Isolation-Based Feature Selection for Unsupervised Outlier Detection

Qibo Yang[1], Jaskaran Singh[2], and Jay Lee[3]

[1,2,3] *NSF I/UCRC Center for Intelligent Maintenance Systems (IMS), Cincinnati, OH, 45221, USA*

yangqb@mail.uc.edu
singh2jn@ucmail.uc.edu
lj2@ucmail.uc.edu

ABSTRACT

For high-dimensional datasets, redundant features and complex interactions between features may increase computational costs and make outlier detection algorithms inefficient. Most feature selection methods are designed for supervised classification and regression. However, limited works have been conducted specifically for unsupervised outlier detection. This paper proposes a novel isolation-based feature selection (IBFS) method for unsupervised outlier detection. It is based on the training process of isolation forest (IFOR). The effectiveness of the proposed methodology is demonstrated on a simulation dataset and benchmarked against variance, Laplacian score and kurtosis. The evaluation results confirm that IBFS is immune to the effects of feature scaling. The performance of the proposed methodology is benchmarked using one-class support vector machine (OCSVM), IFOR and local outlier factor (LOF) on several real-world datasets. The evaluation results demonstrate that the proposed method can improve the performance of IFOR. The performance of IBFS is similar to and even better than the well-known outlier indicator: kurtosis, and better than variance and Laplacian score. Additionally, IBFS can produce good performance using a few high-score features, while other feature selection methods need more features.

1. INTRODUCTION

In high-dimensional dataset scenarios, redundant features and complex interactions result in high computational costs and make outlier detection algorithms inefficient. Various feature selection algorithms for supervised learning have been widely researched for many years, and most of them are specifically for classification and regression. Meanwhile, limited studies have been conducted on feature selection algorithms for unsupervised learning, since it's quite challenging due to lack of class labels. Among these limited set of studies, most of them specifically focused on clustering problems; most notable among them being spectral feature selection (SPEC) (Zhao & Liu, 2007).

There are three general classes of feature selection algorithms: filter, wrapper and embedded methods (Li et al., 2017). Filter methods compute a score of each feature for ranking, and then filter out lowly ranked features. Variance and Laplacian score (He et al., 2006) (Ambusaidi et al, 2015) (Zhang et al., 2008) are used for unsupervised outlier detection. Liu, et al. (Liu et al., 2008) (Liu et al., 2012) used Kurtosis value to select feature subspace for isolation forest. The feature selection method (CBRW_FS) (Pang et al., 2016a) based on Coupled Biased Random Walks (CBRW) provides feature weights for categorical data with diversified frequency distributions and many noisy features. Dense Subgraph-based Feature Selection (DSFS) (Pang et al., 2016b) is a parameter-free work designed for feature subset evaluation. Filter methods are independent of learning algorithms and hence their performance on specific learning algorithms may not be optimal.

Wrapper methods repeat searching for a feature subset and evaluation of these features on the performance of a learning algorithm to select desired features. Pang, et al. (Pang et al., 2017) proposed a wrapper-based detection framework (WrapperOD), which can simultaneously optimize its outlier scoring and feature selection. As search space is quite large, wrapper methods are often computationally expensive.

Embedded methods interact feature selection with the training process of a learning algorithm. As it does not evaluate iteratively like wrapper, it's a trade-off solution between filters and wrappers (Li et al., 2017). However, to the best of the authors knowledge, there is no study conducted using embedded method for unsupervised outlier detection.

This paper proposes an isolation-based feature selection (IBFS) algorithm for unsupervised outlier detection. It computes the score of each feature during the training of isolation forest (IFOR). This paper has two main contributions: (1) A novel embedded method, IBFS, is

proposed based on IFOR to calculate feature scores, which is immune to the effects of feature scaling. (2) This paper compares several feature selection methods on several real-world datasets, and analyzes their characteristics, which are seldom discussed in previous works.

This paper is organized as follows: Section 2 describes the concept of isolation forest, and then proposes IBFS. Section 3 benchmarks IBFS with variance, Laplacian score and kurtosis on a simulated dataset and several real-world datasets. In Section 4, the work and discuss future research are summarized to conclude this paper.

2. METHODOLOGY

The background of IFOR is introduced at first, and then the IBFS algorithm is elaborated.

2.1. Background of Isolation Forest

IFOR (Liu et al., 2008) (Liu et al., 2012) is a tree-based ensemble method for outlier detection. Compared with outlier detection methods which rely on measure of distance, IFOR uses a different mechanism: it isolates data into subspace, and outliers need fewer splitting to be isolated than normal points, which makes IFOR capable to handle large data size and high dimension data with many irrelevant features, so IFOR is considered that it has the potential to identify irrelevant features and to reduce dimensionality.

