aspects: (1) time required for learning the training data; (2) r, i.e., the number of orthonormal basis elements of the subspace spanned by the mapping training data. Usually, a smaller r indicates a reduced time complexity of TP-IKPCA (see Section 4.3). At the same time, we also used two different measures to evaluate the accuracy of TP-IKPCA. The first one is the correlation coefficient between two corresponding principal components (PCs) of TP-IKPCA and KPCA. Since KPCA is performed using all training data in a batch learning model, the PCs of KPCA are the target that TP-IKPCA needs to capture. Ideally, the PCs of TP-IKPCA should be identical with those of KPCA. Therefore, the correlation coefficient between two corresponding PCs is evaluated to show how accurate is TP-IKPCA in comparison to batch KPCA. Specifically, let $\boldsymbol{v}_l$ denote the l-th PC of KPCA after learning all of the N samples and $\boldsymbol{v}_l^{\boldsymbol{TP}}(i)$ is the l-th PC of TP-IKPCA after learning i $(i \leq N)$ samples, we define the correlation coefficient (corr) by

$$corr\left(\boldsymbol{v}_l, \boldsymbol{v}_l^{\boldsymbol{TP}}(\boldsymbol{i})\right) = \frac{\left(\boldsymbol{v}_l\right)^T \boldsymbol{v}_l^{\boldsymbol{TP}}(\boldsymbol{i})}{\left\|\boldsymbol{v}_l\right\| \left\|\boldsymbol{v}_l^{\boldsymbol{TP}}(\boldsymbol{i})\right\|}, \tag{22}$$

The specific computation in (22) can be deduced from (5) and (8). The second measure is to compare the effectiveness of TP-IKPCA and KPCA. Here, for two-dimensional synthesized datasets, we adopt the contour lines of PCs as an evaluation measure. For real datasets, we adopt the practical denoising effect.

5.1. Synthesized Data. In this experiment, we use two-dimensional nonlinear synthetized data to evaluate the accuracy and memory space efficiency of KPCA, IKPCA-RS, INKPCA, and our proposed TP-IKPCA. The data is generated by:

$$y = \left(\frac{x^2}{2} + 1\right) + 0.2\xi, \quad \xi \sim N(0,1), \; x \sim U[0,1], \tag{23}$$

where $N(0,1)$ denotes the standard Gaussian distribution and $U[0,1]$ is the uniform distribution in $[0,1]$. In this experiment, the kernel function is the polynomial kernel form, namely, $k(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\boldsymbol{x_i^T x_j})^{\boldsymbol{m}}$. Here, the parameter m=2.

Table 1: Learning time (sec) for proposed and comparison methods.

	Training stage	Testing stage
KPCA	1.067 ± 0.03	0.203 ± 0.006
IKPCA-RS	0.201 ± 0.0032	0.007 ± 0.0003
INKPCA	0.145 ± 0.0041	0.201 ± 0.004
TP-IKPCA	**0.087 ± 0.0026**	**0.004 ± 0.0002**
Notice	The training number N=500 and the basis number r=9 in TP-IKPCA.	

The contour lines of the first three PCs obtained by each method for N=500 training samples are illustrated in Figure 3 (Top to bottom: KPCA, TP-IKPCA, IKPCA-RS, and INKPCA), where red dots represent samples and green lines represent the contour lines of the corresponding PCs. λ_i $(i = 1, 2, 3)$ denotes the eigenvalue of the corresponding PC. Figure 3 shows no visually discernible differences between our method and the 3 comparison methods. In other words, their contour lines are very similar. Furthermore, the differences in eigenvalues λ_i are also very small. Therefore, it can be concluded that all of the results derived from the three incremental algorithms: IKPCA-RS, INKPCA, and our proposed TP-IKPCA, follow the ground truth results closely.

Figure 4 shows the evolution curves of the correlation coefficients between the first three PCs obtained by TP-IKPCA, IKPCA-RS, INKPCA, and their corresponding PCs obtained by KPCA when increasing the number of training samples. Figure 4 shows for different incremental algorithms that the resulting correlation coefficient gradually converges to 1 as the number of training samples increases. It indicates that the PCs obtained by TP-IKPCA, IKPCA-RS, and INKPCA gradually approximate those obtained by batch mode KPCA with high accuracy.

Table 1 shows the average training time for learning from 500 training samples and testing time for extracting features from 100 testing samples (in seconds). We repeated this procedure 20 times and reported the averaged training and testing times as well as the corresponding standard deviations.

From Table 1, we can clearly see that when using the synthesized data, all of the three incremental KPCA algorithms have faster training speed than KPCA due to low-dimensional features as well as simple distribution of this data. Among these incremental variants, our proposed TP-IKPCA leads to fastest training speed. From the perspective of testing speed, our method can perform prediction in the least time. This can be explained by the fact that the learning time of TP-IKPCA depends on the size of orthonormal basis r (=9) while KPCA and INKPCA both depend on the total number of training samples (=500 in our case). IKPCA-RS also leads to fast testing speed since it uses a reduced set. However, it performs slower than our method in training speed since seeking a set of approximate preimages when new samples arrive using certain optimization techniques or fixed-point iteration is time-consuming.

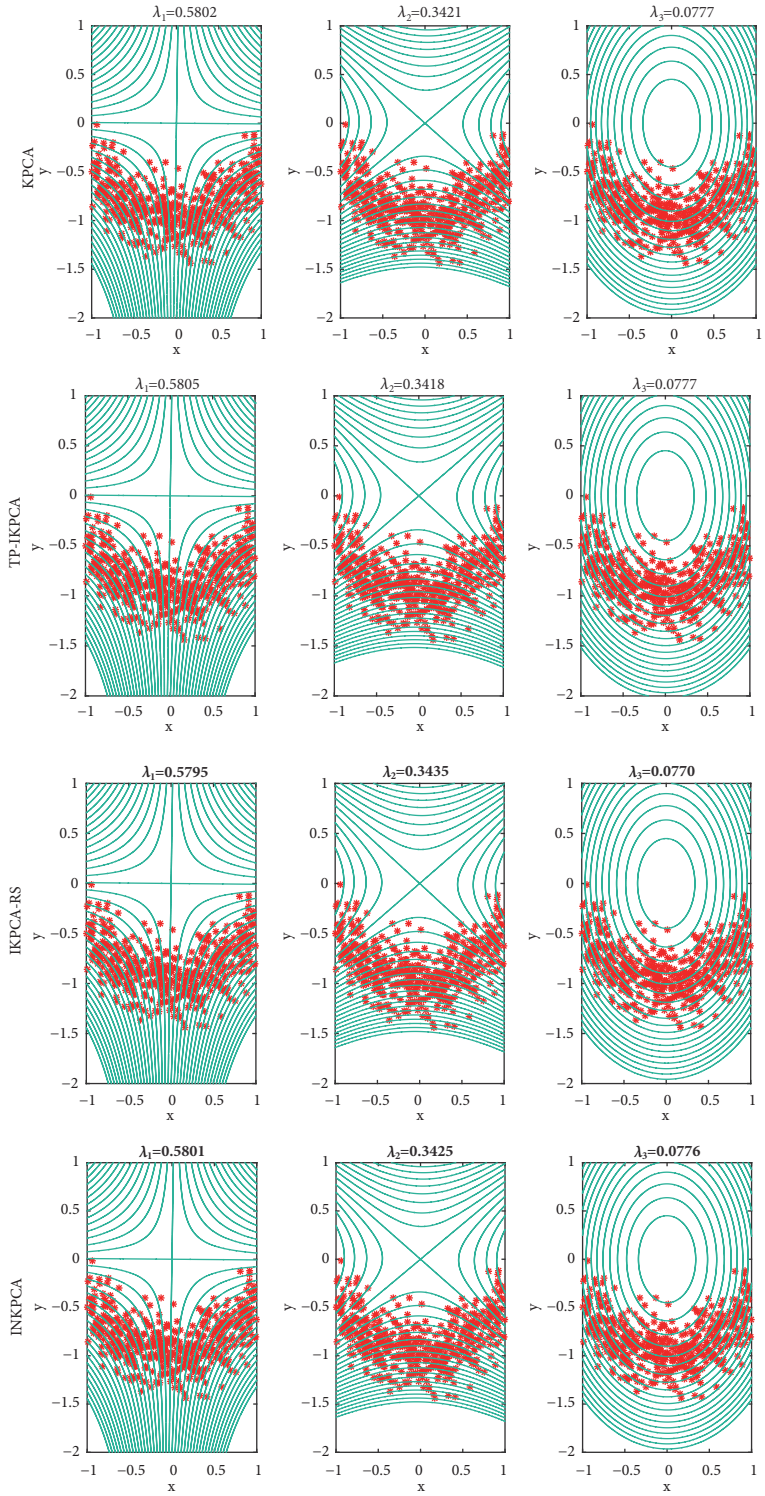
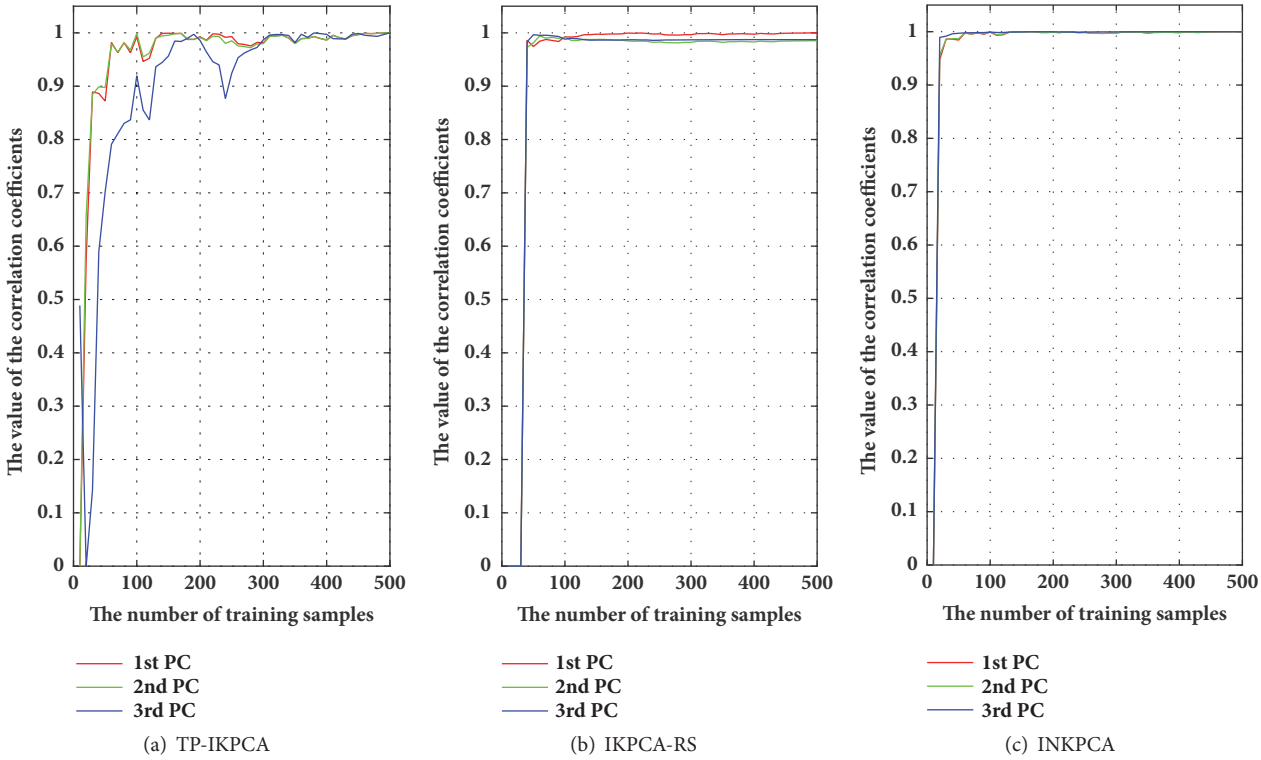


Figure 3: Synthesized data including 500 samples and the contours of the first three principal components drawn using a polynomial kernel. The first row is from KPCA, the second row is from TP-IKPCA, the third row is from IKPCA-RS, and the forth row is from INKPCA. Data points are represented by red dots "∗" and the green lines are the contour lines of constant value of the first three principal components.



(a) TP-IKPCA (b) IKPCA-RS (c) INKPCA

Figure 4: Evolution of the correlation coefficients between the first three PCs of three incremental algorithms and KPCA when increasing the number of training samples.

In what follows, we design two experiments to investigate the behavior of our algorithm when the number of training samples increases. Firstly, in Figure 5, we plot the variation curve of the number of basis, i.e., r, against the number of training samples (N). From this result, we notice that in the beginning, r gradually increases with N. However, after $N \geq 21$, r stops increasing. This shows that the mapped data have strong linear correlation in the kernel space. Hence, although the number of training samples continues to increase, the number of basis remains stable. More importantly, the computational complexity of TP-IKPCA becomes a constant $O(9^3)$ when $N \geq 21$, which is a significant improvement over standard KPCA (in the order of N^3) —particularly when the number of training samples becomes very large.

Then, we compute the acceleration ratio which represents the ratio of the time consumed by KPCA to extract features from 100 test samples to the time consumed by TP-IKPCA. The resulting variation of acceleration ratio for testing speed with respect to the number of training samples is shown in Figure 6. Obviously, a larger ratio indicates a faster test speed of TP-IKPCA compared with KPCA. Figure 6 also shows that the larger the number of training samples, the larger the ratio, implying that TP-IKPCA can significantly improve test speed compared with KPCA.

5.2. MNIST Data. In this section, we consider an image processing application where we process the MNIST database of handwritten digits (http://yann.lecun.com/exdb/mnist/). The database consists of handwritten digits from 0 to 9. Each

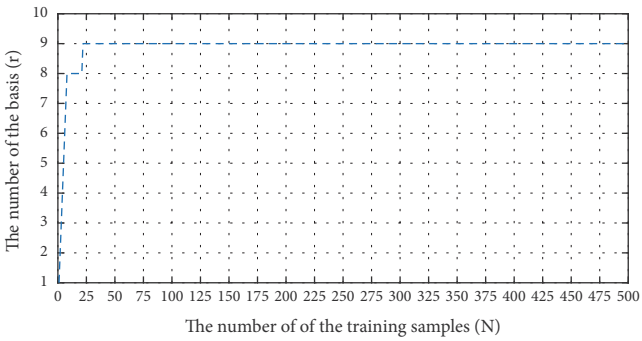


Figure 5: Variation of the number of basis (r) with respect to the number of training samples (N).

digit set includes training set and testing set. Each digit is represented by a 28-by-28 image. In order to evaluate the performance of different approaches, we carried out image denoising experiment to the even number. Firstly, for each digit, we randomly select 500 training samples and 100 testing samples and then add the Gaussian noise and the salt-and-pepper noise, respectively, to the testing images. The mean and variance for the Gaussian noise are 0 and 0.2, respectively, while the level of the salt-and-pepper noise is 0.3. Next, we estimate the first sixteen principal components using KPCA, IKPCA-RS, INKPCA, and our proposed TP-IKPCA, respectively. Finally, we perform denoising experiments on all corrupted testing samples and reconstruct

Figure 6: Changes of the ratio of the test speed with respect to the training sample numbers (N).

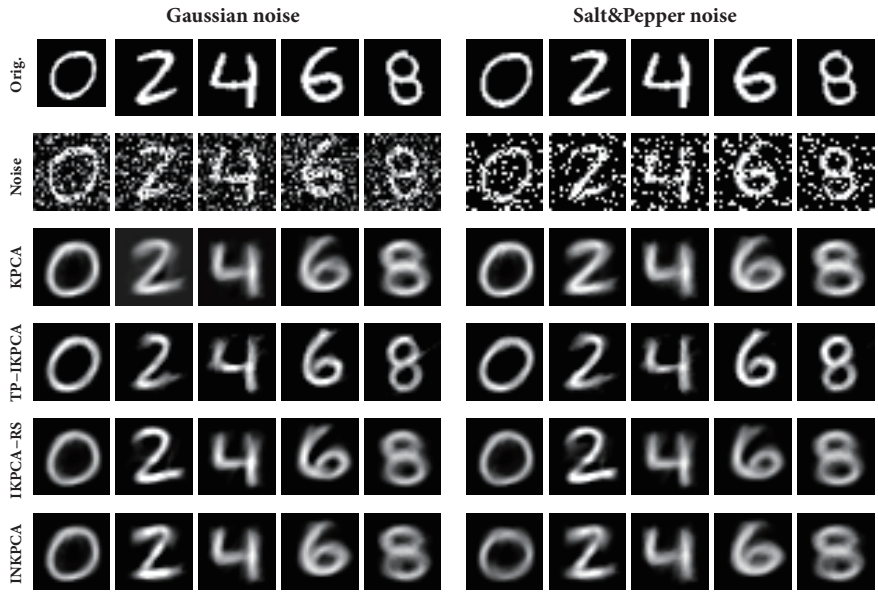


Figure 7: Restoration results by TP-IKPCA, IKPCA-RS, INKPCA, and KPCA.

them using the reconstructive scheme presented in [38]. In the experiment, we use the Gaussian function $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma^2)$ as the kernel function in each method and set the bandwidth $\sigma = 0.2$.

Figure 7 shows some restoration results from the corrupted images by conducting KPCA, TP-IKPCA, IKPCA-RS and INKPCA, respectively. It can be seen from these figures that all of these methods can eliminate noise and reconstruct the images well. Therefore, we can draw the conclusion that different incremental KPCA algorithms can closely approximate batch mode KPCA in reconstruction performance with high accuracy.

Figure 8 shows the evolution curves of the correlation coefficients between the first three PCs by TP-IKPCA, IKPCA-RS, INKPCA, and their corresponding ones by KPCA when the number of training samples increases, where the horizontal axis denotes the number of training samples and the vertical axis is the correlation coefficient computed by (22). Figure 8 shows that all incremental KPCA algorithms lead to good approximation accuracies for the first three PCs since the correlation coefficients gradually converge to 1 as the number of training samples increases. The results indicate that incremental KPCA algorithms can gradually capture PCs in real high-dimensional datasets with good approximation accuracy.

We also display in Table 2 the average training and testing times (in seconds) for training 500 samples and extracting features from 100 testing samples, respectively. We repeated this process 20 times and reported the average values and the corresponding standard deviations.

Firstly, we analyze the training results shown in Table 2. We can derive the following observations: (1) IKPCA-RS consumes much more time in training than other approaches, including batch model KPCA. These results differ from those obtained when using the synthesized data where IKPCA-RS ran faster than batch mode KPCA. Through inspecting the experiments, we found that IKPCA-RS iterates reduced-set expansions many times when computing the preimages for compression. As a result, when handling data with a large number of features (=784 in our case), the calculation of

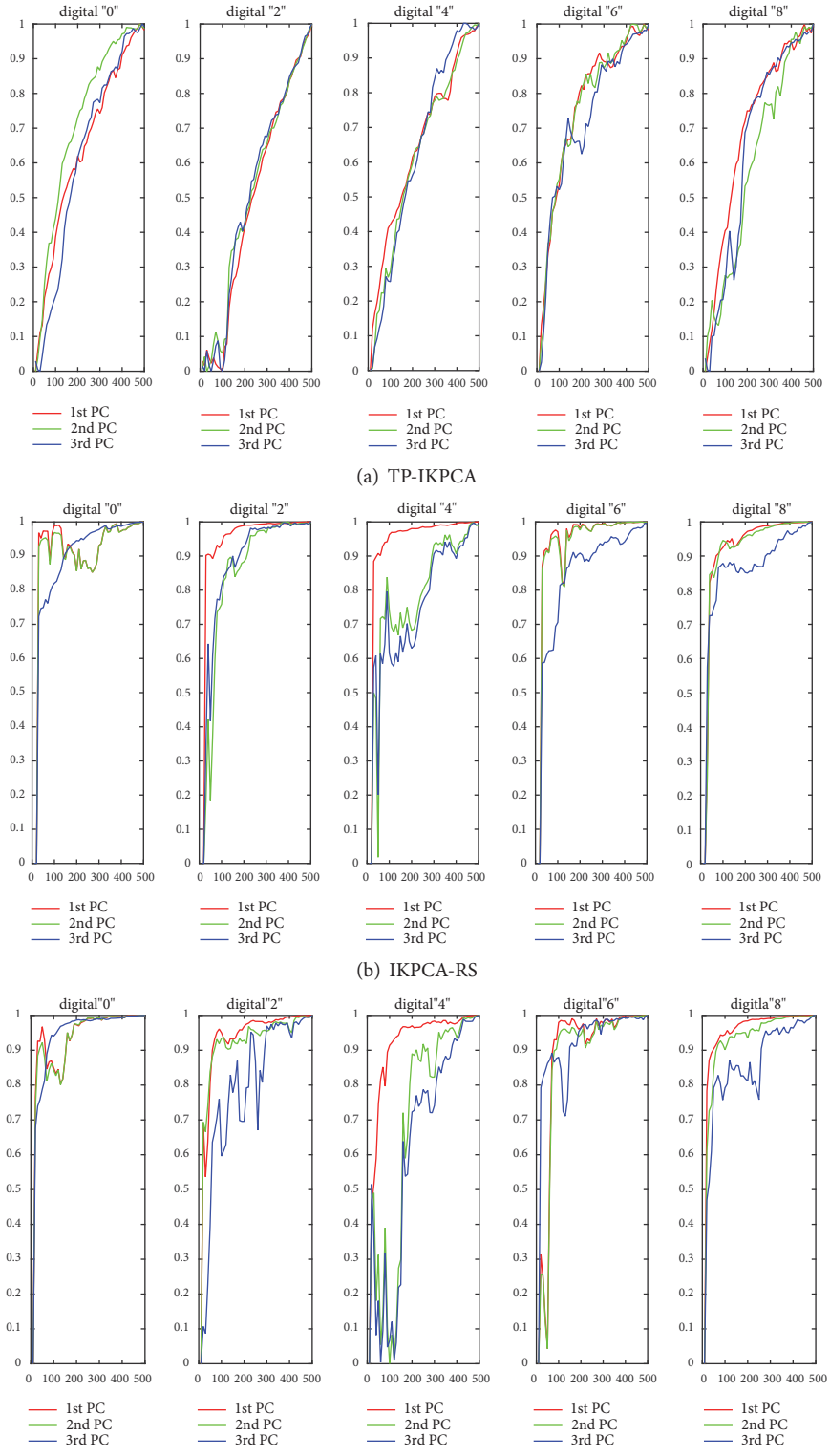


Figure 8: Evolution curves of the correlation coefficients between the first three PCs of (a) TP-IKPCA, (b) IKPCA-RS, (c) INKPCA, and KPCA as the number of training samples increases (x-axis).

Table 2: Experimental results of learning time on different digit (sec).

	Training stage					Testing stage				
	"0"	"2"	"4"	"6"	"8"	"0"	"2"	"4"	"6"	"8"
KPCA	1.75 ± 0.055	1.74 ± 0.018	1.74 ± 0.021	1.76 ± 0.036	1.74 ± 0.030	0.358 ± 0.022	0.351 ± 0.013	0.357 ± 0.012	0.352 ± 0.011	0.354 ± 0.011
IKPCA-RS	43.68 ± 0.049	41.32 ± 0.038	43.21 ± 0.068	44.75 ± 0.074	42.87 ± 0.059	0.221 ± 0.009	**0.243** ± 0.011	0.208 ± 0.014	0.169 ± 0.017	0.214 ± 0.008
INKPCA	23.67 ± 0.032	22.13 ± 0.038	23.91 ± 0.041	23.10 ± 0.047	22.13 ± 0.035	0.360 ± 0.023	0.358 ± 0.014	0.353 ± 0.012	0.355 ± 0.012	0.352 ± 0.011
TP-IKPCA	**1.89** ± 0.019	**3.06** ± 0.017	**1.67** ± 0.043	**1.40** ± 0.024	**2.27** ± 0.031	**0.181** ± 0.012	0.251 ± 0.008	**0.162** ± 0.016	**0.144** ± 0.013	**0.199** ± 0.010
r	242	351	219	190	282	242	351	219	190	282

preimages increases the computational load. This observation is in line with the conclusion drawn in [15]. (2) INKPCA needs to apply rank one update to iteratively calculate eigenvalues and eigenvectors associated with kernel matrix when new data arrive. Although INKPCA consumes more time than batch mode KPCA, it does not need to store the whole kernel matrix in memory and has the advantage of better handling massive data where KPCA rapidly becomes infeasible. (3) As for our proposed TP-IKPCA, it runs much faster than other incremental algorithms, including IKPCA-RS and INKPCA, excluding a few cases where our algorithm may be slower than batch mode KPCA. This can be explained by the computational complexity of TP-IKPCA of the order of r^3. In this experiment, the linear correlation between the mapped digital images is very weak, which results in a very large r and even approximates the number of training samples.

From the testing results shown in Table 2, we can conclude the following: (1) The testing speeds of KPCA and INKPCA are similar since INKPCA cannot select a few yet important samples from the whole data but still makes use of all available samples when calculating the projections of new data. Therefore, their speed is proportional to the total number of the training samples. (2) Regardless of specific digit, the testing speed of IKPCA-RS and TP-IKPCA are much faster than that of KPCA and INKPCA since both methods are able to reduce the number of samples used for kernel evaluation although they adopt different strategies. The testing time of TP-IKPCA is proportional to the size of basis r. In this experiment, the size of basis r is smaller than the number of training samples, thus leading to an improvement in the test speed compared with KPCA. Considering the training time, our proposed TP-IKPCA is obviously preferred over IKPCA-RS.

