IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 21, 2020, accepted February 16, 2020, date of publication March 16, 2020, date of current version April 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981265

Cluster Density Properties Define a Graph for Effective Pattern Feature Selection

KHADIDJA HENNI[1,2], NEILA MEZGHANI[1,2], AND AMAR MITICHE[3]

[1]Centre de Recherche LICEF, Université TÉLUQ, Montréal, QC H2S 3L4, Canada
[2]Laboratoire de recherche en Imagerie et Orthopédie (LIO), Centre de recherche du CHUM, Montréal, QC H2X 0A9, Canada
[3]INRS-Centre Énergie, Matériaux et Télécommunications, Montréal, QC H5A 1K6, Canada

Corresponding author: Khadidja Henni (khenni@teluq.ca)

This work was supported by the Canada research chair on biomedical data mining under Grant 950-231214.

ABSTRACT Feature selection is a challenging problem that occurs in the high-dimensional data analysis of many major applications. It addresses the curse of dimensionality by determining a small set of features to represent high-dimensional data without significant or noticeable loss of information. The purpose of this study is to develop and investigate a new unsupervised feature selection method which uses the k-influence space concept and subspace learning to map features onto a weighted graph and rank them by importance according to the PageRank graph centrality measure. The graph design in this method promotes feature relevance, downgrades redundancy, and is robust to outliers and cluster imbalances. In K-Means classification experiments using the ASU feature selection testing datasets, the method produces better accuracy and normalized mutual information results than state-of-the-art unsupervised feature selection algorithms. In a further evaluation, using a dataset of over 14,000 tweets, conventional classification of features selected by the method gave better sentiment analysis results than deep learning feature selection and classification.

INDEX TERMS Feature selection, projected-clustering, influence-space, graph centrality.

I. INTRODUCTION

Progress in science and technology has allowed the development of applications that use very large data sets of high-dimensional data. These applications occur in various domains, most notably natural language processing, pattern recognition, and computer vision [1], [2]. High-dimensional data analysis suffers from what Duda and Hart called the curse of dimensionality [3], where methods that work well with lower dimensional data may breakdown when faced with high-dimensional data. The curse of dimensionality is likely to overfit training data, and therefore to produce models that do not generalize to new data which they will then fail to interpret [1], [4], [5]. Recent studies have explicitly addressed such issues in various ways [6]–[8], such as dimensionality reduction [9] by feature selection and reduction, done before data analysis, subspace learning to determine data layout and properties to assist clustering [7], classification [10], as well as representation by similarity and kernel functions [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Julien Le Kernec.

Feature selection can be quite useful to high-dimensional data interpretation. The purpose of feature selection is to identify a small set of features in the original high-dimensional data without affecting their intended interpretation in a noticeable way. Feature selection reduces data dimensionality by removing redundant and irrelevant components. It can impact significantly on applications, for instance to speed up machine learning algorithms, improve predictive accuracy, and enhance result interpretation [11], [12].

Feature selection methods can be divided into three categories according to their label information availability: (i) supervised methods, such as Relief [13]; (ii) semi-supervised, such as Feature Selection Via Manifold Regularization (FS-Manifold) [14] and, (iii) unsupervised methods, such as Unsupervised Graph-based Feature Selection (UGFS) and Unsupervised Discriminative Feature Selection (UDFS) [15], [16]. Unsupervised methods have attracted wide attention due to the considerable cost and time required to acquire labelled data. There are three selection strategy models [4]: (i) Filter models, which identify relevant features by statistical or information theoretic criteria; (ii) wrapper models, which involve a learning method and

evaluate features based on its performance; and (iii) embedded models, such as C5.0 [17], which embed feature selection in the learning process and use an objective function to define and select relevant features by minimization. Feature selection returns either a subset of features or the weights of all features measuring their utility. Recently, graph-based feature selection has attracted much research interest and its use has proven valuable in several applications. Graph-based feature selection methods can be divided into two categories: (1) methods based on data graphs; and (2) methods based on feature graphs. The driving idea of the former methods is to search for relevant features while preserving intrinsic data structure, such as the manifold structure or subspace [18]–[20]. These methods use graph representations by mapping data points onto graph nodes in order to characterize local data structure. Linear projection of the data onto new spaces and a minimization of the fitting errors are then applied. Methods in the latter category use features, rather than data, as graph nodes. For instance, the supervised Eigenvector Centrality Feature Selection (ECFS) scheme [21] uses statistical measures (standard deviation and linear correlation) to link features and give a graph structure of the feature space, and then ranks features according to a centrality criterion. The main limitation of this algorithm is that it only accounts for feature redundancy in the graph design, using rather simple statistical considerations. In contrast to the ECFS, UGFS is an unsupervised graph-based method that represents the feature space by a graph that is designed through jointing data neighbourhood information and subspace learning [15].

The features that preserve local data structure are linked, and then a centrality measure is used to rank features based on the most pertinent links. The UGFS scheme has shown a good performance, but it has two important limitations: (i) feature relevance to cluster discrimination is considered in the graph design, but not feature correlation, a redundancy indicator; and (ii) the k-nearest neighbors' set of each point is considered as a cluster representation, which can make this scheme sensitive to outliers and to the occurrence of clusters of unbalanced densities and shapes, which can lead to incorrect data description. The purpose of this study is to investigate a new unsupervised feature selection method, called *Influence Space and Graph-based Feature Selection* (ISGFS), which uses the k-influence space concept [22]–[24] and subspace learning to describe feature relationships and subsequently design a feature selection graph. The particularity of ISGFS is its ability to retrieve discriminative non-redundant features, applicable to data classes of arbitrary shape, size, and density. In subspace learning, the variance of a data cluster projection on a given feature dimension provides a clue to the relevance of that feature in characterizing the cluster [25], [26]. Accordingly, ISGFS links features that preserve the local projected density in core data point neighborhoods. These neighborhoods, and cluster boundaries as well, can be approximated effectively via the k-influence space for arbitrary cluster layouts, including complex cluster shapes, unbalanced clusters, and clusters with significant overlap [22]–[24]. Moreover, graph edges are weighted according to feature correlation to account for feature redundancy in the selection process. Finally, its feature ranking uses PageRank to include various feature interaction characterizations in the selection, such as the number of times features interact.

As in studies [18], [27], [28], ISGFS is informed by nearest neighbor guided subspace learning. However, it improves on these by the novel use of influence space, which makes it robust when dealing with noisy and sparse data. In addition to that, the proposed method exploited subspace learning to establish relevant features relationships. These improvements show in the several validation experiments we describe in Section V. The method does not reduce to a mere processing pipeline of existing tools; instead, it addresses the crucial issue of feature relationship analysis to determine relevant features. Features are ranked by graph centrality analysis using potent criteria related to class separation, ensuring the convergence of PageRank processing.

The remainder of this paper is organized as follows. Section II reviews related graph-based state-of-the-art methods. Section III presents the mathematical framework on which these methods are based. Section IV details the proposed method and Section V describes an experimental demonstration of its performance. Finally, Section VI presents our conclusion and avenues for promising future work.

II. RELATED WORK

This section gives an overview of the prevailing feature selection methods, which can be categorized as filter-driven, embedded cluster analysis-driven, regression-based, and graph mapping-based methods.

A. FEATURE SELECTION

Unsupervised feature selection has attracted increasing attention and various algorithms have been suggested and categorized according to their selection model. Filter-driven methods give each feature a score estimated via a metric. They are univariate and differ by filter type, such as variance in MaxVar [29] and Laplacian in [30]. Filter-based methods suffer mainly from feature interaction omission.

Embedded cluster analysis methods can be quite complex, as they study clustering attributes in the data in order to select features after learning a model (classification or regression), as in Multi Class Feature selection (MCFS) [31], Similarity Preserving Feature Selection (SPFS) [27], and Minimum Redundancy Spectral Feature Selection (MRSF) [32]. These methods' use of clustering makes their algorithms slow and (generally) unscalable. Related methods, such as UDFS [16] and Non-negative Discriminative Feature Selection (NDFS) [33], capture the manifold structure of data by performing a learning step to give scores to the most discriminative features. However, these methods introduce constraints that are too restrictive and

they can be quite affected by data noise and outliers. The sparsity of data in high-dimensional spaces has motivated the use of the $\ell_{2,1}$-norm, which gives a high performance even in the presence of noise. This norm was adopted by the Robust Unsupervised Feature Selection (RUFS) scheme in [34]. RUFS is an embedded method which activates clustering and feature selection simultaneously. The minimization aspect of the $\ell_{2,1}$-norm was utilized in the Regularized Self-Representation (RSR) study of [35]. Some of the more recent feature selection methods are regression-based, where the main feature is to minimize the error between the projected data and the target matrix (as in the RSR) by exploiting orthogonality to decompose the target matrix (Simultaneous Orthogonal basis Clustering Feature Selection(SOFCS) [36]), as well as integrating feature selection, matrix factorization and manifold regularization into a unified framework [19].

