International Journal of Information System Modeling and Design
Volume 13 • Issue 1

Simulated Workbench Design for Characterisation and Selection of Appropriate Outlier Ensemble Algorithm

Divya D., School of Engineering, Cochin University of Science and Technology, India
M. Bhasi, School of Management Studies, Cochin University of Science and Technology, India
Santosh Kumar, School of Engineering, Cochin University of Science and Technology, India

ABSTRACT

Outlier ensembles can enhance the performance of outlier detection algorithms. However, identification of an appropriate outlier ensemble algorithm for the given application is a difficult task. In order to achieve this function, the performance features of outlier ensembles with gathered data types containing a certain set of outliers have been characterised and their ensemble performance analysed in this paper. The proposed methodology suggests comparing data and anomalous features of a given real time application to the data set available in the proposed workbench. Users can use the results of experiments conducted using the workbench to identify highly efficient algorithms for their application without undertaking further analysis once matching data is located in the workbench. Therefore, this research paper act as a recommendation system that enables data scientists from various fields to pick an effective outlier ensemble for their application.

KEYWORDS

benchmark, brute force approach, case study, evaluation metrics, false alarm rate, outlier detection, outlier detection rate, sequential outlier ensemble, workbench

INTRODUCTION

Outlier detection is a field of data mining with a wide range of applications in several domains varying from intrusion detection (Angelo et al., 2018), process monitoring (Wang and Mao, 2019), detection of fake iris images (2018), security threats (Sharma and Gupta, 2021) and identification of credit card frauds (Carcillo et al, 2019). Outlier points might be of several types that may occur in different magnitude, intensity and frequency. Here, the magnitude refers to the extent of variation from the normal behavior and frequency of anomalies can be measured by the percentage of outliers present in the data. Identifying these outlier points is a tough task, data scientists utilise both supervised and unsupervised machine learning models for this purpose. Supervised learning algorithms work based

DOI: 10.4018/IJISMD.315024

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

on the assumption that the test data has similar characteristics as the training data. However, their performance is poor if they encounter a set of data points which is outside the range of training dataset. In unsupervised algorithms, identification of outliers is based on the characteristic difference of the abnormal data points from the rest of the data. Yet, unavailability of ground truth makes it difficult to evaluate these unsupervised algorithms (Dong et al, 2020; Omer and Lior, 2018). Thus, choosing an algorithm that accurately identifies the outlier objects is an essential need for all kinds of applications. Appropriate selection of the evaluation parameters that pick out a good outlier detection algorithm considering all the features of anomalies present in the data is another mandatory requirement.

A number of evaluation parameters are available for analysing the accuracy of outlier detection algorithms, varying from precision and recall (Domingues et al., 2018), Receiver Operator Characteristic (ROC) curve (Goldstein and Uchida, 2016), outlier detection rate and false alarm rate (Shahid et al., 2013). Even so, most of the algorithms that exhibit high detection rate usually express high false alarm rate also. Nevertheless, good outlier detection algorithms ought to have high detection rate as well as low false alarm rate. Hence, the greatest challenge is to design an outlier detection system with high detection rate and low false alarm rate. Several single stage outlier detection algorithms are available in the literature (Domingues et al., 2018; Shahid et al., 2013). These algorithms show lower average performance in the presence of different types of outliers. This leads to the finding of outlier ensembles that exhibit higher performance than individual detection methods. Other than this, deep learning techniques (Vinayakumar et al, 2017) and reinforcement learning (Ghosh et al., 2017) can also be used to improve the detection rate.

Aggarwal and Sathe (2015) put forward the concept of outlier ensemble with suggestions to design an ensemble-based system for outlier detection. Designing an ensemble-based outlier detection system with high detection rate and low false alarm rate is difficult (Aggarwal and Sathe, 2015), because an ensemble design demand understanding the structure of the data, form and frequency of outliers present in the dataset. The accurate evaluation of algorithms in the absence of labelled datasets is another challenge in such algorithm design.The taxonomy of outlier detection is given in Figure 1.