The training stage of IFOR is simply described in Section 2.2. The purpose of training is to construct IFOR $iForest$, which is composed of isolation trees. The training of an isolation tree $iTree$ is the process to generate its nodes until it reaches the height limit or training data are used up. The way to generate a node is to randomly select a feature and a value of this feature, and split data into left and right nodes.

2.2. Feature Selection Algorithm

The procedure to calculate imbalance score using entropy is explained below.

Information entropy is first introduced by Claude Shannon (Shannon, 1948) and then used in Iterative Dichotomiser 3 (ID3) algorithm to measure the impurity of a feature for decision tree (Quinlan, 1986). Equation (1) is the entropy function in the case of two classes.

$$E = -p_1 \, log_2 \, p_1 - p_2 \, log_2 \, p_2 \qquad (1)$$

where p_1 is the proportion of samples from class 1, and p_2 is the proportion of samples from class 2 where $p_2 = 1 - p_1$.

Based on entropy, an imbalance score 's' defined in Equation (2) is proposed to measure the imbalance of the data distribution split by the randomly selected value from a randomly selected feature at a node.

$$s = 1 + p_l \, log_2 \, p_l + (1 - p_l) \, log_2 (1 - p_l) \qquad (2)$$

where p_l is the proportion of samples split into the left node.

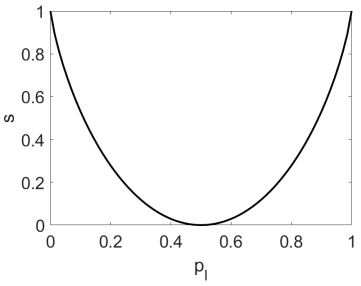

Figure. 1. The imbalance score s for the data distribution while splitting the node.

Figure 1 depicts the variation of score 's' when p_l changes from 0 to 1. It can be observed that the score is large when the data distribution is more imbalanced. Equation (3) defines the penalty of imbalance score θ using entropy. p_0 (Equation (4)) is the proportion of a split point p between the maximum p_{max} and minimum p_{min} of this feature. The penalized score s_θ is defined in Equation (5). It means that if a split point is located at the edge in the range of a feature, a high penalty is given which is a low value θ, so that the penalized score s_θ becomes smaller.

$$\theta = -p_0 \, log_2 \, p_0 - (1 - p_0) \, log_2 (1 - p_0) \qquad (3)$$

$$p_0 = (p - p_{min})/(p_{max} - p_{min}) \qquad (4)$$

$$s_\theta = \theta \times s \qquad (5)$$

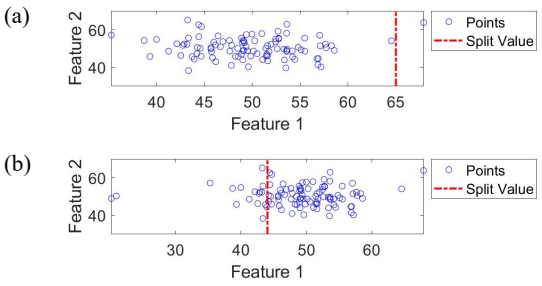

Figure. 2. The illustration of the penalty of imbalance score.

The importance of penalty value is illustrated with the help of a simulation depicted in Figure 2. The feature 1 in Figure 2(b) should be given a higher value θ (lower penalty) than the feature 1 in Figure 2(a). Even though in Figure 2(a) and Figure 2(b), the data split by the feature 1 is imbalanced, giving the similar scores of the feature 1 is improper. It is because in Figure 2(a), the split point of the feature 1 is at the edge in the range of feature 1 and it's a matter of course to be imbalanced so that a high penalty of the imbalanced score should be given, and the penalized score is small. However, in Figure 2(b), the split point of the feature 1 is at the center in the range of the feature 1. If the feature 1 in Figure 2(b) splits imbalanced data, it should be given a low penalty, which produces a rather larger penalized score.

Algorithm 1: S_{forest} from $iForest(X, t, \psi)$

Inputs: X, t, ψ
Output: a set of t $iTrees$ and S_{forest}
Initialize $Forest$
set $l = ceiling(\log_2 \psi)$
for $i = 1$ to t **do**
  $X' \leftarrow sample(X, \psi)$
  $Forest \leftarrow Forest \cup iTree(X', 0, l)$
  $S_{trees} \leftarrow S_{trees} \cup S_{tree_i}$
end for
for $j = 1$ to m **do**
  $s_{forest_j} = \sum_{i=1}^{t} s_{tree_{ij}}/t$
  $S_{forest} \leftarrow S_{forest} \cup s_{forest_j}$
end for
return $Forest$, S_{forest}

Figure. 3. Algorithm 1 of feature selection.