The graph-based investigation of [21] computes feature relationships to realize a mapping of features onto a graph and subsequently rank them according to their importance given by a graph centrality measure. Their study proposed the ECFS algorithm, which ranks features based on three characteristics: (1) mutual information between features and labels, (2) a Fisher criterion to investigate the extent to which features discriminate classes, and (3) the standard deviation, to capture the amount of variation or dispersion of features from the average. By adding the standard deviation matrix and the product of the Fisher and mutual information matrices, the adjacency matrix can be used to construct the graph. Finally, the eigenvector centrality [37] is used to rank features. It is worth noting that the proposed graph design is based on characteristic vector multiplication which only describes pairwise feature relationships and neglects the potential benefits of multi-feature combinations [21]. Several variants of the Eigenvector centrality [38] have been investigated and have exhibited more efficiency in scoring nodes, as with PageRank [39] and HITT [40].

The drawbacks of the ECFS algorithm have motivated the development of the UGFS method [15], where features are graph nodes linked according to subspace preference clusters [6], i.e., features that correspond to significant cluster discrimination are linked. According to subspace learning [28], [41], the variance of the k-nearest neighbors of each data point indicate the most relevant features. These latter are extracted and located on a graph, then the pageRank centrality measure is used to rank features mapped onto this graph. The UGFS performed significantly better than other methods, although three drawbacks have been noted: (i) the elements in a k-nearest neighbors set may belong to different clusters, and therefore, the search for cluster discriminating features may be unduly affected; (ii)considering all data points for feature combination may unduly change the results, especially if the data set contains outliers; and (iii) the correlation between features is not exploited as a means to disfavor the redundant features [15].

B. SUBSPACE LEARNING

A noteworthy property of multidimensional spaces is that the clusters may exist in different subspaces and not in the full-dimensional space [25], [26], [42]. This property has motivated the emergence of studies in subspace and projected clustering learning, which cope with the curse of dimensionality issue by searching for the pair (C; S) where C is a set of points composing a cluster and S is a set of the most characterizing features of the considered cluster. The application of this concept has had considerable success, especially in the unsupervised context where Yip *et al.* [43] have illustrated the existence of projected clusters in real-life datasets. The principal of projected clustering has been widely utilized to account for the effect of large spaces (multidimensional spaces) [28], [41].

The grid-based algorithm CLIQUE [44], which developed subspace clustering, recursively investigates the set of possible subspaces based on an a priori-like method and then retains grid cells with a density greater than a given threshold. Yiu. et al proposed a Monte Carlo algorithm called DOC (Density-based Optimal projective Clustering), that uses the density of points in subspaces iteratively to effectively discover projected clusters [45]. Projected clustering based on K-means (PCKA), proposed by Bouguessa and Wang [25], improves on the K-means algorithm by involving the search for projected clusters. The projected clusters are found by estimating the dense regions in each dimension. Another clustering algorithm based on subspace learning, proposed by Wang *et al.* [46] and called Fast Adaptative K-means (FAKM), integrates feature selection into clustering to identify without eigenvalue decomposition the most representative features subspace. The FAKM objective function accounts also for outliers and noisy data.

Common dissimilarity measures, such as the Euclidean distance, are full-space functions, which makes them highly effective in low-dimensional spaces but significantly less accurate in high-dimensional spaces due to the data sparsity. The notion of subspace analysis has been used to define improved dissimilarity measures, as in [6], where a weighted Euclidean distance was proposed based on subspace learning and the well-founded notion of density connected clusters (more details are given in Section III-C).

C. CENTRALITY MEASURES AND PageRank

Currently, and in light of the ongoing growth of social networks, the characterization of important (central) nodes within graphs is a problem being widely addressed from different perspectives. This characterization is generally related to the calculation of degree centrality measures, i.e., numerical score values that allow the relevant nodes to be ranked according to specific criteria.

While some measures focus on the inherent structure of the network, e.g., the well-known degree and betweenness centrality measures, others bring additional information into

the calculation. The four categories of centrality measures are [38]: 'Degree Centrality', the simplest measure, which is equal to the number of edges connected directly to the considered node; 'Closeness Centrality', based on the distance between nodes and follows the assumption that shorter distances improve information dissemination on a graph; Betweenness Centrality', in which high-scoring nodes are those able to communicate with other nodes, while requiring few intermediaries; and 'Eigenvector centrality', which favors nodes connected to members that are strongly connected to other graph actors. These methods can operate on directed or undirected graphs, and can also take into account graph weights to highlight node relationships.

Google developed an Eigenvector centrality variant called PageRank [47] to locate recognizable and relevant web pages. It is simple, fast, entirely general, and applicable to any type of graph. PageRank has been used in biology and bioinformatics to find and rank genes (GeneRank [48]) and proteins (ProteinRank, AptRank [49], as well as to match protein-protein interactions (IsoRank). It is also used in neuroscience, complex engineering systems (MoniorRank), in Linux kernel, bibliometrics (CiteRank, TimedPageRank, AuthorRank), social networks (BuddyRank and TwitterRank), and several other applications [47].

III. PRELIMINARY CONCEPTS

This section briefly describes some of the fundamental notions of graph analysis theory influence space and subspace learning techniques and their notation conventions.

A. NOTATIONS

Let DB be a dataset of $d-$dimensional points, where the set of features/dimensions is denoted by $F = \{F^1, \ldots, F^d\}$. Let X denotes the set of n data point $X = \{x_1, \ldots, x_n\}$, $X \subset \mathbb{R}^d$. Each point $x_i \in X$ is a vector of d dimensions $x_i = (x_i^1, \ldots x_i^j, \ldots, x_i^d)$, where x_i^j $(i = 1 \ldots n; j = 1 \ldots d)$ is the value of data point x_i on the dimension F^j. Let $dist(p, q)$ be the Euclidean distance between two data points $p, q \in X$.

The problem is to map the features onto a weighted graph. Let $G =< F, E >$ be a graph, where the vertices (nodes) F are the set of features $F = \{F^1, \ldots, F^d\}$, and E indicates the edges linking the vertices. $\mathbf{A}$ is the adjacency matrix corresponding to graph G, where each element $a_{i,j}$ represents a pairwise relationship between features F^i and F^j. Coefficients $a_{i,j}$ are defined via a potential function ϕ:

$$a_{i,j} = \phi(F^i, F^j) \tag{1}$$

Table 1 summarizes the most important notations used in this article.

B. INFLUENCE SPACE

In recent outlier analysis research, local outliers have been mined by computing the density distribution of their neighbors. Cassisi *et al.* proposed the k-influence space (Is_k) concept which improves the separation of clusters with heterogeneous densities [22]–[24]. This concept has been used as a new dimension, with the clustering process carried out in the new residual space. Lv *et al.* have used Is_k to reduce the amount of DBSCAN input parameters [23]. Is_k has also been used to determine the core-points, which supported the is-clustering algorithm in dense regions [24].

TABLE 1. Main notations in the paper.

Notation	Description
d	Dimensionality of samples
n	Number of samples
$\tilde{n}$	Number of core-point
k	Number of considered nearest neigbhors
K	Number of clusters
α	Teleportation parameter
δ	Variance threshold
X	Set of data points
F	Set of features
F^i	ith feature vector
G	Graph
$\mathbf{A}$	Adjacency matrix of graph G
C_s	Projected cluster s
S_s	Relevant features subset to discriminate clusters s

Consider observations $x, p, q \in X$.

Definition 1: The k_{dist} of x, denoted as $k_{dist}(x)$, is the distance $dist(x, p)$ between x and p in R^d, such that: (i) for at least k objects, it holds that $dist(x, q) \leq dist(x, p)$ and (ii) for at most $k-1$ objects, $dist(x, q) < dist(x, p)$.

