Unsupervised Feature Selection for Outlier Detection by Modelling Hierarchical Value-Feature Couplings

Guansong Pang, Longbing Cao, Ling Chen
School of Software
University of Technology Sydney
Ultimo, NSW 2007, Australia
{Guansong.Pang,Longbing.Cao,Ling.Chen}@uts.edu.au

Huan Liu
Computer Science and Engineering
Arizona State University
Tempe, AZ 85287, USA
Huan.Liu@asu.edu

Abstract—**Proper feature selection for unsupervised outlier detection can improve detection performance but is very challenging due to complex feature interactions, the mixture of relevant features with noisy/redundant features in imbalanced data, and the unavailability of class labels. Little work has been done on this challenge. This paper proposes a novel Coupled Unsupervised Feature Selection framework (CUFS for short) to filter out noisy or redundant features for subsequent outlier detection in categorical data. CUFS quantifies the outlierness (or relevance) of features by learning and integrating both the *feature value couplings* and *feature couplings*. Such value-to-feature couplings capture intrinsic data characteristics and distinguish relevant features from those noisy/redundant features. CUFS is further instantiated into a *parameter-free* Dense Subgraph-based Feature Selection method, called DSFS. We prove that DSFS retains a *2-approximation* feature subset to the optimal subset.**

Extensive evaluation results on 15 real-world data sets show that DSFS obtains an average 46% feature reduction rate, and enables three different types of pattern-based outlier detection methods to achieve substantially better AUC improvements and/or perform orders of magnitude faster than on the original feature set. Compared to its feature selection contender, on average, all three DSFS-based detectors achieve more than 20% AUC improvement.

I. Introduction

Outliers are usually rare, i.e., those objects with rare combinations of feature values, compared to the majority of objects. Unsupervised outlier detection in categorical data is essential for broad applications in various domains, such as fraud detection, insider trading, intrusion detection and terrorist detection. In these cases, categorical features are uniquely available or indispensable in data objects [1].

Unsupervised outlier detection faces typical challenges such as sophisticated interactions within and between features, the mixture of relevant features with noisy/redundant features, and the extreme imbalance between normal and outlying objects. In such a complex problem nature, outliers are easily masked as normal objects in *noisy features* - features for which normal objects contain infrequent behaviours while outliers contain frequent behaviours, and only detectable in a subset of features [2], [3]. For example, in fraud detection, suspects may be spotted by partial features, such as marital status, income level and professional, while they may fake themselves as normal with other features, such as gender and education. In addition, many categorical data sets contain a large number of *redundant features* - weakly relevant features that contribute very limited capability, or none, for identifying outliers when combined with other features.

In outlier detection, most unsupervised methods for categorical data (e.g., [4]–[9]) are pattern-based. They search for outlying/normal patterns and employ pattern frequency as a direct outlierness measure. However, these methods fail to perform effectively and efficiently in data sets that have the characteristics discussed above for three main reasons: (i) noisy/redundant features are deeply mixed with relevant features and make it difficult to distinguish outliers from normal objects; (ii) many noisy features mislead the pattern search and result in a large proportion of faulty patterns and a high 'false positive' rate; and (iii) feature redundancy results in numerous redundant patterns and downgrades the efficiency of the pattern search and outlier detection.

Filtering out noisy and redundant features may therefore substantially improve the effectiveness and efficiency of subsequent outlier detection. However, it is very challenging to recognise and remove these features when there are complex interactions between noisy/redundant features and relevant features in highly imbalanced data without class labels.

Little work has been designed to conduct feature selection for unsupervised outlier detection in categorical data. Most feature selection research focuses on classification, regression and clustering [10]–[12]. Existing work on feature selection for very imbalanced data, including studies in [13]–[15], concerns imbalanced classification or supervised outlier detection. The feature weighting method in [5] weights features for outlier detection in categorical data, but it evaluates individual features without considering feature interactions and fails to handle noisy features.

In this paper, we propose a novel *Coupled Unsupervised Feature Selection* framework (CUFS for short) to filter out noisy and redundant features that are coupled with relevant features for outlier detection in categorical data. CUFS first estimates the outlierness of feature values by modelling the low-level intra- and inter-feature value couplings. These *value couplings* reflect the intrinsic data characteristics and facilitate the differentiation between relevant and other features. We further incorporate the value-level outlierness into *feature outlierness* by learning *value-to-feature interactions*. This *value-to-*

feature outlierness is then mapped onto graph representations, on which existing graph mining techniques will be used to identify the desirable relevant feature subset.

We further instantiate CUFS to a *Dense Subgraph-based Feature Selection* method called DSFS, which synthesises the advantages of hierarchical couplings captured in CUFS and the dense subgraph search theories. DSFS computes value outlierness by integrating intra-feature value frequency deviation and inter-feature value correlation and obtains feature outlierness by a linear combination of value outlierness. The max-relevance feature subset evaluation criterion, which is equivalent to the maximum subgraph density of a feature graph, and sequential search strategy are then used to identify the relevant feature subset.

This work makes the following major contributions.

1) We propose a novel and flexible coupled unsupervised feature selection (CUFS) framework for detecting outliers in categorical data, in which relevant features are highly mixed with noisy and redundant features. CUFS captures complex feature interactions by modelling the outlierness (relevance) of features w.r.t. hierarchically intra- and inter-feature couplings, which distinguish relevant features from noisy and redundant features.
2) The performance of CUFS is verified by its instance, i.e., a *parameter-free* feature subset selection method DSFS. We prove that the feature subset selected by DSFS has a *2-approximation* to the optimal subset. This demonstrates the flexibility of CUFS in enabling state-of-the-art graph mining techniques to tackle the feature selection challenge in unlabelled and imbalanced categorical data.

Extensive experiments show that (1) DSFS obtains a large average feature reduction rate (46%) on 15 data sets with a variety of complexities, including different levels of noisy and redundant features, and greatly improves three different types of pattern-based outlier detectors in AUC and/or runtime performance; (2) DSFS substantially defeats its feature weighting-based contender (maximally 94% improvement on a data set); and (3) DSFS achieves good scalability w.r.t. data size (linear to data size, completing execution within one second for a data set with over one million objects) and the number of features (completing the execution within 20 seconds for a data set with over 1000 features).

The rest of this paper is organised as follows. We discuss related work in Section II. CUFS is detailed in Section III. DSFS is introduced in Section IV. Empirical results are provided in Section V. We conclude this work in Section VI.

II. Related Work

Numerous outlier detection methods have been introduced, e.g., distance-based methods, clustering-based methods, and density-based methods [1], but most of them are proximity-based and require a distance/similarity measure. Consequently, they are ineffective for handling data sets with many irrelevant or noisy features due to the curse of dimensionality [2], [3]. Also, it is difficult for proximity-based methods to define an effective distance measure on categorical features with unordered values [16].

Most methods for categorical data are pattern-based, to address the discrete nature of this data. They can be generally classified into three categories: association rule-based [6]–[8], information theory-based [4], [5], and probability test-based methods [9]. Typically, these methods first identify subspaces that contain normal/outlying patterns and then define an outlier score based on the pattern frequency in each subspace. Outlier scores are assigned to objects based on the summation of the outlier scores in subspaces. However, these methods identify a large proportion of misleading patterns when a data set has many noisy features, leading to a high 'false positive' rate. In addition, many pattern-based methods (e.g., [4]–[7]) have at least quadratic time complexity w.r.t. the number of features. The presence of redundant features aggravates the computational cost of pattern discovery and outlier detection whereas detectors receive no improvement in accuracy.