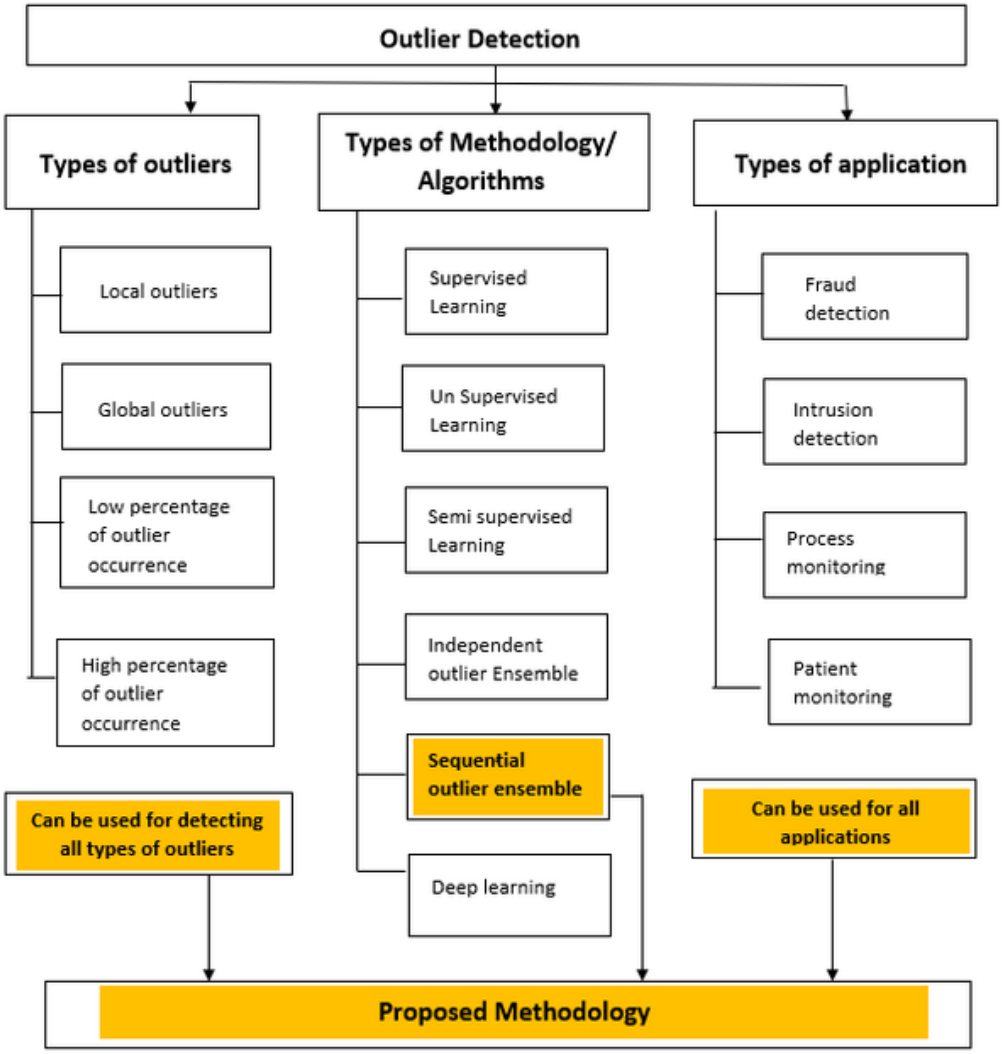
This research proposes a novel sequential outlier ensemble algorithm design procedure (workbench as we call it) that combines supervised and unsupervised algorithms in order to achieve a high detection rate and low false alarm rate. The proposed workbench is capable of testing all the supervised and unsupervised algorithm combinations. Here, supervised algorithms such as K nearest neighbour algorithms (Knn), Random Forest (RF), Adaptive Boosting (AdaBoost), eXtreme Gradient Boosting (XGBoost), and unsupervised algorithms such as Minimum Covariance Determinant Estimator (MCD), Local Outlier Factor (LOF), Isolation Forest (IF), and One Class SVM (OCSVM) are used to create ensembles pairs. For testing the ensemble performance, the proposed workbench's regular dataset is limited to Gaussian and random walk distributions. Analysing the effect of local and global anomalies at various outlier percentages is also included in this study. The workbench design recommended in this study has been successfully demonstrated for choosing an appropriate ensemble-based outlier detection algorithm using Yahoo anomaly benchmark dataset.