In what follows, we gradually increase the number of training samples and summarize the training and testing time required by TP-IKPCA and standard KPCA. We find from extensive experiments that the computational superiority of TP-IKPCA over KPCA increases with the number of training samples. Taking the experiments on digit "0" as an example, we repeated this evaluation 20 times and recorded the resulting training and testing time (in seconds) required by TP-IKPCA and KPCA under different training sample size N. Finally, the averaged time and the standard deviation are shown in Table 3 where the training ratio in Table 3 denotes the ratio of training time of KPCA to that of TP-IKPCA, given a total number of N samples. In a similar way, the testing ratio represents the ratio of testing time of KPCA to that of TP-IKPCA when extracting features from 100 test samples.

Based on Table 3, we can make the following observations: (1) As the number of training samples continues to increase, the number of basis r tends to increase as well. However, the increasing speed of r gradually decreases, which indicates that the larger the size of the training set, the stronger the correlation between the samples. (2) With the increasing of the training set size N, the training time of both KPCA and TP-IKPCA also increases gradually. However, the increasing speed of TP-IKPCA is much slower than that of KPCA. The reason lies in that the time complexity of TP-IKPCA has a close relationship with the number of basis r while that of KPCA depends on the total number of training samples N. As N *increases*, the increasing speed of r progressively decreases since most of the correlation structure among data has been revealed. We also derive a similar conclusion from the ratio's evolution in the training stage with respect to the changes of the number of training samples, where the ratio gradually increases with N in the training stage. (3) As N increases, the testing time of KPCA and TP-IKPCA both increase gradually. However, the increasing speed of TP-IKPCA is much slower than that of KPCA, which is also reflected by the ratio's evolution in the testing stage. The reason is that the test speed of TP-IKPCA is closely related to the number of basis r and the increasing speed of r gradually becomes slower with the increasing of N. Based on the above analysis, we conclude that TP-IKPCA does significantly improve the computational complexity of KPCA. Moreover, TP-IKPCA can deal with dynamic dataset due to its "incremental" nature.

6. Conclusion

In this paper, we proposed a novel incremental feature extraction method termed as TP-IKPCA which endowed KPCA with the capability of handling dynamic or large-scale datasets. The proposed TP-IKPCA differs from the existing incremental approaches in providing an explicit form of the mapped data and the updating process of KPCs is also performed in an explicit space. Specifically, TP-IKPCA is implemented in two phases. First, an incremental algorithm is given to explicitly project the mapped samples in the kernel space. Second, we employed the existing incremental method of PCA to capture KPCs based on the explicit data in the projection space. The computational complexity of TP-IKPCA has a close relationship with the size of basis r of the subspace spanned by the mapped training samples. Usually, r is much smaller than the number of training samples N, and thus TP-IKPCA can greatly improve the computational complexity of KPCA. In the case of large-scale or online dataset, the computational superiority of our approach is remarkable. Experimental results on synthetic and real datasets demonstrate that TP-IKPCA can significantly improve the time complexity of KPCA while preserving a high accuracy as standard KPCA. In comparison with two incremental KPCA algorithms, TP-IKPCA also illustrates superiority in terms of training and testing speed.

TP-IKPCA can be utilized in any application where KPCA needs to be conducted, especially when training data is of large scale, or can only be collected one by one, where the conventional batch-based KPCA cannot be applied. The idea of this study can be extended to other kernel-based methods, such as Kernel Fisher discriminant analysis (KFDA), Kernel independent component analysis (KICA), and so on.

Appendix

A. The Proof of Theorem 1

Let $\{\boldsymbol{\beta}_j\}_{j=1}^{r}$ denote an orthonormal basis of the subspace spanned by $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^{N}$. Equation (1) can be written as

Table 3: Training and testing time (sec) on digit "0" when increasing the number of training samples from 500 to 5000.

N		500	1000	1500	2000	3000	4000	5000
r		242	366	453	514	605	670	735
N/r		**2.07**	**2.73**	**3.31**	**3.89**	**4.96**	**5.97**	**6.80**
training stage	KPCA	1.75 ± 0.055	6.98 ± 0.066	18.72 ± 0.140	35.46 ± 0.185	79.97 ± 0.570	154.65 ± 0.687	255.01 ± 2.083
	TP-IKPCA	1.89 ± 0.019	5.34 ± 0.064	9.86 ± 0.085	14.92 ± 0.078	25.40 ± 0.015	37.46 ± 0.031	50.43 ± 0.411
	ratio	**0.93**	**1.31**	**1.90**	**2.38**	**3.15**	**4.13**	**5.06**
testing stage	KPCA	0.358 ± 0.022	0.650 ± 0.013	1.131 ± 0.018	1.600 ± 0.026	2.406 ± 0.070	3.657 ± 0.170	5.234 ± 0.293
	TP-IKPCA	0.181 ± 0.012	0.266 ± 0.014	0.323 ± 0.017	0.371 ± 0.016	0.429 ± 0.429	0.465 ± 0.011	0.511 ± 0.010
	ratio	**1.98**	**2.44**	**4.06**	**4.31**	**5.61**	**8.02**	**10.24**

$$\boldsymbol{Y} = \boldsymbol{B}^T \boldsymbol{\varphi}(\boldsymbol{X}), \tag{A.1}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]$ is the projection matrix, $\mathbf{B} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r]$ denotes the orthonormal basis matrix, and $\boldsymbol{\varphi}(\boldsymbol{X}) = [\boldsymbol{\varphi}(\boldsymbol{x}_1), \boldsymbol{\varphi}(\boldsymbol{x}_2), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_N)]$ is the mapped sample matrix. The covariance matrix $\boldsymbol{C}$ of $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^N$ (see (2)) can be expressed using the following formula:

$$\boldsymbol{C} = \boldsymbol{\varphi}(\boldsymbol{X}) \boldsymbol{H}_N \boldsymbol{H}_N^T \boldsymbol{\varphi}(\boldsymbol{X})^T, \tag{A.2}$$

where $\boldsymbol{H}_N$ is a $N \times N$ matrix and its element h_{ij} can be written as

$$h_{ij} = \begin{cases} \dfrac{(N-1)}{N} & i = j \\ \dfrac{-1}{N} & i \neq j \end{cases} \quad (1 \leq i, j \leq N). \tag{A.3}$$

Combining (A.1) and (A.2), we have

$$\boldsymbol{C} = \boldsymbol{B}\left(\boldsymbol{Y}\boldsymbol{H}_N\boldsymbol{H}_N^T\boldsymbol{Y}^T\right)\boldsymbol{B}^T. \tag{A.4}$$

Let $\overline{\boldsymbol{C}} = \boldsymbol{Y}\boldsymbol{H}_N\boldsymbol{H}_N^T\boldsymbol{Y}^T$, then $\overline{\boldsymbol{C}}$ is the covariance matrix of $\boldsymbol{Y}$. We have

$$\boldsymbol{C} = \boldsymbol{B}\overline{\boldsymbol{C}}\boldsymbol{B}^{\boldsymbol{T}}. \tag{A.5}$$

For the relationship of the eigenvalues and the corresponding eigenvector between $\boldsymbol{C}$ and $\overline{\boldsymbol{C}}$, we give a derivation as follows.

Lemma A.1. *Let $\lambda_l \neq 0$ be the l-th nonzero eigenvalue of $\boldsymbol{C}$ and $\boldsymbol{v}_l$ be the eigenvector, then $\boldsymbol{u}_l = \boldsymbol{B}^T\boldsymbol{v}_l$ is the eigenvector of $\overline{\boldsymbol{C}}$ and its eigenvalue is λ_l. Scilicet, $\overline{\boldsymbol{C}}\boldsymbol{u}_l = \lambda_l\boldsymbol{u}_l$.*

Proof. Based on the above definitions, we have $\boldsymbol{C}\boldsymbol{v}_l = \lambda_l\boldsymbol{v}_l$ and$\boldsymbol{v}_l = \boldsymbol{B}\boldsymbol{u}_l$. On the other hand, $\boldsymbol{C} = \boldsymbol{B}\overline{\boldsymbol{C}}\boldsymbol{B}^T$ is established. So, $\boldsymbol{B}\overline{\boldsymbol{C}}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{u}_l = \lambda_l\boldsymbol{B}\boldsymbol{u}_l$, thus, $\boldsymbol{B}\overline{\boldsymbol{C}}\boldsymbol{u}_l = \lambda_l\boldsymbol{B}\boldsymbol{u}_l$. Hence, $\boldsymbol{B}^T\boldsymbol{B}\overline{\boldsymbol{C}}\boldsymbol{u}_l = \lambda_l\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{u}_l$, Scilicet, $\overline{\boldsymbol{C}}\boldsymbol{u}_l = \lambda_l\boldsymbol{u}_l$. □

Lemma A.2. *Let $\lambda_l \neq 0$ be the l-th nonzero eigenvalue of $\overline{\boldsymbol{C}}$ and $\boldsymbol{u}_l$ be the corresponding eigenvector, then $\boldsymbol{v}_l = \boldsymbol{B}\boldsymbol{u}_l$ is the eigenvector of $\boldsymbol{C}$ and its eigenvalue is λ_l. Scilicet, $\boldsymbol{C}\boldsymbol{v}_l = \lambda_l\boldsymbol{v}_l$.*

Proof.

$$\boldsymbol{C}\boldsymbol{v}^l = \boldsymbol{B}\overline{\boldsymbol{C}}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{u}_l = \boldsymbol{B}\overline{\boldsymbol{C}}\boldsymbol{u}_l = \boldsymbol{B}\lambda_l\boldsymbol{u}_l = \lambda_l\boldsymbol{v}_l. \tag{A.6}$$

Based on Lemmas A.1 and A.2, we know the nonzero eigenvalue of $\boldsymbol{C}$ is the same with that of $\overline{\boldsymbol{C}}$ and the corresponding eigenvector satisfies $\boldsymbol{v}_l = \boldsymbol{B}\boldsymbol{u}_l$. Hence, the eigendecomposition of $\boldsymbol{C}$ can be converted into the corresponding process of $\overline{\boldsymbol{C}}$. And because $\boldsymbol{v}_l/\|\boldsymbol{v}_l\| = \boldsymbol{B}\boldsymbol{u}_l/\|\boldsymbol{B}\boldsymbol{u}_l\| = \boldsymbol{B}\boldsymbol{u}_l/\|\boldsymbol{u}_l\|$, so (8) still holds after the eigenvector is unitized. So, Theorem 1 is proven. □

B. The Proof of Lemma 2

Let $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_r]$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r)$ be the corresponding matrix which are, respectively, from the eigenvector and the eigenvalue of the kernel matrix $\boldsymbol{K}_{rr} = [k(\boldsymbol{x}_{bs}, \boldsymbol{x}_{bt})]_{1\leq s,t\leq r}$. We have $\boldsymbol{\Gamma}^T\boldsymbol{K}_{rr}\boldsymbol{\Gamma} = \boldsymbol{\Lambda}$. Let $\boldsymbol{B} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r]$ and $\boldsymbol{\Delta} = \mathrm{diag}(1/\sqrt{\varepsilon_1}, 1/\sqrt{\varepsilon_2}, \ldots, 1/\sqrt{\varepsilon_r})$; we have $\boldsymbol{D} = \boldsymbol{\Gamma}\boldsymbol{\Delta}$. Thus $\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{D}^T\boldsymbol{K}_{rr}\boldsymbol{D} = \boldsymbol{\Delta}^T\boldsymbol{\Gamma}^T\boldsymbol{K}_{rr}\boldsymbol{\Gamma}\boldsymbol{\Delta} = \boldsymbol{\Delta}^T\boldsymbol{\Lambda}\boldsymbol{\Delta} = \boldsymbol{I}$. So, Lemma 2 is proven.

C. The Proof of Lemma 3

If $\{\boldsymbol{\beta}_j\}_{j=1}^r$ is the orthonormal basis of the subspace spanned by $\{\boldsymbol{\varphi}(\boldsymbol{x}_i)\}_{i=1}^{N+1}$, then $\boldsymbol{\varphi}(\boldsymbol{x}_{N+1}) = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r][\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r]^T\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})$. Based (9), we have

$$\begin{aligned} &\boldsymbol{\varphi}(\boldsymbol{x}_{N+1}) = [\boldsymbol{\varphi}(\boldsymbol{x}_{b1}), \boldsymbol{\varphi}(\boldsymbol{x}_{b2}), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{br})] \\ &\quad \cdot \boldsymbol{D}\boldsymbol{D}^T\boldsymbol{k}_{b(N+1)} \Longleftrightarrow \\ &\left\|\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})\right. \\ &\quad \left. - [\boldsymbol{\varphi}(\boldsymbol{x}_{b1}), \boldsymbol{\varphi}(\boldsymbol{x}_{b2}), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{br})]\, \boldsymbol{D}\boldsymbol{D}^T\boldsymbol{k}_{b(N+1)}\right\| \\ &\quad = 0 \Longleftrightarrow \\ &k_{N+1} - \boldsymbol{k}_{b(N+1)}^T\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{k}_{b(N+1)} = 0. \end{aligned} \tag{C.1}$$

So, conclusion (1) in Lemma 3 is proven. Subsequently, if $\delta = k_{N+1} - \boldsymbol{k}_{b(N+1)}^T\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{k}_{b(N+1)} \neq 0$, this means $\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})$ cannot be linearly expressed by $\{\boldsymbol{\beta}_j\}_{j=1}^r$. Based the Gram-Schmidt orthogonalization process, $\boldsymbol{\beta}_{r+1}$ can be determined using the following formula:

$$\begin{aligned} &\boldsymbol{\beta}_{r+1} \\ &= \frac{\boldsymbol{\varphi}(\boldsymbol{x}_{N+1}) - [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r][\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r]^T\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})}{\left\|\boldsymbol{\varphi}(\boldsymbol{x}_{N+1}) - [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r][\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r]^T\boldsymbol{\varphi}(\boldsymbol{x}_{N+1})\right\|}. \end{aligned} \tag{C.2}$$

Combined with (9) and kernel function, (C.2) can be written using the following formula:

$$\begin{aligned} \boldsymbol{\beta}_{r+1} &= [\boldsymbol{\varphi}(\boldsymbol{x}_1), \boldsymbol{\varphi}(\boldsymbol{x}_2), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_N), \boldsymbol{\varphi}(\boldsymbol{x}_{N+1})] \\ &\quad \cdot \begin{bmatrix} \dfrac{-\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{k}_{b(N+1)}}{\sqrt{|\delta|}} \\ 1 \\ \dfrac{}{\sqrt{|\delta|}} \end{bmatrix}. \end{aligned} \tag{C.3}$$

Then, we can obtain (11) and (12). So Lemma 2 is proven.

Data Availability

In our manuscript, we used two datasets to support the findings of our study. One dataset is the synthetic toy data, which can be generated by the following way: $\boldsymbol{y} = (\boldsymbol{x}\ 2\ 2 + 1) + 0.2\boldsymbol{\xi}$, $\boldsymbol{\xi} \sim \boldsymbol{N}(0,1)$, $\boldsymbol{x} \sim \boldsymbol{U}[0,1]$ (1) where $\boldsymbol{N}(0,1)$ denotes the standard Gaussian distribution and $\boldsymbol{U}[0,1]$ is the uniform distribution in $[0,1]$. The second data set is the MNIST database of handwritten digits, which are available on this site: http://yann.lecun.com/exdb/mnist.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grants nos. 61773244, 61373079, and 61572344), National Institutes of Health in USA (AG041721, MH107815, EB006733, EB008374, and EB009634), and Provincial Natural Science Foundation of Shanxi in China (2018JM4018).

References

[1] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, NY, USA, 1986.

[2] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[3] B. Chen, J. Yang, B. Jeon, and X. Zhang, "Kernel quaternion principal component analysis and its application in RGB-D object recognition," *Neurocomputing*, vol. 266, pp. 293–303, 2017.

[4] X. Deng and L. Wang, "Modified kernel principal component analysis using double-weighted local outlier factor and its application to nonlinear process monitoring," *ISA Transactions®*, vol. 72, pp. 218–228, 2018.

[5] Y. Yang, W. Sheng, Y. Han, and X. Ma, "Multi-beam pattern synthesis algorithm based on kernel principal component analysis and semi-definite relaxation," *IET Communications*, vol. 12, no. 1, pp. 82–95, 2018.

[6] K. I. Kim, M. O. Franz, and B. Schölkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1351–1366, 2005.

[7] A. R. Teixeira, A. M. Tomé, K. Stadlthanner, and E. W. Lang, "KPCA denoising and the pre-image problem revisited," *Digital Signal Processing*, vol. 18, no. 4, pp. 568–580, 2008.

[8] W. Soh, H. Kim, and B.-J. Yum, "Application of kernel principal component analysis to multi-characteristic parameter design problems," *Annals of Operations Research*, vol. 263, no. 1-2, pp. 69–91, 2018.

[9] R. Rosipal and M. Girolami, "An expectation-maximization approach to nonlinear component analysis," *Neural Computation*, vol. 13, no. 3, pp. 505–510, 2001.

[10] G. Simon, N. S. Nicol, and S. V. N. Vishwanathan, "Fast iterative kernel principal component analysis," *Journal of Machine Learning Research (JMLR)*, vol. 8, no. 4, pp. 1893–1918, 2007.

[11] W. Zheng, C. Zou, and L. Zhao, "An improved algorithm for kernel principal component analysis," *Neural Processing Letters*, vol. 22, no. 1, pp. 49–56, 2005.

[12] F. Vojtěch and H. Václav, "Greedy algorithm for a training set reduction in the kernel methods," in *Proceedings of the 10th International Conference on Computer Analysis of Image and Patterns*, vol. 2756 of *Lecture Notes in Comput. Sci.*, pp. 426–433, Springer, Groningen, Netherlands, August 2003.

[13] F. Vojtech, *Optimization Algorithms for Kernel Methods [Ph.D. Dissertation]*, Center for Machine Perception, Czech Technical University, Prague, Czech Republic, 2005.

[14] T. Chin and D. Suter, "Incremental kernel PCA for efficient non-linear feature extraction," in *Proceedings of the 17th British Machine Vision Conference*, pp. 4–7, Edinburgh, Scotland, September 2006.

[15] T.-J. Chin and D. Suter, "Incremental kernel principal component analysis," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1662–1674, 2007.

[16] B. J. Kim and I. K. Kim, "Incremental nonlinear PCA for classification," in *Proceedings of the European Conference on Knowledge Discovery in Databases (PKDD)*, vol. 3202 of *Lecture Notes in Computer Science*, pp. 291–300, Springer, 2004.

[17] B.-J. Kim, "Active visual learning and recognition using incremental kernel PCA," in *Proceedings of the 18th Australian Joint Conference on Advances in Artificial Intelligence AI'05*, vol. 3809 of *Lecture Notes in Comput. Sci.*, pp. 585–592, Springer, 2005.

[18] P. M. Hall, D. Marshall, and R. R. Martin, "Incremental eigenanalysis for classification," in *Proceedings of the British Machine Vision Conference*, pp. 286–295, 1998.

[19] S. Kimura, S. Ozawa, and S. Abe, "Incremental Kernel PCA for online learning of feature space," in *Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation*, vol. 1, pp. 595–600, Vienna, Austria, November 2005.

[20] Y. Takeuchi, S. Ozawa, and S. Abe, "An efficient incremental kernel principal component analysis for online feature selection," in *Proceedings of the 2007 International Joint Conference on Neural Networks*, pp. 2346–2351, Orlando, FL, USA, August 2007.

[21] O. Seiichi, Y. Takeuchi, and A. Shigeo, "A fast incremental kernel principal component analysis for online feature extraction," in *Proceedings of the Pacific Rim International Conference on Trends in Artificial Intelligence*, vol. 6230 of *Lecture Notes in Computer Science*, pp. 487–497, Springer, 2010.

[22] T. Takaomi and O. Seiichi, "A fast incremental kernel principal component analysis for learning stream of data chunks," in *Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN 2011 - San Jose)*, pp. 2881–2888, San Jose, CA, USA, July 2011.

[23] A. A. Joseph and S. Ozawa, "A fast incremental kernel principal component analysis for data streams," in *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014.

[24] A. A. Joseph, T. Tokumoto, and S. Ozawa, "Online feature extraction based on accelerated kernel principal component analysis for data stream," *Evolving Systems*, vol. 7, no. 1, pp. 15–27, 2016.

[25] H. Fredrik and N. Paul, "Incremental kernel PCA and the Nyström method," 2018, https://arxiv.org/abs/1802.00043.

[26] G. Baudat and F. Anouar, "Kernel-based methods and function approximation," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'01)*, pp. 1244–1249, Washington, DC, USA, July 2001.

[27] H. Zhao, P. C. Yuen, and J. T. Kwok, "A novel incremental principal component analysis and its application for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 4, pp. 873–886, 2006.

[28] J. Weng, Y. Zhang, and W. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1034–1040, 2003.

[29] S. Nicole, "Feedforward neural networks for principal components extraction," *Computational Statistics & Data Analysis*, vol. 33, no. 4, pp. 425–437, 2000.

[30] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, no. 6, pp. 459–473, 1989.

[31] E. Oja, "A simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology*, vol. 15, no. 3, pp. 267–273, 1982.

[32] Y. Li, "On incremental and robust subspace learning," *Pattern Recognition*, vol. 37, no. 7, pp. 1509–1518, 2004.

[33] M. Artac, M. Jogan, and A. Leonardis, "Incremental PCA for on-line visual learning and recognition," in *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 781–784, Quebec City, Canada, 2002.

[34] O. Seiichi, P. Shaoning, and K. Nikola, "A modified incremental principal component analysis for on-line learning of feature space and classifier," in *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*, pp. 231–240, Auckland, New Zealand, 2004.

[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, NY, USA, 2013.

[36] S. Ling, X. Cheng, and T. Jiang, "An algorithm for coneigenvalues and coneigenvectors of quaternion matrices," *Advances in Applied Clifford Algebras (AACA)*, vol. 25, no. 2, pp. 377–384, 2015.

[37] C. Han, Y. Wang, and G. He, "On the convergence of asynchronous parallel algorithm for large-scale linearly constrained minimization problem," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 434–441, 2009.

[38] S. B. Mike, B. Scholkopf, and A. J. Smola, "Kernel PCA and denoising in feature space," in *Advances in Neural Information Processing System*, pp. 524–536, MTI press, Cambridge, UK, 1999.

Hindawi
Complexity
Volume 2019, Article ID 6876173, 23 pages
https://doi.org/10.1155/2019/6876173

Research Article

A Novel Semi-Supervised Learning Method Based on Fast Search and Density Peaks

Fei Gao,[1] **Teng Huang**,[1] **Jinping Sun**,[1] **Amir Hussain**,[2] **Erfu Yang**,[3] **and Huiyu Zhou**[4]

[1] *School of Electronic and Information Engineering, Beihang University, Beijing 101191, China*
[2] *Cognitive Big Data and Cyber-Informatics (CogBID) Laboratory, School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, Scotland, UK*
[3] *Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow G1 1XJ, UK*
[4] *Department of Informatics, University of Leicester, Leicester LE1 7RH, UK*

Correspondence should be addressed to Teng Huang; huangteng1220@buaa.edu.cn and Jinping Sun; sunjinping@buaa.edu.cn

Received 5 October 2018; Revised 7 December 2018; Accepted 23 December 2018; Published 3 February 2019

Guest Editor: David Tomás

Radar image recognition is a hotspot in the field of remote sensing. Under the condition of sufficiently labeled samples, recognition algorithms can achieve good classification results. However, labeled samples are scarce and costly to obtain. Our major interest in this paper is how to use these unlabeled samples to improve the performance of a recognition algorithm in the case of limited labeled samples. This is a semi-supervised learning problem. However, unlike the existing semi-supervised learning methods, we do not use unlabeled samples directly and, instead, look for safe and reliable unlabeled samples before using them. In this paper, two new semi-supervised learning methods are proposed: a semi-supervised learning method based on fast search and density peaks (S^2DP) and an iterative S^2DP method (IS^{2}DP). When the labeled samples satisfy a certain requirement, S^2DP uses fast search and a density peak clustering method to detect reliable unlabeled samples based on the weighted kernel Fisher discriminant analysis (WKFDA). Then, a labeling method based on clustering information (LCI) is designed to label the unlabeled samples. When the labeled samples are insufficient, IS^{2}DP is used to iteratively search for reliable unlabeled samples for semi-supervision. Then, these samples are added to the labeled samples to improve the recognition performance of S^2DP. In the experiments, real radar images are used to verify the performance of our proposed algorithm in dealing with the scarcity of the labeled samples. In addition, our algorithm is compared against several semi-supervised deep learning methods with similar structures. Experimental results demonstrate that the proposed algorithm has better stability than these methods.