Feature selection has been shown critical for removing irrelevant and redundant features (note that all features that are not relevant to learning tasks are defined as irrelevant features, including noisy features [17]), but most existing methods focus on regression, classification and clustering [10]–[12]. Very few feature selection methods have been specifically designed for outlier detection. Some relate work has been on feature selection for imbalanced data classification and supervised outlier detection [13]–[15]. However, they fail in the context without class label information or being costly to obtain class labels. Unfortunately, many real-life outlier detection applications fall in this scenario.

Even less work is available on unsupervised feature selection for outlier detection. Two related studies are [18] and [5]. In [18], the partial augmented Lagrangian method simultaneously selects objects from the minority class and features that are relevant to minority class detection. While it shows to be effective in selecting features for unsupervised rare class detection, as pointed out by its authors, this method assumes that the objects of rare classes are strongly self-similar. This assumption does not apply to the nature of outlier detection, where many outliers are isolated objects and distributed far away from each other in data space. The unsupervised entropy-based feature weighting in [5] for categorical data is most closely related to this paper. It weights features and highlights strongly relevant features for subsequent outlier detection. However, as shown in the experiments, it evaluates individual features without considering underlying feature interactions, and thus wrongly treats noisy features as relevant.

Recently, learning value-to-object coupling relationships has shown valuable and been successfully applied to various problems, e.g., abnormal group behaviour detection [19] and similarity learning [20]. This work builds on their methodology to learn value-to-feature outlierness in unlabelled categorical data and integrate the outlierness with graph mining techniques to select features for unsupervised outlier detection.

III. The CUFS Framework

In this section, we introduce the CUFS framework. CUFS builds and integrates two-level hierarchical couplings, i.e., feature value couplings and feature couplings, toward a proper estimation of the feature relevance to outlier detection. Specifically, it learns the intra- and inter-feature value couplings to compute outlierness on the feature value level and constructs a *value graph* with the outlierness being the edge weights. We then feed the value graph to feature-level coupling analysis and construct a *feature graph* by aggregating the value-level outlierness. Our coupled feature selection framework for unsupervised outlier detection (i.e., CUFS) is shown in Fig. 1.

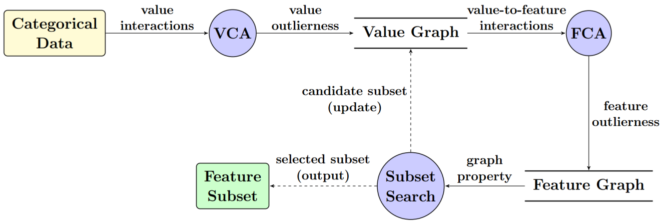

Fig. 1: The Proposed CUFS Framework. VCA and FCA are short for Value Coupling Analysis and Feature Coupling Analysis, respectively.

The value coupling analysis captures the intrinsic interactions between the values of data objects, which enables a proper estimation of the value outlierness in data and distinguish outlying values from noisy values. As the features build their capability on their values, feature outlierness is thus modelled by aggregating value outlierness in terms of the value-to-feature interactions. Such feature couplings distinguish useful features from noisy and redundant features.

As a result of these factors, CUFS builds on the deep understanding of intrinsic data characteristics in outlying data, and effectively combines the advantages of data-driven complex feature relation analysis with unsupervised feature selection and graph theories for outlier detection. It has the graph properties and a feature subset search strategy as input to search and select a feature subset for outlier detection. Table I presents major notations used throughout this paper.

TABLE I: Symbols and Definitions

Symbol	Definition
$\mathcal{X}$	A set of data objects with size $N = \lvert\mathcal{X}\rvert$
$\mathcal{F}$	The set of $D = \lvert\mathcal{F}\rvert$ categorical features in $\mathcal{X}$
$\mathcal{V}$	The whole set of feature values contained in $\mathcal{F}$
$\mathcal{S}$	Feature subset of $\mathcal{F}$ with $D' = \lvert\mathcal{S}\rvert$ features
G	Value graph in which each node is a feature value
$\mathbf{A}$	The weighted adjacent matrix of G
G^*	Feature graph in which each node is a feature
$\mathbf{A}^*$	The weighted adjacent matrix of G^*

A. Value Graph Construction

The outlying behaviours of a feature value are captured by intra-feature and inter-feature value couplings. Accordingly, we define *value couplings* and *value graph* as follows.

Definition 1 (Value Coupling): The couplings in a value v of feature f are represented by a three-dimensional tuple $VC = (f, \delta(\cdot), \eta(\cdot,\cdot))$, where

- $f \in \mathcal{F}$, where $\mathcal{F}$ is the feature space.
- $\delta(\cdot)$ captures the outlying behaviours of the value v w.r.t. the value interactions within feature f. For example, $\delta(\cdot)$ may be a function of deviations of value frequencies from the mode frequency or value similarities, etc.
- $\eta(\cdot,\cdot)$ captures the outlying behaviours of the value v w.r.t. interactions with the values in the rest of features in $\mathcal{F}$. For example, $\eta(\cdot,\cdot)$ may be a function of value co-occurrence frequency, conditional probabilities or other value correlation quantisation methods.

With the value couplings of all feature values, a *value graph* can be built to present their relationship.

Definition 2 (Value Graph): The value graph G is defined as $G =< \mathcal{V}, \mathbf{A}, g(\delta(\cdot), \eta(\cdot,\cdot)) >$, where a value $v \in \mathcal{V}$ represents a node, the entry of the weighted adjacent matrix $A(v, v')$ (i.e., edge weight) is determined by function $g(\cdot,\cdot)$, which is a joint function of $\delta(v)$ and $\eta(v, v')$, $\forall v, v' \in \mathcal{V}$.

The graph G can be an undirected or directed graph depending on how the edge weight is defined.

One major benefit of mapping the value couplings to the value graph is that we can utilise the value graph properties (e.g., ego-network, shortest path, node centrality, or random walk distance [21]) to infer deeper value interactions and to further explore feature interactions by building the following feature graph.

B. Feature Graph Construction

The feature couplings are derived from the value couplings to capture the value-to-feature interactions.

Definition 3 (Feature Coupling): The couplings within a feature f are described as a three-dimensional tuple $FC = (dom(f), \delta^*(\cdot), \eta^*(\cdot,\cdot))$, where

- $dom(f)$ is the domain of the feature f, which consists of a finite set of possible feature values contained in f.
- $\delta^*(\cdot)$ computes the outlying degree of f based on its value outlierness $\delta(\cdot)$. For example, $\delta^*(f)$ may be a linear or non-linear function for combining all $\delta(v), \forall v \in dom(f)$.
- $\eta^*(\cdot,\cdot)$ captures the outlying degree of f w.r.t. its value interactions with other features in $\mathcal{F}$. Specifically, given $\forall f' \in \mathcal{F} \setminus f$, $\eta^*(f, f')$ may be a linear or non-linear function for incorporating $\eta(v, v')$ for $\forall v \in dom(f)$ and $\forall v' \in dom(f')$.