Definition 2: The *k-nearest neighborhood* of an observation $x \in X$, $NN_k(x)$ is the set of observations p such that $dist(x, p) \leqslant k_{dist}(x)$, which means:

$$NN_k(x) := \{p \in X \setminus \{x\} \mid dist(x, p) \leqslant k_{dist}(x)\}.$$

Definition 3: The *reverse k-nearest neighborhood* of an element x is defined as:

$$RNN_k(x) := \{p \in D \mid x \in NN_k(p)\}.$$

Definition 4: The *k-influence space* of the observation x is defined as:

$$Is_k(x) := NN_k(x) \cap RNN_k(x).$$

The density of neighbouring data around an observation $x \in X$ can be estimated through the k-influence space ($Is_k(x)$) [22]–[24]. The k-nearest neighbours set $NN_k(x)$ is never empty, whereas the size of $RNN_k(x)$ depends on how many times x is classified as k-nearest neighbor of an object x. In cluster analysis, the size of $k-$influence space of a data point indicates the importance of the corresponding data point: for $p \in X$, if $|Is_k(p)| > \frac{2}{3}k$, p is a core point, and if $|Is_k(p)| = 0$, p is a noise point (see figure 1), where $\frac{2}{3}k$ is a threshold used in the literature [22]–[24].

In this paper, we use the k-influence space to define the neighbourhood of data points independently of the geometrical distances, and to eliminate noisy data, which have empty $k-$influence space sets and can negatively influence the feature selection results.

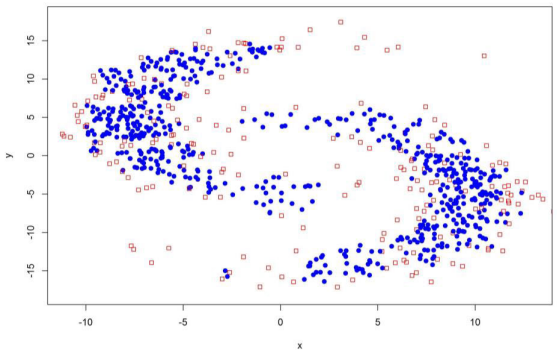

FIGURE 1. The distribution of core (blue points) and noise (red points) data classified via the Is_k cardinality.

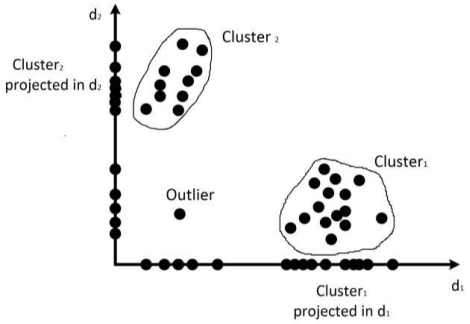

FIGURE 2. Variation of projected densities along dimensions.

C. PROJECTED CLUSTER DENSITY

Projected clustering is a class of subspace learning, which highlights and exploits the existing correlation of different clusters along different subsets of dimensions. The clusters are defined as a subset of data points C which are densely clustered together a subspace S of features [25], [26].

Let x_i^j a 1-dimension point representing the projected value of the point $\mathbf{x_i}$ on the dimension F^j. A projected cluster C_s, $s = 1, \ldots, K$, is a pair (C_s, S_s), where K is the number of clusters, C_s is a subset of data points composing the cluster and S_s is a set of pertinent features that characterize the considered cluster. The variance of the projection of a data cluster on a given feature can indicate its relevance in the detection of the latter cluster. The subset of dimensions $\{S_s\}_{\{s=1,\ldots,K\}}$, are not disjointed and may have different cardinalities. This assumption has motivated subspace learning in feature selection [15]. For instance, as illustrated in Figure 2, feature d_1 is able to discriminate cluster$_2$ by means of the projected data variances better than feature d_2 can discriminate cluster$_2$.

Figure 3 illustrates an artificial data set composed of 2,000 data points represented by 12 dimensions. Data points are clustered into 5 clusters; rows represent clusters boundaries. Each cluster has a subset of relevant dimensions that exhibit its structure; for example, cluster 1 exists in dimensions F^1, F^3, F^6, F^8, F^1 and F^{11}. Note that some irrelevant dimensions can be identified visually (marked in red color), such as F_2, F_5 and F_{12}, where data points are uniformly distributed and where no cluster structure has been identified.

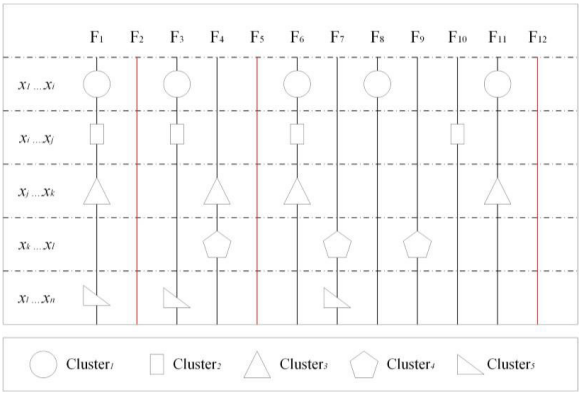

FIGURE 3. Illustration of an artificial dataset with projected clusters.

In an unsupervised context, the neighborhood of data points is considered to be the points that belong to its cluster. Bohom *et al.* [6] have proposed a weighted Euclidean distance based on subspace concepts and the notion of density connected clusters. Their study proposed the notion of a subspace preference cluster based on the variance of the data neighborhood along features, and then used these variances to weight the distances.

D. PageRank FOR GRAPH CENTRALITY

PageRank, Google's web page ranking system proposed by Brin and Page [50], is a common ranking system. It uses a variant of the eigenvector centrality measure, which iteratively computes the normalized and propagated value for each node in a graph, where each graph node rank depends on the rank of nodes pointing to it. The iterative process to compute the PageRank (pr) was initially defined by a simple sum as follows:

$$pr_{iter+1}(E_i) = \sum_{E_j \in B_{E_i}} \frac{pr_{iter}(P_j)}{|E_j|}, \tag{2}$$

where pr_{iter+1} is the PageRank of graph node E_i at iteration $iter + 1$, and B_{E_i} is the set of nodes pointing to E_i. The process is initialized with $pr_0(E_i) = \frac{1}{\hat{n}}$, for all E_i, where $\hat{n}$ is the number of nodes in the graph, and is repeated with the expectation that the scores will converge to stable values.

Eq. 2 computes the PageRank one node at a time. The following matrix representation computes the PageRank vector of all nodes:

$$\pi^{(iter+1)T} = \pi^{(iter)T}\mathbf{H}, \tag{3}$$

where π is the PageRank vector, $\mathbf{H}$ is a $\hat{n} \times \hat{n}$ binary adjacency matrix of the graph, $H_{i,j} = 1/|E_i|$ if there is a link from node i to j, and is 0 otherwise.

The convergence of the iterative process to a stable PageRank value has been verified by the Markov properties. In fact, Eq.3 is a simple "linear stationary process", it is equivalent to a power method applied to $\mathbf{H}$. The later matrix $\mathbf{H}$ is

like a stochastic transition probability matrix for a Markov chain. According to the Markov properties, if the Markov matrix is stochastic, irreducible and aperiodic, the process will converge to a unique positive vector (stationary vector).

The Google Matrix, $\mathbf{G}$, is defined as:

$$\mathbf{G} = \alpha\mathbf{H} + \frac{1-\alpha}{n}\mathbf{e}\mathbf{e}^T, \tag{4}$$

where $\mathbf{e}$ is a vector with all entries equal to 1, and α is an input parameter $\alpha \in (0, 1)$, called the "teleportation parameter", used to control the "diffusion" of random walks combinations. The proposed matrix is a Markov matrix: (1) a convex combination of two stochastic matrices $\mathbf{H}$ and $\frac{1}{n}\mathbf{e}\mathbf{e}^T$; (2) irreducible, each node directly connected to every other node; (3) aperiodic, due to the self-loops $\mathbf{G}_{ii>0}$, for all i and; (4) sparse. For a discussion of the convergence process, please refer to [39], [51]. Coefficient α is typically fixed to 0.85. Convergence is typically obtained quickly for this value. However, convergence generally requires more time when α is larger, and instability can arise when α is close to 1 [52].

IV. INFLUENCE SPACE GRAPH-BASED FEATURE SELECTION (ISGFS)

The main idea driving ISGFS is to map features onto a weighted graph and then use PageRank to score them according to their importance in that graph. In essence, the greater the number of links pointing to a feature (node) and the larger these links' weights, the higher that feature's rank.