Based on the training process of IFOR, two pseudo codes (Algorithm 1 & Algorithm 2 in Figure 3 & 4 respectively) are provided below to calculate the feature scores. Algorithm 1 uses training data X with m features to compute a set of feature scores S_{forest} by building an isolation forest model $iForest$ including t isolation trees ($iTree$) while ψ is a hyperparameter to define the training data size of each tree. Algorithm 2 is the procedure to grow an $iTree$ for Algorithm 1 and compute a set of feature scores S_{tree} for a tree using a subset of training data X' including ψ selected samples from X, noting that e is the current tree height of the tree, and l is the height limit of a tree.

In Algorithm 2, during the split of each node of each tree, a penalized score s_θ is calculated for each node defined by Equation (2), (3), (4) and (5).

According to Liu et al. (Liu et al., 2008), the number of trees 't' is recommended to be 100, and before the algorithm reaches this number, the forest often converges well. Meanwhile, during the training process of isolation forest, as the data are randomly split by different features, the feature scores computed by the IFOR with small number of trees may change sharply. Therefore, sufficient number of trees is needed to get a stationary distribution of scores. Liu et al. (Liu et al., 2008) empirically found that subsampling size ψ = 256 is enough to perform anomaly detection, so ψ = 256 is used as default value.

3. RESULTS AND DISCUSSION

3.1. Evaluation on A Simulated Dataset

In this section, the performance of variance, Laplacian score, kurtosis and IBFS is compared on a simulated dataset. Variance is the simplest unsupervised feature evaluation. Larger variance often implies stronger representative power. Laplacian Score (He et al., 2006) is another unsupervised feature selection method. It not only considers large variances, but also takes locality preserving ability into consideration. Kurtosis measures the "tailedness" of a data distribution and large kurtosis tells the propensity to produce outliers (Westfall, 2014). Variance and kurtosis can be easily calculated in any computing environments. Laplacian score is computed by using the Feature Selection Library (Roffo, 2018).

Algorithm 2: S_{tree} from $iTree(X, e, l)$

Inputs: X, e, l
Output: an $iTree$ and S_{tree}
if $e \geq l$ or $|X| \leq 1$ **then**
  for $j = 1$ to m **do**
    search s_{θ_j} in S_θ and its n_{n_j} in N_n
    calculate $s_{\text{tree}_j} = \sum_{k=1}^{\sigma} s_{\theta_{kj}} \times n_{n_{kj}}/\psi$
  end for
  return $exNode\{Size \leftarrow |X|\}$
else
  let Q be a list of features in X, n_n is number of samples of X
  randomly select a feature $q \in Q$
  randomly select a split point p between p_{max} and p_{min} of feature q in X
  $X_l \leftarrow filter(X, q < p)$
  $X_r \leftarrow filter(X, q \geq p)$
  calculate s_θ
  return $inNode\{Left \leftarrow iTree(X_l, e + 1, l)$,
    $Right \leftarrow iTree(X_r, e + 1, l)$,
    $SplitFeature \leftarrow q$,
    $SplitValue \leftarrow p$,
    $S_\theta \leftarrow s_\theta$
    $N_n \leftarrow n_n\}$
end if

Figure. 4. Algorithm 2 of feature selection.

The simulated data contain three features and 100 samples. Feature 1 is simulated based on a gaussian distribution with 100 normal data points with mean = 20. Feature 2 is simulated based on a gaussian distribution with 96 data points with mean = 20 and 4 outlier data points with mean = 70. Feature 3 is simulated based on a gaussian distribution with 92 data points with mean = 20, 4 outliers data points with mean = 70and 4 data points with mean = -30. Figure 5 depicts the histograms of the 3 features. Since outliers can be seen from feature 2 and 3; therefore, it should be reasonable to give a higher score to feature 2 and feature 3.

This comparison also considers the influence of feature scaling on feature selection methods. Min-max scaling has been widely used in outlier detection (Cousineau & Chartier, 2010), and it normalizes each feature into the range of [0, 1].

The evaluation results are displayed in Table 1. In Table 1, the value corresponding to "F1" and "Variance" represents the score for the feature 1 given by variance. A higher score means this feature is better for outlier detection. With no

feature scaling, all methods correctly regard the feature 1 as the weakest. But the methods provide different assessments on feature 2 and feature 3. With min-max scaling, the rankings of variance and Laplacian score are disordered. Meanwhile, the values of kurtosis and IBFS do not change with feature scaling.