1. Introduction

Radar image recognition is a popular research area in the field of remote sensing [1–3]. With the development of imaging technologies and the expansion of radar image data, the requirement of real-time and accuracy of data processing becomes higher and higher. Under the condition where the number of the labeled samples is sufficient, a recognition algorithm can generally achieve satisfactory classification results with a strong sample representation ability [2, 4]. However, the labeled radar images are scarce compared to the case of optical images, and the cost of labeling is also very expensive. They can usually be interpreted by an experienced expert [5, 6]. Therefore, it is unrealistic to obtain a large number of labeled samples by manual annotation.

This paper focuses on how to use these unlabeled samples to improve the performance of a recognition algorithm in the case of limited labeled samples. This is a semi-supervised learning problem. Currently, semi-supervised deep learning achieves promising recognition performance, such as Ladder Network [7] and Temporal Ensembling [8]. However, unlike those existing semi-supervised learning methods, we do not use unlabeled samples directly and, instead, look for safe and reliable unlabeled samples and then use these unlabeled samples to enhance the performance of the recognition algorithm. This is because the unlabeled radar

images need to go through the detection stage in the process of acquisition [9, 10]. These samples may deteriorate the semi-supervised algorithms' learning, especially when the number of the labeled samples and that of the unlabeled samples are somehow unbalanced. This will influence the performance of the semi-supervised algorithm. The negative effects of these unreliable and unlabeled samples on semi-supervised algorithms are analyzed comprehensively in [11, 12]. Therefore, it is very important for a semi-supervised algorithm to identify reliable unlabeled samples before we learn unlabeled samples' features.

Effective use of unlabeled samples is a new and interesting topic for semi-supervised methods. These emerging semi-supervised methods are mainly divided into two categories: semi-supervision based on integrated resources and safe semi-supervision based on weights. Semi-supervised methods, based on integration resources, usually combine multiple semi-supervised models, comprehensively analyse the predictions of unlabeled samples, and choose reliable unlabeled samples to improve the recognition performance of the system. For example, Li et al. [13] proposed the S^3VM-us method, which consists of a semi-supervised support vector machine (S^3VM) [14] and a standard support vector machine (SVM) [15]. The confidence of unlabeled samples is determined by both classifiers. If the evaluation results are consistent, the unlabeled samples are identified. Li et al. [16] also proposed a safe S^3VM method (S^4VM). We understand that the S^3VM is based on the low-density hypothesis in order to detect a significant interval along the low-density boundary from the feature space to identify unlabeled samples. Unlike S^3VM, S^4VM was based on the fact that there may be more than one low-density boundary in the feature space. This approach considers all the possible situations, equivalently, integrating multiple S^3VMs to pinpoint reliable unlabeled samples. Wang et al. [17] proposed a safety-aware semi-supervised method. It consists of a semi-supervised model and a supervised model, which minimized the square loss between the two models in order to detect reliable unlabeled samples. Similar to [17], Gan et al. [18] proposed a safe semi-supervised method which added a Laplace regularization term to the square loss function to enhance the reliability of unlabeled sample selection. Persello et al. [19] proposed a progressive S^3VM with diversity (PS^3VM-D) method. On the basis of multiple confidence measurements, reliable unlabeled samples were obtained by querying the samples nearby the margin band.

Weight-based semi-supervisory is based on the fact that the more unlabeled samples with similar weights to the labeled samples, the more reliable the system becomes. Therefore, the influence of unreliable unlabeled samples on the algorithmic performance is suppressed by reducing their weights. For example, [20] considered the unlabeled samples nearby the classification plane and suppressed their influence on the system performance by reducing their weights. In addition, [21–23] controlled the weights by density estimation, weighted likelihood maximization, and graph modelling.

The above semi-supervised methods use unlabeled samples to some extent, however, they also ignore the number of the labeled samples. If the labeled samples are too few, the performance of these algorithms is difficult to be guaranteed, which will inevitably affect the evaluation of the reliability of unlabeled samples. In addition, they lack investigating variability and similarity between unlabeled and labeled samples, which makes it difficult to understand the dynamics and interaction of unlabeled samples. Therefore, in this paper, two new semi-supervised learning methods are proposed: a semi-supervised learning method based on fast search and density peaks (S^2DP) and an iterative S^2DP method (IS^2DP).

When the labeled samples satisfy a certain number, S^2DP is used directly to identify reliable unlabeled samples. For one thing, it works with a new sample weighted kernel Fisher discriminant analysis (WKFDA) supervision method. Using the difference between the samples, the WKFDA method extracts the features of the labeled samples to help formulate the distribution of the unlabeled samples' features, solving the problem of mismatch between them. And for another, it is combined with a clustering method: fast search and determination of density peaks (DP) proposed by Rodriguez and Laio in 2014 [24]. Then, unlabeled sample features are further investigated so that the reliable unlabeled sample features are identified. Finally, an unlabeled sample labeling method based on clustering information (LCI) is designed to retrieve the labels of the unlabeled sample features.

When the labeled samples are insufficient, IS^2DP is used to iteratively render reliable unlabeled samples. Since the labeled and the unlabeled samples may be uneven in numbers, the unreliable unlabeled samples tend to deteriorate the semi-supervised algorithm. The IS^2DP first divides the unlabeled learning set into different subsets according to the size of the labeled sample set. This not only prevents the deterioration of the semi-supervised algorithm by a large number of unreliable samples but also speeds up the processing of the semi-supervised algorithm. Then, the S^3VM is exploited to go through the semi-supervised samples which may be away from the hyperplane of the S^3VM as the reliable semi-supervised samples are added to the labeled samples to improve the performance of the semi-supervised algorithm.

The rest of this paper is organized as follows. Section 2 gives a brief review of the approaches involved. Section 3 describes the proposed method in detail. Section 4 presents the experiments for the SAR images targets recognition. The conclusion is drawn in Section 5.

2. Preliminary

2.1. DP Algorithm. Clustering by fast search and detection of density peaks (DP)[24] can quickly realize accurate detection and clustering of various shapes. Moreover, it is used to evaluate each cluster membership so as to determine reliable cluster members. The DP algorithm is mainly divided into the following three steps.

(1) Determination of Cluster Centers. In the DP, it is assumed that the cluster centers are surrounded by the neighbors with the lower local density and they are at a relatively large distance from any points with a higher local density. Based on the above cluster center assumption, for each sample i, two

quantities are calculated: the local density ρ_i of the sample and the distance σ_i from a sample to the other with a high local density. In the decision map with ρ_i and σ_i as the horizontal and vertical coordinates, respectively, their product is

$$\gamma_i = \rho_i \sigma_i \tag{1}$$

where the sample point with the larger γ_i is more likely to be the cluster center. Therefore, only γ_i is sorted in a descending order, and several corresponding samples are selected as the clustering center from the largest value.

(2) Clustering of Samples. After the clustering center has been determined, all the samples are assigned to be the nearest cluster centers. Compared with the other clustering algorithms, DP clustering process is simple and does not require iterative optimization of the loss function.

(3) Automated Evaluation of Cluster Members. In the clustering results, it is important to quantitatively evaluate the credibility of each sample cluster. The DP algorithm has this capability, compared to other clustering algorithms. It firstly defines a neighbourhood for each cluster. Then, the maximum value ρ_b of the local density of the samples is found in each neighbourhood. Finally, in each cluster, all the samples with local density greater than ρ_b are considered as the cluster core candidates, otherwise, they are considered as the cluster halo of the cluster. The samples in the cluster core are very similar to the central samples and belong to reliable samples. The samples in the cluster halo have a certain distance from the central sample, which is very likely to be noise and belongs to unreliable samples. In addition, there are some cross-clustering and isolated samples that are also unreliable.

In summary, after having clustered by the DP, the samples located at the cluster core are considered to be reliable cluster samples, whilst the others are unreliable samples. Compared to the conventional clustering algorithms, such as Clara [25] or Fanny [26], the DP has lower computational complexity and less computational time. It also well characterizes the distribution of the samples and achieves more accurate clustering results. Besides, the reliability of the clustering results can be provided, which makes the DP easy to be interactive with other algorithms. However, only considering the distance between the sample points can insufficiently characterize the data because it cannot accurately describe the samples with small difference between two categories. When the sample dimension is high, the distance matrix is large, which can reduce the efficiency of the algorithm. Therefore, choosing the appropriate feature extraction method is a key in the DP.

2.2. S^3VM Method. The S^3VM is the extension of the support vector machine (SVM). A standard SVM is based on the structural risk minimization to classify the learning set by extracting the support vectors from the training set to find the optimal hyperplane. In case of the binary SVM, given the training set **L** and the testing set **U**, we have the following constriction optimization problem:

$$\begin{aligned} \min \quad & \Phi(\boldsymbol{w}) = \frac{1}{2}\left(\boldsymbol{w}^T \boldsymbol{w}\right) + \sum_{i=1}^{n} c_i \xi_i \\ s.t. \quad & \mathrm{y}_i \left[\boldsymbol{w}^T \Phi(x_i) + b\right] \geq 1 - \xi_i, \\ & \xi_i \geq 0, \ \mathrm{i} = 1, 2, \cdots, \mathrm{n} \end{aligned} \tag{2}$$

where x_i is the training sample and y_i is the corresponding label, $(x_i, y_i) \in$ **L**; $\Phi(\cdot)$ maps the data into the feature space; $\boldsymbol{w}$ is the orthogonal vector between x_i and the hyperplane; b is the bias to measure the distance between **L** and the hyperplane; ξ_i is the slack variable to represent the offset of x_i; c_i is the cost factor to measure the weight between the optimal hyperplane and the minimum offset; n is the number of the training samples.

For the S^3VM, the iterative process is operated and the semi-labeled samples (selected from **U** in the previous step) are added to **L**. Their confidence is diverse in different iterative steps and they are given different cost factors, leading to the following function:

$$\begin{aligned} \min \quad & \Phi(\boldsymbol{w}) = \frac{1}{2}\left(\boldsymbol{w}^T \boldsymbol{w}\right) + \sum_{i=1}^{n} c_i \xi_i + \sum_{i=1}^{m} c_j \varepsilon_j \\ s.t. \quad & \mathrm{y}_i \left[\boldsymbol{w}^T \Phi(x_i) + b\right] \geq 1 - \xi_i, \\ & \xi_i \geq 0, \ \mathrm{i} = 1, 2, \cdots, \mathrm{n} \\ & \hat{y}_j \left[\boldsymbol{w}^T \Phi\left(\hat{x}_j\right) + b\right] \geq 1 - \varepsilon_j, \\ & \varepsilon_j \geq 0, \ \mathrm{j} = 1, 2, \cdots, \mathrm{m} \end{aligned} \tag{3}$$

where $\hat{x}_j$ is the semi-labeled sample selected from **U**, with the slack variable (ε_j), cost factor (c_j) and semi-label ($\hat{y}_j$) and m is the number of the semi-labeled samples.

The S^3VM can deal with the nonlinear problem using the kernel methods and its semi-supervised samples with the bigger c_j. But when the sample dimension is high, the computation speed would decrease. Therefore, the dimension reduction and effective semi-supervised samples are the critical aspects to the S^3VM.

3. Proposed Methods

This paper presents two methods: S^2DP and IS^2DP. When the labeled samples exceeds a certain number, S^2DP directly performs screening and classification of the reliable unlabeled samples. When the labeled sample is insufficient, IS^2DP is used to continuously query reliable unlabeled samples and generate necessary samples to be added to the labeled samples in order to improve the recognition performance of S^2DP. The S^2DP and IS^2DP are described below, respectively.

3.1. S^2DP. Figure 1 shows the flowchart of the proposed S^2DP. First, we use WKFDA to extract the labeled sample

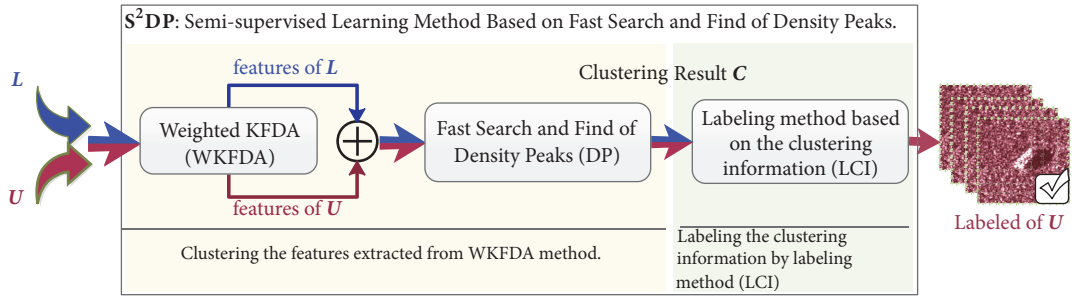


FIGURE 1: Basic flowchart of the S²DP.

features to build a new space. New features are obtained by projecting unlabeled samples into this new space. In this space, the new feature distributions are as close as possible between the intraclass features with a certain weight, and the interclass features are as far apart as possible to enhance the separability between the features. Secondly, the DP is used to cluster the generated features. Finally, the unlabeled samples are identified by the labeling method based on the DP clustering information (LCI). In Figure 1,$\boldsymbol{L}$ represents a set of the labeled samples, and$\boldsymbol{U}$ represents a set of the unlabeled samples, which respectively generate features with the labeled information (i.e., labeled features) and features without labeled information (i.e., unlabeled features) after going through WKFDA;$\boldsymbol{C}$ represents the clustering results of the DP. The WKFDA and LCI methods are described in the following section.

(1) WKFDA. Assume that $\boldsymbol{X}$ represents all the samples of $\mathbf{L}$ and the ith category $\boldsymbol{X_i} = [x_1^i, x_2^i, \ldots, x_{N_i}^i]$ is the subset of $\boldsymbol{X}$, where N_i is the samples' number of $\boldsymbol{X_i}$. $\boldsymbol{v_i} = [v_1^i, v_2^i, \ldots, v_{N_i}^i]$ is the sample weight vector of $\boldsymbol{X_i}$. It is used to control intraclass samples as close as possible with certain weights.

In case of the binary classification, it cannot simply multiply the weight by the corresponding sample. Firstly, the weight matrices $\boldsymbol{V_i}$ and $\boldsymbol{H_i}$ are generated:

$$\begin{aligned} \boldsymbol{V_i} &= \operatorname{diag}\left(v_1^i, v_2^i, \ldots, v_{N_i}^i\right)_{N_i * N_i} \\ \boldsymbol{H_i} &= [\boldsymbol{v_i}, \boldsymbol{v_i}, \ldots, \boldsymbol{v_i}]_{N_i * N_i} \end{aligned} \tag{4}$$

Secondly, the weight vector and the weight matrix are normalized using

$$\begin{aligned} \boldsymbol{v_i} &= \frac{\boldsymbol{v_i}}{sum\left(\boldsymbol{v_i}\right)} \\ \boldsymbol{V_i} &= \frac{N_i \boldsymbol{V_i}}{sum\left(\boldsymbol{V_i}\right)} \\ \boldsymbol{H_i} &= \frac{\boldsymbol{H_i}}{sum\left(\boldsymbol{H_i}\right)} \end{aligned} \tag{5}$$

where $sum(\cdot)$ represents the summation. The above weight matrix can be used to measure the information of the sample itself. Although $\boldsymbol{V_i}$ and $\boldsymbol{H_i}$ are made up of $\boldsymbol{v_i}$, their elements are different. The sum of each column's elements of $\boldsymbol{H_i}$ is equal to 1, and the trace of $\boldsymbol{V_i}$ is equal to N_i. Thirdly, the projection direction $\boldsymbol{w}$ is calculated by Equation (6):

$$\begin{aligned} \boldsymbol{w} &= \sum_{j=1}^{N} a_j \Phi\left(x_j\right) v_j = \sum_{j=1}^{N} a_j v_j \Phi\left(x_j\right) = \sum_{j=1}^{N} \beta_j \Phi\left(x_j\right) \\ &= \Phi\left(\boldsymbol{X}\right) \boldsymbol{\beta} \end{aligned} \tag{6}$$

where $\beta_j = a_j v_j$. $\Phi(\cdot)$ is nonlinear mapping that maps the samples to a new feature space. In this new space, the sample's mean, before and after the projection has been made, can be calculated by

$$\begin{aligned} \boldsymbol{m_i}^{\phi} &= \frac{1}{N_i} \sum_{j=1}^{N_i} \Phi\left(x^i{}_j\right) v^i{}_j = \frac{1}{N_i} \Phi\left(\boldsymbol{X_i}\right) \boldsymbol{v_i} \\ \boldsymbol{w}^T \boldsymbol{m_i}^{\phi} &= \frac{1}{N_i} \sum_{j=1}^{N} \sum_{k=1}^{N_i} \beta_j \Phi\left(x_j\right) \Phi\left(x^i{}_k\right) v^i{}_k = \frac{1}{N_i} \boldsymbol{\beta}^T \boldsymbol{K_i} \boldsymbol{v_i} \end{aligned} \tag{7}$$

The interclass scatter matrix $\boldsymbol{w}^T \boldsymbol{S}^{\phi}{}_b \boldsymbol{w}$ and intraclass scatter matrix $\boldsymbol{w}^T \boldsymbol{S}^{\phi}{}_w \boldsymbol{w}$, after the projection has been achieved, are calculated by

$$\begin{aligned} &\boldsymbol{w}^T \boldsymbol{S}^{\phi}{}_b \boldsymbol{w} = \boldsymbol{w}^T \left(\boldsymbol{m}^{\phi}{}_1 - \boldsymbol{m}_2{}^{\phi}\right)\left(\boldsymbol{m}^{\phi}{}_1 - \boldsymbol{m}_2{}^{\phi}\right)^T \boldsymbol{w} \\ &= \boldsymbol{\beta}^T \boldsymbol{M} \boldsymbol{\beta} \\ &\boldsymbol{w}^T \boldsymbol{S}^{\phi}{}_w \boldsymbol{w} \\ &= \sum_{i=1}^{2} \sum_{j=1}^{N_i} \boldsymbol{w}^T \left[\Phi\left(\boldsymbol{x}^i{}_j\right) v^i{}_j - \boldsymbol{m}_i{}^{\phi}\right]\left[\Phi\left(\boldsymbol{x}^i{}_j\right) v^i{}_j - \boldsymbol{m}_i{}^{\phi}\right]^T \boldsymbol{w} \\ &= \boldsymbol{\beta}^T \boldsymbol{G} \boldsymbol{\beta} \end{aligned} \tag{8}$$

where $\boldsymbol{M} = (\boldsymbol{K}_1 \boldsymbol{v}_1 / \boldsymbol{N}_1 - \boldsymbol{K}_2 \boldsymbol{v}_2 / \boldsymbol{N}_2)(\boldsymbol{K}_1 \boldsymbol{v}_1 / \boldsymbol{N}_1 - \boldsymbol{K}_2 \boldsymbol{v}_2 / \boldsymbol{N}_2)^{\mathrm{T}}$ and $\boldsymbol{G} = \sum_{i=1}^{2} \boldsymbol{K}_i (\boldsymbol{V}_i - \boldsymbol{H}_i)(\boldsymbol{V}_i - \boldsymbol{H}_i)^{\mathrm{T}} \boldsymbol{K}_i{}^T$. In order to satisfy the requirements of the maximum interclass interval and the minimum intraclass interval, this goal can be expressed as follows:

$$\max J\left(\boldsymbol{w}\right) = \frac{\boldsymbol{\beta}^T \boldsymbol{M} \boldsymbol{\beta}}{\boldsymbol{\beta}^T \boldsymbol{G} \boldsymbol{\beta}} \tag{9}$$

which is called the generalized Rayleigh quotient. Then $\boldsymbol{\beta}$ can be calculated according to the flowchart of the KFDA by solving the following optimization problem:

$$\begin{aligned} \max \quad & \boldsymbol{\beta}^T \boldsymbol{M} \boldsymbol{\beta} \\ s.t. \quad & \boldsymbol{\beta}^T \boldsymbol{G} \boldsymbol{\beta} = c \neq 0 \end{aligned} \tag{10}$$

where c is the constant. By introducing the Lagrange multiplier, the function can be transformed to a Lagrange unconstrained extremum problem:

$$L(\boldsymbol{w}, \lambda) = \boldsymbol{\beta}^T \boldsymbol{M} \boldsymbol{\beta} - \lambda \left(\boldsymbol{\beta}^T \boldsymbol{G} \boldsymbol{\beta} - c \right) \tag{11}$$

Let $\partial L(\boldsymbol{w}, \lambda)/\partial \boldsymbol{\beta} = 0, \partial(\cdot)$ represent the partial derivative. This function solution $\boldsymbol{\beta}$ is the eigenvector of $\boldsymbol{G}^{-1}\boldsymbol{M}$. Once solving $\boldsymbol{\beta}$, for any sample $\boldsymbol{x}$, its projection is

$$y = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle = \sum_{i=1}^{N} a_j v_j k(\boldsymbol{x}_i, \boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{K}_x \tag{12}$$

where $\boldsymbol{K}_x$ is the kernel matrix of all the training samples and $\boldsymbol{x}$.

Adding weights to KFDA algorithm is a common way to improve the KFDA algorithm. The aim is to make the WKFDA algorithm better learn sample features. However, different ways of adding weights make the WKFDA algorithm focus on learning sample features differently. For example, [27] added weights to each kernel function. The purpose was to introduce the prior knowledge of samples to enhance the learning of sample features in the WKFDA algorithm. Reference [28] added weights to the within-class scatter matrix. The purpose was to make the WKFDA algorithm not only learn the features of different types of samples but also learn the features of same types of samples in the process of finding the best vector. Unlike these algorithms, the WKFDA algorithm in this paper adds weights to samples, and these weights can be calculated by using the similarity or iterative difference of the samples. The purpose is to make the intraclass samples close to a certain distance, so that the WKFDA algorithm can not only suppress overfitting due to the small number of labeled samples but also facilitate the absorption of spectral information of samples to improve the learning of sample features. Although the binary WKFDA is shown, the multi-WKFDA can be obtained in accordance with the promotion of the kernel Fisher discriminant analysis (KFDA) [29].

(2) LCI. After the labeled sample set$\boldsymbol{L}$ and the unlabeled sample set$\boldsymbol{U}$ have been extracted by the WKFDA method, the labeled and unlabeled features are obtained. Next, the labeled and unlabeled features go into the DP to produce a clustering result$\boldsymbol{C}$. The clustering result$\boldsymbol{C}$ includes features such as cluster center, clustering core, clustering halo, and cross-clustering, but is insufficient to determine the labels of the unlabeled features. To solve this problem, we develop the LCI by using the clustering results and labeling information of the labeled features. LCI is able to label the clustering results of the unlabeled features. Because the unlabeled features are generated from the unlabeled samples, the unlabeled features and the unlabeled samples share the same labels. The basic flowchart of the LCI is shown in Figure 2.

We know that the features of clustering halo and cross-clustering are unreliable. Therefore, in Figure 2, the interference features in$\boldsymbol{C}$ need to be cleared to ensure that the subsequent unlabeled features are reliable. The$\boldsymbol{C}$ clearing the interference is processed separately according to whether the labeled features are included in the cluster core. If there are labeled features in a certain cluster core, the unlabeled features of the cluster core are very similar to the labeled features. These unlabeled features are regarded as the best learning features, combined with the corresponding labeled features, for training the S^3VM. At this time, in each iteration of the S^3VM, the labeled features from the unlabeled features are added to the next iteration to improve the robustness of the S^3VM algorithm. For the clustering cores which do not contain any labeled feature, the cluster centers are extracted and sent to the trained S^3VM to obtain their labels. Once the unlabeled cluster centers are labeled, the unlabeled features of the corresponding clustering core will be assigned the label. In this way, all the clustering cores' features are labeled, and the features that are not labeled are removed as noise. Finally, the unlabeled samples corresponding to the unlabeled features also have corresponding labels.

3.2. IS^{2}DP. When the number of the labeled samples is small but reaches a certain amount, the S^2DP uses the labeled features to investigate the distribution of the unlabeled features and also use the labeled features and the clustering result of the DP to obtain reliable unlabeled samples. However, when the number of the labeled samples is small, after the DP clustering has been achieved, the labeled features are not necessary in the cluster core, resulting in a low correlation between the labeled and unlabeled features. At this time, the labelled samples are difficult to represent the unlabeled samples, and S^2DP is no longer applicable. In this case, the common solution is that the semi-labeled sample from the unlabeled set is queried in order to increase the number of the original labeled samples. In order to obtain the reliable semi-labeled samples, the S^2DP needs to be modified iteratively.

The iterative semi-supervised method of the S^2DP, namely, the IS^{2}DP, is shown in Figure 3, where$\boldsymbol{U}$ is the unlabeled learning set to query the semi-labeled samples,$\boldsymbol{L}$ is the labeled training set, $\boldsymbol{L}^*$ is the semi-labeled samples set in each iteration, $\boldsymbol{L}'$ represents the final labeled training set, and$\boldsymbol{T}$ is the testing set.