By exploiting feature selection constraints, these graphs are designed to maximize features' relevance and minimize their redundancy, without being influenced by the existence of outlying data and unbalanced clusters. This is accomplished by analyzing the projected clusters and ranking features according to their ability to preserve the neighborhood densities and shape of informative data points by linking features with smaller variance in the projected neighborhood data of core data points. These neighborhoods are estimated using the influence space concept [22]–[24]. This process is carried out in four basic steps:

First, ISGFS searches the neighborhood of each data point p, which is its k-influence space $Is_k(p)$, where data points are assumed to belong to the same cluster. $Is_k(p)$ is composed of data belonging to both $NN_k(p)$ (the k-nearest neighbors of p) and $RNN_k(p)$ (the reverse k-nearest neighbors of p) (see section III-B and [22]–[24]).

Second, ISGFS computes the variances along features as follows:

$$Var^i(Is_k(p)) = \frac{\sum_{q\in Is_k(p)}(dist(x_p^i, x_q^i))^2}{|Is_k(p)|}, \tag{5}$$

where $p \in X$, $k \in N$, $Var^i(Is_k(p))$ is the variance of $Is_k(p)$ along a feature i, x_p^i is the value that takes the point p on the feature F^i (projected value).

Third, it searches for each non-noisy point p ($|Is_k(p)| > \frac{2}{3}k$), the subspace preference dimensionality S_p. This is the set of features with variances lower than a user input threshold δ.

Fourth and last, ISGFS maps the features onto the graph and applies PageRank centrality to rank them. The graph edges relating features F^i and F^j are weighted by the negative of their pairwise correlations, thereby downgrading feature redundancy. Thus, the potential function is given by:

$$\phi(F^i, F^j) = \begin{cases} 1/corr(F^i, F^j), & \text{if } F^i, F^j \in S_p \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where S_p is the core point p subspace preference dimensionality, $Var^i(Is_k(p)) \leq \delta$, $Var^j(Is_k(p)) \leq \delta$ and δ is the variance threshold.

The adjacency matrix $\mathbf{A}$, with elements $a_{i,j} = \phi(F^i, F^j)$, associated to the designed graph, is stochastic, sparse, and reducible. Matrix $\mathbf{A}$ is used as the transition probability matrix of a Markov process to compute the features PageRank.

More details are provided in Algorithm 1.[1] Note that i, j, l, and m are indexes, $\tilde{n}$ is the number of core-points, and $Var_{Binarized}$ is a binary matrix in which rows and columns correspond to core points and features, respectively. Also, $Var_{Binarized}(i, j) = 1$ if feature F^j is relevant to core point x_i.

V. EXPERIMENTS

This section focuses on the evaluation of the feature selection performance of ISGFS, in both supervised and unsupervised tasks, i.e., clustering and classification, on several public datasets. In the experiments, ISGFS is evaluated in two ways: (i) It is compared to state-of-the-art unsupervised feature selection algorithms; and (ii) The classification of ISGFS features by conventional algorithms is compared to deep learning feature extraction and classification.

A. EVALUATION METRIC

Classification performance, in either supervised or unsupervised mode, on the selected features datasets, is the criterion to evaluate the feature selection effectiveness of given algorithms.

For all methods compared, the classifiers are applied on the selected features and four evaluation metrics are used to assess the classification performance (1) the classification accuracy (ACC), (2) $precision$, (3) $recall$ and, (4) normalized mutual information (NMI) defined by:

Classification performance in the selected feature datasets, in either supervised or unsupervised mode, is the criterion for evaluating the feature selection effectiveness of a given algorithm. The classifiers are applied on the selected features of the six methods compared here, and four evaluation metrics are used to assess the classification performance: (1) the classification accuracy (ACC), (2) precision, (3) recall, and (4) the normalized mutual information (NMI), defined by:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

[1]The source code will be posted online to provide the needed material for the use of ISGFS.

Algorithm 1 : ISGFS Algorithm

Input: Observed data: $X = \{x_1, \ldots, x_n\}$, k, δ.
Output: *RankedFeatures*: a vector of ranked features.

Compute $Is_k(x_i)$, with $i = 1, \ldots, n$.
Compute $Var^j(Is_k(x_i))$, with $i = 1, \ldots, n$ and $j = 1, \ldots, d$ (see Equation 3).
$\tilde{n} = 0, A(i, j) = 0, i = 1, \ldots, d$ and $j = 1, \ldots, d$.
for $i = 1 : n$ & $|Is_k(x_i)| \geq 2/3k$ **do**
  for $j = 1 : d$ **do**
    if $Var^j(Is_k(x_i)) \leq \delta$ **then**
      $Var_{Binarized}(\tilde{n}, j) = 1$
    else
      $Var_{Binarized}(\tilde{n}, j) = 0$
    end if
  end for
  $\tilde{n} = \tilde{n} + 1$
end for
for $i = 1 : \tilde{n}$ **do**
  $S_p = \{\}$
  for $j = 1 : d$ **do**
    if $Var_{Binarized}(i, j) == 1$ **then**
      $S_p = S_p \cup \{F^j\}$
    end if
  end for
  for $l = 1 : size(S_p)$ **do**
    for $m = 1 : size(S_p)$ **do**
      $A(l, m) = 1/corr(F^{S_p(l)}, F^{S_p(m)})$
    end for
  end for
end for
$G = (\{F^1, \ldots, F^d\}, A)$
$RankedFeatures = pageRank(G, 0.85)$

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP + TN}{TP + FN}, \tag{9}$$

where TP, TN, FP and FN denote the number of true positives, true negatives, false positives and false negatives, respectively. The NMI is defined as:

$$NMI = \frac{MI(C, G)}{max(H(C), H(G))}, \tag{10}$$

where C and G are classification labels and ground truth labels, $MI(C; G)$ is the mutual information between C and G, and $H(C)$ and $H(G)$ denote the entropy of C and G, respectively. The standard unsupervised classifier K-means algorithm is used in these experiments. Each experiment was repeated 20 times and the average results are reported. Supervised classifiers (Logistic regression model and Random forests) are assessed by 20-fold cross validation. As random sampling is used for both K-means initialization and the splitting of data into learning and test subsets, when dealing with the supervised classifiers, all experiments were repeated 20 times and the means of ACC, NMI, precision, and recall are reported to ensure that the results were not influenced by sampling.

TABLE 2. Dataset descriptions.

Dataset	Features	Instances	Classes	Types
USPS	256	9298	10	Digit images
Orl64	4096	400	50	Face image
Yale	1024	165	15	Face image
Isolet	617	1560	26	Speech signal
Prostate	5966	102	2	Microarrays data
Relathe	4322	1427	2	Text data
BaseHock	4862	1993	2	Text data
US Twitter Airline	-	14640	2	Text data

B. DATASETS

Several experiments were conducted to show the effectiveness of ISGFS in selecting relevant features. Two types of experiments were utilized to achieve a more accurate appraisal.

- Experiment 1: Comparison of ISGFS with state-of-the-art feature selection methods. Here, ISGFS and other feature selection algorithms are applied on 7 publicly available datasets [2] that are generally used to evaluate feature selection techniques. The p first features in the rankings provided by these methods are then used to classify data by k-means, and the classification is evaluated according to the ACC and NMI values (see section V-F.1).
- Experiment 2: Comparison of ISGFS with deep learning feature extraction methods. Here, a public text dataset used in sentiment analysis, composed of 14640tweets, is the input to compare the sentiments classification given by: (1) Conventional methods using ISGFS features in which the tweets are preprocessed and ISGFS selects features and reduces the dataset, then uses common text mining classifiers, namely logistic regression and random forests, to analyze the sentiments in the tweets; and (2) Deep learning feature extraction and classification in which two models are used to extract relevant features and then classify the tweets.

The description of these datasets is summarized in Table. 2.

In order to assess the robustness of ISGFS compared to that of UGFS when dealing with noisy data (outlier data points and redundant features), we added two other experiments: (i) modifying the USPS dataset by adding a set of noisy instances (200), called "USPS + NI"; and (ii) adding a set of redundant features (20 features) to the USPS, called "USPS + NF".

C. FEATURE SELECTION ILLUSTRATION

The Yale dataset, composed of 165 gray-scale images of 15 individuals, was used here [53]. This facial feature dataset was created by capturing 11 images of each subject, with different facial expressions and configurations: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light,

[2] http://featureselection.asu.edu/

FIGURE 4. Feature selection illustration on the Yale dataset. Each row refers to a number of selected features $p \in \{50, 100, 200, 300\}$ and each column corresponds to the selected human face image in 11different configurations.

sad, sleepy, surprised, and wink. In order to visually illustrate the behavior of ISGFS, a subject is randomly selected from the Yale dataset, and the selected features (i.e., pixels) are highlighted in white. For each sample; the number of selected features p has been chosen as $p \in \{50, 100, 200, 300\}$.