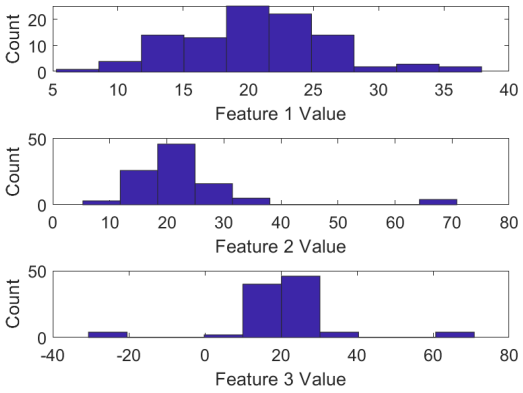

Figure. 5. Simulated features.

Table 1. The comparison of feature scaling.

Method	Feature	IBFS	Kurtosis	Variance	Laplacian score
No scaling	F1	0.210	3.330	33.441	1.491
	F2	0.264	13.104	123.588	10.763
	F3	0.291	9.542	228.238	10.465
Min-max scaling	F1	0.210	3.330	0.031	0.002
	F2	0.264	13.104	0.029	0.003
	F3	0.291	9.542	0.022	0.001

* F1: Feature 1, F2: Feature 2, F3: Feature 3.

Figure 6 depicts how the scores of the three features change while the number of trees in the IFOR model increases. It can be observed that a forest built with about 50 trees or more provides stable scores.

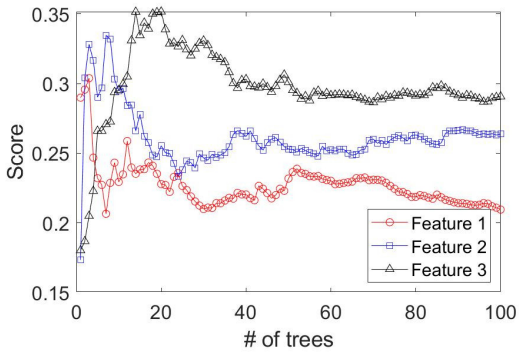

Figure. 6. Feature scores with increasing number of trees

3.2. Evaluation on Real-World Datasets

In this section, the feature selection methods are evaluated on three real-world datasets from ODDS (Outlier Detection DataSets) Library (Rayana, 2018). These datasets are usually used to evaluate outlier detection methods. Table 2 provides the descriptions of datasets, including number of samples and number of features.

Table 2. Dataset descriptions.

ID	Dataset	# of samples	# of features
1	Annthyroid	7200	6
2	Arrhythmia	452	274
3	Breastw	683	9
4	Cardio	1831	21
5	Glass	214	9
6	Ionosphere	351	33
7	Letter	1600	32
8	Lympho	148	18
9	Mammography	11183	6
10	Mnist	7603	100
11	Musk	3062	166
12	Thyroid	3772	6
13	Vertebral	240	6
14	Vowels	1456	12

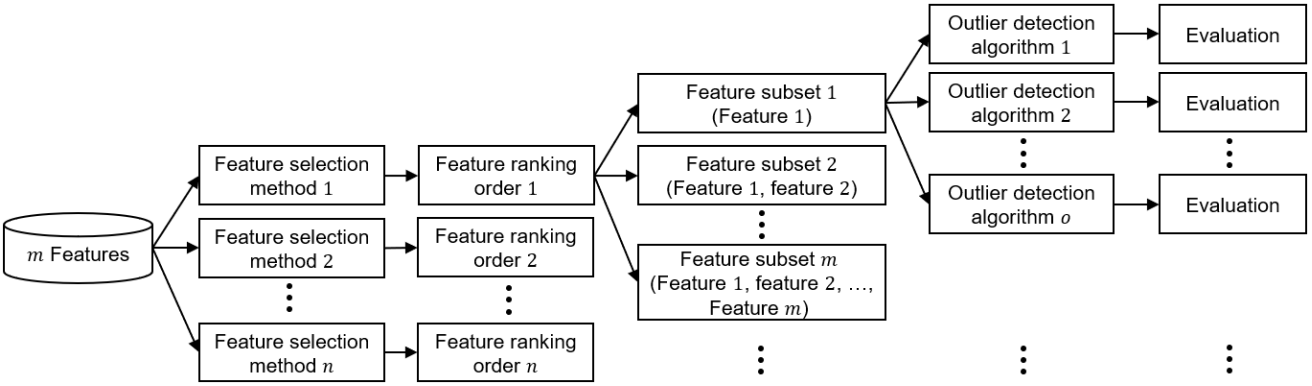

Figure 7. The working flowchart for the evaluation of feature selection methods.