These couplings are then mapped into a feature graph G^*.

Definition 4 (Feature Graph): The feature graph G^* is defined as $G =< \mathcal{F}, \mathbf{A}^*, h(\delta^*(\cdot), \eta^*(\cdot,\cdot)) >$, where a feature $f \in \mathcal{F}$ represents a node and the entry of the weighted adjacent matrix $A^*(f, f')$ is determined by $h(\cdot,\cdot)$, a function combining $\delta^*(f)$ and $\eta^*(f, f')$ for $\forall f, f' \in \mathcal{F}$.

With the feature graph, existing graph mining algorithms and theories (e.g., dense subgraph discovery, graph partition and frequent graph pattern mining [21]) can then be applied to identify the most relevant feature subset for outlier detection. As presented in Section IV, by utilising dense subgraph discovery theories, the CUFS instance can efficiently retain a 2-approximation feature subset.

C. Feature Subset Selection

Our goal here is to find a feature subset, i.e., a subgraph of the feature graph, which reserves feature nodes with high outlierness while at the same time reduces redundancy between the reserved features.

The feature subset searching contains two major ingredients: *search strategy* and *objective function* (i.e., subset evaluation criteria) [22]. Typical search strategies include complete search, sequential forward or backward search, and random search. *Complete search* can obtain an optimal feature subset, but its runtime is prohibitive for high-dimensional data. *Sequential search* and *random search* are heuristic and result in a suboptimal subset, but they are more practical than complete search as they have much better efficiency.

A generic objective function for this context is:

$$\max J(\mathcal{S}) \tag{1}$$

where $J(\cdot)$ is a function evaluating the outlierness in the feature subset $\mathcal{S}$, which needs to be specified based on the chosen search strategy.

As illustrated in Fig. 1, we may need to iteratively update the value graph and feature graph during the subset searching, e.g., when adding or removing features in sequential search, before obtaining an optimal subset.

IV. The CUFS Instance: DSFS

The CUFS framework can be instantiated by first specifying the three functions δ, η and g for constructing the value graph and the other three functions δ^*, η^* and h for building the feature graph. A subset search strategy can then be formed by utilising the graph properties of the feature graph to identify the desired feature subset.

We illustrate the instantiation of CUFS by identifying the dense subgraph of the feature graph, i.e., DSFS. DSFS uses the recursive backward elimination search with the subgraph density as the objective function.

A. Specifying Functions δ, η and g for the Value Graph

Per the definition of outliers, the frequencies of values are closely related to the degree of outlierness. Hence, the outlierness of feature values is dependent on its intra-feature frequency distribution and inter-feature value co-occurrence frequencies. Motivated by this, we specify the intra- and inter-feature value outlierness in terms of frequency deviation and *confidence* values [23].

Definition 5 (Intra-feature Value Outlierness δ): The intra-feature outlierness $\delta(v)$ of a feature value $v \in dom(f)$ is defined as the extent to which its frequency deviates from the frequency of the mode:

$$\delta(v) = \frac{freq(m) - freq(v) + \epsilon}{freq(m)} \tag{2}$$

where m is the mode of the feature f, $freq(\cdot)$ is a frequency counting function and $\epsilon = \frac{1}{N}$.

In Equation (2), the mode frequency is used as a benchmark, and the more the frequency of a feature value deviates from the mode frequency, the more outlying the value is. We use $\epsilon = \frac{1}{N}$ to estimate the outlierness of the mode, which is proportional to the data size. $\delta(\cdot)$ makes the outlierness of values from different frequency distributions more comparable, which differs from many existing work [4], [6], [7] in which the outlierness of each pattern is measured without considering its associated frequency distributions.

Definition 6 (Inter-feature Value Outlierness η): The inter-feature outlierness $\eta(v, v')$ of a value $v \in dom(f)$ and another value $v' \in dom(f')$ is defined as follows:

$$\eta(v, v') = \delta(v) conf(v, v') \delta(v') \tag{3}$$

where $conf(v, v') = \frac{freq(v,v')}{freq(v')}$.

The component $\delta(v) conf(v, v')$ can be interpreted as the outlierness of the value v' w.r.t. its coupling with the value v. That is, a value has high outlierness if it has strong correlation with outlying values. For example, taking obese persons as outliers compared with persons of average weights, if someone has an overweight friend, his/her chance of becoming obese increases by 57% [24].

Definition 7 (Edge Weighting Function g for Value Graph G): The edge weight of the value graph G, i.e., the entry (v, v') of the weight matrix $\mathbf{A}$, is defined as follows:

$$A(v, v') = g(v, v') = \begin{cases} \delta(v), & v = v' \\ \eta(v, v'), & \text{otherwise} \end{cases} \tag{4}$$

We have $\delta(\cdot) \in (0, 1)$ and $\eta(\cdot, \cdot) \in [0, 1)$ according to Equations (2) and (3), and thus $\delta(v)\eta(v, v')\delta(v') \in [0, 1)$. That is, the edge weight would be zero iff two distinctive nodes v and v' have no association.

Note that although the two cases in Equation (4) are in slightly different ranges, they will be used independently in the next section to avoid incomparable issues. We will also discuss in Section IV-D how this function helps us to distinguish noisy features from relevant features.

Overall, the value graph G has the following properties.

1) G is a directed graph with self loops, as $A(v, v') \neq A(v', v)$ and $A(v, v) \neq 0$.
2) Its adjacent matrix $\mathbf{A}$ is a value outlierness matrix, representing outlying degree of individual values and pairs of distinctive values. The larger a matrix entry is, the higher the outlierness of the incoming value node is.

B. Specifying Functions δ^*, η^* and h for the Feature Graph

For simplicity and the consideration of common scenarios, we assume that the intra-feature and inter-feature value outlierness measures are linearly dependent. Accordingly, we estimate the intra- and inter-feature outlierness of a feature and their integration for feature-level outlierness by simply summing its associated δ and η values.

Definition 8 (Intra-feature Outlierness δ^*): The intra-feature outlierness of a feature $f \in \mathcal{F}$ is specified below:

$$\delta^*(f) = \sum_{v \in dom(f)} \delta(v) \tag{5}$$

Definition 9 (Inter-feature Outlierness η^*): The inter-feature outlierness of a feature f w.r.t. feature f' is quantified as:

$$\eta^*(f, f') = \sum_{v \in dom(f), v' \in dom(f')} \eta(v, v') \tag{6}$$

Similar to g, we specify the function h using intra-feature outlierness as diagonal entries and inter-feature outlierness as off-diagonal entries in the weight matrix $\mathbf{A}^*$.

Definition 10 (Edge Weighting Function h for Feature Graph G^*): The edge weight $A^*(f, f')$ of the feature graph G^*, i.e., the entry (f, f') of $\mathbf{A}^*$, is measured as:

$$A^*(f, f') = h(f, f') = \begin{cases} \delta^*(f), & f = f' \\ \eta^*(f, f'), & \text{otherwise} \end{cases} \tag{7}$$

Note that, to make the entries in $\mathbf{A}^*$ comparable, δ^* and η^* are normalised into the same range $[0, 1]$ for further use in feature subset searching.

The feature graph G^* has the following key properties.