The IS^{2}DP specific process is described as follows. Firstly, in each iteration,$\boldsymbol{U}$ is randomly divided into several subsets $(\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_n)$, which are combined with$\boldsymbol{L}$ to obtain $(\boldsymbol{U}_1, \boldsymbol{L}), (\boldsymbol{U}_2, \boldsymbol{L}), \ldots, (\boldsymbol{U}_n, \boldsymbol{L})$ as the input of the S^2DP. Secondly, the cluster cores are selected from $(\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_n)$ after the S^2DP as the candidate semi-labeled samples, and their cluster centers are added to the training set as the labeled samples to train S^3VM. For one thing, the number of the labeled samples sets is increased. And for another, it ensures that the labeled samples match the unlabeled samples since the cluster center represents the features of all the other samples in the

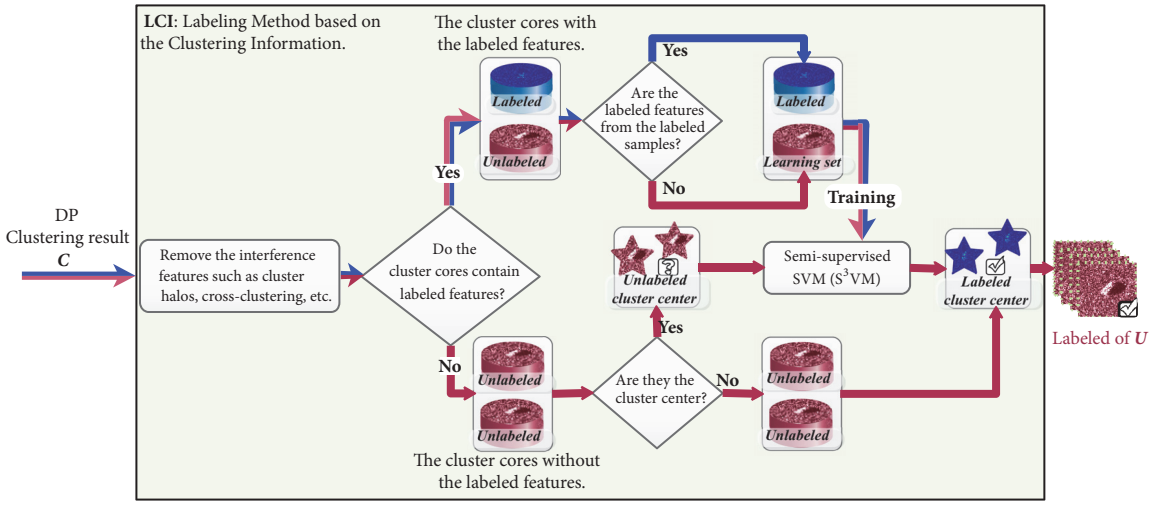


FIGURE 2: Basic flowchart of the LCI.

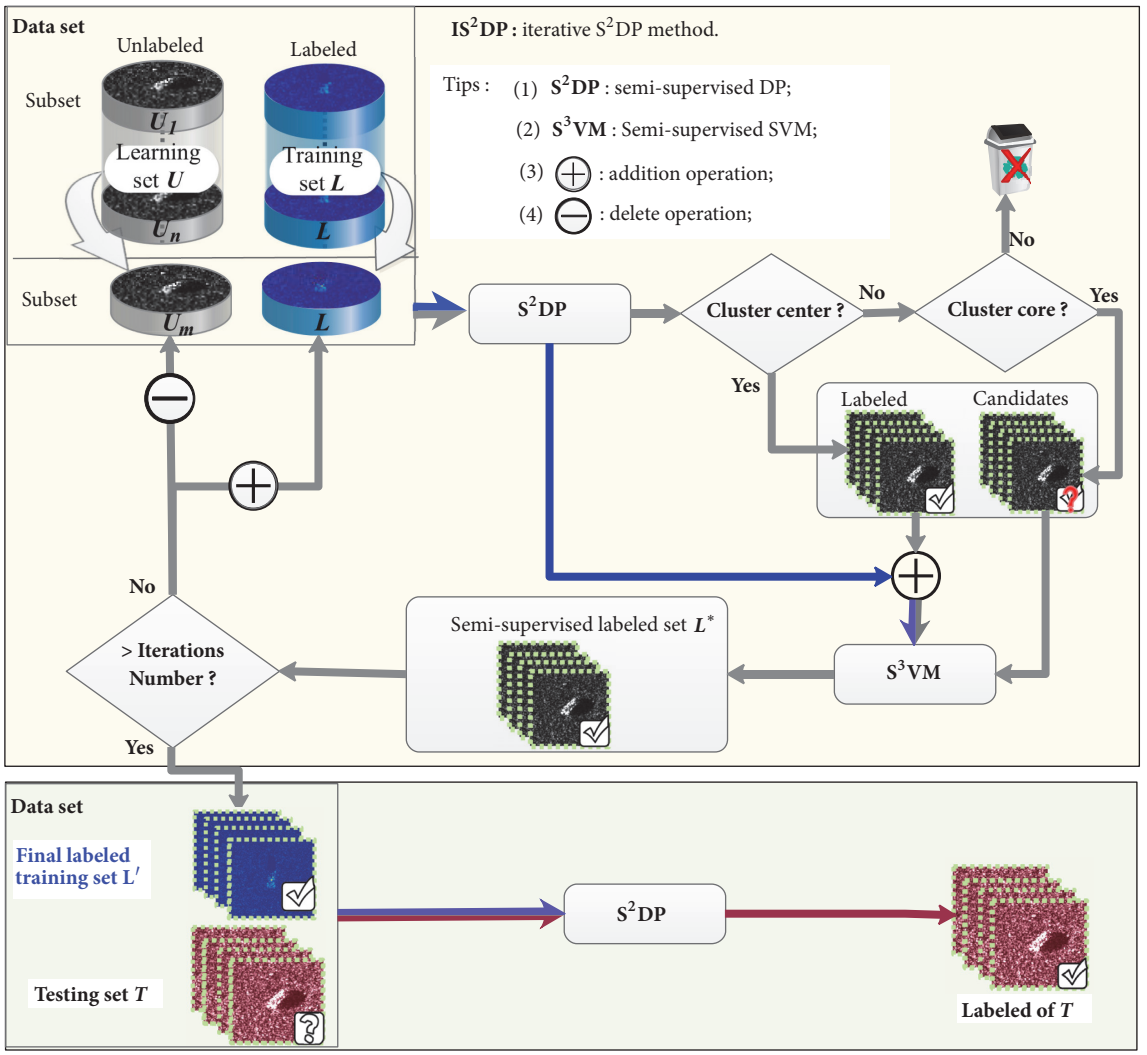


FIGURE 3: Flowchart of the IS^{2}DP.

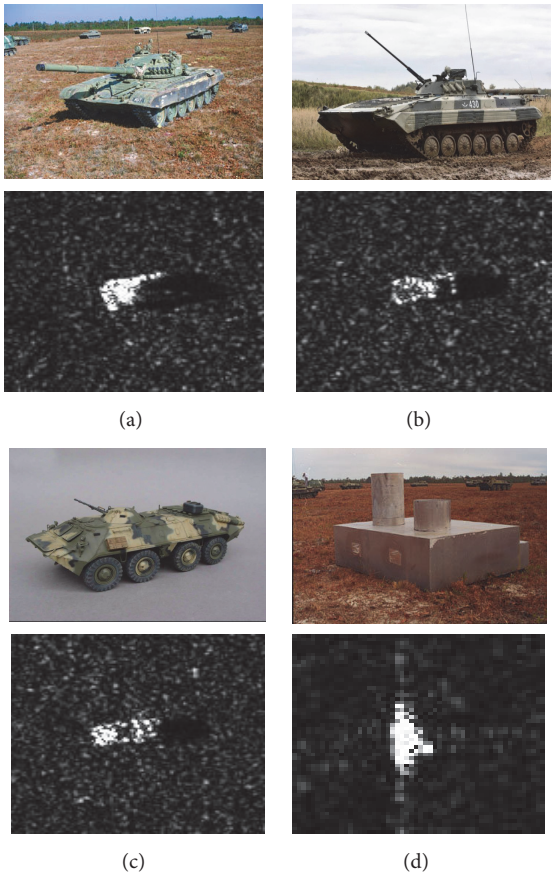


Figure 4: Optical images of four types of targets and corresponding SAR images: (a) T72, (b) BMP2, (c) BTR70, and (d) SLICY.

cluster core. Therefore, the robustness of the S^3VM is ensured. Thirdly, the semi-supervised sample $\boldsymbol{L}^*$ of each iteration is obtained by S^3VM. Finally, it needs to determine whether or not the iteration's termination condition is met so that the number of the iteration is greater than the threshold. If not, $\boldsymbol{L}$ is updated, $\boldsymbol{U}$ is reduced, and the iteration process continues. Otherwise, the final labeled training $\boldsymbol{L}'$ is undertaken to classify the testing set $\boldsymbol{T}$ by the S^2DP.

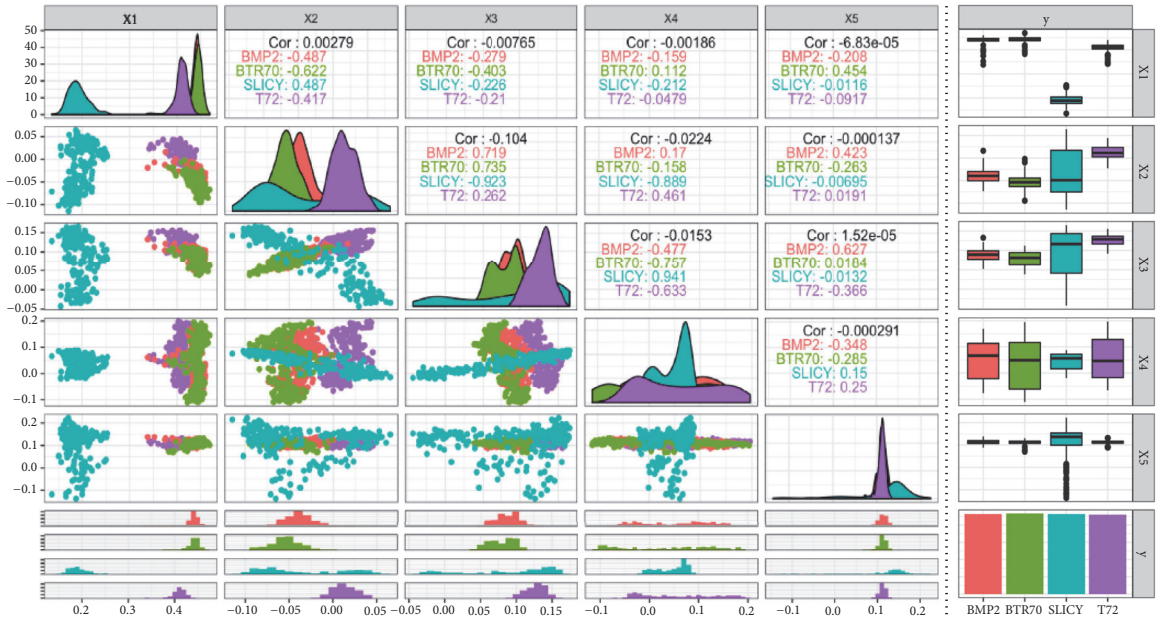
When the labeled sample is insufficient and necessary to query the semi-supervised samples, the IS^2DP can query the reliable semi-supervised samples and classify the unlabeled samples. In fact, the IS^2DP is equivalent to the S^2DP when the labeled samples reach a certain number.

4. Experiments

Our experiments use the SAR images from the Moving and Stationary Target Acquisition and Recognition (MSTAR) database, cofounded by National Defense Research Planning Bureau and the US Air Force Research Laboratory. The military targets contained in the database are collected at 15° and 17° depression angles, covering 360° azimuth angles. To display the intermediate experimental results in geometric space and highlight the significance and effectiveness of our method, the experiments in this paper use three types of military targets and one type of interference targets, which are T72, BMP2, BTR70, and SLICY. Of course, you can also choose other targets. Among these three types of military targets, BMP2 and T72 also contain different version variants. These variants have the same design blueprint, but from different manufacturers, they are slightly different in color and shape.

The optical images of the T72, BMP2, BTR70, and SLICY targets and the corresponding SAR images are shown in Figure 4. From optical images, the difference between these four types of targets is significant. However, the corresponding SAR images are difficult to distinguish by human vision due to speckle noise and similar spatial and spectral characteristics. The original resolution of these SAR image slices are 128∗128 and 45∗45. To facilitate the processing, we only take the 32∗32 resolution that contains the target and flatten these 2D images into one dimension. In order to show the separability of these data, we perform covariance operations on them in order to establish correlations between two-dimensional features. Figure 5 shows the correlation and box plot of the first 5-dimensional features. Figure 5(a) is the correlation of two dimension features, and Figure 5(b) is the corresponding box plot. In Figure 5(a), the lower left corner part is the scatter plot of two-dimensional features, and the upper right part is the correlation coefficient corresponding to the two-dimensional features. *Cor* represents the total correlation coefficient of the relevant two-dimensional features. Positive numbers indicate positive correlations and negative numbers indicate negative correlations. The greater the absolute value of these numbers, the more relevant the features of the corresponding two dimensions. From the correlation coefficient, *Cor*'s absolute value is small which shows that the correlation is low, indicating that they are independent of each other. From the scatter plot, we observe that they are very similar, which increases the difficulty of the recognition algorithm. In addition, from the box plot, there are abnormal points in the upper and lower bounds of the data. If these points are not removed in the learning set features, the performance of the algorithm will be affected.

In order to evaluate the performance of the proposed method, we design three sets of experiments: the evaluation experiment of effectiveness, the evaluation experiment of generalization ability, and the experiment compared with the semi-supervised deep learning method. Among them, the first set of the experiments will be carried out under standard operating conditions (SOC), the latter two sets under different extended operating conditions (EOC). The SOC mean that the testing and the training conditions are very similar. For example, the target types of the training, the learning, and the test sets are the same. On the basis of SOC, the gap between the training and testing conditions is gradually extended to form different EOC. For example, the target types of training set, learning set, and test set are different variants. Even the learning set contains other interfering targets. Compared with SOC, EOC significantly increases the recognition difficulty of the algorithm. We will set up one SOC and two EOCs (EOC_1 and EOC_2) to carry out the above three sets of experiments. The specific configuration of these conditions is as follows.



(a) Correlation between every two dimensions (b) Box plot

FIGURE 5: The correlation between the first 5-dimensional features after flattening SAR images of T72, BMP2, BTR70 and SLICY. (a)Correlation between every two dimensions and (b) box plot.

(1) Data Configuration of the SOC. Table 1 shows the data configuration of SOC. It contains two sets of data: data and test sets. The data set is used for the algorithm training. According to the label of the samples, the data set is divided into the labeled and unlabeled samples. The labeled samples, also known as the training set, have a number ranging from 3 to 40 per class. The unlabeled samples, also known as the learning set, have 190 samples per class. Regardless of the training or learning set, their target depression angle is 17°. The testing set is used for algorithm testing. Its target depression angle is 15°. Regardless of data or test set, we use the same variants of the targets, that is, T72 series sn_132 tanks, BMP2 series sn_c21 armored vehicles, and BTR70 series sn_c71 armored vehicles. We will verify the effectiveness of the S^2DP and IS^2DP under these conditions in Section 4.1, including their core components (LCI and WKFDA).

(2) Data Configuration of the EOC_1. Table 2 shows the data configuration of EOC_1. In Table 2, the training set is the same as that of Table 1. And the testing set is not the same version variants as the training set and the learning set. For example, the T72 is the sn_s7 version in the test, but it is the sn_132 and sn_812 versions in the training and the learning sets, respectively. These conditions will help increase the recognition difficulty of the algorithm. Other conditions shown in Table 2, such as the number of data sets, the depression angle of data sets, and the depression angle of the test set, are the same as those shown in Table 1 and are not described here. We will verify the generalization ability of the S^2DP and IS^2DP under these conditions presented in Section 4.2.

(3) Data Configuration of the EOC_2. Table 3 shows the data configuration of EOC_2. It is formed by adding the interference target SLICY to the learning set of Table 2, further increasing the recognition difficulty of the algorithm. To highlight the advantages of the proposed algorithm, we will compare the S^2DP based IS^2DP algorithm with the semi-supervised depth learning method under EOC_2 in Section 4.3.

4.1. Effectiveness Evaluation Experiment

4.1.1. The Effectiveness of the WKFDA Feature Extraction. To verify the effectiveness of the WKFDA feature extraction, it is compared with the KFDA, kernel local linear discriminant analysis (KLFDA) [30], semi-supervised KLFDA (Semi-KLFDA) [31] and kernel principal component analysis (KPCA) [32]. After these algorithms have extracted features, they all use the standard SVM as the final classifier. The experimental data configuration is shown in Table 1, and with the change of the number of the labeled samples, the overall accuracy rates (OA) of different methods are obtained, as shown in Figure 6. The horizontal axis represents the number of each type of target labeled samples corresponding to different experiments and the vertical axis represents the overall accuracy rate.

In Figure 6, the classification accuracy difference between the different algorithms is very clear. The WKFDA and KFDA both show higher accuracy, followed by the KLFDA and Semi-KLFDA, and finally KPCA. For the WKFDA and KFDA, when the number of the labeled samples is less than

Table 1: Data configuration of the SOC.

	Data set						Testing set		
	Training set (Labeled samples)			Learning set (Unlabeled samples)					
Target	T72	BMP2	BTR70	T72	BMP2	BTR70	T72	BMP2	BTR70
Type	sn_132	sn_c21	sn_c71	sn_132	sn_c21	sn_c71	sn_132	sn_c21	sn_c71
Quantity	3~40	3~40	3~40	190	190	190	196	195	196
Depression	17°			17°			15°		

Table 2: Data configuration of the EOC_1.

	Data set						Testing set		
	Training set (Labeled samples)			Learning set (Unlabeled samples)					
Target	T72	BMP2	BTR70	T72	BMP2	BTR70	T72	BMP2	BTR70
Type	sn_132	sn_c21	sn_c71	sn_812	sn_9566	sn_c71	sn_s7	sn_9563	sn_c71
Quantity	3~40	3~40	3~40	190	190	190	196	195	196
Depression	17°			17°			15°		

Table 3: Data configuration of the EOC_2.

	Data set							Testing set		
	Training set (Labeled samples)			Learning set (Unlabeled samples)						
Target	T72	BMP2	BTR70	T72	BMP2	BTR70	SLICY	T72	BMP2	BTR70
Type	sn_132	sn_c21	sn_c71	sn_812	sn_9566	sn_c71	—	sn_s7	sn_9563	sn_c71
Quantity	3~40	3~40	3~40	190	190	190	190	196	195	196
Depression	17°			17°				15°		

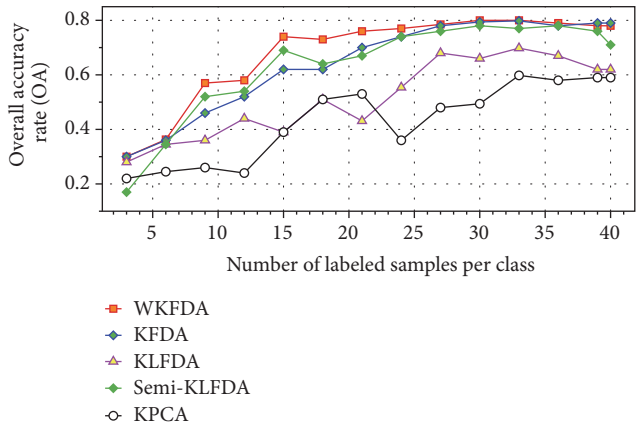


Figure 6: The OA trend chart of different feature extraction algorithms with the number of labeled samples changes in SOC experiment. Here, these feature extraction algorithms use SVM as classifier.

24, the WKFDA's classification results are better than the KFDA. When the number of the labeled samples is greater than 24, their classification results are almost the same. It shows that KFDA and WKFDA have good feature extraction capabilities, while the WKFDA is suitable for dealing with a small quantities of labeled samples. For the KFDA and Semi-KLFDA, when the number of the labeled samples is less than 20, the Semi-KLFDA's classification results are better than the KFDA. When the number of labeled samples is greater than 20, their classification results are almost the same. For the KPCA, as the number of the labeled samples increases, its classification results are always poor.

In order to understand the above experimental results, we take a close look at the projections of the learning samples under the condition that the same number of the labeled samples is taken. Figures 7(a), 7(b), 7(c), 7(d), and 7(e) shows the projection of the learning set for KPCA, KLFDA, Semi-KLFDA, KFDA, and WKFDA algorithms respectively when the number of the labeled samples is 20. As can be seen from Figure 7, the projection result shown in (e) is the best, where we can classify the three targets, second best is (d) and then (c), (b), and (a). The quality of the projection results mainly depends on whether or not the feature extraction algorithm can effectively extract features from the SAR images.

For the KPCA algorithm, it only reduces the original features of the SAR images. As the number of the labeled samples increases, the classification accuracy of the KPCA features continues to increase. The original features of the SAR images are difficult to identify. Therefore, the classification accuracy of SVM based on the KPCA features is poor, shown in Figure 7(a).

For the KLFDA and Semi-KLFDA algorithms, they take advantage of the difference between the sample classes and extract features that are easily identifiable from the SAR images to certain extent. Therefore, their projection looks better than the KPCA algorithm. However, in the case where the overall features are not separable, the KLFDA and Semi-KLFDA algorithms overemphasize the local features,

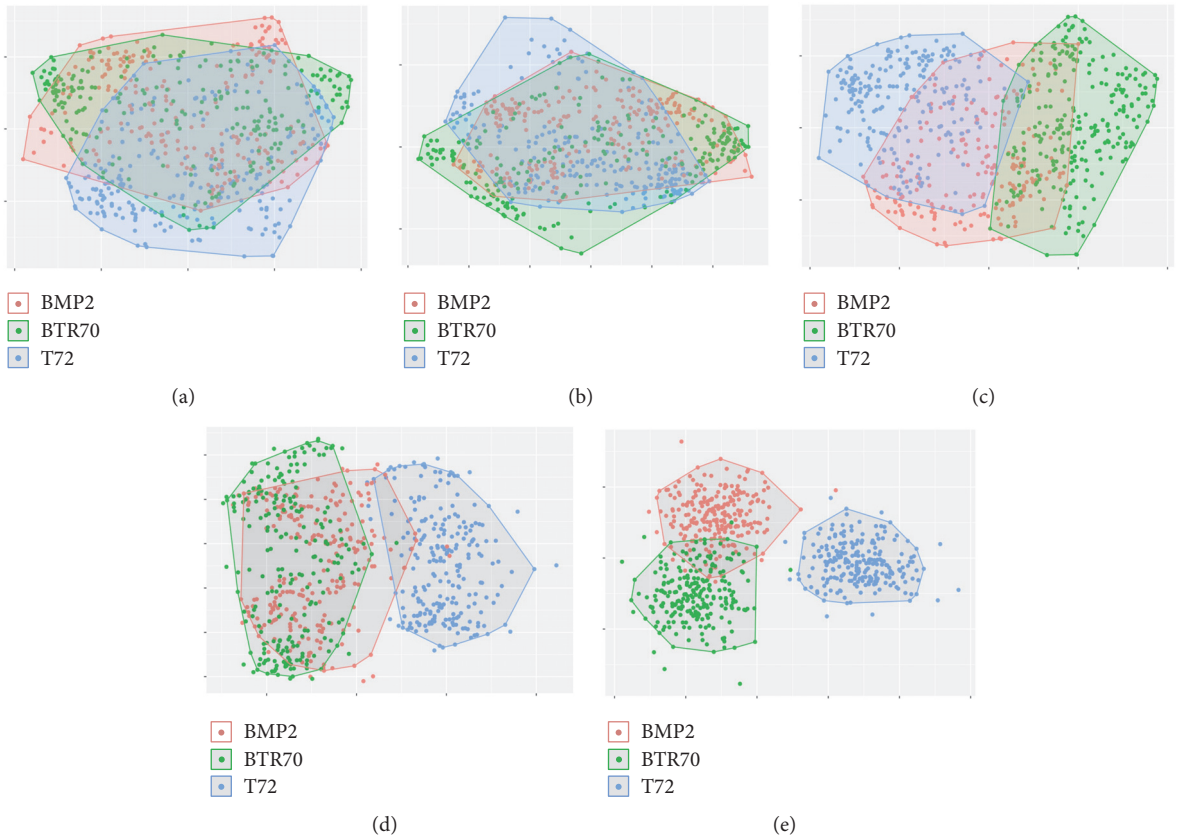


Figure 7: The projections of the learning set for KLFDA, Semi-KLFDA, and KPCA, respectively, when the number of the labeled samples is 20. (a)KPCA; (b)KLFDA; (c)Semi-KLFDA; (d)KFDA; (e)WKFDA.

resulting in more confusing clutters in the projection space. This is observed from Figures 7(b) and 7(c).

For the KFDA and WKFDA algorithms, both of them well use the difference between different classes and the similarities in the same classes. Therefore, the projections shown in Figures 7(d) and 7(e) are better than those of the other methods. We know that the KFDA and WKFDA algorithms are supervised algorithms which guide the projection of the unlabeled sample features based on the features of the labeled samples. Therefore, whether or not these algorithms are good at learning the labeled sample features will affect the quality of the projection of the unlabeled sample features. In the process of the labeled sample feature learning, the KFDA algorithm forces the interclass samples to be as far apart as possible in addition to forcing the samples intraclasses to be as close as possible. At the same time, it may cause the algorithm to overfit and is difficult to guide the unlabeled sample features to be projected onto the optimal direction. The WKFDA algorithm is able to give the samples different weights so that the intraclass samples are close to each other with a certain weight. This can balance the concentration characteristics of the samples (the intraclass samples aggregate with each other and have a certain spatial structure) and can fully utilise the spectral information of the samples and reduces the algorithm's overfitting. Therefore, the results shown in Figure 7(e) seem better than those of Figure 7(d).