Rest of the paper is organised as follows, Section 2 details the related work in this area, Section 3 describes the workbench design used in this research, Section 4 presents the case study and Section 5 is discussion and Section 6 is devoted to conclusion.

RELATED WORK

The challenging problem in outlier detection is to find the decision boundaries between normal data objects and outlier points and giving a proper definition for outliers across various application domains (Kiani et al., 2020). Dong et al., (2020) categorised outlier points into rare events, anomalies or novelty and reviewed different supervised outlier detection techniques in this area. Zhang et al (2020) proposed an outlier adapted training dataset, which is a new technique to enhance the system's ability to model the outlier embedded in the dataset while avoiding possibility of over fitting. Due to scarcity

Figure 1. Taxonomy of outlier



of anomalous instances in the data an anomaly detection problem can be viewed as an unbalanced case of classification (Emmott et al., 2013). If you consider anomaly detection as an unsupervised learning scenario, then difficulty arises in the evaluation of algorithms due to the unavailability of ground truth. Hence, scientists used synthetic dataset for evaluation of different unsupervised algorithms (Steinbuss and Bohm, 2021). Researchers proposed a set of rules for data generation. These rules ensure that the generated dataset follows the characteristics of a real time dataset. Emmott et al. (2013) generated and used multivariate Gaussian distribution for regular instances. Li et al. (2016) created normal data objects using Gaussian distribution. All these researchers have utilised a separate uniform distribution for generation of anomalous points (Emmott et al., 2013; Steinbuss and Bohm, 2021; Li et al. 2016).

After setting up the evaluation data sets, next challenge is identifying the right algorithm with a high detection rate for the application. Since independent anomaly detection techniques exhibits

lower detection rate, Aggarwal and Sathe (2015) put forward the concept of outlier ensembles, which brings the idea of ensemble-based outlier detection that combines multiple outlier detectors to increase the outlier detection rate and decrease false alarm rate of the overall system. Main challenge in the creation of an ensemble is the selection of different algorithms as ensemble members and the method used to combine various detector outputs (Campos et al, 2018). As outlier ensembles are expected to outperform the independent counter parts, we must take care to ensure that combining one algorithm with another will not bring down the overall detection rate (Rayana and Akoglu, 2016). Therefore, care must be taken during component selection and aggregation of outlier ensembles.

In an independent ensemble, various executions of the same algorithm with different parameters or different algorithms are executed on the whole dataset or portions of the data (Aggarwal and Sathe, 2017). One of the earliest works in an independent outlier ensemble is a feature bagging, where every independent algorithm selects a random set of features at each execution (Lazarevic and Kumar, 2005). Other methods under this category combines multiple algorithms with random sub samples of the data and random subset of features (Pasillas-Díaz and Sylvie, 2016; Zimek et al., 2013). Setting up an appropriate threshold for distinguishing outliers can also be handled by independent ensembles, where the first stage of the algorithm determines the threshold for outlier detection and second stage deals only with evaluation procedure (Smolyakov et al., 2019).

In sequential ensembles, algorithms are applied sequentially, and output of the previous algorithms impact the output of the future algorithms, since second stage algorithms are applied on a refined dataset (Aggarwal and Sathe, 2017). The result will be obtained from the output of final algorithms. Liu et al. (2014) designed a sequential outlier detection algorithm in which local outliers are identified in the first stage and global outliers are detected in the second stage. In these methods separate identification of local and global outliers can enhance the overall outlier detection rate of the algorithm. Zhao et al. (2015) proposed a sequential ensemble algorithm in which the initial algorithm is applied on the whole dataset to get outlier ranks of all the data points and the subsequent stage select a data set that has highest anomaly scores in the first stage. Similarly, this same definition is used by Nguyen et al. (2017) with a difference in the selection of second stage input, where the most likelihood normal samples are given as the second stage input. Some other researchers combine supervised and unsupervised methods to enhance the overall detection rate of the algorithm (Carcillo et al., 2019; Veeramachaneni, 2016). Thus, using sequential outlier detection, several researchers used intermediate results and increased the overall algorithm detection rate (Zhang et al., 2019).