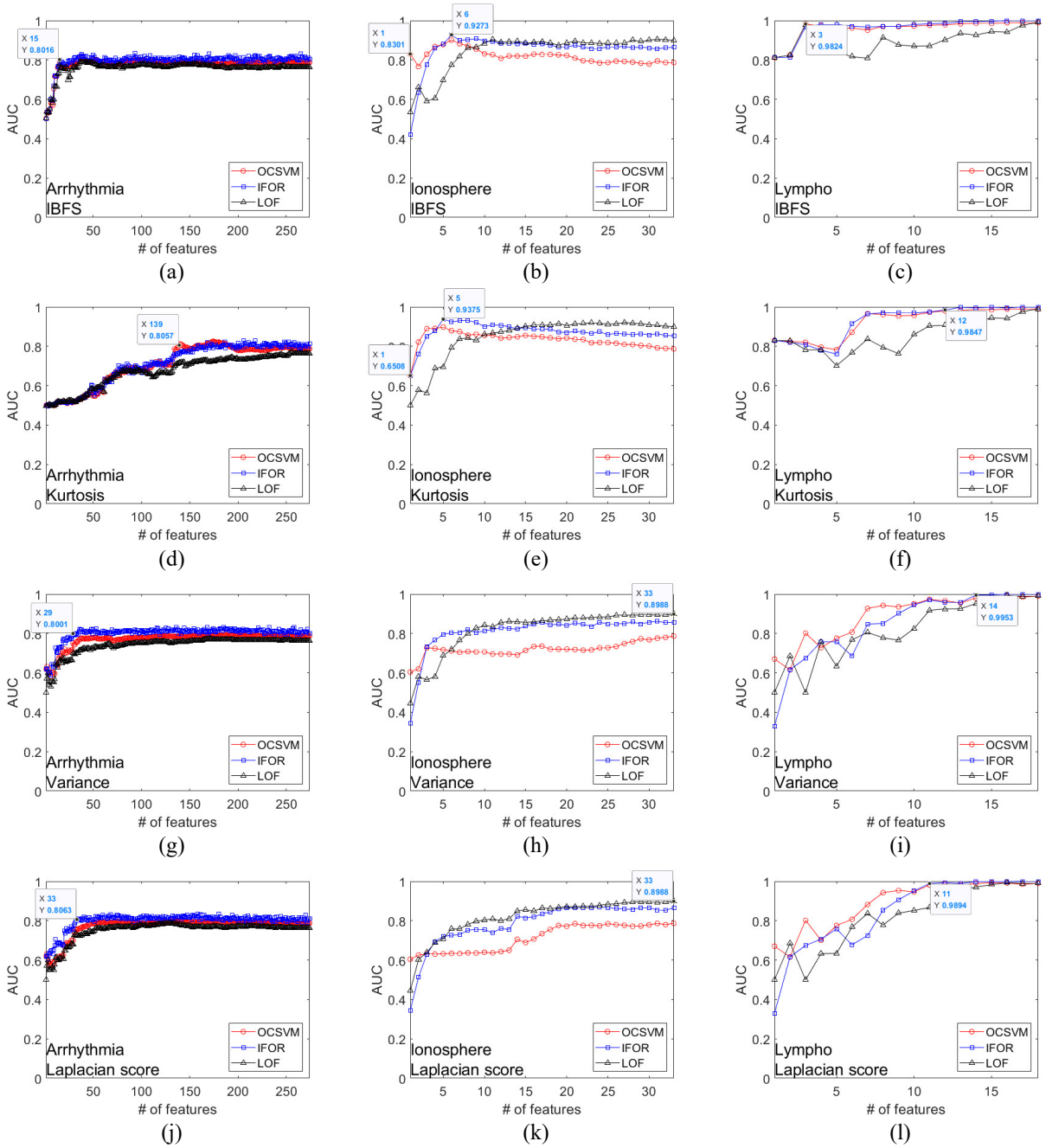

Figure 8. The performance evaluation using AUC curves generated by adding features iteratively.

The feature selection methods are evaluated by the working flowchart shown in Figure 7 with following steps: 1) Data are normalized by min-max scaling. 2) Feature scores are given by n different feature selection method. (Here is variance, Laplacian score, kurtosis and IBFS). 3) Features are ranked in a descending order according to their feature scores for each feature selection method. 4) For each feature ranking order, m feature subsets are generated. For ith feature subset, features with i top feature rankings are selected. 5) For each feature subset, their performance using l outlier detection algorithms are evaluated (Here is one-class support vector machine (OCSVM), IFOR and local outlier factor (LOF). They are implemented by scikit-learn (INRIA, 2018) in Python 3.6). AUC (Area under the Receiver Operating Characteristic Curve) is the metric to evaluate the performance of the algorithms. Higher AUC represents better performance.