To further explore the impact of weighting on the WKFDA algorithm, taking the same labeled samples, Figure 8 shows the results of the KFDA and WKFDA algorithms for learning the features of the labeled samples. Figure 8(a) shows the KFDA features. Although the interclass distance is significant, the intraclass samples are concentrated, almost grouping to a point, which is easy to cause overfitting of the algorithm. Figure 8(b) shows the WKFDA features. Under the condition that the interclasses is separable, the intraclass distance is relatively large, which is easy to learn the sample information and suppress the overfitting of the algorithm. Here, Figure 8(b) also shows that different weights can result in different intraclass distance. Compared with the weight of 100, the intraclass sample space is larger when the weight is 50, and the interclasses can be well spaced, which makes it easier to learn sample information.

4.1.2. The Effectiveness of the LCI for Labeling Unlabeled Samples. To verify the effectiveness of the LCI for labeling unlabeled samples, using the WKFDA features of the Section 4.1.1 experiments, LCI is compared with the SVM and S^3VM classifiers under the DP clustering conditions. Using Table 1 as the experimental data, the same number of the labeled samples is selected from each type of targets in the training set. With the change of the number of the labeled samples, the OA trend chart of three methods is obtained, as

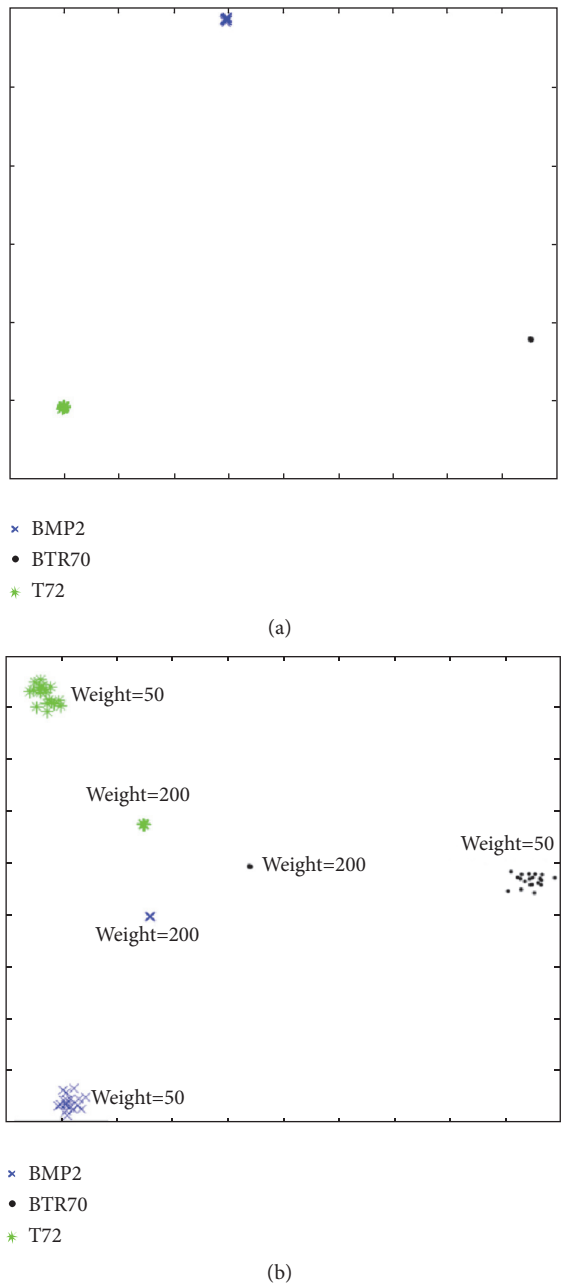


Figure 8: The projection of the KFDA and WKFDA with the same number of the labeled samples. (a)KFDA and (b)WKFDA.

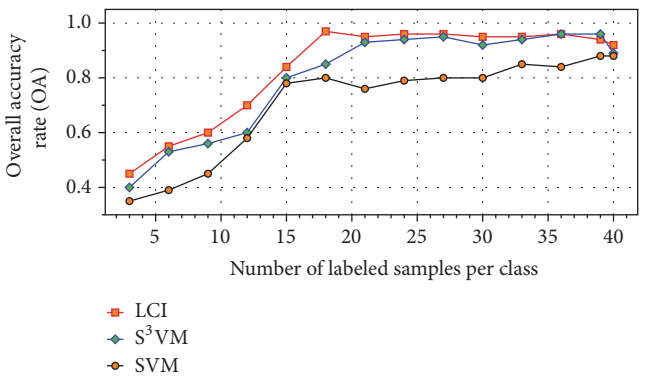


Figure 9: The OA trend chart of the LCI, S^3VM, and SVM algorithms as the number of labeled samples changes in the SOC experiment.

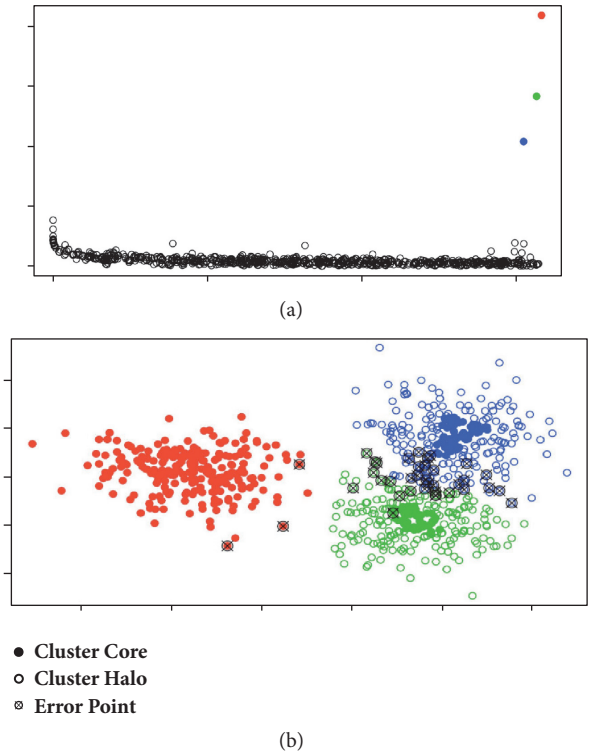


Figure 10: The clustering results of DP on the learning set when the number of the labeled samples is 21. (a) Red circle, green circle, and blue circle are the cluster centers selected by the DP; (b)the clustering results of the DP. • are the cluster cores, ◦ are clustered halos, and ⊗ are clustering error samples.

shown in Figure 9. The horizontal axis represents the number of each type of the labeled target samples corresponding to different experiments, and the vertical axis represents the overall accuracy rate. As can be seen from Figure 9, the accuracy of LCI and S^3VM is better than SVM. With more and more labeled samples, the accuracy of LCI and S^3VM is almost the same.

We know that the SVM, as a supervised learning method, requires a large number of labeled samples. Because the DP algorithm cannot provide enough labeled samples for the SVM, the SVM classification results are poor. For S^3VM and LCI, as a semi-supervised method, when the DP clustering outcome is reliable, they collect enough labeled samples to improve the recognition performance. Figure 10 shows the clustering results of the DP on the learning set when the number of the labeled samples is 21. In Figure 10(a), red circle, green circle, and blue circle are the cluster centers selected by the DP. The DP algorithm recommends that the learning set be divided into 3 categories, which is consistent with the actual situation. Figure 10(b) shows the clustering results of the DP. • are the cluster cores, ◦ are clustered halos, and ⊗

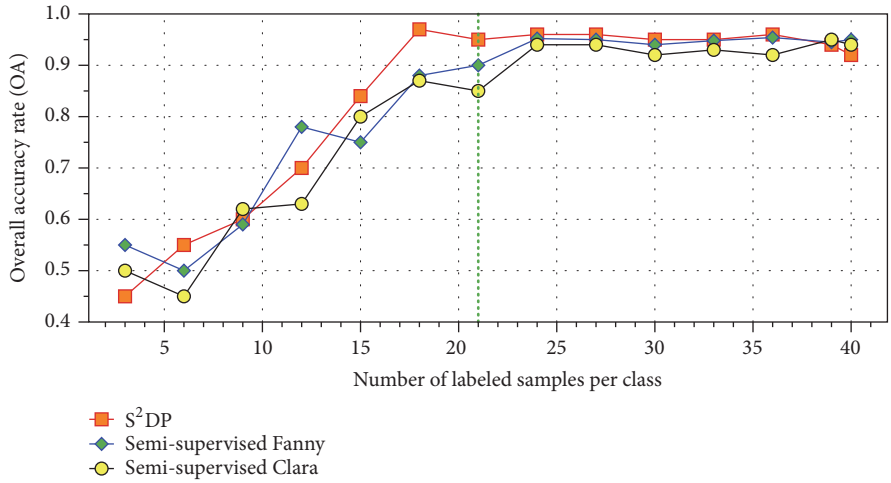


Figure 11: The OA trend chart of the S²DP, semi-supervised Clara, and semi-supervised Fanny with the number of labeled samples changes in SOC experiment.

are clustering errors. From Figure 10(b), the DP clustering has only minor errors and the result is quite accurate, further demonstrating that the recognition accuracy of the S³VM and LCI is equivalent. In addition, these errors are located in the cluster halos. In the LCI algorithm, the cluster halos and the cross-clustering samples will be deleted to ensure that the final labeled samples are reliable. Therefore, compared to the S³VM, the LCI recognition results are more consistent.

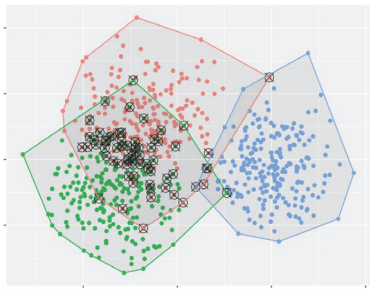
4.1.3. Verifying the Recognition Performance of S²DP. In order to verify the recognition performance of S²DP, S²DP is compared with its similar semi-supervised methods. These similar semi-supervised methods are the semi-supervised algorithms that replace DP in S²DP with other classical clustering algorithms: Clara [25] and Fanny [26], namely, semi-supervised Clara and semi-supervised Fanny. The experimental data configuration is shown in Table 1. With the changing numbers of the labeled samples, the OA trend chart of three methods is obtained, as shown in Figure 11. The horizontal axis represents the number of each type of labeled target samples corresponding to different experiments, and the vertical axis represents the overall accuracy rate.

By comparing the S²DP with the semi-supervised Clara and semi-supervised Fanny, the classification results of the different methods are greatly influenced by the number of the labeled samples. When the number of the labeled samples is less than 24, the overall accuracy of the three methods is continuously improved with the increase of the labeled samples. For the curve smoothness, the curve of the S²DP looks consistent over the curves of the other two methods. When the number of the labeled samples reaches 15, the recognition accuracy of S²DP is higher than that of the other two methods. When the number of the labeled samples reaches 24, the three methods have the same recognition accuracy and the curve trend is stable, but the S²DP is still better than the other two methods. Therefore, the S²DP is superior to the other two algorithms in terms of stability and classification accuracy.

When the labeled samples are very few, the DP clustering results of S²DP are too divergent to represent the unlabeled samples. Only few labeled samples are generated from the cluster core samples. In the end, the classification accuracy of the S²DP will not be high. As the number of the labeled samples increases, more and more labeled samples are generated by the cluster cores, which are also quite reliable. The S²DP classification accuracy is greatly improved. The other two methods are similar. However, as the number of the labeled samples increases, it is difficult for semi-supervised Clara and semi-supervised Fanny to guarantee the reliability of the labeled samples from the unlabeled samples during the clustering process. Therefore, their stability is not as good as that of S²DP. Figure 12 shows the three algorithms generate labeled samples from the learning set when the number of the labeled samples is 21. ⊗ are clustering errors. Obviously, the labeled samples generated by the S²DP algorithm are more reliable than the other two methods.

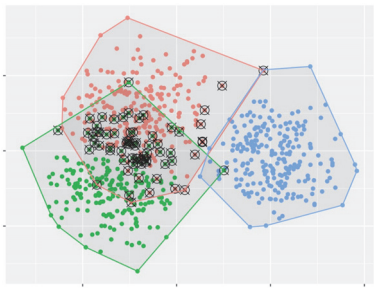
The Sections 4.1.1–4.1.3 experimental results show the relationship between the number of the labeled samples and the S²DP, verifying the validity of the WKFDA, LCI, and DP as the key step in the S²DP. It shows that the S²DP, compared with the other two methods, can achieve the best classification result when the initial labeled samples reach a certain number. But when the labeled samples are too few, its classification precision decreases. Therefore, the ability of the modified IS²DP to query semi-supervised samples needs to be verified.

4.1.4. Verifying the IS²DP in Ability to Query the Semi-Labeled Samples. When the labeled samples are few, the IS²DP can select the semi-labeled samples from the unlabeled samples as the labeled samples. In the Section 1, we know that PS³VM-D is also a semi-supervised method, which considers reliable incremental samples as semi-supervised samples by sample similarity. Therefore, PS³VM-D is selected as a comparative semi-supervised algorithm. They are all based on the extracted features by WKFDA. The experimental data



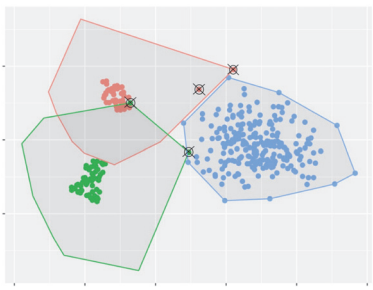
BMP2
BTR70
T72

(a)



BMP2
BTR70
T72

(b)



BMP2
BTR70
T72

(c)

Figure 12: The semi-supervised Clara, semi-supervised Fanny, and S^2DP generate reliable labeled samples from the learning set when the number of the labeled samples is 21. ⊗ are clustering errors. (a) The Clara clustering results; (b) the Fanny clustering results; (c) the DP cluster cores.

configuration is shown in Table 1. With the change of the number of the labeled samples, the OA trend chart of two methods is obtained, as shown in Figure 13. The horizontal axis represents the number of each type of the labeled target samples corresponding to different experiments and the vertical axis represents the overall accuracy rate.

When the number of the labeled samples is less than 25, the classification accuracy of PS^3VM-D is obviously lower than that of IS^2DP, indicating that IS^2DP is more suitable for the case of too few labeled samples. When the number of the labeled samples is more than 25, the IS^2DP and PS^3VM-D have the same accuracy. It shows that the PS^3VM-D also gets enough labeled sample information, and the classification accuracy is improved.

We know that the core of PS^3VM-D is SVM. The optimal classification surface of PS^3VM-D is mainly influenced by SVM. The PS^3VM-D relies heavily on the labeled samples. It needs enough quantity to obtain a universal classification surface. Therefore, its classification performance varies significantly with the number of labeled samples and cannot remain stable until the labeled samples are sufficient. The classification performance of the IS^2DP is largely determined by the DP and WKFDA, which makes the IS^2DP more stable and accurate when the labeled samples are very few due to the sample description ability of the DP and the effective use of the labeled samples by the WKFDA.

4.2. Evaluation of Generalization Ability. The following will verify the generalization capabilities of S^2DP and IS^2DP under the EOC_1.

4.2.1. Verifying the S^2DP Generalization Capabilities. In Section 4.1.2, the comparison between LCI and S^3VM algorithm is actually the comparison of S^2DP with S^3VM based on WKFDA and DP (WKFDA+DP+S^3VM). The recognition accuracy of S^2DP and WKFDA+DP+S^3VM is equivalent in the SOC experiments. Here, we continue to compare the S^2DP and WKFDA+DP+S^3VM. The experimental data configuration is shown in Table 2. With the change of the number of labeled samples, the OA trend chart of two methods is obtained, as shown in Figure 14. The horizontal axis represents the number of each type of the labeled target samples corresponding to different experiments, and the vertical axis represents the overall accuracy rate.

In Figure 14, the recognition accuracy of S^2DP and WKFDA+DP+S^3VM algorithms increases with the increasing number of the labeled samples, and their final accuracy is equivalent. However, the curve of the S^2DP is relatively smooth. This shows that our method is stable and robust.

To verify this conclusion, we perform visual analysis of the key steps of the two methods, when the number of samples is 21. Figure 15(a) shows the features of the training and learning sets after the WKFDA processing. • represents the learning samples and ∗ represents the initial labeled sample. Figure 15(b) is the actual classification map of the WKFDA features after the DP clustering has been achieved. • represents the clustering core and ◦ represents the clustering halo. Figure 15(c) is the true classification map of Figure 15(b). • represents the clustering core, ⊗ represents the clustering error sample, and **?** represents the sample of the next step of the algorithm to be identified. As can be seen from Figure 15(a), the three types of targets are more confused at the boundary, which means that, in the future, they will affect the performance of the recognition algorithm if these

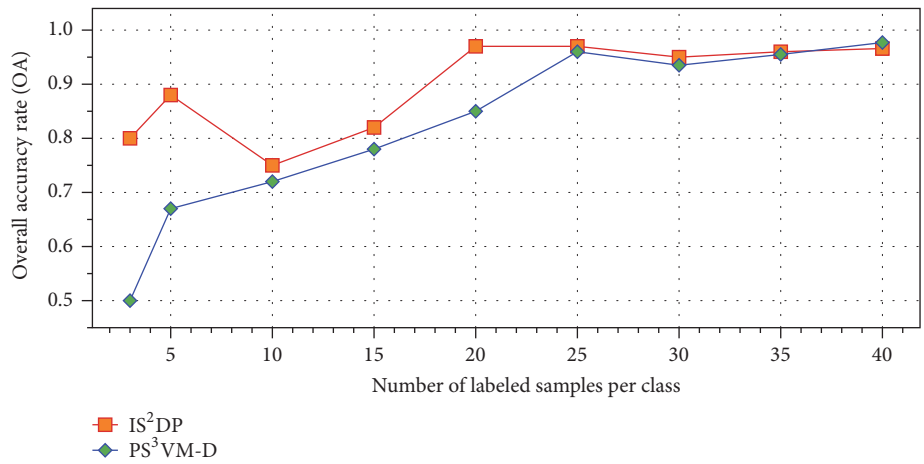


Figure 13: The OA trend chart of the IS^{2}DP and PS^{3}VM-D as the number of the labeled samples changes in the SOC experiment.

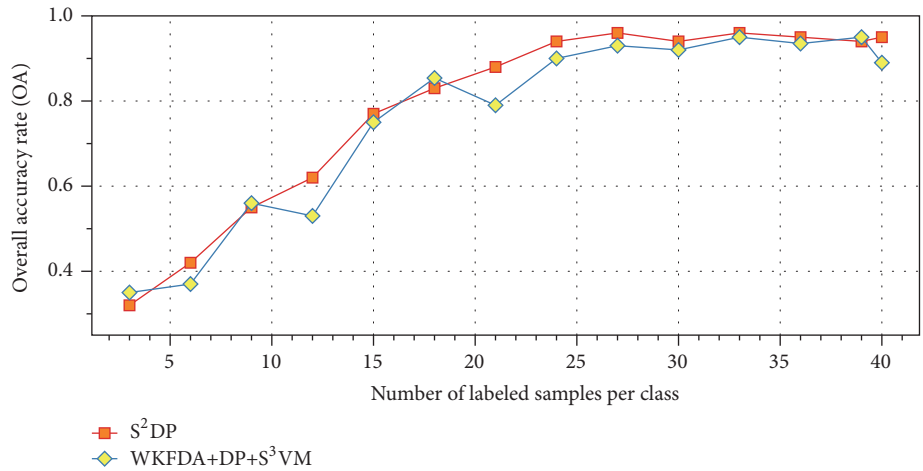


Figure 14: The OA trend chart of the S^{2}DP and WKFDA+DP+S^{3}VM as the number of labeled samples changes in the EOC_1 experiment.

samples are not cleared. As can be seen from Figure 15(b), the DP algorithm divides the WKFDA features into 5 clusters. Among these 5 clusters, clusters 1 and 3 have cluster haloes, and clusters 2, 4, and 5 are all clustered cores. As can be seen from Figure 15(c), the initial labeled samples ($*$ samples) are not included in clusters 4 and 5 and, therefore, the samples of clusters 4 and 5 need to wait for the next step of the algorithm to identify and label. Clusters 1, 2, and 3 contain initial labeled samples ($*$ samples), so they get the same label as the initial labeled sample. In the clustering halos of clusters 1 and 3, there are many clustering error samples ($\otimes$ samples) caused by the confused samples shown in Figure 15(a). This means that, in the future, they will affect the performance of the recognition algorithm if these $\otimes$ samples are not cleared.

For the WKFDA+DP+S^{3}VM algorithm, in the S^{3}VM training process, for one thing, the S^{3}VM cannot clear the $\otimes$ samples in Figure 15(c). And for another, for the **?** samples in Figure 15(c), the S^{3}VM can only identify them by traversing the samples. Therefore, the WKFDA+DP+S^{3}VM algorithm is unstable and inefficient. For the S^{2}DP algorithm, once the features of Figure 15(c) are input into the LCI, the LCI algorithm removes the unreliable features such as the clustering halos and cross-clustering features and makes full use of the cluster cores as reliable samples. Once the labeled samples are included in the cluster core, the other unlabeled samples are labeled with the labels of the labeled samples. For cluster cores that do not contain labeled samples, only the clustering center is identified, and the label of the whole cluster core can be obtained, which greatly improves the recognition efficiency. Figure 16 is a sequence diagram showing the recognition of the DP clustering result of Figure 15(b) by the LCI in the S^{2}DP algorithm. Figure 16(a) is the visualization of the DP clustering results after removing the interference samples. Figure 16(b) is the result diagram of LCI's final recognition of the DP clustering. As can be seen from Figure 16(a), both the confusing sample in Figure 15(a) and the $\otimes$ sample in Figure 15(c) are removed, greatly improving the reliability of sample identification. As can be seen from Figure 16(b), clusters 4 and 5 are correctly identified, and at the same time, only 5 samples with incorrect identification are in the

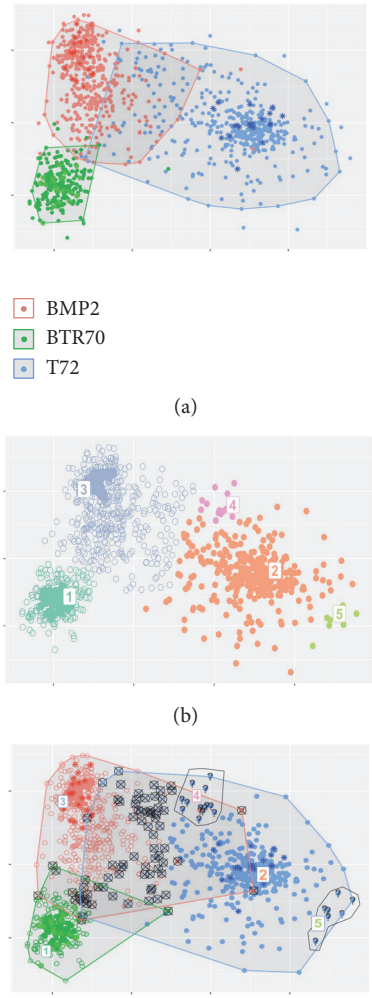


Figure 15: In EOC_1 experiment, the visualization results of the first two steps of the S^{2}DP and WKFDA+DP+S^{3}VM algorithms when the number of the samples is 21. (a) WKFDA features: • represents the learning set and ∗ represents the initial labeled samples; (b) actual classification map of WKFDA features after DP clustering has been carried: • represents the clustering core and ◦ represents the clustering halo; (c) true classification map of (b): • represents the clustering core, ⊗ represents the clustering error sample, and **?** represents the sample of the next step of the algorithm to be identified.

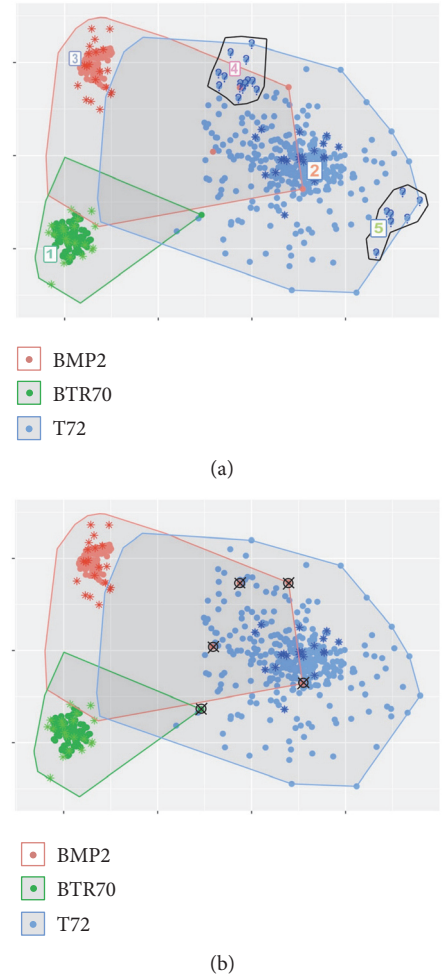


Figure 16: The sequence diagram of the LCI algorithm for identifying DP clustering results in Figure 15(b). (a) Visualization of DP clustering results after removing the interference samples; (b) LCI's final recognition visualization of DP clustering. • represents the clustering core; ∗ represents the initial labeled sample; **?** represents the sample of the next step of the algorithm to be identified; ⊗ represents the sample for labeling errors.

expanded labeled samples. Thus, S^{2}DP is quite reliable. In this way, the above conclusions are verified.