Both independent ensembles and sequential ensembles require a combination function, which combines outlier scores from different independent detectors (Aggarwal and Sathe, 2017; Gao and Tan, 2006). Important factor in this case is the selection of method for combining the outlier scores. Techniques such as maximum function (Lazarevic and Kumar, 2005), averaging (Muller et al, 2011), damped averaging, pruned averaging (Aggarwal and Sathe, 2017) are commonly used for combining outlier scores. Some researchers accept the final output of the last phase as the result of the combination function (Ryana et al., 2016). Normalisation of different forms of output is also a major concern in outlier ensembles since the data analyst needs a unified outlier score to combine the multi-stage outputs (Aggarwal and Sathe, 2017; Gao and Tan, 2006).

Another important factor in the creation of outlier ensembles is evaluating the suitability of the algorithm for the given dataset. Methods like Dynamic Combination of Detector Scores for Outlier Ensembles (DSCO) (Zhao and Hryniewicki, 2018) used ROC curves as evaluation parameter (Zimek and Filzmoser, 2018). Outlier ranking coefficients (Muller et al., 2011), advanced correlation coefficients (Blest et al. 2000; Kumar and Vassilvitski, 2010), visual analytic system (Xu et al., 2019) are some advanced techniques used for evaluation of outlier ensembles. All these techniques try to reduce the bias and variance of the outlier detection system (Ryana et al., 2016).

Zhang et al. (2021) used a weighted voting mechanism to create an outlier adapted training dataset for credit scoring system. Also, researchers used local outlier factor algorithm enhanced with bagging strategy for effectively identifying the outliers (Zhang et al., 2021; Reunanen et al., 2020). Yahaya et

al. (2019) introduced Consensus Novelty Detection Ensemble for human activity recognition, which is a data driven and environment invariant method for outlier detection. Carcillo et al. (2019) used a hybrid approach that combines supervised and unsupervised outlier detection techniques. Wang et al. (2020) proposed a dynamic selection mechanism, which selects most dynamic independent base detectors for each test pattern. In order to handle class imbalance problems in the anomaly samples ensemble approach that combines One Class Classifiers (OCC) is used for outlier detection (Fujita et al., 2020).

However, identifying which outlier ensemble is suitable for the given application data is a daunting task, since it depends on the characteristics of regular instances and the statistics of the anomalies present in the data. As a result, a workbench must be developed to test outlier ensemble algorithms in various regular and anomalous data combinations. This workbench must be structured in such a way that it can test all conceivable outlier ensemble combinations, and a user must be able to select an outlier ensemble algorithm from this workbench without having to do any more processing for ensemble algorithm selection. Main contributions of the proposed system compared to the existing systems are explained in Table 1.

WORKBENCH DESIGN

There is a need for creating a workbench for the evaluation of outlier detection algorithms due to the unavailability of ground truth (Steinbuss and Bohm, 2021). Hence, we developed a workbench structure and process, which has been demonstrated here for the analysis of two types of regular instances, Gaussian distribution and random walk model. The reason for selecting Gaussian distribution is that most of the real time applications follow Gaussian distribution (Domingues et al, 2017) and random walk model is selected to simulate a data that has highest variance.

After generating normal data objects, outlier points are inserted from a separate uniform distribution (Steinbuss and Bohm, 2021). In the current analysis, experiments are conducted using two types of anomalies with different frequencies. Local and global anomalies are the anomaly types utilised in this paper. Current workbench design uses three to eight percent of outliers in the data. Here, the percentage of outliers represents the frequency of outliers in data. After generating relevant dataset, a set of anomaly detection algorithms are selected for creating the evaluation methodology part of the workbench.