The evaluation results of the best performing algorithm for each feature selection method and each outlier detection

Table 3. The performance evaluation using maximum AUCs.

ID	IBFS			Kurtosis			Variance			Laplacian score		
	OCSVM	IFOR	LOF	OCSVM	IFOR	LOF	OCSVM	IFOR	LOF	OCSVM	IFOR	LOF
1	0.7826	**0.9885**	0.8321	0.8142	**0.9885**	0.8178	0.5842	0.8426	0.7692	0.5842	0.8426	0.7123
2	0.8117	**0.8363**	0.7918	0.8253	0.8287	0.7719	0.7938	0.8329	0.7721	0.7994	0.8353	0.7864
3	0.8052	**0.9880**	0.8174	0.8052	0.9874	0.7570	0.8143	0.9857	0.6329	0.8143	0.9860	0.6818
4	0.9430	0.9371	0.6582	0.9465	0.9328	0.6582	0.9286	0.9508	0.6902	0.9286	**0.9513**	0.6582
5	**0.9878**	0.7488	0.8770	**0.9878**	0.7539	0.8770	0.8783	0.8827	0.7837	0.8873	0.8827	0.7837
6	0.9011	0.9273	0.9043	0.8980	**0.9375**	0.9206	0.7867	0.8613	0.8988	0.7867	0.8693	0.8988
7	0.6070	0.7167	0.9013	0.5824	0.7106	0.9013	0.5065	0.7269	**0.9044**	0.5132	0.7559	0.9027
8	0.9906	**1.0000**	0.9906	0.9906	**1.0000**	0.9894	0.9918	**1.0000**	0.9930	0.9930	**1.0000**	0.9930
9	0.8412	0.8629	0.7683	0.8412	0.8635	0.7683	0.8412	0.8700	0.7627	0.8434	**0.8807**	0.7662
10	0.8376	0.8343	0.7704	0.8382	0.8766	0.7656	**0.9032**	0.8377	0.6684	0.8544	0.8384	0.7031
11	**1.0000**	**1.0000**	0.7190	**1.0000**	**1.0000**	0.8737	0.9997	**1.0000**	0.7989	**1.0000**	**1.0000**	0.6105
12	0.9895	**0.9924**	0.8554	0.9895	**0.9924**	0.8554	0.8437	0.9823	0.9363	0.8437	0.9823	0.9363
13	**0.8489**	0.4141	0.5954	**0.8489**	0.4743	0.5954	0.7240	0.4444	0.5301	0.7173	0.4858	0.5263
14	0.7290	0.8165	**0.9622**	0.7681	0.8350	**0.9622**	0.5507	0.8135	0.9485	0.5507	0.8173	0.9485

algorithm are depicted in Table 3. In Table 3, the highest AUC for each dataset are highlighted in 'bold'. The scores of OCSVM come from the highest AUC of the following models: OCSVM with linear kernel, OCSVM with polynomial kernel with three degrees, and OCSVM with Gaussian kernel. The scores of IFOR stem from the better outputs from IFOR with the subsampling size equaling to 256, and IFOR with the subsampling size equaling to the number of samples. The scores of LOF arise from the best performance of LOFs with 5, 15 and 25 number of neighbors.

From Table 3, the merits and potential of IBFS can be easily understood. (1) IFOR can often be improved by IBFS: among the 9 datasets that IFOR perform best, IBFS can perform best for 6 datasets. (2) IFOR can also improve other outlier detection methods like OCSVM and LOF, like dataset 5, 11, 13 and 14. (3) IBFS can get similar performance with kurtosis for dataset 1, 8, 11, 12, 13, 14, and as kurtosis is a well-known statistic to show outliers, the capability of IBFS is also valid.

Figure 8 shows another way to evaluate the results: assessing the change of AUC with the increasing number of features. Due to page limitations, only three datasets are displayed. Figure 8(a), (d), (g) and (j) are the results for "Arrhythmia" dataset. Overall, there are many redundant features, so that with a few features, the algorithms can achieve good performance. Specifically, IBFS can exceed AUC 0.8 with 15 high-score features, but Laplacian score needs 33 high-score features, variance needs 29 features and kurtosis needs 139 features to get the same performance.