Figure 4 shows the features selected by ISGFS. It demonstrates that ISGFS is able to capture the most discriminative parts of human faces such as the eyes, nose, and mouth.

D. EXPERIMENTAL SETUP

As noted in the previous sub-section, the evaluation of ISGFS was done using two types of experiments. The different experimental setups are described below.

1) EXPERIMENT 1- COMPARISON OF ISGFS WITH OTHER FEATURE SELECTION METHODS

The proposed ISGFS algorithm is compared with the following feature selection algorithms:

- Baseline: All of the original features are adopted, i.e., no selection;
- Laplacian Score (LS): The Laplacian score is used to choose features that preserve the similarity of the original data [30];
- Multi-cluster feature selection (MCFS): Selects features by spectral information regression based on ℓ_1-norm regularization [31];
- Regularized self-representation feature selection (RSR): Selects features from sparse spaces through the $\ell_{2,1}$-norm regularization [35];
- Multi-Task Feature Learning Via Efficient $\ell_{2,1}$-norm Minimization (LL$\ell_{2,1}$): Considers the $\ell_{2,1}$-norm regularized regression model for joint feature selection from multiple tasks [54];
- Eigenvector Centrality for Feature Selection (ECFS): Ranks features by measuring the eigenvector centrality of the pairwise features graph [21]; and
- Unsupervised Graph-based Feature Selection (UGFS): Ranks features by applying PageRank centrality on the feature graph designed via subspace preference clusters [15].

ISGFS and the other algorithms compared here use input parameters, which need to be properly adjusted. The parameter that needs to be adjusted for all algorithms is the number, p, of the selected features. In these experiments, the top p features are selected for all algorithms, where $p \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%\}$: for each specific value of p and for each dataset, the parameters are tuned by the "grid search" strategy to achieve the best results among all possible combinations. For the algorithms that use the KNN graph in their search strategy, such as ISGF, LS, etc, the number of the nearest neighbors is fixed at 5. The parameter λ of RSR takes its value in the set $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$. For both the *ACC* and the *NMI*, 20 clustering experiments are performed, each with random initialization, and the corresponding mean and standard deviation are computed.

2) EXPERIMENT 2: COMPARISON OF ISGFS AND DEEP LEARNING:

This experiment focuses on sentiment analysis in a US Twitter Airline dataset by exploring conventional classification techniques combined with ISGFS and deep learning.

We compare the performance of the logistic regression model and random forest classification, with and without features selected by ISGFS, as well as the performance of the Long Short-Term Memory (LSTM) network and the Convolutional neural network (CNN).

Logistic regression provides a simple classification algorithm commonly used in data mining. It is simple and does not require tuning, but for text classification it is less accurate than other models due to its sensitivity to feature correlation [55]. The random forest classifier is an ensemble learning method that has given the best accuracy in textual data classification. We have used these two classifiers to illustrate that the features selected by ISGFS improve the performance of classifiers whatever their ability to classify textual data [55].

To assess the quality of sentiment analysis by conventional classification techniques, the US Twitter Airline dataset was processed using some text mining techniques for sentiment analysis. First, the text was cleaned by removing mentions, hashtag signs, punctuation including question and exclamation marks, URLs, digits and stop words, as well as emojis. The emojis were converted into one word and all words were normalized to lower case. Next, we applied a stemming algorithm to keep the words' stems. After this step, the text is considered to be cleaned and prepared for feature extraction and classification. Two feature extractor methods, widely used in text mining, were used: term frequency–inverse document frequency (TF-IDF Vectorizer) and Word2Vec.

TF-IDF Vectorizer is a statistical method used by the most popular recommender systems, which are information filtering systems based on text mining to customize users' information according to their interests and recommend pertinent items. TF-IDF Vectorizer evaluates word importance in a document/corpus and provides a weighted vector of those words. The word importance is evaluated based on how frequently they appear across multiple documents.

Word2vec is a word-embedding method that computes high-dimensional vectors representing word semantics. Words with similar semantics will have similar vectors. The method trains a two-layer neural network to reconstruct the linguistic contexts of words. Words sharing a common context in a given corpus are neighbors in the semantic vector space.

For both vectorization methods, TF-IDF and Word2vec, we retain 3000 of the 14640 original features. Numerical matrices are used by the logistic regression model and the random forest classifiers, with and without using ISGFS to reduce the high-dimensional space. Classification is assessed by 20-fold cross-validation, and results are compared using the *ACC*, *precision*, and *recall*. values. To apply deep learning for sentiment classification, two popular network architectures were used on the tokenized dataset in order to compare the classifiers and to assess the ISGFS effectiveness: the Long Short-Term Memory network (LSTM) [56] and the Convolutional neural network (CNN) [57].

The LSTM network is a recurrent neural network characterized by its memory information for long periods of time, which explains its successful application on sequential data, particularly in text and sentiment analysis. The LSTM network adds or removes information using structures called gates: input, forget, and output. The LSTM model is created with a sigmoid activation function, which has shown better results than Softmax.

The CNN was initially proposed for image classification and has become a versatile model used for a wide range of tasks. It recognizes pertinent local features in a multidimensional space by utilizing convolutional layers composed of filters. CNNs were used in text mining by Kim [57] to exploit the fact that text is structured and organized.

TABLE 3. Computational complexity of ISGFS against the background of other methods: d is the dimension of data, n the number of data points, $\tilde{n}$ the number of core points, t the number of algorithmic iterations, and K is the number of clusters.

Method	Time Complexity
Laplacian Score (LS)	$O(dn^2)$
MCFS	$O(n^2d + Kn^3 + nKd^2 + d\log(d))$
RSR	$O(td^2n + td^3)$
LL$\ell_{2,1}$	$O(nd + nt)$
ECFS	$O(nd + d^2)$
UGFS	$O(d\log(n) + nd^2)$
ISGFS	$O(d\log(n) + nd^2)$

TABLE 4. Comparison of feature selection and classification runtimes.

Method	Feature selection	K-means
Baseline	-	197.4 s
Laplacian Score (LS)	585.2 s	107.8 s
MCFS	1982.5 s	64.7 s
RSR	1023.1 s	57.6 s
LL$\ell_{2,1}$	515.1 s	38.9 s
ECFS	634.8 s	126.6 s
UGFS	976.4 s	45.1 s
ISGFS	798.4s	35.4 s

E. TIME COMPLEXITY

Here following is a computational complexity analysis of the ISGFS algorithm. The time complexity for calculating the influence space set is approximately $O(d\log n + n)$ because ISGFS computes in $O(d\log n)$ time the $k-$nearest neighbors by the kD-tree method, and the reverse $k-$nearest neighborhood in $O(n)$ time.

Feature graph generation, which follows the k-nearest neighbors and reverse k-nearest neighbors analyses, is run in $O(\tilde{n}d^2)$ time for $\tilde{n}$ core points, which is at most $O(nd^2)$. Finally, PageRank requires $O(\frac{1}{\alpha}.\log(\tilde{n}))$ time. Therefore, the computational complexity of ISGFS feature selection is $O(d\log(n)+n) + O(nd^2) + O(\log(n)) = O(d\log(n) + nd^2)$.

Table 3 shows the time complexity of the ISGFS method against the background of others. Note that ISGFS and UGFS have the same theoretical time complexity, but UGFS is generally slower in practice due to the influence of noise in data. ISGFS has higher complexity than filter feature selection methods (LS, LL$\ell_{2,1}$, ECFS), but lower than other embedded algorithms such as MCFS.

ISGFS is implemented in Python, under the Windows operating system. Experimental evaluation is done on a laptop

TABLE 5. ACC±std clustering results of different feature selection algorithms on different datasets. The best results are highlighted in bold and the second best results are underlined.