1) G^* is a complete graph with self loops, as $\delta^*(\cdot) > 0$ and $\eta^*(\cdot, \cdot) > 0$.
2) G^* is an undirected graph, as we always have $A^*(f, f') = A^*(f', f)$ for $\forall f', f \in \mathcal{F}$.
3) Its adjacent matrix $\mathbf{A}^*$ is a feature outlierness matrix, representing outlying degree of features and their combinations. Larger values in $\mathbf{A}^*$ indicate higher outlierness.
4) The total edge weight of a feature node f is large if both of its intra- and inter-feature outlierness are high.

C. The Search Strategy

Our target is to find a subset of features with the highest relevance to outlier detection, i.e., with the highest outlierness. A feature has high outlierness if it has large edge weights in the feature graph G^*, according to the properties (3) and (4) of G^*. However, simply selecting the top-ranked k features does not necessarily obtain the best feature subset, since the outlierness of a feature also depends on its coupled features. This distinguishes our design from most of the existing methods, which overlook hierarchical feature interactions.

Motivated by the *max-relevance* idea in [25], the following *max-relevance objective function* is designed to search for the most relevant feature subset $\mathcal{S}$.

$$\max \frac{1}{|\mathcal{S}|} \sum_{f \in \mathcal{S}} \sum_{f' \in \mathcal{S}} A^*(f, f') \tag{8}$$

In other words, we specify $J(\cdot)$ in Equation (1) as $J(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{f \in \mathcal{S}} \sum_{f' \in \mathcal{S}} A^*(f, f')$.

Searching the exact $\mathcal{S}$ is computationally intractable for high dimensional (D) data, as the search space is 2^D. A heuristic sequential search strategy, namely Recursive Backward Elimination (RBE), is used to search for an approximately best subset. RBE conducts an iterative search as shown in Algorithm 1. In the next section, we prove that the resultant subset is a 2-approximation to the optimum.

Algorithm 1 *RBE ($\mathcal{F}$)*

```
Input: $\mathcal{F}$ - full feature set
Output: $\mathcal{S}$ - the feature subset selected
1: while $|\mathcal{F}| > 0$ do
2:     for $f \in \mathcal{F}$ do
3:         Compute $J(\mathcal{F} \setminus f)$
4:     end for
5:     Remove the feature $f$ that results in the largest $J(\mathcal{F} \setminus f)$
6: end while
7: return Return the subset with the largest $J(\cdot)$ as $\mathcal{S}$
```

D. Theoretical Analysis

Theoretical analysis is provided for DSFS in the first subsection and we then discuss why DSFS can handle noisy and redundant features in the remaining two subsections.

1) Approximation: Following the definition of subgraph density for *unweighted graphs* in [26], [27], we define the subgraph density for *weighted graphs* by replacing the total number of edges with the total weight defined in our graph.

Definition 11 (Subgraph Density): The density of an undirected weighted subgraph $\mathcal{S}$ is its average weighted degree:

$$den(\mathcal{S}) = \frac{vol(\mathcal{S})}{|\mathcal{S}|} \tag{9}$$

where $vol(\mathcal{S}) = \frac{\sum_{f \in \mathcal{S}} \sum_{f' \in \mathcal{S}} A^*(f, f')}{2}$ is the volume of $\mathcal{S}$.

With Equations (8) and (9), we have the following lemma.

Lemma 1 (Equivalence to the Densest Subgraph Discovery): Equation (8) is equivalent to calculating the maximum of $den(\mathcal{S})$, i.e., the densest subgraph of the feature graph G^*.

Proof: It is easy to see that Equation (8) is equivalent to $2den(\mathcal{S})$, and thus the densest subgraph of G^* is the exact solution $\mathcal{S}$ to Equation (8). ■

We show below that the RBE search with quadratic time complexity can be simplified to an equivalent procedure with linear time complexity. Following theorems of dense subgraph discovery in unweighted graphs [26], [27], we further prove that the RBE search on the weighted graph G^* achieves a feature subset with a 2-approximation to the optimum.

Lemma 2 (Search Strategy Equivalence): Steps (2-5) of RBE in Algorithm 1 are equivalent to the removal of the feature node f with the smallest weighted degree.

Proof: If the feature node f has the smallest weighted degree, then $\sum_{f' \in \mathcal{F} \setminus f} \sum_{f'' \in \mathcal{F} \setminus f} A^*(f', f'')$ is the largest in the current iteration. Since $\frac{1}{|\mathcal{F} \setminus f'|}$ is the same $\forall f' \in \mathcal{F}$, the removal of f results in the largest $J(\cdot)$. ■

Instead of recursively computing $J(\cdot)$ for each feature in each iteration, we therefore remove the feature node with the smallest weighted degree to achieve the same result, which avoids the inner loop and has linear time complexity.

Theorem 1 (2-Approximation): The feature subset $\mathcal{S}$ created by the RBE search is a 2-approximation to the optimal subset.

Proof: Let $\mathcal{S}_{opt}$ be the set of feature nodes in the densest subgraph. According to Lemma 1, below we show $den(\mathcal{S}) \geq \frac{den(\mathcal{S}_{opt})}{2}$ to prove the theorem.

Since $\mathcal{S}_{opt}$ forms the densest subgraph, we have

$$den(\mathcal{S}_{opt}) = \frac{vol(\mathcal{S}_{opt})}{|\mathcal{S}_{opt}|} \geq \frac{vol(\mathcal{S}_{opt}) - d(f)}{|\mathcal{S}_{opt}| - 1},\ \forall f \in \mathcal{S}_{opt}$$

, where $d(f) = \sum_{f' \in \mathcal{S}_{opt}} A^*(f, f')$ denotes the weighted degree of a feature node. After some replacements we have $d(f) \geq den(\mathcal{S}_{opt})$, $\forall f \in \mathcal{S}_{opt}$, i.e., every node in $\mathcal{S}_{opt}$ has weighted degree at least $den(\mathcal{S}_{opt})$.

Let $\mathcal{T}_i$ be the set of feature nodes left after the i-th node is removed. Considering the iteration of RBE, let $\mathcal{T}_j$ be the set of remaining nodes when the first node f contained in the optimal subset $\mathcal{S}_{opt}$ is removed, so $\mathcal{T}_{j-1}$ is the set of remaining nodes before the node f is removed, which indicates that $d(f) \geq den(\mathcal{S}_{opt})$, $\forall f \in \mathcal{T}_{j-1}$, according to Lemma 2. Since G^* is a complete graph with self loops, we have

$$2vol(\mathcal{T}_{j-1}) \geq den(\mathcal{S}_{opt})|\mathcal{T}_{j-1}| + \sum_{f \in \mathcal{T}_{j-1}} A^*(f, f)$$

. We then have

$$\frac{vol(\mathcal{T}_{j-1})}{|\mathcal{T}_{j-1}|} \geq \frac{den(\mathcal{S}_{opt})}{2} + \frac{\sum_{f \in \mathcal{T}_{j-1}} A^*(f, f)}{2|\mathcal{T}_{j-1}|}$$

, and accordingly

$$den(\mathcal{T}_{j-1}) = \frac{vol(\mathcal{T}_{j-1})}{|\mathcal{T}_{j-1}|} \geq \frac{den(\mathcal{S}_{opt})}{2}$$

. Since RBE returns the feature subset $\mathcal{S}$ with the largest subgraph density over all iterations and $\mathcal{T}_{j-1}$ is one of the feature subset candidates, $den(\mathcal{S})$ has at least $\frac{den(\mathcal{S}_{opt})}{2}$. ■

2) Handling Noisy Features: According to Equation (4), a value node has high outlierness if δ and η are high. Given a noisy feature value that occurs infrequently but is contained by normal objects, since it has low frequency, its intra-feature value outlierness δ is high. However, since these noisy values tend to be more frequently or only contained by normal objects, they are presumed to have stronger couplings with normal values versus weak/no couplings with outlying values. On the other hand, truly outlying values have high outlierness in terms of both δ and η, because the frequency is low and the couplings with other outlying values are strong, and thus the overall value outlierness is often much higher than that of noisy feature values. Since the intra- and inter-feature outlierness is linearly correlated to intra- and inter-feature value outlierness respectively, the intra- and inter-feature outlierness of outlying features is also higher than that of noisy features. As a result, the noisy features are removed during the iterative procedure in RBE, while the relevant features are reserved in order to maximise $J(\cdot)$.