Figure 8(b), (e), (h), (k) illustrate the results of "Ionosphere" dataset. IBFS and kurtosis are over AUC 0.9 with 6 and 5 features respectively, but the performance of variance and Laplacian increase slowly, and does not go beyond AUC 0.9. This demonstrates that bad features hamper the performance of outlier detections and IBFS and kurtosis can reduce their negative effects. Compared with kurtosis, the feature with highest score can achieve AUC 0.8301 while the feature with highest score of kurtosis can only achieve AUO 0.6508.

Figure 8(b), (e), (h), (k) presents the results of "Lympho" dataset. On the whole, AUC 1 could be achieved by all four criteria's and only a few redundant features influence the efficiency. With 3 high-score features scoring by IBFS, AUC over 0.98 can be achieved. Correspondingly, kurtosis needs 12 features, variances needs 14 features and Laplacian needs 11 features to go beyond AUC 0.98.

Based on the discussion of Figure 8, it can be observed that IBFS can perform well with a fewer high-score features, while kurtosis, Laplacian score and variance need more features.

4. CONCLUSIONS

This paper proposes a novel isolation-based feature selection method (IBFS) where an embedded feature selection method is developed specifically for unsupervised outlier detection. The proposed methodology is validated on real-world datasets and benchmarked against variance, Laplacian score and kurtosis criteria. The evaluation results confirm that the proposed method and kurtosis are immune to the effects of feature scaling in comparison to variance and Laplacian score based criteria's. However, IBFS can achieve good results using a fewer high-score features, while kurtosis, Laplacian score and variance need more features. The evaluation results further confirm that IBFS can often improve the performance of isolation forest (IFOR), and its results are similar to and even better than the well-known outlier indicator: kurtosis.

However, the scores from IBFS were observed to be less stable with a small number of trees. Making the proposed methodology more stable and measuring the convergence of the scores with the increasing number of trees is part of the author's future scope of research.

NOMENCLATURE

E	Entropy
e	current tree height
$Forest$	$iForest(X, t, \psi)$
$iForest$	an isolation forest
$iTree$	an isolation tree
l	height limit of a tree
m	number of features
N_n	a set of number of samples for m features before each splitting, $N_n = \{n_{n_1}, n_{n_2}, \dots, n_{n_m}\}$
n	number of feature selection methods
n_n	number of samples of a node before splitting
n_{n_j}	number of samples for jth feature before each splitting
$n_{n_{kj}}$	number of samples of kth node using jth feature before splitting
n_l	number of samples split into the left node
o	number of outlier detection algorithms
p	a split point
p_0	a proportion of a split point p
p_1	a proportion of samples from class 1
p_2	a proportion of samples from class 2
p_l	a proportion of samples split into the left node, $p_l = n_l/n_n$
p_{max}	a maximum value of a feature
p_{min}	a minimum value of a feature
Q	a list of features in X
q	a feature randomly selected for splitting
S_{forest}	a set of feature scores computed from $iForest$ for m features, $S_{forest} = \{s_{forest_1}, s_{forest_2}, \dots, s_{forest_m}\}$
s_{forest_j}	a score of jth feature from $iForest$
S_{tree}	a set of feature scores computed from an $iTree$ for m features, $S_{tree} = \{s_{tree_1}, s_{tree_2}, \dots, s_{tree_m}\}$
S_{trees}	a set of feature scores computed from t $iTrees$, $S_{trees} = \{S_{tree_1}, S_{tree_2}, \dots, S_{tree_t}\}$
S_{tree_i}	a set of feature scores computed from ith $iTree$ for m features, $S_{tree_i} = \{s_{tree_{i1}}, s_{tree_{i2}}, \dots, s_{tree_{im}}\}$
s_{tree_j}	a score of jth feature from a tree
$s_{tree_{ij}}$	a score of jth feature from ith $iTree$
S_θ	a set of penalized scores of a node for m features, $S_\theta = \{s_{\theta_1}, s_{\theta_2}, \dots, s_{\theta_m}\}$
s_{θ_j}	a penalized score of jth feature
s	an imbalanced score
s_θ	a penalized score
$s_{\theta_{kj}}$	a penalized score of kth node using jth feature for splitting
t	number of trees
X	input training data
X'	ψ selected samples from X
X_l	samples split into left node
X_r	samples split into right node
θ	penalty
σ	number of nodes using jth feature for splitting
ψ	sub-sampling size that determines the training data size of each tree

REFERENCES

Ambusaidi, M. A., He, X., & Nanda, P. (2015, August). Unsupervised feature selection method for intrusion detection system. In *2015 IEEE Trustcom/BigDataSE/ISPA* (Vol. 1, pp. 295-301). IEEE.