Dataset	Baseline	LS	MCFS	RSR	LL$\ell_{2,1}$	ECFS	UGFS	ISGFS
USPS	63.56±1.54	63.48±1.12	63.56±1.54	65.16±3.56	66.12±3.41	64.23±1.32	66.24±1.54	**66.84±1.91**
Orl64	56.09±2.14	52.61±1.54	56.64±1.89	54.89±2.18	56.8±1.54	54.86±2.18	55.76±1.58	**56.71±1.84**
Yale	42.8±2.4	43.12±1.41	46.37±1.68	45.51±1.24	46.01±1.24	42.74±1.45	46.24±1.84	**46.45**±1.45
Isolet	58.43±2.75	61.58±1.54	62.84±2.35	60.98±1.84	**63.41±2.68**	59.38±2.7	60.54±1.18	63.28±2.56
Prostate	58.64±1.03	62.54±1.5	68.42±2.14	62.78±2.37	68.76±1.54	66.43±2.04	68.47±1.86	**70.61±1.42**
Relathe	51.1±2.34	53.63±1.31	56.48±2.54	54.58±1.38	57.14±1.62	53.64±1.57	56.75±3.52	**58.15±1.65**
BaseHock	51.21±1.34	52.63±1.75	54.68±1.76	**56.45±3.06**	54.68±2.68	55.71±2.68	55.98±2.81	56.87±1.86
USPS+NI	58.45±5.61	49.34±3.67	59.28±3.39	60.98±3.68	58.17±1.84	61.84±2.37	60.86±5.53	**73.02±2.53**
USPS+NF	53.34±3.49	45.63±6.62	46.21±6.7	45.89±6.42	47.06±2.36	65.45±1.49	58.65±4.38	**68.92**±4.85

TABLE 6. NMI±std clustering results of different feature selection algorithms on different datasets. The best results are highlighted in bold and the second best results are underlined.

Dataset	Baseline	LS	MCFS	RSR	LL$\ell_{2,1}$	ECFS	UGFS	ISGFS
USPS	61.63±1.62	59.47±1.60	66.56±0.44	66.16±1.77	61.36±1.82	67.53±2.02	60.32±1.63	**68.37±1.01**
Orl64	74.74±1.75	74.35±1.05	75.16±1.67	72.46±1.29	75.16±1.67	74.63±1.22	74.65±1.62	**75.41±1.19**
Yale	59.2±1.86	56.8±2.42	63.12±1.73	62.14±1.73	64.35±2.57	57.6±1.15	65.35±2.57	**66.04±2.42**
Isolet	74.68±1.21	76.9±1.4	72.4±0.95	77.46±1.64	75.94±1.87	75.46±1.65	**77.27±1.24**	77.86±1.19
Prostate	2.17±0.16	2.94±0.31	6.35±0.42	4.42±0.65	5.71±1.62	6.15±0.35	7.06±1.56	**7.81±0.64**
Relathe	1.3±0.05	2.15±0.85	1.34±0.17	3.18±0.21	2.62±0.68	2.95±0.27	2.95±0.18	**3.21±0.61**
BaseHock	3.8±0.53	3.94±0.34	5.15±0.29	**8.53±0.57**	5.65±0.72	4.68±0.52	7.49±0.58	8.17±0.97
USPS+NI	47.56±2.32	39.78±2.45	37.61±2.89	43.88±1.48	44.84±2.63	38.62±3.17	44.64±2.64	**59.37±1.81**
USPS+NF	59.47±1.60	59.47±1.60	66.56±0.44	66.16±1.77	61.36±1.82	67.53±2.02	60.32±1.63	**68.57±1.95**

i7 Intel dual processor 2.4 GHz/CPU and 16 GB DRAM. We generated an artificial big dataset of 10000 objects (20% noisy data point) and 5000 features, and compared the run times of k-means when it uses the optimal selection of features by ISGFS and others. The optimal selection, a for any given method, is the one that gives the best clustering by k-means for the method of selection, i.e., the selection is optimized individually for each selection method. Table 4 shows the results. Features selected by ISGFS enabled the fastest clustering runtime. Note that, in general, faster feature selection does not necessarily translate into better clustering runtime.

F. FEATURE SELECTION EVALUATION AND DISCUSSION

1) EXPERIMENT 1

To show the effectiveness of feature selection on classification, features are selected as described in Section V-D.1, and then their classification is evaluated according to the *ACC* and *NMI*. The experimental results are summarized in Table 5 and Table 6.

For each dataset, the best results are highlighted in bold. The comparison of *ACC* and *NMI* values shows that the proposed ISGFS method outperforms most of the state-of-the-art methods. Experiments where outlier data are added (see section V-B), illustrated the ability of ISGFS to select relevant features without being influenced by outliers, which is not the case for most of the other methods.

ISGFS with UGFS show competitive results for most of the datasets, however, the ISGFS gave better results when dealing with datasets with additive noisy data (outlier data points and correlated features). These results confirm the effectiveness of using Is_k, which made the algorithm robust against outliers. In addition, the negative of the pairwise correlation to the weighting feature' links in the graph was effective in eliminating correlated features.

Finally, the comparisons of ISGFS and UFGS to the supervised ECFS show that the use of the subspace cluster preference in designing graph features improves performance. Figures 5 and 6 illustrate the *ACC* and *NMI* values according to different numbers of selected features, p. In most cases, the obtained results reveal that the proposed ISGFS gives the best results.

Note that, in most cases, ISGFS obtained higher clustering accuracy than the baseline method when fewer features were retained, which confirms its ability to improve clustering accuracy and selection time. The ISGFS outperforms RSR in all experiments, supporting its ability to preserve local geometrical structure. The ISGFS also outperformed the L$\ell_{2,1}$-norm, a robust algorithm with sparse datasets. In summary, the experimental results support the conclusion that ISGFS can be used advantageously to improve both runtime and accuracy of classification.

2) EXPERIMENT 2

Table 7 summarizes the comparative results of several sentiment analysis techniques. As noted in Section V-D.2, we compare conventional classification methods with deep learning based methods, while using or not using the ISGFS algorithm and also by using two feature extraction methods.

Some conclusions can be drawn from Table 7: (1) The logistic regression model provides improved classification results with both vectorization methods, with and without

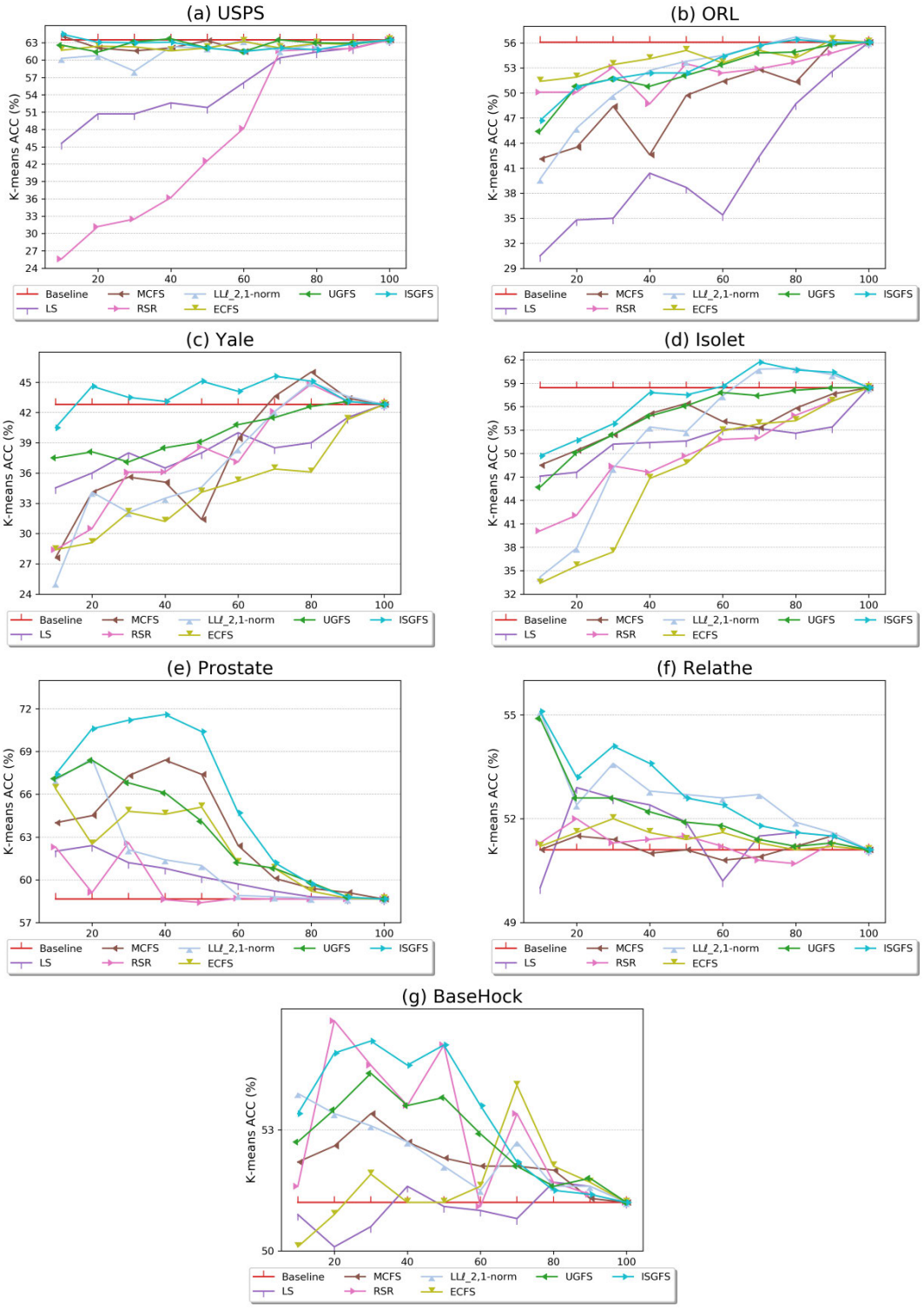

FIGURE 5. ACC (%) of different feature selection algorithms, over a varied number of features.

using ISGFS; (2) The Word2Vec vectorization method supports classifiers to achieve better classification results; (3) ISGFS improves the classification results in all cases; (4) The LSTM network has shown the best classification accuracy, better than CNN and conventional methods; and (5) The LSTM and the logistic regression model show competitive results when combined with Word2Vec and ISGFS feature selection algorithms.