4.2.2. Verifying the IS^{2}DP Generalization Capabilities. In Section 4.2.1, the S^{2}DP is relatively stable, but its recognition accuracy is relatively low when the number of the labeled samples is less than 21. Therefore, the IS^{2}DP is required to generate a large number of the labeled samples to improve the recognition accuracy of S^{2}DP. Here, we compare the IS^{2}DP+S^{2}DP and S^{2}DP. The experimental data configuration is shown in Table 2 with the change of the number of the labeled samples, and the OA trend chart of the two methods is obtained, as shown in Figure 17. The horizontal axis represents the number of each type of target labeled samples corresponding to different experiments; the vertical axis represents the overall accuracy rate.

From Figure 17, we can see that when the number of the labeled samples is less than 21, the recognition performance of IS^{2}DP+S^{2}DP is 10% higher than that of S^{2}DP. With the number of labeled samples larger than 21, their classification accuracy is equivalent. To verify this conclusion, we apply 100 iterations onto IS^{2}DP when the number of labeled samples is 15. The labeled samples generated by IS^{2}DP are counted, as shown in Table 4. The accuracy rate of labeled samples generated from learning set is over 85%.

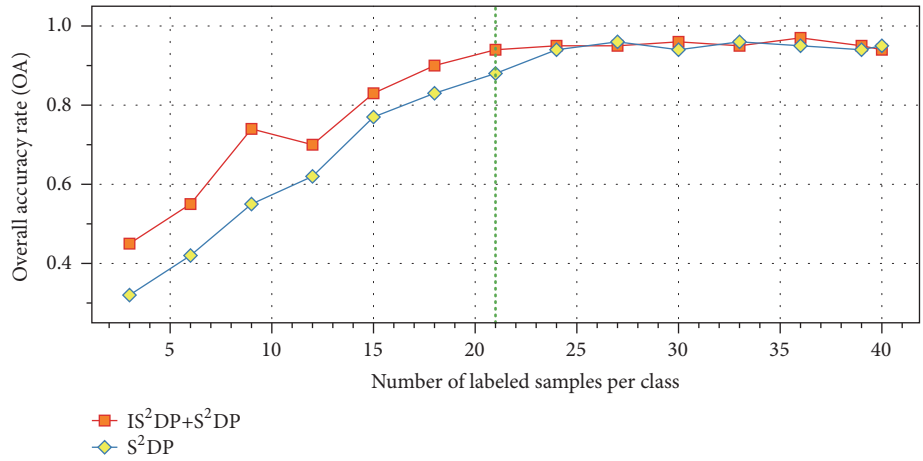


Figure 17: The OA trend charts of the IS^{2}DP+S^{2}DP and S^{2}DP with the number of labeled samples changes in EOC_1 experiment.

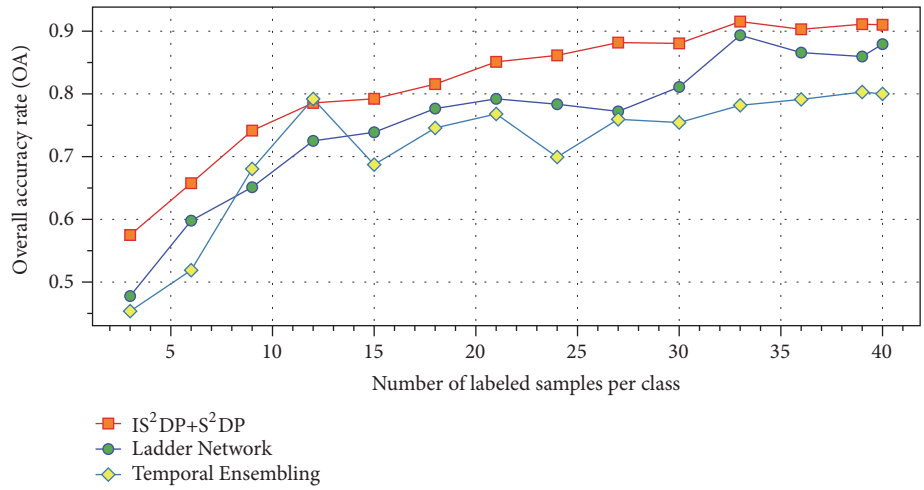


Figure 18: The OA trend charts of the IS^{2}DP+S^{2}DP, Ladder Network, and Temporal Ensembling with the number of labeled samples changes in EOC_2 experiment.

4.3. Comparison with Semi-Supervised Deep Learning. The semi-supervised deep learning algorithms, Ladder Network [7] and Temporal Ensembling [8], which contain a supervised and unsupervised learning process, similar to our algorithm. Therefore, we choose these two methods to compare with the IS^{2}DP-based S^{2}DP algorithm (IS^{2}DP+S^{2}DP). In addition to using SAR images as experimental data, we also use a set of publicly available optical image data to verify the effectiveness of our algorithm.

4.3.1. Testing with SAR Images. The experimental data configuration is shown in Table 3, and with the change of the number of the labeled samples, the OA trend chart of three methods is obtained, as shown in Figure 18. The horizontal axis represents the number of each type of labeled target samples corresponding to different experiments, and the vertical axis represents the overall accuracy rate.

In Figure 18, the recognition accuracy of the three methods is increasing with the increase of the labeled samples. From the curve smoothing, the accuracy curves of the Ladder Network and the Temporal Ensembling are fluctuating, especially the Temporal Ensembling. Comparing them, the accuracy curve of the IS^{2}DP+S^{2}DP is relatively consistent. From the classification accuracy, when the number of the labeled samples is less than 33, the results obtained by Ladder Network and Temporal Ensembling are not much different, but significantly lower than that of IS^{2}DP+S^{2}DP. When the number of the labeled samples reaches 33, the classification accuracy of IS^{2}DP+S^{2}DP is slightly better than that of Ladder Network. These results indicate that the learning set containing the interference samples has a great influence on the recognition performance of the Ladder Network and Temporal Ensembling. Because the Ladder Network and Temporal Ensembling were unable to remove these interference samples during the training process, their recognition accuracy was unstable and not high. Different from them, the IS^{2}DP+S^{2}DP can select reliable unlabeled samples and remove those interference samples, so its recognition performance is relatively stable and the accuracy is improved. When the number of the labeled samples is equal

Table 4: The IS^{2}DP generates labeled samples from the learning set when the number of labeled samples is 15 in EOC_1 experiment.

Target	Learning set	Generate labeled samples		Reject	Accuracy of each type of target(%)
		Correct	Error		
T72	190	167	11	12	87.89
BMP2	190	163	13	14	85.79
BTR70	190	172	8	10	90.53
Overall accuracy					88.07

Table 5: Under EOC_2, when labeled samples number 21, recognition results (confusion matrix) of the learning set by trained Temporal Ensembling.

	T72	BMP2	BTR70	Accuracy of each type of target(%)
T72	65	3	122	34.21
BMP2	0	174	16	91.58
BTR70	0	0	190	100
SLICY	0	0	190	0
Overall accuracy				56.45

to 21, we will analyze the use of the learning set by the three methods below.

In the Temporal Ensembling algorithm, one neutral network conducts two different works, supervised learning and unsupervised learning. Figures 19(a)–19(c), respectively, show losses in these two processes and in the whole method. Observed from the curve fluctuation, supervised learning loss in Figure 19(a) is the most stable while unsupervised learning in Figure 19(b) fluctuates significantly. It demonstrates that neural network performs well in learning labeled samples, but is still unstable to handle the learning set, thus resulting in unstable overall loss as shown in Figure 19(c). Finally, the Temporal Ensembling algorithm utilizes the learning set by 56.45% only, which is calculated based on the trained neutral network's recognition of the learning set. Recognition results of the learning set by the Temporal Ensembling algorithm is displayed in Table 5 (confusion matrix). Observed from the confusion matrix, the remaining 43.55% disturbs the learning process, for instance, by misrecognizing SLICY as BTR70 targets.

Similar to temporal ensembling algorithm, the neutral network in the Ladder Network algorithm consists of supervised learning and unsupervised learning as well. Figures 20(a)–20(c), respectively, show losses in these two processes and by the whole method. Observed from the curve fluctuation shown in Figure 20(a), supervised learning loss significantly fluctuates, probably because of inadequate labeled samples; in Figure 20(b), unsupervised learning performs stably, probably resulting from unsupervised learning (Autoencoder) embedded in the Ladder Network algorithm which could learn and recognize unlabeled samples and reduce certain interference. Thus, the overall loss shown in Figure 20(c) performs stably. Therefore, comparing with temporal ensembling, Ladder Network improves the utilization of the learning set to 68.42% (as shown in Table 6 confusion matrix), enhancing its recognition performance as well.

Differing from temporal ensembling and ladder network, the IS^{2}DP+S^{2}DP algorithm identifies reliable unlabeled samples by iterations before implementing feature learning, instead of directly learning features from the unlabeled samples. Here we employ 300 iterations on the IS^{2}DP+S^{2}DP algorithm for fair comparison. Figures 21(a)–21(c) show the screening of the reliable samples in the learning set during one iteration: (a) projection of the WKFDA algorithm on the learning set; (b) DP clustering result; (c) reliable samples labeled by LCI. In Figure 21, red circle, green circle, light blue circle, and blue circle represent BMP2, BTR70, T72, and SLICY target samples, respectively, and $*$ represents the labeled samples. Confused by SLICY interference targets, the WKFDA algorithm has some issue in projecting the learning set but performs well in dividing different samples during the DP clustering, and successfully identify SLICY during the LCI labeling process. Finally, IS^{2}DP+S^{2}DP improves the utilization of the learning set to 82.76% (as shown in Table 7). As 28.95% unreliable sample rejecting recognition will be deleted, only 10% false samples affects the performance; thus IS^{2}DP+S^{2}DP's recognition performance can be improved.

4.3.2. Testing with Optical Images. To verify the effectiveness of the proposed method on other data sets, we use optical image data to test IS^{2}DP+S^{2}DP. These optical image data come from some publicly available databases, and the detailed data configuration is shown in Table 8. The images of cats and dogs are from the database of the Kaggle competition platform [33]; the images of panda are from the ImageNet database [34]; the images of airplanes, motorbike, and faces are from the caltech101 database [35].

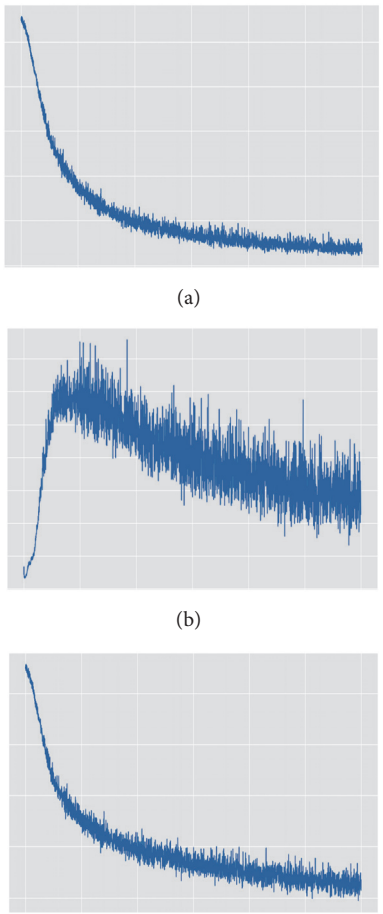
In Table 8, we set more stringent conditions than EOC_2 for SAR images, which is closer to the reality. Specifically, our interested targets are cats, dogs and panda. However, our learning set contains not only unlabeled interested targets,

Table 6: Under EOC_2, when the labeled samples' number is 21, the recognition results (confusion matrix) of the learning set by the trained ladder network.

	T72	BMP2	BTR70	Accuracy of each type of target(%)
T72	166	10	4	87.37
BMP2	12	178	0	93.68
BTR70	2	12	176	92.63
SLICY	67	53	70	0
Overall accuracy				68.42

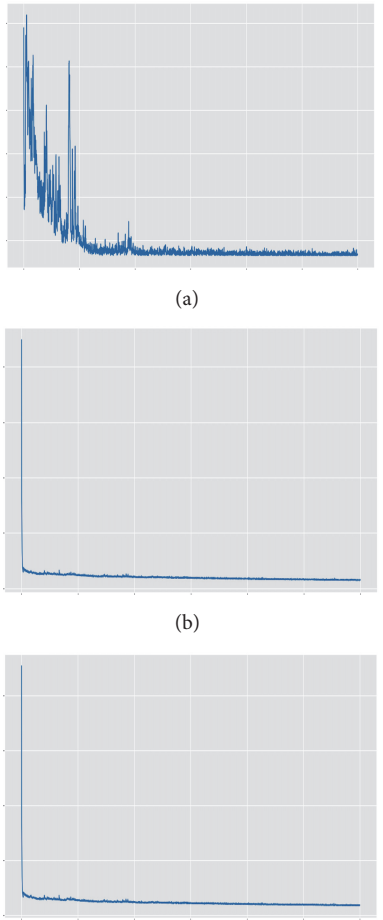
Table 7: Under EOC_2, when labeled samples number 21, recognition results of the learning set by IS^{2}DP+S^{2}DP after 300 iterations.

Target	Learning set	Generate labeled samples		Reject	Accuracy of each type of target(%)
		Correct	Error		
T72	190	155	12	23	81.58
BMP2	190	162	17	11	85.26
BTR70	190	147	22	21	77.37
SLICY	190	—	25	165	86.84
Overall accuracy					82.76



(a) (b) (c)

Figure 19: Under EOC_2, when labeled samples number 21, Temporal Ensembling losses during training: (a) supervised learning loss; (b)unsupervised learning loss; (c) overall loss.



(a) (b) (c)

Figure 20: Under EOC_2, when labeled samples number 21, Ladder Network losses during training: (a) supervised learning loss; (b) unsupervised method loss; (c) overall loss.

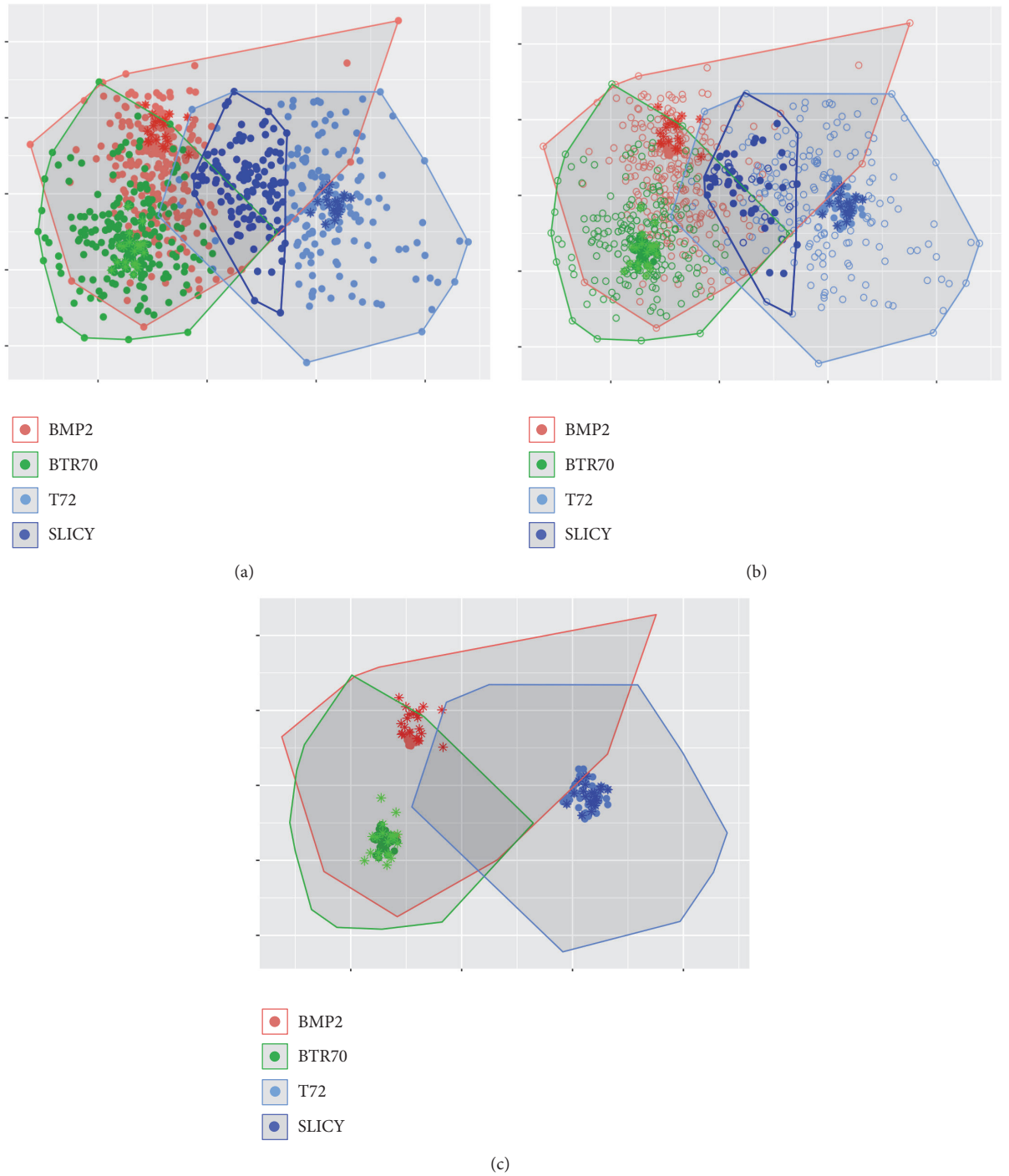


Figure 21: Under EOC_2, when the labeled samples number is 21, IS^{2}DP+S^{2}DP's outcomes of the learning set: (a) WKFDA's projection of the learning set; (b) DP clustering result; (c) reliable samples selected and labeled by LCI. Red circle, green circle, light blue, and blue circle represent BMP2, BTR70, T72, and SLICY target samples, respectively, and $*$ represents labeled sample.

but also other 3 types of interference targets (airplanes, motorbike, and faces) with the same number of unlabeled interested targets. Under such conditions, the Ladder Network, Temporal Ensembling, and IS^{2}DP+S^{2}DP are tested and compared. With the change of the number of the labeled samples, the OA trend chart of three methods is obtained, as shown in Figure 22. The horizontal axis represents the number of each type of target labeled samples corresponding to different experiments; the vertical axis represents the overall accuracy rate.

In Figure 22, the identification accuracy of our method IS^{2}DP+S^{2}DP is significantly better than Ladder Network and

Table 8: Optical image data set-up of the EOC.

		Data set						Testing set		
		Training set (labeled samples)			Learning set (unlabeled samples)					
Interested	Target	cats	dogs	panda	cats	dogs	panda	cats	dogs	panda
	Quantity	3~40	3~40	3~40	200	200	200	200	200	200
Interferential	Target	—	—	—	airplanes	motorbike	faces	—	—	—
	Quantity	—	—	—	200	200	200	—	—	—

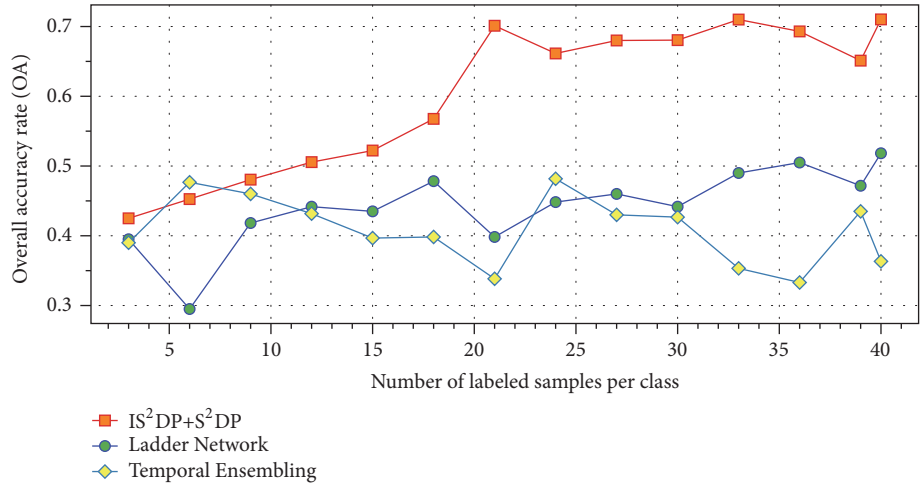


Figure 22: The OA trend charts of the IS^{2}DP+S^{2}DP, Ladder Network, and Temporal Ensembling with the number of the labeled samples changes in the optical images testing experiment.

Temporal Ensembling. From the OA trend, the recognition accuracy of Ladder Network and Temporal Ensembling does not increase significantly with the increase of the number of the labeled samples, while IS^{2}DP+S^{2}DP is significantly improved. Compared with the results of the SAR image test (Figure 18), the results of the three algorithms in the optical image test are significantly lower. This may be because in the learning set, we both increase the numbers of the target types and the number of the confusion targets, which leads to the less satisfactory results in learning the target features. From Figure 22, the Ladder Network and Temporal Ensembling algorithms are subject to more serious interference, and their average recognition accuracy is about 45%, respectively. Our algorithm IS^{2}DP+S^{2}DP is also subject to certain interference, but when the number of samples per class reaches 21, its average recognition accuracy is about 70%, which is significantly higher than the Ladder Network and Temporal Ensembling algorithms. When the number of the labeled samples is equal to 21, we will analyze the use of the learning set by the three methods.

Figure 23 shows the use of the learning set by the Ladder Network in the last 270 iterations during 1000 iterations of training. Figure 23(a) is the recognition accuracy of the learning set by Ladder Network; Figure 23(b) is the Ladder Network's loss value, where the blue line with square is the overall loss, the black line with circle is the supervised loss, and the green line with diamond is the unsupervised loss. From Figure 23(a), we know that the recognition accuracy is very low, about 33%. From Figure 23(b), we know that the supervised loss is low, while the unsupervised loss is high, which makes the overall loss difficult to reduce. Ladder Network is a complex network, which is intertwined by many components, but its core part mainly includes adding noise to samples, reconstructing samples and "skip connection" [36]. It first augments the unlabeled samples by adding noise to obtain a wider range of generalization information, secondly retains the sample information as much as possible by reconstructing the unlabeled samples in a regularized manner, and finally combines unsupervised learning with supervised learning to form semi-supervised learning by skip connection. Compared with supervised learning, unsupervised learning is more important in Ladder Network. Therefore, although Ladder Network has been well learned in the labeled samples, it has not been well learned in using the unlabeled samples, resulting in the whole algorithm has not been well trained. Finally, the Ladder Network recognition accuracy is neither stable nor high.

Figure 24 shows the use of the learning set by the Temporal Ensembling in the last 270 iterations during 1000 iterations of training. Compared with Figure 23, the recognition accuracy of Temporal Ensembling for the learning set is increased, about 48%, but it is still relatively low. Different from the Ladder Network, Temporal Ensembling adds noise to all the samples, which makes the labeled samples augmented. At the same time, in the initial stage of the training, the Temporal Ensembling's supervised learning plays an important role

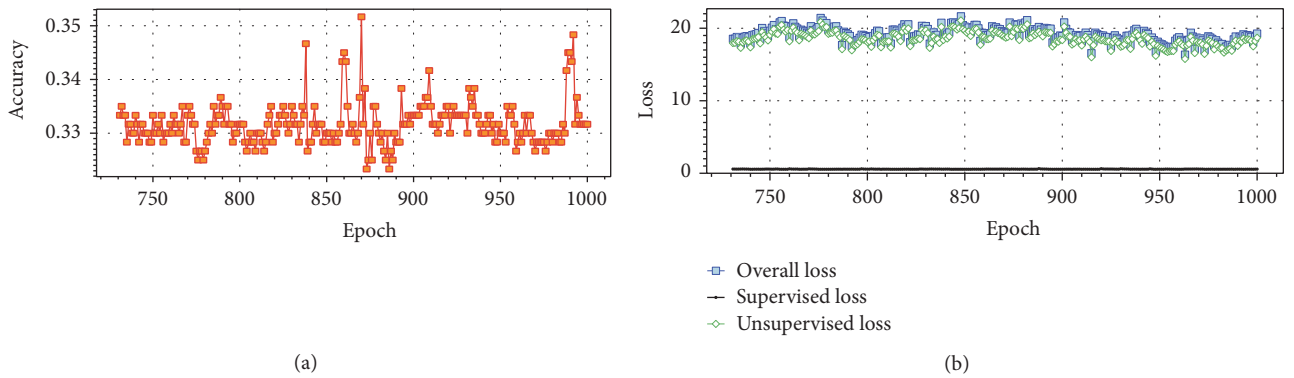


FIGURE 23: The use of the learning set by the Ladder Network in the last 270 iterations during 1000 iterations of training. (a) Recognition accuracy of learning sets and (b) Ladder Network loss value.