3) Handling Redundant Features: Many redundant features are often only weakly relevant to outlier detection and have very limited or no capability for identifying outliers when they are combined with strongly relevant features. In other words, their intra-feature outlierness is quite high, but its inter-feature outlierness is low. This results in a low overall feature outlierness, and consequently these features are not retained in $\mathcal{S}$ since all the features in $\mathcal{S}$ have high outlierness.

Algorithm 2 *DSFS ($\mathcal{X}$)*

```
Input: $\mathcal{X}$ - data objects
Output: $\mathcal{S}$ - the feature subset selected
 1: Initialise $\mathbf{A}$ as a $|V| \times |V|$ matrix
 2: for $f \in \mathcal{F}$ do
 3:     Compute $\delta(v)$ for each $v \in dom(f)$
 4:     for $f' \in \mathcal{F}$ do
 5:         $A(v, v') \leftarrow g(v, v'),\ \forall v' \in dom(f')$
 6:     end for
 7: end for
 8: Initialise $\mathbf{A}^*$ as a $|D| \times |D|$ matrix
 9: for $f \in \mathcal{F}$ do
10:     for $f' \in \mathcal{F}$ do
11:         $A^*(f, f') \leftarrow h(f, f')$
12:     end for
13: end for
14: Set $\mathcal{S} \leftarrow \mathcal{F}$ and $s \leftarrow den(A^*)$
15: for $i = 1$ to $D$ do
16:     Find $f$ that has the smallest weighted degree in $\mathbf{A}^*$
17:     $\mathcal{F} \leftarrow \mathcal{F} \setminus f$ and update $\mathbf{A}^*$
18:     $\mathcal{S} \leftarrow \mathcal{F}$ and $s \leftarrow den(\mathbf{A}^*)$ if $s \leq den(\mathbf{A}^*)$
19: end for
20: return $\mathcal{S}$
```

E. The DSFS Algorithm

Algorithm 2 presents the procedures of the proposed instantiation DSFS. Steps (1-7) and (8-13) construct the value graph G and the feature graph G^*, respectively. Steps (14-19) obtain the feature subset $\mathcal{S}$. As proved in Lemma 2, Steps (16-17) are equivalent to Steps (2-5) in RBE in Algorithm 1.

DSFS requires only one database scan to compute the intra- and inter-feature value outlierness in Steps (1-7), and thus has $O(N)$. DSFS has $O(D^2)$, as two loops are required in order to generate the adjacent matrices of the value graph and the feature graph. However, the computation within the inner loop, i.e., Steps (5) and (11), is a very simple multiplication and value assignment, enabling it to complete the execution quickly in high dimensional data. Hence, DSFS has good scalability w.r.t. data size and the number of features.

V. Experiments and Evaluation

A. Data Sets

Due to the lack of benchmark data for performance evaluation in outlier detection, balanced or imbalanced classification data is transformed to create outlier and normal classes for the evaluation in literature [4], [28]–[30]. Following the above, 15 publicly available real-world data sets [1] are used, which cover

[1] aPascal and CelebA are available at http://vision.cs.uiuc.edu/attributes/ and http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html, respectively. Sylva is available at http://www.agnostic.inf.ethz.ch/datasets.php. The other 12 data sets are from the UCI machine learning repository at http://archive.ics.uci.edu/ml/.

diverse domains, e.g., intrusion detection, image object recognition, advertising and marketing, population and ecological informatics, as shown in Table II. Eleven of these data sets are directly transformed from highly imbalanced data, where the smallest class is treated as outliers and the rest of classes as normal [28], [30]. For the other four data sets, *Probe* and *U2R* are derived from the KDDCUP99 data sets which integrates multiple types of *probing* and *user-to-root* attacks as outliers; following [4], [29], we transform two balanced classification data sets (i.e., *Mushroom*, and *Optdigits* with classes '1' and '7') by sampling a small subset of the small class as outliers, resulting in 5% outliers in the created data sets. The above transformation methods guarantee that the outlier class chosen is either a rare class or a class with outlying semantics.

All data sets are used with categorical features only. Features with only one feature value are removed. Missing values are replaced with the mode.

B. Baselines and Settings

We first evaluate the feature selection method DSFS by examining its capability of improving the effectiveness and efficiency of unsupervised outlier detectors. Three different types of representative pattern-based outlier detection methods, MarP [9], Comprex (denoted by COMP) [4] and FPOF [6], are compared.

- MarP is a probabilistic method. It uses the inverse of marginal probabilities of feature values of individual features as an outlier measure. It has linear time complexity w.r.t. the number of features and is parameter-free.
- COMP is an information-theory-based method. It combines minimum description length models with information gain to automatically partition the features and builds coding tables based on feature groups to detect objects with high compression cost as outliers. It has quadratic time complexity w.r.t. the number of features and requires no parameter settings.
- FPOF is an association rule-based method. It uses the inverse of the frequencies of frequent patterns as an outlier measure. It has exponential time complexity w.r.t. the number of features. Following [6], FPOF is set with the minimum *support* threshold $supp = 0.1$ and the maximum pattern length $l = 5$.

We further compare DSFS with the entropy-based feature weighting method (denoted by ENFW) [5] for outlier detection by the above three detectors. Feature weighting methods only assign relevance weights to features and require a decision threshold to select a feature subset. To have a fair comparison, the top-ranked k features are selected, where k is the number of features in the feature subset selected by DSFS.

The scalability of DSFS w.r.t. data size and the number of features is evaluated on six subsets of the two UCI data sets *LINK* and *AD*, which have the largest number of objects and features in our data sets. For LINK, the smallest subset contains 1,000 objects, and subsequent subsets are increased by a factor of four until the largest subset which contains 1,024,000 objects. For AD, the data with the smallest feature subset contains 40 features, and subsequent subsets are increased by a factor of two, until the largest feature subset which contains 1,280 features.

DSFS, ENFW, FPOF and MarP were implemented in JAVA in WEKA [31]. COMP was obtained from the authors of [4] in MATLAB. All the experiments were performed at a node in a 3.4GHz Phoenix Cluster with 32GB memory.