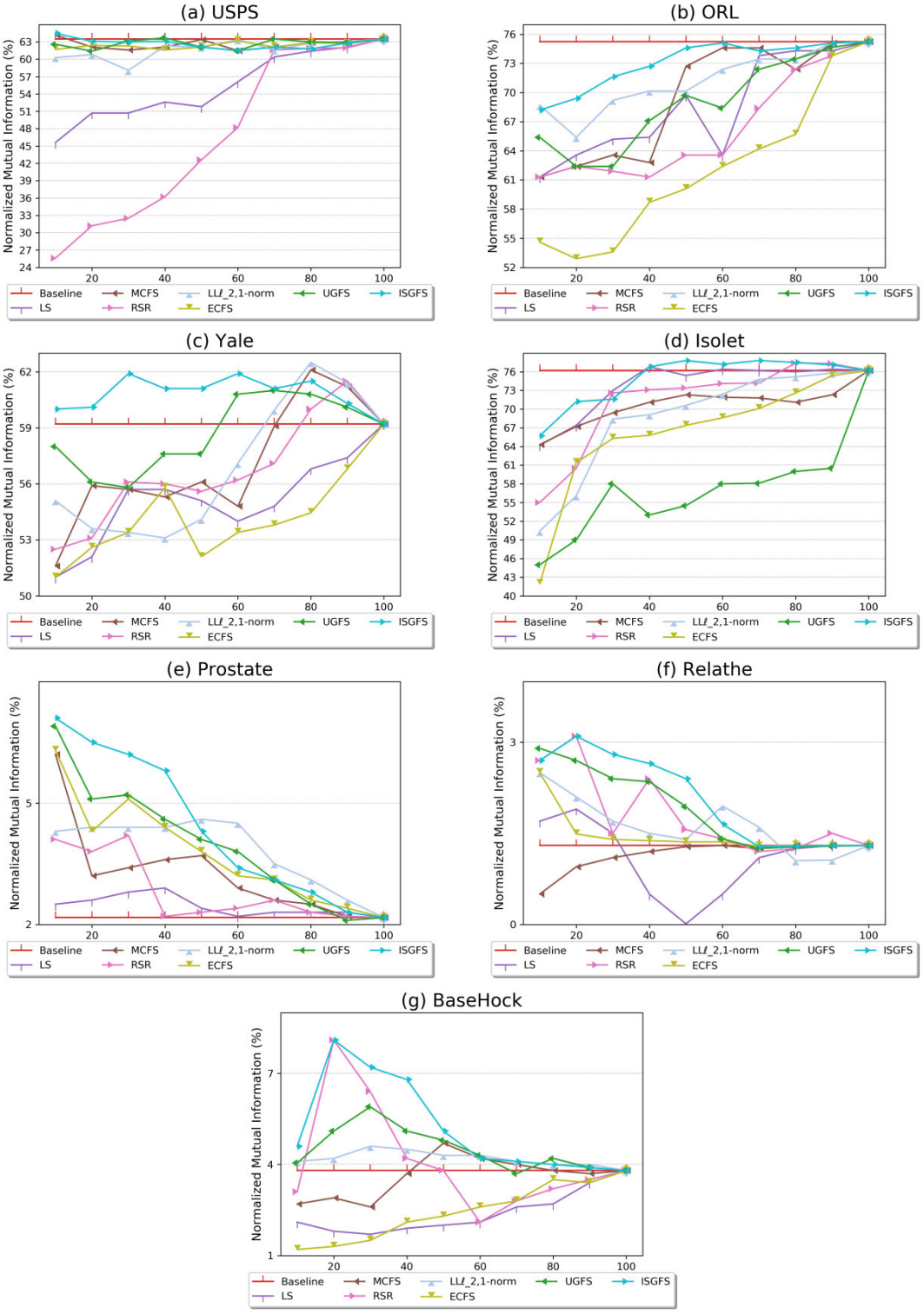

FIGURE 6. NMI (%) of different feature selection algorithms, over a varied number of features.

In summary: Although it can be slower than a few other methods, feature selection by ISGFS is better able to identify relevant class separation features in the presence of adverse conditions common in practice, such as feature redundancy, unbalanced class data amounts and densities, and noisy data.

TABLE 7. Comparison of different classifiers (conventional and deep) for sentiment analysis of US Twitter Airline dataset.

Classifiers	ACC	Precision	Recall
TF-IDF vectorizer + Conventional classifiers			
Logistic Regression	79.96%	72.12%	75.34%
Random Forest	70.74%	69.12%	71.53%
ISGFS + Logistic Regression	81.68%	76.45%	76.17%
ISGFS + Random Forest	76.74%	70.28%	71.93%
Word2vec vectorizer + Conventional classifiers			
Logistic Regression	82.34%	74.38%	76.18%
Random Forest	72.53%	69.83%	71.91%
ISGFS + Logistic Regression	**84.38**%	78.31%	77.16%
ISGFS + Random Forest	79.644%	71.32%	72.53%
Deep learning classifiers			
LSTM	**84.42**%	78.66%	77.3%
CNN	78.23%	71.18%	72.29%

VI. CONCLUSION

The purpose of this study was to develop and investigate an unsupervised feature selection method based on subspace learning and graph analysis. The proposed method identifies feature relationships through cluster density properties and subspace learning. Feature relationships are used to design a feature selection graph weighed by feature correlations. PageRank centrality ranks features by their ability to separate clusters, and feature correlations eliminate redundant features. The richer analysis of data in this method, especially data with structure and noise, as is common in the datasets of major current applications, explains in part the better results it yields. Several examples have been given, in which the proposed algorithm outperformed state-of-art methods in both classification and clustering. As future work, we plan to investigate density threshold selection and a learning method to extract feature relationships, as well as graph direction impact. We also plan to extend the proposed method to investigate gene-gene/drug-target interactions. The source code will be posted online to provide all of the material needed to reproduce our experiments.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. 2001.

[2] A. Flexer and D. Schnitzer, "Choosing ℓ^p norms in high-dimensional spaces based on hub analysis," *Neurocomputing*, vol. 169, no. 2, pp. 281–287, 2015.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2000.

[4] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[5] N. Altman and M. Krzywinski, "The curse (s) of dimensionality," *Nature Methods*, vol. 15, pp. 399–400, Jun. 2018.

[6] C. Bohm, K. Kailing, H.-P. Kriegel, and P. Kroger, "Density connected clustering with local subspace preferences," in *Proc. 4th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2004, pp. 27–34.

[7] K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong, "A survey on enhanced subspace clustering," *Data Mining Knowl. Discovery*, vol. 26, no. 2, pp. 332–397, Mar. 2013.

[8] K. Henni, O. Alata, L. Zaoui, B. Vannier, A. E. Idrissi, and A. Moussa, "ClusterMPP: An unsupervised density-based clustering algorithm via marked point process," *Intell. Data Anal.*, vol. 21, no. 4, pp. 827–847, Aug. 2017.

[9] X. Chen and X. Hao, "Feature reduction method for cognition and classification of IoT devices based on artificial intelligence," *IEEE Access*, vol. 7, pp. 103291–103298, 2019.

[10] J. Chen, S. Ji, B. Ceran, Q. Li, M. Wu, and J. Ye, "Learning subspace kernels for classification," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, p. 106.