Cousineau, D., & Chartier, S. (2010). Outliers detection and treatment: a review. *International Journal of Psychological Research*, 3(1), 58-67.

He, X., Cai, D., & Niyogi, P. (2006). Laplacian score for feature selection. In *Advances in neural information processing systems* (pp. 507-514).

INRIA. *Scikit-learn v0.20.0.* http://scikit-learn.org/stable.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2018). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6), 94.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413-422). IEEE.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1), 3.

Pang, G., Cao, L., & Chen, L. (2016). Outlier Detection in Complex Categorical Data by Modeling the Feature Value Couplings. In *IJCAI*, pp. 1902-1908.

Pang, G., Cao, L., Chen, L., & Liu, H. (2016, December). Unsupervised feature selection for outlier detection by modelling hierarchical value-feature couplings. In *2016 IEEE 16th International Conference on Data Mining (ICDM)* (pp. 410-419). IEEE.

Pang, G., Cao, L., Chen, L., & Liu, H. (2017, January). Learning Homophily Couplings from Non-IID Data for Joint Feature Selection and Noise-Resilient Outlier Detection. In *IJCAI* (pp. 2585-2591).

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1), 81-106.

Rayana, S. *ODDS Library*. Stony Brook, NY: Stony Brook University, Department of Computer Science. http://odds.cs.stonybrook.edu.

Roffo, G. *Feature Selection Library (MATLAB Toolbox)*. https://it.mathworks.com/matlabcentral/fileexchange/56937-feature-selection-library.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3), 379-423.

Sphinx-Gallery. *Scikit-learn: Outlier detection with several methods.*

https://sklearn.org/auto_examples/covariance/plot_outlier_detection.html.

Westfall, P. H. (2014). Kurtosis as peakedness, 1905–2014. RIP. *The American Statistician*, 68(3), 191-195.

Zhang, D., Chen, S., & Zhou, Z. H. (2008). Constraint Score: A new filter method for feature selection with pairwise constraints. *Pattern Recognition*, 41(5), 1440-1451.

Zhao, Z., & Liu, H. (2007, June). Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning* (pp. 1151-1157). ACM.

BIOGRAPHIES

Qibo Yang received the B.S. degree in Quality and Reliability Engineering from Beihang University, Beijing, China, in 2013, and the M.E. degree in Control Science and Engineering from Beihang University, Beijing, China, in 2016. He is currently a Ph.D. Candidate in Mechanical Engineering at the Center for Intelligent Maintenance Systems, University of Cincinnati, Cincinnati, OH, USA. His research interests include prognostics and health management, applied statistics and machine learning.

Dr. Jaskaran Singh is currently a post-doctoral fellow at the NSF I/UCRC for Intelligent Maintenance Systems (IMS) at the Dept. of Mechanical and Materials Engineering at the University of Cincinnati. His current role in the IMS center includes conducting research in the field of Prognostics and Health Management for rotating machinery, condition monitoring of industrial equipment, mentoring graduate students and formulating projects with IMS industry members. During his PhD studies, Dr. Singh worked at the Vibration Research Lab (IIT Delhi, India) as a Research Scholar with a research focus on the design of some novel approaches emphasizing on fault diagnosis, degradation assessment and life prediction of rolling element bearings.

Dr. Jay Lee is an Ohio Eminent Scholar, L.W. Scott Alter Chair Professor, and Univ. Distinguished Professor at the Univ. of Cincinnati and is founding director of National Science Foundation (NSF) Industry/University Cooperative Research Center (I/UCRC) on Intelligent Maintenance Systems (www.imscenter.net) which consists of the Univ. of Cincinnati (lead institution), the Univ. of Michigan, Missouri Univ. of S&T, and the Univ. of Texas-Austin. Since its inception in 2001, the Center has been supported by over 100 global companies. IMS was selected as the most economically impactful I/UCRC in the NSF Economic Impact Study Report in 2012 which reported that the Center has delivered to its members a combined benefit of $847.6 million in cost savings, and that the Center returned $238.30 of benefits for every $1 invested by the National Science Foundation. Currently, he is leading a new Industrial AI Center. His current research focuses on predictive big data analytics and cyber physical systems for intelligent maintenance, prognostics and health management (PHM), and Industry 4.0 systems. He was selected to be one of the 30 Visionaries in Smart Manufacturing in U.S. by SME in Jan. 2016. In addition, he is coFounder of Predictronics--a start-up company from NSF IMS Center of the Univ. of Cincinnati through NSF ICorp award in 2012.