C. Performance Evaluation Method

We measure the detector effectiveness in terms of the area under ROC curve (AUC). All the three outlier detectors assign an outlier score to each data object and thus rank all objects w.r.t. their degree of outlierness. AUC is then computed based on the ranking using the Mann-Whitney-Wilcoxon test [32]. Higher AUC indicates better detection accuracy.

The unsupervised detectors are trained and evaluated on the same data set, but the class labels are not employed in training; rather they are used in testing for computing AUC.

The runtime of feature selection and outlier detection is recorded to evaluate their efficiency. Here *runtime* is the time for executing the core algorithms, excluding the runtime for data loading and outputting results.

Two *data indicators* are introduced to describe the underlying data characteristics, which are sensitive to the performance of learning methods. They provide some insights into our design, and their quantisation is reported in Table II.

- Feature noise level κ_{nos}. Based on the AUC measured by using MarP for each feature, a feature is regarded as noisy if AUC is less than 0.5. We report the percentage of noisy features as κ_{nos}.
- Feature redundancy level κ_{rdn}. Features are retained if their corresponding AUC is more than 0.5 (i.e., redundant features need to be relevant features). The pairs of selected features are checked to compare the AUC by using pairwise feature combinations with that using individual features. One feature is thought to be redundant to another if the AUC difference is less than 0.01. We report the percentage of such combinations as κ_{rdn}.

Having an accurate estimation of the data complexity itself is a very challenging task. Although the above two indicators are based on low-order information only, they assist us in understanding data complexity and our empirical results.

D. Findings and Analysis

The feature selection results are presented in the first subsection. The next two subsections discuss the AUC performance and runtime of three outlier detectors with or without using DSFS and compare DSFS with its contender ENFW, respectively. Lastly, a scale-up test is conducted.

1) Large Average Feature Reduction Rate: We record the number of selected features by DSFS, D', and the *reduction rate*, RED. The reduction rate is defined as the rate of the reduced number of features in the feature subset selected by DSFS to that in the full feature set, which is shown in the last column in Table II. The results show that DSFS leads to a significant reduction rate, ranging from 13% up to 97% across 15 data sets. On average, DSFS obtains 46% reduction rate.

The two data indicators κ_{nos} and κ_{rdn} demonstrate that nearly all data sets have a large proportion of noisy or redundant features. These noisy and redundant features make the three types of pattern-based outlier detectors less effective and efficient. We show in the next section that proper feature selection is essential for enabling the detectors to handle the data complexities.

2) Improving Three Different Types of Pattern-based Outlier Detectors in AUC and/or Efficiency: The AUC performance and runtime of three detectors: MarP, COMP and FPOF compared with their editions by incorporating DSFS: MarP*, COMP* and FPOF* are presented in Table III [2]. On average, MarP*, COMP* and FPOF* obtain 7%, 4% and 4% AUC improvements respectively while they only use averagely 54% features compared to their counterparts. In particular, MarP* achieves maximally 42% improvement on aPascal, COMP* makes maximally 33% improvement on aPascal, and FPOF* gains maximally 18% on Census. It is interesting to see that less improvement is made on UCI data sets, which is understandable as UCI data sets tend to be highly manipulated and simpler.

TABLE II: Feature Selection Results on Data Sets with Different Characteristics. The data sets are sorted by κ_{nos}. The middle horizontal line roughly separates data sets with many noisy features (i.e., $\kappa_{nos} > 35\%$) from other data sets. $RED = \frac{D-D'}{D}$ (%) denotes the reduction rate by DSFS. N is the number of data objects in a data set, D is the number of features, and D' is the number of reserved features by DSFS. Avg. indicates the averaged figure.

Data Set	**Acronym**	κ_{nos}	κ_{rdn}	N	D	D'	**RED**
BankMarketing	BM	90%	0%	41188	10	7	30%
aPascal	-	81%	0%	12695	64	20	69%
Sylva	-	78%	0%	14395	87	66	24%
Census	-	58%	0%	299285	33	10	70%
CelebA	-	49%	4%	202599	39	34	13%
CMC	-	38%	4%	1473	8	5	38%
CoverType	CT	34%	22%	581012	44	5	89%
Chess	-	33%	0%	28056	6	4	33%
U2R	-	17%	7%	60821	6	3	50%
SolarFlare	SF	9%	0%	1066	11	8	27%
Optdigits	DIGIT	8%	26%	601	64	46	28%
Mushroom	MRM	5%	2%	4429	22	13	41%
Advertisements	AD	5%	78%	3279	1555	49	97%
Linkage	LINK	0%	0%	5749132	5	4	20%
Probe	-	0%	7%	64759	6	2	67%
Avg.		34%	10%	470986	131	18	46%

With regard to efficiency, MarP*, COMP* and FPOF* run orders of magnitude faster than their counterparts as they work on the highly reduced feature subsets. For example, FPOF* runs six orders of magnitude faster than FPOF on *CT*. DSFS enables COMP and FPOF to perform outlier detection on high dimensional data, such as *Sylva* with 87 features and *AD* with 1555 features, where these detectors are otherwise prohibitive in terms of runtime and/or space requirements.

A more *straightforward benefit* is that the simplest detector MarP empowered by DSFS can obtain the AUC performance that is the same as, or very competitive with, that of the two other detectors COMP and FPOF, while at the same time saving several orders of magnitude runtime. However, COMP and FPOF have much higher time and spaces complexities in complex data sets (e.g., *aPascal*, *CMC* and *AD*).

Next two subsections further explore the performance of these three detectors in data sets with many noisy or redundant features, respectively.

2.1) Substantially Enhancing both AUC and Runtime on Data Sets with High Feature Noise Level: In data with many noisy features, e.g., *BM* (90% w.r.t. κ_{nos}), *aPascal* (81%), *Sylva* (78%), *Census* (58%), *CelebA* (49%) and *CMC* (38%) (see Table II), on average, DSFS removes 43% features and enables MarP, COMP and FPOF to respectively obtain 14%, 10% and 9% AUC improvements as shown in Table III, compared to their counterparts. This is because DSFS successfully removes many noisy features from these highly noisy data, and enables pattern-based detectors to work on much cleaner data, which thus perform more effectively.

In other data sets (e.g., *Sylva* and *CelebA*) where feature reduction rates are smaller, resulting in a number of noisy features retained in the selected feature subset, it is very difficult to separate them from the relevant features. As a result, the detectors make very limited, or none, AUC improvements. This shows that such tough noisy features are deeply mixed with the outlier-discriminative features, and generate higher outlierness than truly outlying features. In these cases, it is too difficult for DSFS to distinguish them from outlying features.

In addition to the AUC improvement, the DSFS-enabled detectors can also have a significant speedup due to the significant feature reduction rate, e.g., FPOF runs 409 times slower than FPOF* on *Census*.

2.2) Achieving a Substantial Speedup on Data Sets with High Feature Redundancy Level: In data sets with a high feature redundancy level, e.g., *CT* (22% w.r.t. κ_{rdn}) and *AD* (78% w.r.t. κ_{rdn}), DSFS generates a very aggressive feature reduction, removing 89% and 97% features, respectively. Although this massive feature reduction might result in little loss in terms of AUC, e.g., 1% on *CT*, the outlier detectors can obtain up to six orders of magnitude speedup by working on a substantially smaller feature set, e.g., FPOF on *CT* and COMP on *AD*. On the other hand, MarP using DSFS obtains 6% AUC improvement on *AD* even if it works on the data with only 3% original features left.