[11] S. Ding, H. Zhu, W. Jia, and C. Su, "A survey on feature extraction for pattern recognition," *Artif. Intell. Rev.*, vol. 37, no. 3, pp. 169–180, 2012.

[12] P. Sun, D. Wang, V. C. Mok, and L. Shi, "Comparison of feature selection methods and machine learning classifiers for Radiomics analysis in Glioma grading," *IEEE Access*, vol. 7, pp. 102010–102020, 2019.

[13] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *J. Biomed. Informat.*, vol. 85, pp. 189–203, Sep. 2018.

[14] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010.

[15] K. Henni, N. Mezghani, and C. Gouin-Vallerand, "Unsupervised graph-based feature selection via subspace and pagerank centrality," *Expert Syst. Appl.*, vol. 114, pp. 46–53, Dec. 2018.

[16] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011, pp. 1–6.

[17] Y. Guo, F.-L. Chung, G. Li, and L. Zhang, "Multi-label bioinformatics data classification with ensemble embedded feature selection," *IEEE Access*, vol. 7, pp. 103863–103875, 2019.

[18] N. Zhou, Y. Xu, H. Cheng, J. Fang, and W. Pedrycz, "Global and local structure preserving sparse subspace learning: An iterative approach to unsupervised feature selection," *Pattern Recognit.*, vol. 53, pp. 87–101, May 2016.

[19] S. Du, Y. Ma, S. Li, and Y. Ma, "Robust unsupervised feature selection via matrix factorization," *Neurocomputing*, vol. 241, pp. 115–127, Jun. 2017.

[20] S. Feng and M. F. Duarte, "Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation," *Neurocomputing*, vol. 312, pp. 310–323, Oct. 2018.

[21] G. Roffo and S. Melzi, "Ranking to learn: Feature ranking and selection via eigenvector centrality," in *Proc. Int. Workshop New Frontiers Mining Complex Patterns*, in Lecture Notes in Computer Science, 2017, pp. 19–35.

[22] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, "Enhancing density-based clustering: Parameter reduction and outlier detection," *Inf. Syst.*, vol. 38, no. 3, pp. 317–330, May 2013.

[23] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "An efficient and scalable density-based clustering algorithm for datasets with complex structures," *Neurocomputing*, vol. 171, pp. 9–22, Jan. 2016.

[24] K. Henni, P.-Y. Louis, B. Vannier, and A. Moussa, "Is-ClusterMPP: Clustering algorithm through point processes and influence space towards high-dimensional data," *Adv. Data Anal. Classification*, Nov. 2019.

[25] M. Bouguessa and S. Wang, "Mining projected clusters in high-dimensional spaces," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 4, pp. 507–522, Apr. 2009.

[26] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park, "Fast algorithms for projected clustering," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 1999, pp. 61–72.

[27] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.
[28] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.
[29] W. J. Krzanowski, "Selection of variables to preserve multivariate data structure, using principal components," *J. Roy. Stat. Soc., C, Appl. Statist.*, vol. 36, no. 1, pp. 22–33, 1987.
[30] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 2005, pp. 507–514
[31] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2010, pp. 333–342.
[32] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 1–6.
[33] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *Proc. 26th AAAI Conf. Artif. Intell. Unsupervised*, 2012, pp. 1–7.
[34] J. R. Adhikary and M. N. Murty, "Feature selection for unsupervised learning," Lecture Notes in Computer Science, 2012.
[35] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. K. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognit.*, vol. 48, no. 2, pp. 438–446, Feb. 2015.
[36] D. Han and J. Kim, "Unsupervised simultaneous orthogonal basis clustering feature selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5016–5023.
[37] T. Martin, X. Zhang, and M. E. Newman, "Localization and centrality in networks," *CoRR*, 2014.
[38] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social Netw.*, vol. 32, no. 3, pp. 245–251, Jul. 2010.
[39] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton, NJ, USA: Princeton Univ. Press, 2006.
[40] P. De Meo, E. Ferrara, G. Fiumara, and A. Ricciardello, "A novel measure of edge centrality in social networks," *Knowl.-Based Syst.*, vol. 30, pp. 136–150, Jun. 2012.
[41] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
[42] G. Moise, A. Zimek, P. Kröger, H.-P. Kriegel, and J. Sander, "Subspace and projected clustering: Experimental evaluation and analysis," *Knowl. Inf. Syst.*, vol. 21, no. 3, pp. 299–326, Dec. 2009.
[43] K. Y. Yip, D. W. Cheung, M. K. Ng, and K.-H. Cheung, "Identifying projected clusters from gene expression profiles," *J. Biomed. Informat.*, vol. 37, no. 5, pp. 345–357, Oct. 2004.
[44] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data," *Data Mining Knowl. Discovery*, vol. 11, no. 1, pp. 5–33, Jul. 2005.
[45] M. Lung Yiu and N. Mamoulis, "Iterative projected clustering by subspace mining," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 2, pp. 176–189, Feb. 2005.
[46] X.-D. Wang, R.-C. Chen, F. Yan, Z.-Q. Zeng, and C.-Q. Hong, "Fast adaptive K-Means subspace clustering for high-dimensional data," *IEEE Access*, vol. 7, pp. 42639–42651, 2019.
[47] D. F. Gleich, "PageRank beyond the Web," *SIAM Rev.*, vol. 57, no. 3, pp. 321–363, Jan. 2015.
[48] J. L. Morrison, R. Breitling, D. J. Higham, and D. R. Gilbert, "GeneRank: Using search engine technology for the analysis of microarray experiments," *BMC Bioinf.*, vol. 6, no. 1, p. 233, 2005.
[49] B. Jiang, K. Kloster, D. F. Gleich, and M. Gribskov, "AptRank: An adaptive PageRank model for protein function prediction on bi-relational graphs," *Bioinformatics*, vol. 33, no. 12, pp. 1829–1836, 2017.
[50] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, nos. 1–7, pp. 107–117, Apr. 1998.
[51] A. K. Srivastava, R. Garg, and P. K. Mishra, "Discussion on damping factor value in PageRank computation," *Int. J. Intell. Syst. Appl.*, vol. 9, no. 9, pp. 19–28, Sep. 2017.
[52] P. Boldi, "TotalRank: Ranking without damping," in *Proc. Special Interest Tracks Posters 14th Int. Conf. World Wide Web (WWW)*, New York, NY, USA, 2005, pp. 898–899.
[53] R. Gross, "Face databases," in *Handbook of Face Recognition*. 2005, p. 22.
[54] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization," in *Proc. 25th Conf. Uncertainty Artif. Intell. (UAI)*, Arlington, VA, USA, 2009, pp. 339–348.
[55] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.
[56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[57] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1746–1751.

KHADIDJA HENNI received the Ph.D. degree in computer science from the Oran University of Science and Technology, Oran, Algeria, in February 2017.

She joined the LICEF Research Center, TELUQ University, in June 2017, as a Postdoctoral Fellow. She is currently a Professor with Université TELUQ (Quebec University) and a Researcher with the LICEF Institute. Her current research interests include clustering, classification, feature selection, and bioinformatic.

NEILA MEZGHANI received the degree in telecommunications engineering from the Higher School of Telecommunications of Tunis (Sup'Com), the master's degree in information technology from the National School of Engineers of Tunis, and the Ph.D. degree from the Institut National de Recherche Scientifique—Centre Énergie Matériaux et Télécommunications de Montréal.

She is currently a Data Scientist Professor with Université TELUQ (Quebec University) and a Researcher with the Centre de recherche du Centre hospitalier de l'Université de Montréal (CR-CHUM). She is the author of two patents and about 100 peer-reviewed publications in renowned scientific journals and international conferences. Her research interests include biomedical data mining and classification, artificial intelligence, decision support systems in the medical field, and mobile health. She has the Canada Research Chair in biomedical data mining.

AMAR MITICHE received the Licence Es Sciences degree in mathematics from the University of Algiers and the Ph.D. degree in computer science from the University of Texas at Austin. He is currently a Professor with the Department of Telecommunications (INRS-EMT), Institut National de la Recherche Scientifique (INRS), Montreal, QC, Canada. His research is in computer vision and pattern recognition. He has written several articles on the subjects, as well as three books: *Computational Analysis of Visual Motion* (Plenum Press, 1994), *Variational and Level Set Methods in Image Segmentation* (Springer, 2011), with Ismail Ben Ayed, and *Computer Vision Analysis of Image Motion by Variational Methods* (Springer, 2014), with J. K. Aggarwal. His current interests include image segmentation, image motion analysis, and pattern classification by neural networks.