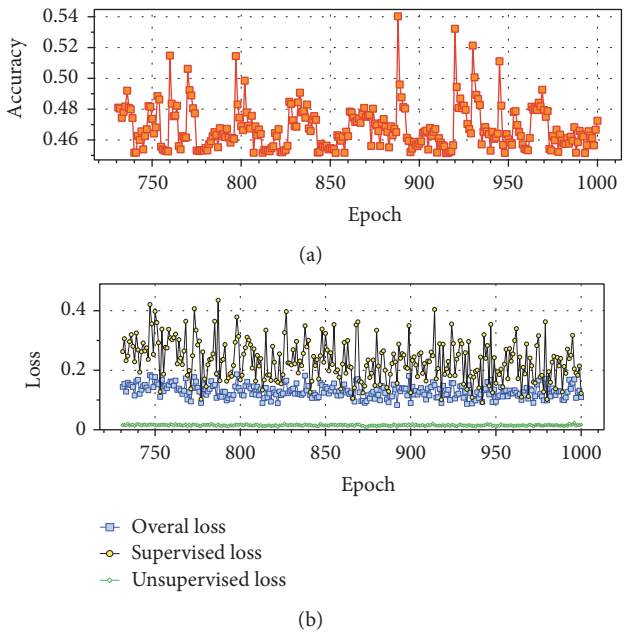


FIGURE 24: The use of the learning set by the Temporal Ensembling in the last 270 iterations during 1000 iterations of training. (a) Recognition accuracy of learning sets and (b) Temporal Ensembling loss value.

because of the small value of the unsupervised loss weighting function [8]. Therefore, the Temporal Ensembling is well trained to some extent. As the value of the loss weighting function increases, the unsupervised learning gradually plays an important role in Temporal Ensembling. Although the unsupervised loss is very low, Temporal Ensembling has not been well trained in learning interested target features because of the large number of unreliable samples in the learning set. Finally, Temporal Ensembling still has low recognition accuracy for interested targets.

Unlike the Ladder Network and the Temporal Ensembling algorithms, the IS^{2}DP+S^{2}DP algorithm first removes the interference samples in the process of using the learning set and then learns the selected reliable samples. Table 9 shows the recognition results of IS^{2}DP+S^{2}DP algorithm for learning set after 300 iterations. The average accuracy is 80%, which is significantly higher than that of Ladder Network and Temporal Ensembling algorithms. Compared with Table 7, the average accuracy of Table 9 is lower. However, the correct rate of rejection of the 3 types interference target samples has not been reduced, and these correct rates have reached more than 80%. In addition, the rejection error rate of the IS^{2}DP+S^{2}DP algorithm for the target samples is quite low; for example, cats is 18/200 = 0.09; dogs is 15/200 = 0.075; panda is 20/200 = 0.1. These experimental results show that the proposed algorithm is effective in optical image testing.

5. Conclusions

In order to accurately identify remote sensing images when there are few labeled samples, two new semi-supervised learning algorithms have been proposed in this paper: S^{2}DP and IS^{2}DP. They use labeled sample information to filter out reliable unlabeled samples to improve the performance of the semi-supervised algorithms.

The novelty of this paper lies in the following: (a) the WKFDA has been derived to explore the features of the images; (b) based on the clustering information of the DP, the labeling method LCI has been designed to query reliable unlabeled samples and accurately classify the unlabeled samples; (c) in IS^{2}DP, the unlabeled training set is divided into different subsets, which suppresses the deterioration of the algorithm by too many unreliable unlabeled samples in the learning process. Moreover, IS^{2}DP uses S^{3}VM twice to ensure reliable semi-supervised samples.

In the experiments for the actual SAR images recognition from the MSTAR database, the S^{2}DP has made a significant improvement in terms of the classification accuracy and the stability in comparison with other existing methods. In addition, the IS^{2}DP is effective and has applicable values to query the semi-labeled samples and is more suitable to deal with the situation where it lacks labeled samples.

How to make full use of remote sensing images to improve the performance of recognition algorithm has always been an open problem. Although the semi-supervised deep learning

Table 9: Recognition results of optical image learning set obtained after 300 iterations of the IS^{2}DP+S^{2}DP algorithm when the number of samples equals 21.

Target	Learning Set	Generate labeled samples		Reject	Accuracy of each type of target(%)
		Correct	Error		
cats	200	143	39	18	71.50
dogs	200	154	31	15	77.00
panda	200	165	15	20	82.50
airplanes	200	—	32	168	84.00
motorbike	200	—	40	160	80.00
faces	200	—	27	173	86.50
Overall accuracy					80.00

algorithm is susceptible to interfering samples, it has strong feature learning capabilities once the interfering samples have been removed. In the near future, we will try to further improve the feature learning ability of the S^{2}DP and IS^{2}DP algorithms by virtue of the semi-supervised deep learning.

Abbreviations

The following abbreviations are used in this manuscript:

DP:	Clustering by fast search and find of density peaks
EOC:	Extended operating conditions
IS^{2}DP:	Iterative S^{2}DP
KFDA:	Kernel Fisher discriminant analysis
KLFDA:	Kernel local Fisher discriminant analysis
KPCA:	Kernel principal component analysis
LCI:	Labeling method based on the DP clustering information
MSTAR:	Moving and Stationary Target Acquisition and Recognition database
OA:	Overall accuracy rate
PS^{3}VM-D:	Progressive semi-supervised SVM with diversity
SAR:	Synthetic aperture radar
Semi-KLFDA:	Semi-semisupervised KLFDA
S^{2}DP:	Semi-supervised learning method based on DP
SOC:	Standard operating conditions
SVM:	Support vector machine
S^{3}VM:	Semi-supervised SVM
S^{4}VM:	Safe S^{3}VM
WKFDA:	Weighted Kernel Fisher discriminant analysis.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61771027; 61071139; 61471019; 61171122; 61501011; 61671035), the Scientific Research Foundation of Guangxi Education Department (KY2015LX444), the Scientific Research and Technology Development Project of Wuzhou, Guangxi, China (201402205), the Guangxi Science and Technology Project (Guike AB16380273), and the Research and Practice on Teaching Reform of Web Page Making and Design Based on the Platform of "E-Commerce Pioneer Park" (Guijiao Zhicheng [2014]41). Professor A. Hussain was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) Grant no. EP/M026981/1. E. Yang was supported in part under the RSE-NNSFC Joint Project (2017-2019), grant number 6161101383, with China University of Petroleum (Huadong). H. Zhou was supported by UK EPSRC under Grant EP/N011074/1, Royal Society-Newton Advanced Fellowship under Grant NA160342, and European Union's Horizon 2020 research and innovation program under the Marie-Sklodowska-Curie Grant agreement no. 720325.

References

[1] M. Kang, K. Ji, X. Leng, X. Xing, and H. Zou, "Synthetic aperture radar target recognition with feature fusion based on a stacked autoencoder," *Sensors*, vol. 17, no. 1, p. 192, 2017.

[2] F. Gao, T. Huang, J. Sun, J. Wang, A. Hussain, and E. Yang, "A New Algorithm of SAR Image Target Recognition Based on," *Cognitive Computation*, pp. 1–16, 2018.

[3] F. Zhang, X. Yao, H. Tang, Q. Yin, Y. Hu, and B. Lei, "Multiple Mode SAR Raw Data Simulation and Parallel Acceleration for Gaofen-3 Mission," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–12, 2018.

[4] A. Verma, H. Qassim, and D. Feinzimer, "Residual squeeze CNDS deep learning CNN model for very large scale places image recognition," in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 463–469, New York, NY, USA, October 2017.

[5] A. Popella, G. Eiben, W. Geile, and S. Meltzer, "Knowledge-Based Interpretation of SAR Imagery," in *International Archives of Photogrammetry and Remote Sensing*, vol. 29, pp. 52-52, 1993.

[6] F. W. Rohde, P. F. Chen, and R. A. Hevenor, "Automated radar image analysis research in support of military needs,"

Automated Radar Image Analysis Research in Support of Military Needs, p. 87, 1986.

[7] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko, "Semi-Supervised Learning with Ladder Networks," *Computer Science*, vol. 9, supplement 1, p. 1, 2015.

[8] S. Laine and T. Aila, "Temporal Ensembling for Semi-Supervised Learning," 2016, https://arxiv.org/abs/1610.02242.

[9] G. Wang, S. Tan, C. Guan, N. Wang, and Z. Liu, "Multiple model particle filter track-before-detect for range ambiguous radar," *Chinese Journal of Aeronautics*, vol. 26, no. 6, pp. 1477–1487, 2013.

[10] Z. Huang, Z. Pan, and B. Lei, "Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data," *Remote Sensing*, vol. 9, no. 9, p. 907, 2017.

[11] T. Yang and C. E. Priebe, "The effect of model misspecification on semi-supervised classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2093–2103, 2011.

[12] H. Gan, Z. Li, Y. Fan, and Z. Luo, "Dual Learning-Based Safe Semi-Supervised Learning," *IEEE Access*, vol. 6, pp. 2615–2621, 2017.

[13] Y.-F. Li and Z.-H. Zhou, "Improving semi-supervised support vector machines through unlabeled instances selection," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence and the 23rd Innovative Applications of Artificial Intelligence Conference, AAAI-11 / IAAI-11*, pp. 386–391, USA, August 2011.

[14] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Proceedings of the 12th Annual Conference on Neural Information Processing Systems, NIPS 1998*, pp. 368–374, USA, December 1998.

[15] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[16] Y.-F. Li and Z.-H. Zhou, "Towards making unlabeled data never hurt," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pp. 1081–1088, USA, July 2011.

[17] Y. Wang and S. Chen, "Safety-aware semi-supervised classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1763–1772, 2013.

[18] H. Gan, Z. Luo, Y. Sun, X. Xi, N. Sang, and R. Huang, "Towards designing risk-based safe Laplacian Regularized Least Squares," *Expert Systems with Applications*, vol. 45, pp. 1–7, 2016.

[19] C. Persello and L. Bruzzone, "Active and semisupervised learning for the classification of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6937–6956, 2014.

[20] D. Wang, F. Nie, and H. Huang, "Large-scale adaptive semi-supervised learning via unified inductive and transductive model," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014*, pp. 482–491, USA, August 2014.

[21] N. Sokolovska, O. Cappé, and F. Yvon, "The asymptotics of semi-supervised learning in discriminative probabilistic models," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 984–991, New York, NY, USA, July 2008.

[22] M. Kawakita and J. Takeuchi, "Safe semi-supervised learning based on weighted likelihood," *Neural Networks*, vol. 53, pp. 146–164, 2014.

[23] Y. M. Zhang, Y. Zhang, D. Y. Yeung, C. L. Liu, and X. Hou, "Transductive Learning on Adaptive Graphs," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, Georgia, Atlanta , USA, July 2010.

[24] A. Laio and A. Rodriguez, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[25] L. Kaufman and P. J. Rousseeuw, *Clustering Large Applications (Program CLARA)*, John Wiley & Sons, 2008.

[26] A. Jatram and B. Biswas, "Dimension reduction using spectral methods in FANNY for fuzzy clustering of graphs," in *Proceedings of the 8th International Conference on Contemporary Computing, IC3 2015*, pp. 93–96, India, August 2015.

[27] T. Wen, J. Yan, D. Huang et al., "Feature extraction of electronic nose signals using QPSO-based multiple KFDA signal processing," *Sensors*, vol. 18, no. 2, p. 388, 2018.

[28] J.-M. Yin, M. Yang, and J.-W. Wan, "A kernel fisher linear discriminant analysis approach aiming at imbalanced data set," *Moshi Shibie yu Rengong Zhineng/Pattern Recognition and Artificial Intelligence*, vol. 23, no. 3, pp. 414–420, 2010.

[29] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Muller, "Fisher discriminant analysis with kernels," in *Proceedings of the 9th IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing (NNSP '99)*, pp. 41–48, Madison, Wis, USA, August 1999.

[30] Z. Shi and J. Hu, "A kernel approach to implementation of local linear discriminant analysis for face recognition," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 12, no. 1, pp. 62–70, 2017.

[31] M. Sugiyama, T. Ide, S. Nakajima, and J. Sese, "Semi-supervised local Fisher discriminant analysis for dimensionality reduction," *Machine Learning*, vol. 78, no. 1-2, pp. 35–61, 2010.

[32] Q. Wang, *Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models*, Computer Science, 2012.

[33] L. Smith and Y. Gal, "Understanding Measures of Uncertainty for Adversarial Example Detection," 2018, https://arxiv.org/abs/1803.08533.

[34] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[35] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[36] M. Pezeshki, L. Fan, P. Brakel, A. Courville, and Y. Bengio, "Deconstructing the ladder network architecture," in *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, pp. 3527–3539, USA, June 2016.

Hindawi
Complexity
Volume 2019, Article ID 2095063, 8 pages
https://doi.org/10.1155/2019/2095063

Research Article

LMC and SDL Complexity Measures: A Tool to Explore Time Series

José Roberto C. Piqueira [1] **and Sérgio Henrique Vannucchi Leme de Mattos** [2]

[1] *Escola Politécnica da Universidade de São Paulo, Avenida Prof. Luciano Gualberto, travessa 3, n. 158, 05508-900 São Paulo, SP, Brazil*
[2] *Universidade Federal de São Carlos, Rod. Washington Luís km 235, SP-310, 13565-905 São Carlos, SP, Brazil*

Correspondence should be addressed to José Roberto C. Piqueira; piqueira@lac.usp.br

Received 21 September 2018; Revised 24 November 2018; Accepted 10 December 2018; Published 2 January 2019

Guest Editor: Jose Garcia-Rodriguez

This work is a generalization of the López-Ruiz, Mancini, and Calbet (LMC) and Shiner, Davison, and Landsberg (SDL) complexity measures, considering that the state of a system or process is represented by a continuous temporal series of a dynamical variable. As the two complexity measures are based on the calculation of informational entropy, an equivalent information source is defined by using partitions of the dynamical variable range. During the time intervals, the information associated with the measured dynamical variable is the seed to calculate instantaneous LMC and SDL measures. To show how the methodology works generating indicators, two examples, one concerning meteorological data and the other concerning economic data, are presented and discussed.

1. Introduction

The word complexity, in the common sense meaning, represents systems that are difficult to describe, design, or understand. However, since Kolmogorov presented the concept of computational complexity [1], new ideas have been associated with this word, mainly in life sciences [2], relating complexity, and information [3].

As a consequence, complexity started to be associated to with systems and with the emergence of unexpected behaviors, due to nonlinearities [4, 5] and, concerning system theory [6], a new meaning was carved, postulating that complexity is half way of the equilibrium and disequilibrium [7].

Developing this idea, in a seminal paper [8], López-Ruiz, Mancini, and Calbet proposed the LMC (López-Ruiz, Mancini, and Calbet) complexity measure for a random distribution by using informational entropy [9] to evaluate equilibrium, and the quadratic deviation from the uniform distribution to evaluate disequilibrium.

However, there has been some criticism about the LMC measure, considering that it is inaccurate for some classes of systems obeying Markovian chains and cannot be considered to represent an extensive variable. Feldman and Crutchfield [10] proposed a correction for the disequilibrium term, replacing it by the relative entropy with respect to the uniform distribution.

Shiner, Davison, and Landsberg proposed another modification of the LMC measure, replacing the disequilibrium term by the complement of the equilibrium term. This measure is called SDL (Shiner, Davison, and Landsberg) [11] and presents conclusions similar to that obtained by using LMC, for the majority of usual statistical distributions [2].

The main restriction to LMC and SDL complexity measures is due to Crutchfield, Feldman, and Shalizi, as they argue that an equilibrium system can be structurally complex [12], but this problem could be solved by weighting order and disorder, according to the specific problem to be analyzed.

Since the early 2000s, the idea of adapting LMC and SDL to dynamical systems was successfully applied to different types of time evolution problems: bird songs [13], neural plasticity [14], interactions between species in ecological systems [2], physiognomies of landscapes [15], economic series [16], spread depression [17], and quantum information [18].

With these ideas in mind, this article presents a systematization of the methodology used in the referred papers,

based on LMC and SDL measures, to be applied to temporal series, by defining and calculating the dynamic complexity measures.

The procedure, applied to a temporal series representing some organizational or functional aspect of a system, provides insights regarding the evolution of its complexity.

As the LMC and SDL dynamical measures are based on informational entropy [16], the first task, described in the next section, is to define an alphabet source, associating a probability distribution with the possible system states.

Following the definition of the probability distribution, a new section defines how dynamical LMC and SDL measures can be calculated at each time, based on the individual information associated with the system state at this time, generating temporal series for LMC and SDL measures.

To illustrate the calculation procedure, two examples are presented: one related to a meteorological time series and the other to an economic time series. In both cases section, a practical discussion about how to divide the range of the values assumed by the system state is presented.

The examples were chosen to show that the methodology can be applied to different types of phenomena: precipitation (first example) with strong periodic component and economic time series (second example) that seems to be random.

The work is closed with a conclusion section, emphasizing that the same procedure can be applied to any kind of temporal real numbers series, even with different temporal scale, to calculate complexity measures.

2. Defining Source and Probability Distribution for a Temporal Series

Considering Shannon's model [9] for an information source, a time series $x(n)$ is considered to be a function of the nonnegative integers into a real interval, i.e., $x(n) : Z_+ \longrightarrow (a, b)$, associating with each time $t_0 + nT$ a real number belonging to (a, b), with $t_0 > 0$ being the initial instant and $T > 0$ an arbitrary period, depending on the data availability.

The set $x(t_0), x(t_0 + T), \ldots x(t_0 + nT)$ is assumed to be a sequence of independent random variables and the stochastic process $x(n)$ as a whole is stationary [19].

The first step is to divide the interval (a, b) into N subintervals. For the sake of simplicity, N is chosen equal to $2^k, k \in Z_+$.

At this point, it could be asked how to choose N, as there is a compromise between precision (high values of N) and speed of calculation (low values of N). This question will not be addressed theoretically; however, in the example section, practical hints about this choice are presented.

Consequently, the source alphabet is defined by the intervals $A_i, i = 1, \ldots, N$, with $\bigcup_{i=1}^{N} A_i = (a, b)$ and $A_i \cap A_j = \phi, \ \forall i \neq j$.

Then, a time interval defined by a given n must be chosen, and for the time sequence $t_0, t_0 + T, \ldots t_0 + nT$ the values of the variable $x(n)$ must be read and associated with the intervals A_i, containing their respective value.

Therefore, for the whole set $t_0, t_0 + T, \ldots t_0 + nT$, each interval A_i belonging to the source alphabet is associated with $x(n)$ a certain number of times n_i, which defines a relative frequency $p_i = n_i/(n + 1)$.

As $p_i \geq 0$ and $\sum_{i=1}^{N} p_i = 1$, it can be taken as a probability, associated with each interval A_i.

Following the definition, for each subinterval $A_i \subset (a, b)$, its individual contribution to the whole information entropy is given by $S_i = -p_i \log_2 p_i$; and the maximum value of the informational entropy for the whole source, $S_{max} = \log_2 N = k$, can be calculated [9].

3. Dynamical LMC and SDL

As the source alphabet and individual information were defined, the instantaneous values of $x(n)$ are associated with their respective S_i, allowing the calculation of the instantaneous value of the equilibrium (disorder) term:

$$\Delta(n) = \frac{S_i}{k}. \tag{1}$$

Combining (1) with the different definitions of the disequilibrium (order) terms, dynamical LMC and SDL measures are defined.

3.1. LMC Dynamical Measure. As indicated by López-Ruiz, Mancini, and Calbet [8], the dynamic disequilibrium (order) term can be calculated as the quadratic deviation of the source alphabet probability distribution from the uniform distribution and, consequently, the individual contribution of each interval A_i is

$$D(n) = \left(p_i - \frac{1}{N}\right)^2. \tag{2}$$

Extending the definition of LMC measure, dynamical LMC, calculated in $t_0 + nT$, is given by

$$C_{LMC}(n) = \Delta(n).D(n). \tag{3}$$

3.2. SDL Dynamical Measure. As proposed by Shiner, Davison, and Landsberg [11], the dynamic disequilibrium (order) term can be calculated as the complement of the dynamic equilibrium term:

$$D(n) = (1 - \Delta(n)). \tag{4}$$

Extending the definition of SDL measure, dynamical SDL, calculated in $t_0 + nT$, is given by

$$C_{SDL}(n) = \Delta(n).D(n). \tag{5}$$

4. Applying the Method to Meteorological Data

A monthly meteorological temporal series is studied in this section, showing that the described method can be applied, independently of the natural time scale and periodicity of the phenomenon.

The meteorological data series relative to rain precipitation in Dourados-MS-Brazil [20] is analyzed, only in a

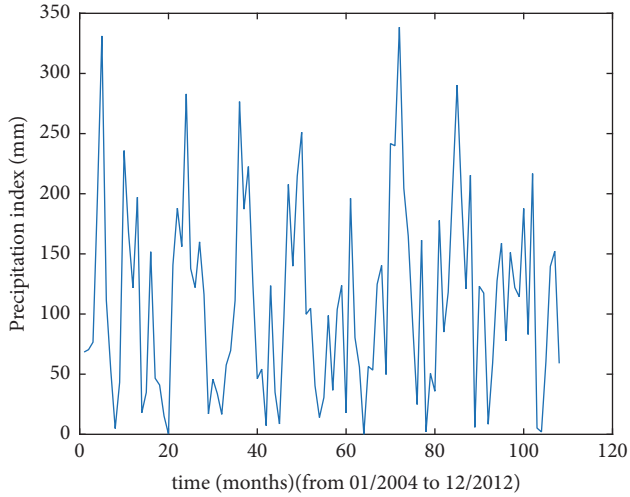


Figure 1: Precipitation index (Dourados-MS-Brasil).

methodological point of view, without any meteorological conjecture about the results.

The monthly precipitation index temporal series, from January 2004 to September of 2012, shown in Figure 1, represents the value of $x(n)$ [20], whose complexity is analyzed.

Consequently, the interval (a, b) related to the excursion of x is $(0; 338)$. It is divided into $8(k = 3)$, $16(k = 4)$, $32(k = 5)$, and $64(k = 6)$ subintervals to build the sources and their respective probability distributions.

Based on these probability distributions, $C_{LMC}(n)$ and $C_{SDL}(n)$ are calculated and plotted giving an idea about how the measure choice and the interval division affect the results.

4.1. Equivalence between LMC and SDL. Dividing the range of $x(n)$ into 8 parts, the results of the calculation of $C_{LMC}(n)$ and $C_{SDL}(n)$ measures are shown in Figures 2(a) and 2(b), respectively.

As Figures 2(a) and 2(b) show, in spite of the numerical differences, the time evolutions of $C_{LMC}(n)$ and $C_{SDL}(n)$ are represented by similar curves, in the eight-part division case.

Observing the figures, it is possible to infer that the LMC measure captures the periodic character of the precipitation along the years in a better way. However, the SDL measure assumes its maximum value (.25).

If the range of $x(n)$ is divided into 16 parts, Figures 3(a) and 3(b) show the results for $C_{LMC}(n)$ and $C_{SDL}(n)$.

It can be observed that, in this case (sixteen-division case), $C_{LMC}(n)$ and $C_{SDL}(n)$ differ by a scale factor, with LMC measures presenting better accuracy to express the periodic character of the rain seasons. SDL measure presents high value of peaks but the maximum value (.25) is not reached.

Comparing Figures 2(a) and 3(a), $C_{LMC}(n)$ for different range partitions, the global aspects of the curves are the same and, by increasing the number of divisions, the dynamical range of the measures decreases, and some rapid oscillatory variations similar to noise appear.

Comparing Figures 2(b) and 3(b), $C_{SDL}(n)$ for different range partitions, the whole aspects of the curves are the same and the noisy aspect due to the increasing number of interval divisions is similar to the presented by $C_{LMC}(n)$.

4.2. Range Interval Partition. As it was observed, the dynamical range of the both measures decreases as the number of divisions increases, as a consequence of the fact that the number of elements of the source increases provoking more uniform distribution of the possible measures.

To better understand this phenomenon, the measures are recalculated by increasing the number of intervals of $x(n)$, and the result for a thirty-two partition is shown in Figure 4(a) for $C_{LMC}(n)$ and in Figure 4(b) for $C_{SDL}(n)$.

By analyzing the results from Figures 2(a), 3(a), and 4(a), it could be observed that, by increasing the number of intervals, the dynamical range of $C_{LMC}(n)$ decreases but, apparently, for this long series, the temporal evolution of $C_{LMC}(n)$ maintains its qualitative behavior mixing noise with accuracy.

By analyzing the results from Figures 2(b), 3(b), and 4(b), it could be observed that, by increasing the number of intervals, the dynamical range of $C_{SDL}(n)$ decreases and its maximum value (.25) is not reached. Apparently, for this long series, the temporal evolution of $C_{SDL}(n)$ maintains its qualitative behavior mixing noise with accuracy.

5. Applying the Method to Economic Data

In this section, the economic series relative to the conversion of currencies studied in [16] is taken as an example, showing the applicability of the methodology for random phenomena.

The temporal series related to the daily dollar to Brazilian real (USD/BR) conversion rate, from January 1999 to September of 2015, shown in Figure 5 [16] is analyzed, only in a methodological point of view, without any economic conjecture about the results.