For data sets such as *U2R*, *SF*, *MRM*, *Probe* and *LINK*, the reduction rates are more than the sum of κ_{nos} and κ_{rdn}. It should be noted that we only have a conservative estimation of κ_{nos} and κ_{rdn}, so the true feature noise and redundancy levels might be much higher than the estimated values. This

[2] All runtime refers to the runtime of the detectors only, excluding that of DSFS, but our empirical results show that the runtime of DSFS is within one second in most data sets and that is almost negligible in practice.

TABLE III: AUC and Runtime of the Three Detectors with or without DSFS. Three baseline detectors are MarP, COMP and FPOF. Their editions using DSFS are MarP*, COMP* and FPOF*, respectively. IMP and SU indicate the AUC improvement and runtime speedup of the detectors combined with DSFS. Avg. indicates the averaged figure.

	AUC Performance									**Runtime (s)**								
	MarP	MarP*	IMP	COMP	COMP*	IMP	FPOF	FPOF*	IMP	MarP	MarP*	SU	COMP	COMP*	SU	FPOF	FPOF*	SU
BM	0.56	0.59	5%	0.63	0.64	2%	0.55	0.59	7%	0.17	0.15	1	212.46	170.43	1	0.85	0.57	1
aPascal	0.62	0.88	42%	0.66	0.88	33%	○	0.88	○	0.31	0.12	3	451.36	41.00	11	○	53.29	○
Sylva	0.96	0.96	0%	0.95	0.96	1%	○	○	○	0.21	0.20	1	1137.07	498.59	2	○	○	○
Census	0.59	0.69	17%	0.64	0.71	11%	0.61	0.72	18%	1.62	0.51	3	18174.49	12878.14	1	30790.78	75.23	409
CelebA	0.74	0.74	0%	0.76	0.76	0%	0.74	0.75	1%	0.89	0.82	1	1647.47	1169.27	1	159377.51	50188.65	3
CMC	0.54	0.66	22%	0.57	0.66	16%	0.56	0.65	16%	0.14	0.01	11	5.14	2.42	2	0.10	0.06	2
CT	0.98	0.97	-1%	0.98	0.97	-1%	0.98	0.97	-1%	3.14	0.36	9	3914.33	341.98	11	410016.55	1.09	377547
Chess	0.64	0.64	0%	0.64	0.63	-2%	0.62	0.61	-2%	0.12	0.08	1	95.35	49.30	2	0.42	0.18	2
U2R	0.88	0.92	5%	0.99	0.99	0%	0.92	0.97	5%	0.28	0.13	2	318.95	255.28	1	0.39	0.22	2
SF	0.84	0.85	1%	0.85	0.86	1%	0.86	0.86	0%	0.02	0.01	1	6.33	4.40	1	0.39	0.09	4
DIGIT	0.95	0.95	0%	0.97	0.97	0%	0.96	0.94	-2%	0.04	0.03	1	217.10	111.51	2	10196.85	31.99	319
MRM	0.89	0.89	0%	0.93	0.94	1%	0.91	0.91	0%	0.07	0.07	1	48.72	32.18	2	19.32	2.70	7
AD	0.70	0.74	6%	●	0.75	●	○	0.74	○	0.85	0.10	9	●	126.35	●	○	54088.52	○
Probe	0.98	0.98	0%	0.98	0.98	0%	0.99	0.98	-1%	0.28	0.11	3	576.08	456.00	1	0.47	0.20	2
LINK	1.00	1.00	0%	1.00	1.00	0%	1.00	1.00	0%	2.74	2.27	1	6365.26	5203.67	1	23.56	17.93	1
Avg.			7%			4%			4%			3			3			31525

'○' indicates out-of-memory exceptions.

'●' indicates that we cannot obtain the results within four weeks, i.e., 2,419,200 seconds.

explains why the three detectors empowered by DSFS can still perform equally well or very competitively on these data sets, compared to their counterparts not using DSFS.

3) Defeating the Feature Weighting-based Contender: The performance comparison between two feature selection methods ENFW and DSFS in terms of the three detectors is shown in Table IV. On average, MarP, COMP and FPOF using DSFS obtain 24%, 26% and 24% AUC improvements, compared to MarP, COMP and FPOF using ENFW, respectively. Impressively, DSFS-empowered MarP gains maximally 91% improvement over ENFW-based MarP on aPascal, DSFS-empowered COMP makes maximally 94% improvement over ENFW-based COMP on CT, and DSFS-empowered FPOF achieves maximally 91% improvement over ENFW-based FPOF on aPascal.

3.1) Beating ENFW in Data Sets with Noisy Features: We further explore the power of DSFS on noisy features. As shown in IV, DSFS generally performs much better than ENFW on almost all data sets, as long as a data set contains noisy features. This is mainly because ENFW evaluates features independently and wrongly takes noisy features as relevant features. However, DSFS estimates the outlierness of features based on the intra- and inter-feature couplings embedded within/between features, thus can much better filter out noisy features than ENFW.

The exceptional cases are on *CelebA* (with 39 features) and *Chess* (with 6 features), where DSFS and ENFW perform equally well. This is because both DSFS and ENFW cannot remove a sufficient number of noisy features, and as a result the three detectors not using DSFS and ENFW obtain equally good performance as their counterparts using either DSFS or ENFW. This also shows the significant challenge of identifying intrinsic characteristics and sophisticated interactions between features for outlier detection.

TABLE IV: AUC Performance Comparison of the Three Detectors Using ENFW and DSFS respectively. IMP denotes the improvement of DSFS over ENFW.

	MarP			**COMP**			**FPOF**		
	ENFW	DSFS	IMP	ENFW	DSFS	IMP	ENFW	DSFS	IMP
BM	0.53	0.59	12%	0.56	0.64	15%	0.53	0.59	12%
aPascal	0.46	0.88	91%	0.46	0.88	92%	0.46	0.88	91%
Sylva	0.82	0.96	17%	0.82	0.96	18%	○	○	○
Census	0.43	0.69	61%	0.43	0.71	65%	0.46	0.72	55%
CelebA	0.74	0.74	0%	0.76	0.76	0%	0.75	0.75	0%
CMC	0.50	0.66	31%	0.52	0.66	27%	0.51	0.65	28%
CT	0.51	0.97	90%	0.50	0.97	94%	0.51	0.97	90%
Chess	0.64	0.64	0%	0.63	0.63	0%	0.61	0.61	0%
U2R	0.86	0.92	8%	0.83	0.99	20%	0.86	0.97	13%
SF	0.81	0.85	5%	0.82	0.86	5%	0.83	0.86	4%
DIGIT	0.93	0.95	2%	0.95	0.97	2%	0.93	0.94	1%
MRM	0.89	0.89	1%	0.93	0.94	1%	0.90	0.91	0%
AD	0.56	0.74	33%	0.56	0.75	35%	0.56	0.74	33%
Probe	1.00	1.00	0%	1.00	1.00	0%	1.00	1.00	0%
LINK	0.93	0.98	6%	0.88	0.98	12%	0.93	0.98	5%
Avg			24%			26%			24%

'○' indicates out-of-memory exceptions.

4) Good Scalability: The scalability test results of DSFS against ENFW as a baseline are illustrated in Fig. 2. As expected, DSFS has linear time complexity with respect to data size and is quadratic to the number of features. Although DSFS runs slower than ENFW, it still has quite good scalability with respect to both data size and the number of features, given that DSFS completes its execution within one second for the largest data set with 1,024,000 objects and less than 20 seconds for the high-dimensional data with 1,028 features.

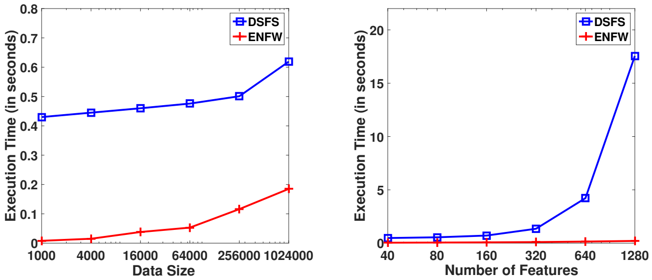

Fig. 2: Scale-up Test Results of DSFS against ENFW w.r.t. Data Size and the Number of Features.

VI. Conclusions

This paper proposes a novel and flexible unsupervised feature selection framework for outlier detection (CUFS). Unlike existing feature selection and unsupervised outlier detection, CUFS effectively captures the low-level hierarchical interactions embedded in relevant features which are mixed with noisy and redundant features. We further introduce a parameter-free instantiation (DSFS) of the CUFS framework. DSFS combines the advantage of CUFS with graph-based strategies. We prove that the feature subset selected by DSFS achieves a 2-approximation to the optimum.

Our extensive evaluation results show that, on average, (i) DSFS obtains 46% feature reduction rate on 15 data sets with different levels of noisy features and redundant features, and (ii) DSFS enables three different types of pattern-based outlier detectors (i.e., MarP, COMP and FPOF) to respectively obtain 7%, 4% and 4% AUC improvements compared to their counterparts not using DSFS.

On data sets with high noise level, in particular, DSFS is able to remove a large proportion of noisy features, resulting in higher AUC improvements of the three detectors (14%, 10% and 9%, respectively). Moreover, by working on data sets with significantly smaller feature subsets, COMP and FPOF, which have at least quadratic time complexity with respect to the number of features, perform orders of magnitude faster than on the original full feature set.

Compared to its feature selection contender ENFW, DSFS performs substantially better in most data sets with noisy features. On average, all three DSFS-based detectors obtain more than 20% AUC improvements compared to ENFW.

As expected, DSFS has linear time complexity to data size and completes the outlier scanning of a data set containing 1,024,000 objects within one second. Although DSFS has quadratic time complexity to the number of features, it completes the data set containing 1,280 features within 20 seconds. This enables DSFS to scale up well with respect to data size and the number of features.

We are working on enhancing CUFS and DSFS by deeply understanding the sophisticated interactions between relevant and noisy/redundant relations to address the challenges in extremely imbalanced data with highly noisy features.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, no. 3, 2009.

[2] C. C. Aggarwal and P. S. Yu, “Outlier detection for high dimensional data,” in *ACM Sigmod Record*, vol. 30, no. 2, 2001, pp. 37–46.

[3] A. Zimek, E. Schubert, and H.-P. Kriegel, “A survey on unsupervised outlier detection in high-dimensional numerical data,” *Statistical Analysis and Data Mining*, vol. 5, no. 5, pp. 363–387, 2012.

[4] L. Akoglu, H. Tong, J. Vreeken, and C. Faloutsos, “Fast and reliable anomaly detection in categorical data,” in *CIKM*, 2012, pp. 415–424.

[5] S. Wu and S. Wang, “Information-theoretic outlier detection for large-scale categorical data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 589–602, 2013.

[6] Z. He, X. Xu, Z. J. Huang, and S. Deng, “FP-outlier: Frequent pattern based outlier detection,” *Computer Science and Information Systems*, vol. 2, no. 1, pp. 103–118, 2005.

[7] K. Smets and J. Vreeken, “The odd one out: Identifying and characterising anomalies,” in *SDM*, 2011, pp. 109–148.

[8] G. Tang, J. Pei, J. Bailey, and G. Dong, “Mining multidimensional contextual outliers from categorical relational data,” *Intelligent Data Analysis*, vol. 19, no. 5, pp. 1171–1192, 2015.

[9] K. Das and J. Schneider, “Detecting anomalous records in categorical datasets,” in *SIGKDD*, 2007, pp. 220–229.

[10] K. Yu, X. Wu, W. Ding, and J. Pei, “Towards scalable and accurate online feature selection for big data,” in *ICDM*, 2014, pp. 660–669.

[11] L. Du and Y.-D. Shen, “Unsupervised feature selection with adaptive structure learning,” in *SIGKDD*, 2015, pp. 209–218.

[12] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, “Feature selection: A data perspective,” *arXiv preprint arXiv:1601.07996*, 2016.

[13] X. Chen and M. Wasikowski, “FAST: A ROC-based feature selection metric for small samples and imbalanced data classification problems,” in *SIGKDD*, 2008, pp. 124–132.

[14] S. Maldonado, R. Weber, and F. Famili, “Feature selection for high-dimensional class-imbalanced data sets using support vector machines,” *Information Sciences*, vol. 286, pp. 228–246, 2014.

[15] F. Azmandian, A. Yilmazer, J. G. Dy, J. Aslam, D. R. Kaeli *et al.*, “GPU-accelerated feature selection for outlier detection using the local kernel density ratio,” in *ICDM*, 2012, pp. 51–60.

[16] C. C. Aggarwal, “Outlier detection in categorical, text and mixed attribute data,” in *Outlier Analysis*, 2013, pp. 199–223.

[17] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[18] J. He and J. Carbonell, “Coselection of features and instances for unsupervised rare category analysis,” *Statistical Analysis and Data Mining*, vol. 3, no. 6, pp. 417–430, 2010.

[19] L. Cao, Y. Ou, and P. S. Yu, “Coupled behavior analysis with applications,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1378–1392, 2012.

[20] C. Wang, X. Dong, F. Zhou, L. Cao, and C.-H. Chi, “Coupled attribute similarity learning on categorical data,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 781–797, 2015.

[21] D. Chakrabarti and C. Faloutsos, “Graph mining: Laws, generators, and algorithms,” *ACM Computing Surveys*, vol. 38, no. 1, p. 2, 2006.

[22] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[23] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *VLDB*, 1994, pp. 487–499.

[24] N. A. Christakis and J. H. Fowler, "The spread of obesity in a large social network over 32 years," *New England Journal of Medicine*, vol. 357, no. 4, pp. 370–379, 2007.

[25] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[26] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Approximation Algorithms for Combinatorial Optimization*, 2000, pp. 84–95.

[27] S. Khuller and B. Saha, "On finding dense subgraphs," in *Automata, Languages and Programming*, 2009, pp. 597–608.

[28] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *SIGKDD*, 2005, pp. 157–166.

[29] A. Zimek, M. Gaudet, R. J. Campello, and J. Sander, "Subsampling for efficient and effective unsupervised outlier detection ensembles," in *SIGKDD*, 2013, pp. 428–436.

[30] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, pp. 1–37, 2016.

[31] I. H. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[32] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Machine Learning*, vol. 45, no. 2, pp. 171–186, 2001.