This conversion rate represents the value of $x(n)$, whose complexity is analyzed.

Consequently, the interval (a, b) related to the excursion of x is $(1.207; 4.178)$. It is divided into $8(k = 3)$, $16(k = 4)$, $32(k = 5)$, and $64(k = 6)$ subintervals to build the sources and the respective probability distributions.

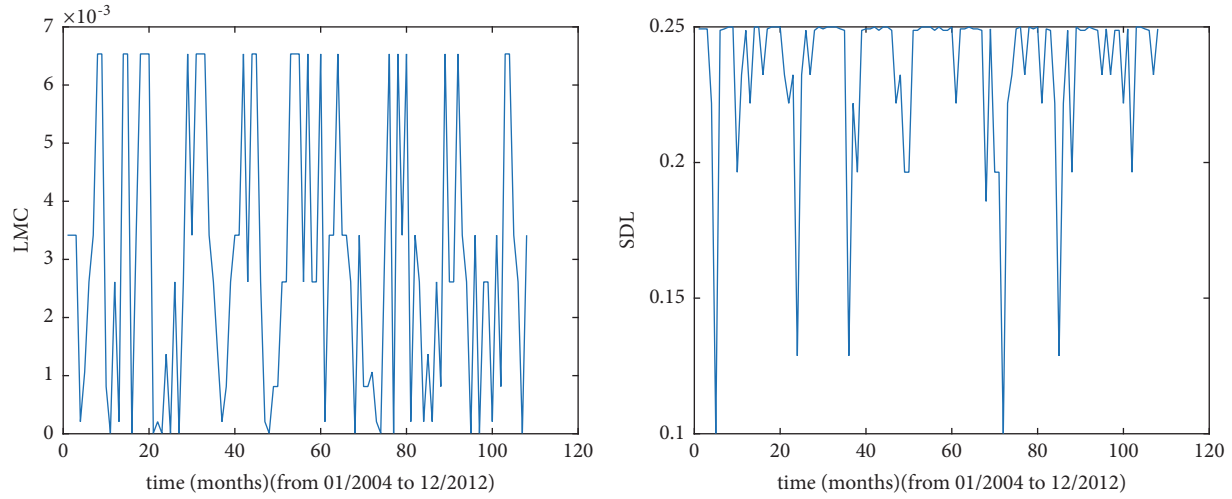
Based on these probability distributions, $C_{LMC}(n)$ and $C_{SDL}(n)$ are calculated and plotted giving an idea about how the measure choice and the interval division affect the results.

5.1. Equivalence between LMC and SDL. Dividing the range of $x(n)$ into 8 parts, the results of the calculation of $C_{LMC}(n)$ and $C_{SDL}(n)$ measures are shown in Figures 6(a) and 6(b), respectively.

As Figures 6(a) and 6(b) show, in spite of the numerical differences, the time evolution of $C_{LMC}(n)$ and $C_{SDL}(n)$ are qualitatively the same and represented by very similar curves, in the eight-part division case.

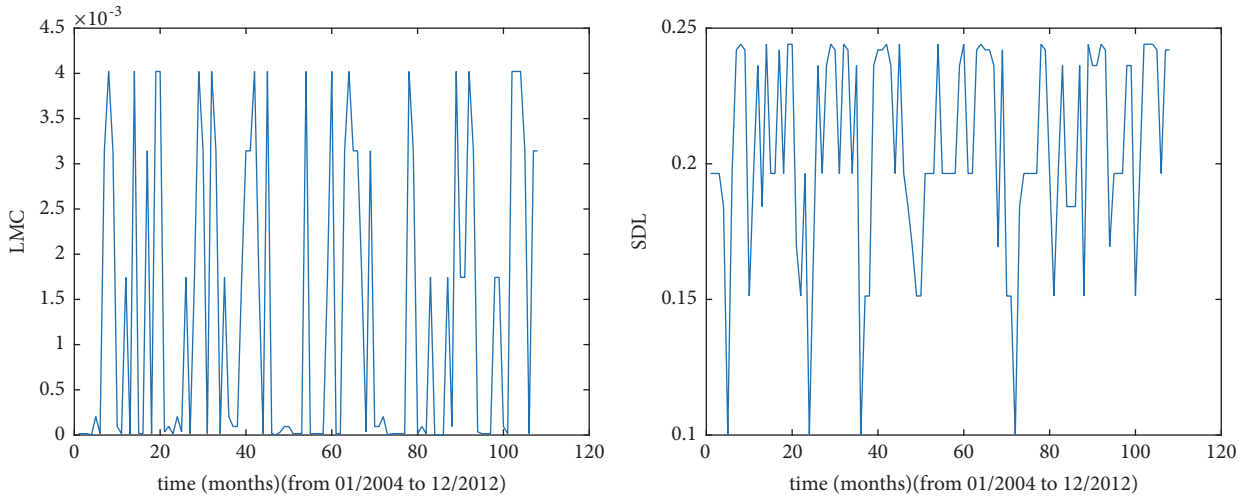
If the range of $x(n)$ is divided into 16 parts, Figures 7(a) and 7(b) show the results for $C_{LMC}(n)$ and $C_{SDL}(n)$.

It can be observed that, in this case (sixteen-division case), $C_{LMC}(n)$ and $C_{SDL}(n)$ differ only by a scale factor, with the same qualitative time evolution.



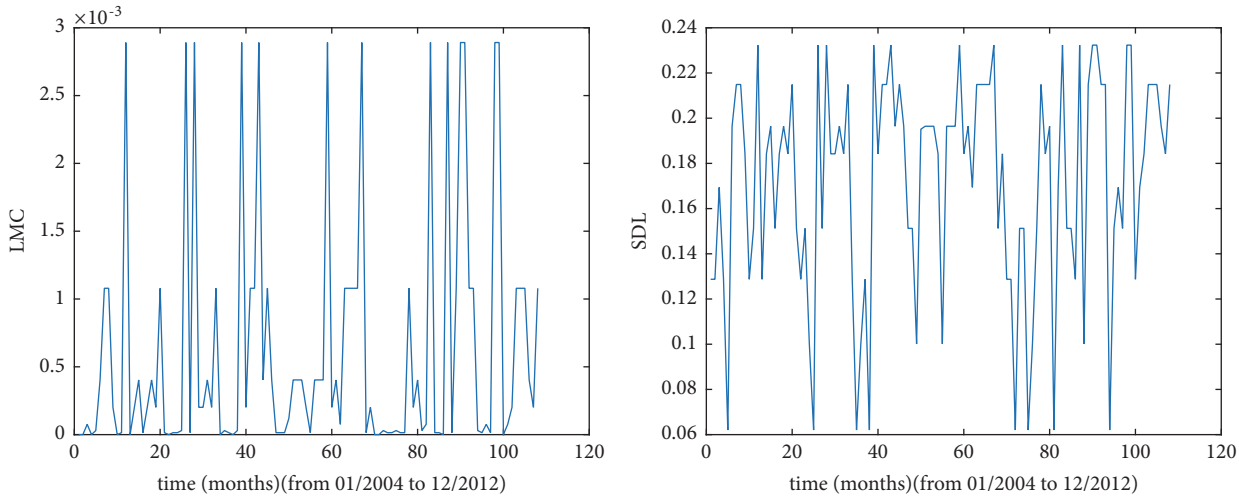
(a) LMC (b) SDL

Figure 2: Temporal evolution of complexity (8-part division).



(a) LMC (b) SDL

Figure 3: Temporal evolution of complexity (16-part division).



(a) LMC (b) SDL

Figure 4: Temporal evolution of complexity (32-part division).

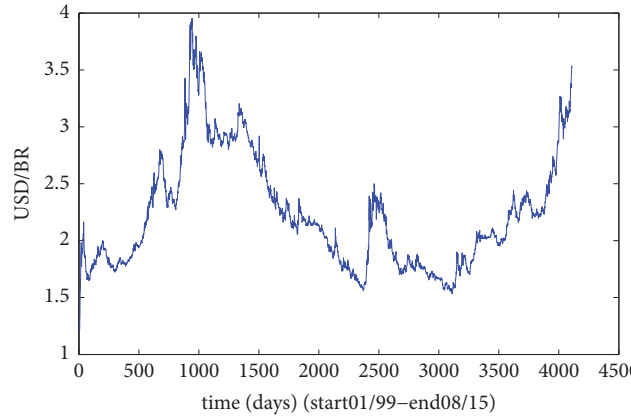
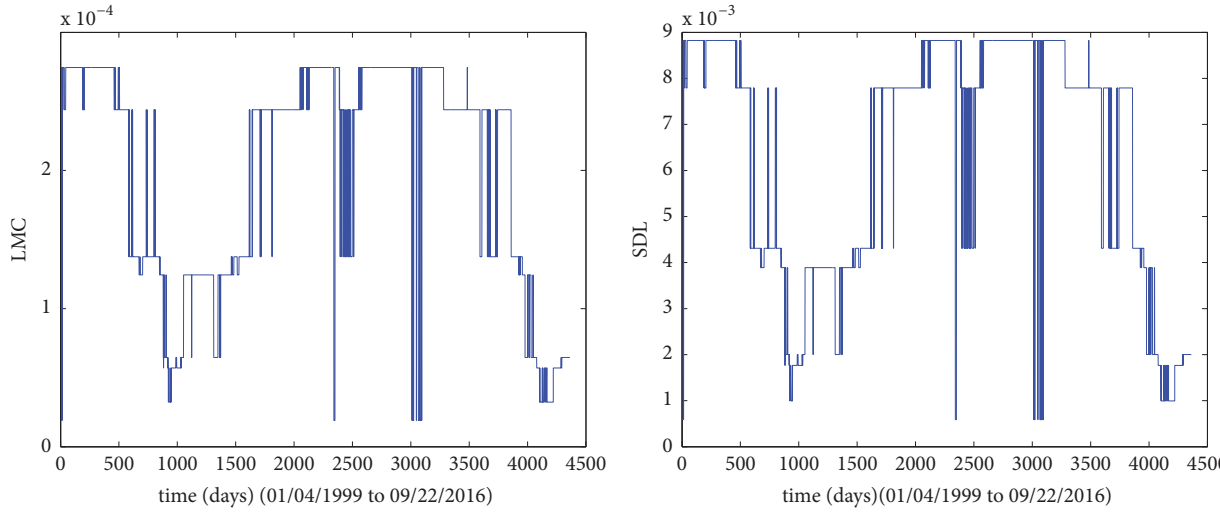
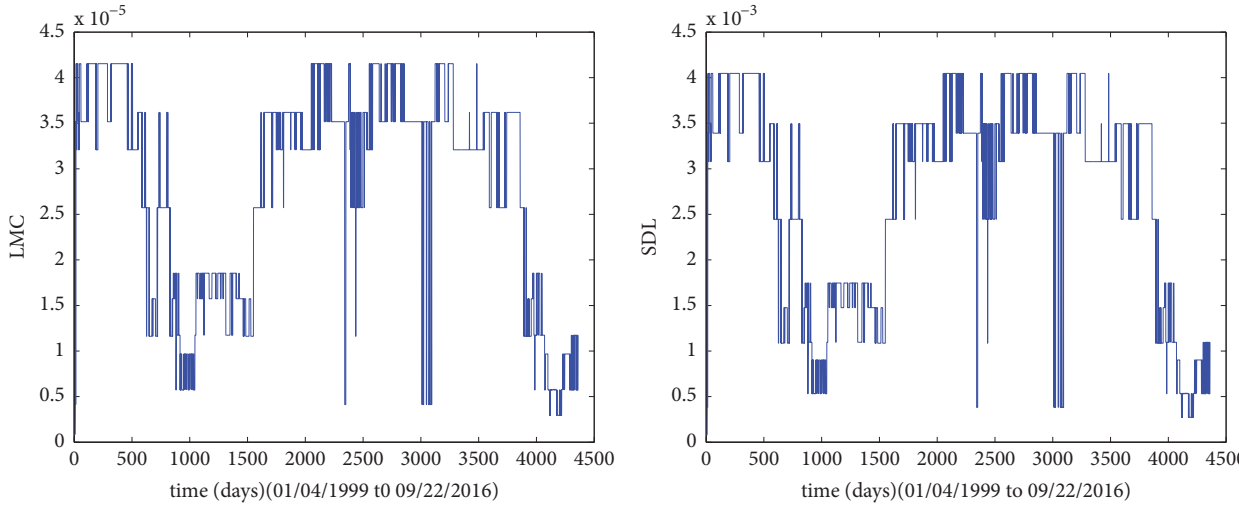


Figure 5: USA Dollar-Brazilian real exchanging rate.



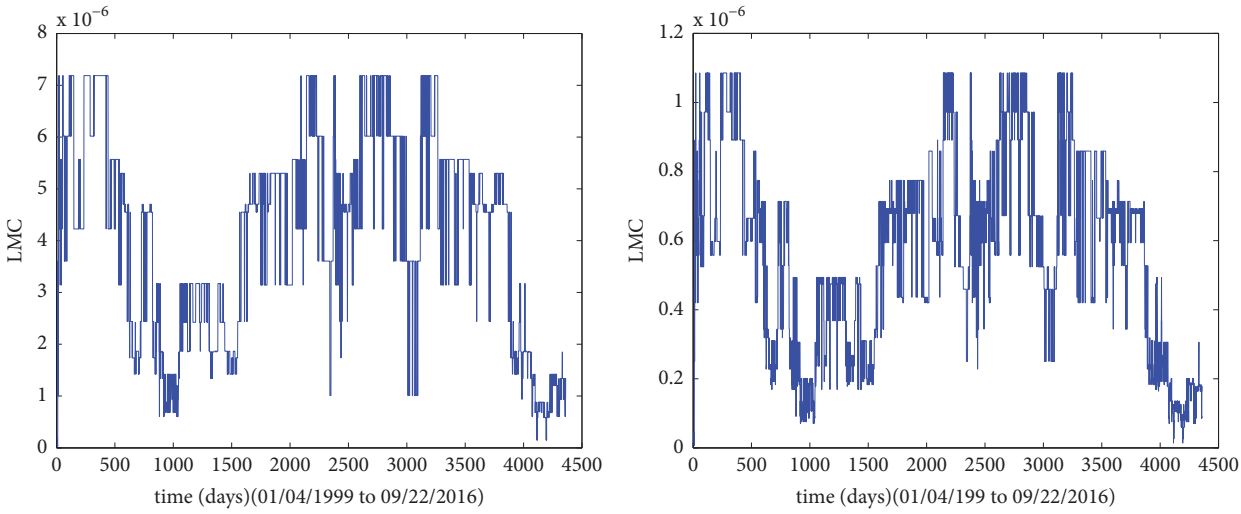
(a) LMC (b) SDL

Figure 6: Temporal evolution of complexity (8-part division).



(a) LMC (b) SDL

Figure 7: Temporal evolution of complexity (16-part division).



(a) 32-part partition

(b) 64-part partition

Figure 8: Temporal evolution of LMC complexity.

Comparing Figures 6(a) and 7(a), $C_{LMC}(n)$ for different range partitions, the whole qualitative aspects of the curves are the same and by increasing the number of divisions, the dynamical range of the measures changes, implying some rapid oscillatory variations, similar to noise.

Comparing Figures 6(b) and 7(b), $C_{SDL}(n)$ for different range partitions, the whole qualitative aspects of the curves are the same and the noisy aspect due to the increasing number of interval divisions is maintained.

Consequently, from now on, only LMC measure will be analyzed, since SDL presents the same qualitative dynamical behavior and partition sensitivity.

5.2. Range Interval Partition. By increasing the number of intervals of $x(n)$ and recalculating $C_{LMC}(n)$, the result for a thirty-two partition is shown in Figure 8(a) and, for a sixty-four partition, in Figure 8(b).

By analyzing the results from Figures 6(a), 7(a), 8(a), and 8(b), it could be observed that, by increasing the number of intervals, the maximum value of $C_{LMC}(n)$ decreases improving the precision but, apparently, for this long series, the temporal evolution of $C_{LMC}(n)$ maintains its qualitative behavior mixing noise with accuracy.

Attempting to be more precise about how the range interval partition, $C_{LMC}(n)$ is calculated for the several partitions, but considering a shorter time period for the data. The interval between July and December of 2002 is chosen, because, as explained in [16], it is critical concerning the conversion rates in Brazil.

Figures 9(a), 9(b), 9(c), and 9(d) show the LMC dynamical measure calculated, for the initial set of data, with the range interval divided into 8, 16, 32, and 64 parts, respectively.

It can be observed from these results that, for shorter intervals, the general qualitative characteristics of the time evolution appear, independently on the partition. However, as the number of subintervals increases, the instantaneous numerical values change but the precision increases, allowing more accurate analysis.

6. Conclusions

A methodology for calculating LMC and SDL dynamical complexity was developed, starting with the construction of a source and a probability distribution, for any temporal series. The contribution is just concerning to extend ideas, mainly applied to static situations, to temporal evolution of variables representing some kind of organization phenomenon.

LMC and SDL measures were observed to be equivalent in some temporal analyses but, when there is a strong oscillatory component, the LMC measure seems to be more accurate to express the temporal evolution of the complexity, as the meteorological data analysis shows.

For more randomly distributed data, the two measures (LMC and SDL) present the same accuracy, as the economic data analysis shows.

A point that is always an object of discussion is the range interval partition. The choice of the number of subintervals is a matter of experience.

Long time intervals are not so sensitive to the increase of the number of divisions, in spite of meteorological data being more sensitive than economic. However, for short time intervals, increasing the number of divisions produces a less precise analysis, introducing noise.

The examples presented were just to illustrate the methodological approach, without any compromise with meteorologic or economic conclusions that can be inferred by a specialist, by using the developed tool.

Data Availability

All the data used in this paper are available in the Internet by following the links given in the references.

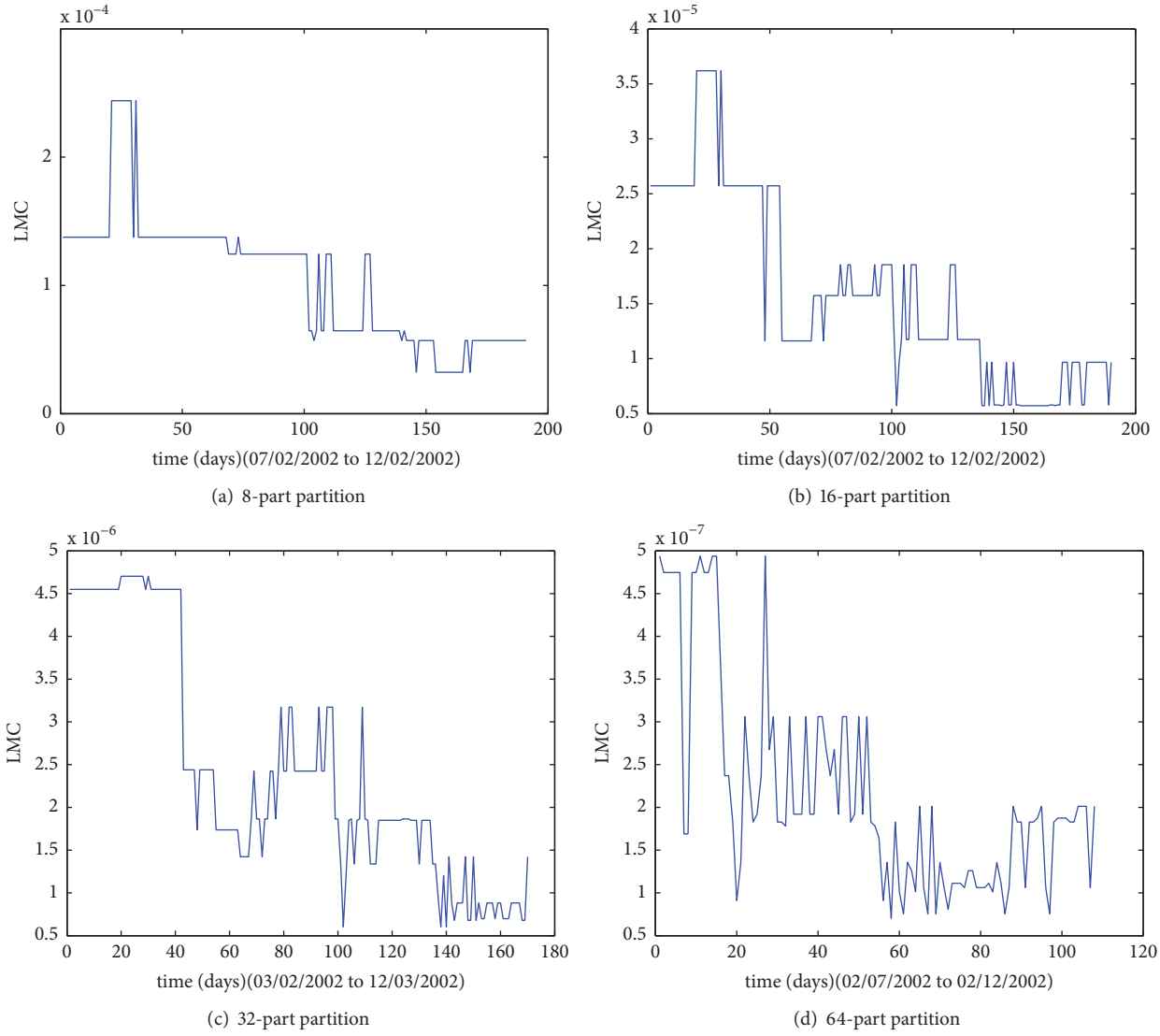


Figure 9: LMC temporal evolution for shorter time intervals.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

All the data used in this paper are available in the Internet by following the links given in the references. This work was supported by CNPq, Brazil.

References

[1] A. N. Kolmogorov, "Three approaches to the definition of the concept "quantity of information"," *Akademiya Nauk SSSR. Institut Problem Peredachi Informatsii Akademii Nauk SSSR. Problemy Peredachi Informatsii*, vol. 1, no. vyp. 1, pp. 3–11, 1965.

[2] M. Anand and L. Orlóci, "Complexity in plant communities: The notion and quantification," *Journal of Theoretical Biology*, vol. 179, no. 2, pp. 179–186, 1996.

[3] H. Haken, *Information and Self-Organization*, Springer Series in Synergetics, Springer-Verlag, Berlin, Second edition, 2000.

[4] E. Morin, *On Complexity*, Hampton Press, New York, NY, USA, 2008.

[5] G. Nicolis and I. Prigogine, *Self-Organization in Nonequilibrium Systems*, John Wiley & Sons, USA, 1977.

[6] L. Von Bertalanffy, *General System Theory: Foundations, Development, Applications*, George Braziller Inc., New York, NY, USA, 1968.

[7] K. Kaneko and I. Tsuda, *Complex Systems: chaos and beyond*, Springer Verlag: Berlin, Germany, 2001.

[8] R. López-Ruiz, H. L. Mancini, and X. Calbet, "A statistical measure of complexity," *Physics Letters A*, vol. 209, no. 5-6, pp. 321–326, 1995.

[9] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, llini Books Edition, Chicago, USA, 1993.

[10] D. P. Feldman and J. P. Crutchfield, "Measures of statistical complexity: Why?" *Physics Letters A*, vol. 238, no. 4-5, pp. 244–252, 1998.

[11] J. S. Shiner, M. Davison, and P. T. Landsberg, "Simple measure for complexity," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 59, no. 2, pp. 1459–1464, 1999.

[12] J. P. Crutchfield, D. P. Feldman, and C. R. Shalizi, "Comment I on "Simple measure for complexity"," *Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 62, no. 2, pp. 2996-2997, 2000.

[13] M. L. Da Silva, J. R. C. Piqueira, and J. M. E. Vielliard, "Using Shannon entropy on measuring the individual variability in the Rafous-bellied thrush Turdus rufiventris vocal communication," *Journal of Theoretical Biology*, vol. 207, no. 1, pp. 57–64, 2000.

[14] M. Pinho, M. Mazza, J. R. C. Piqueira, and A. C. Roque, "Shannons entropy applied to the analysis of tonotopic reorganization in a computational model of classic conditioning," *Neurocomputing*, pp. 923–928, 2002.

[15] S. H. V. L. De Mattos, L. E. Vicente, A. P. Filho, and J. R. C. Piqueira, "Contributions of the complexity paradigm to the understanding of Cerrado's organization and dynamics," *Anais da Academia Brasileira de Ciências*, vol. 88, no. 4, pp. 2417–2427, 2016.

[16] L. P. D. Mortoza and J. R. C. Piqueira, "Measuring complexity in Brazilian economic crises," *PLOSONE*, vol. 12, no. 3, Article ID e0173280, 2017.

[17] J. R. C. Piqueira, V. M. F. De Lima, and C. M. Batistela, "Complexity measures and self-similarity on spreading depression waves," *Physica A: Statistical Mechanics and Its Applications*, vol. 401, pp. 271–277, 2014.

[18] Y. C. Campbell-Borges and J. R. C. Piqueira, "Complexity Measure: A Quantum Information Approach," vol. 10, p. 19, 2012.

[19] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, Mc Graw Hill, USA, 4th edition, 2002.

[20] "EMBRAPA (Empresa Brasileira de Produ ção Agropecuária)," https://www.embrapa.br/agropecuaria-oeste/biblioteca/acervo, 2018.