Table 1. Main contributions of the proposed system

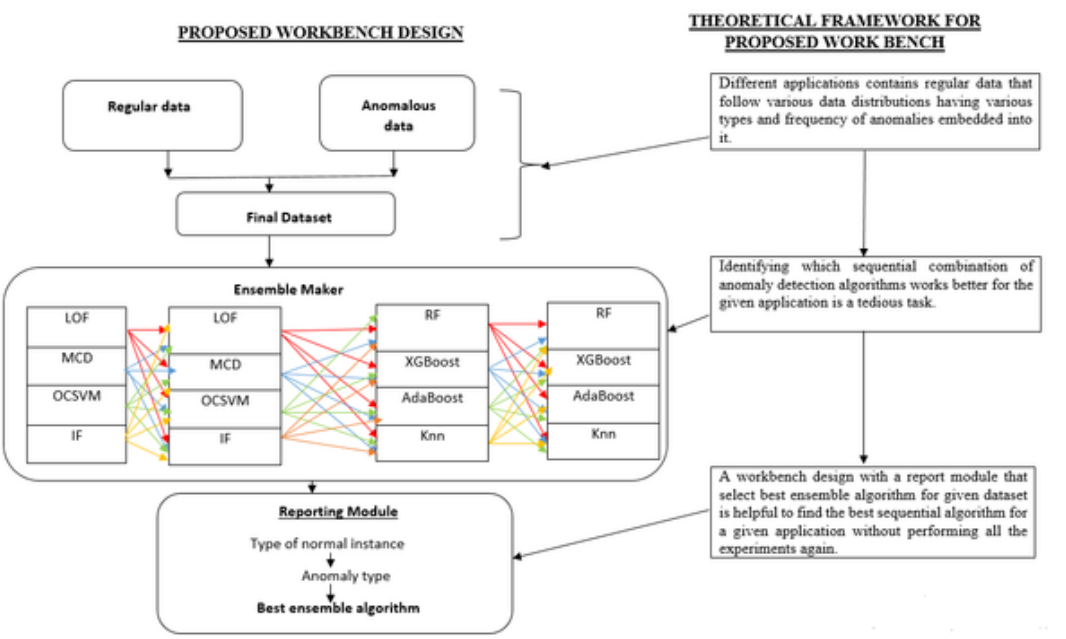
Limitation of existing systems	Contribution of the proposed system
Both independent and sequential ensembles require a combination function, which combines the outlier scores from different detection techniques (Aggarwal and Sathe, 2017; Gao and Tan, 2006). Important factor in this case is the selection of method for combining the outlier scores	The proposed system uses a confusion matrix as a scoring mechanism. In the proposed workbench, unsupervised algorithm has a simulated ground truth for comparison. This rectifies the difficulty in combining the outlier scores.
Main challenge in the creation of an ensemble is the selection of different algorithms as ensemble members and the method used to combine various detector outputs (Campos et al, 2018)	This paper proposes a workbench that works with different datasets and select best outlier ensemble algorithm by combing the selected supervised and unsupervised algorithms. Hence, for a given application user need not select an ensemble member for their application. Best combination for the given application is available in the proposed workbench report.
All the existing algorithms are designed to work on a single real time application. No generalised approaches are available in the area of ensemble-based outlier detection algorithms (Zhang et al., 2021; Carcillo et al., 2019).	Proposed system designed to work on two different types of data, Gaussian distribution and random walk model. This workbench acts as a generalised design for all applications that follows the given data distribution.

Aggarwal and Sathe (2017) classified outlier ensembles into sequential ensembles and independent ensembles. In a sequential outlier ensemble, the same algorithm or a different algorithm is applied sequentially. Succeeding algorithms are applied on the results of the preceding ones. This workbench operates on the basic concepts of sequential ensembles by utilising the data obtained from the generator in the workbench.

Figure 2 shows connection between the theoretical motivations for this research and how the problems are solved using the proposed workbench design. First motivation for this research states that, different applications contain regular data that follow various data distributions having various types and frequency of anomalies embedded into it. In order to simulate this situation in the proposed study make use of the data generator that produces two types of regular data sets, Gaussian distribution and random walk distribution. Dataset is generated by following certain rules adopted from the previous studies in this area (Steinbuss and Bohm, 2021; Li et al. 2016; Emmott et al., 2013). Anomalies are inserted into this generated dataset. There can be two types of outliers, local outliers that have less deviation from normal data objects and global outliers that have higher deviation from the normal dataset. Outlier points are labelled as '1' and normal objects are labelled as '0' during the generation process to get a set of labelled data which will be helpful for accurate evaluation of algorithms. After combining normal data objects and anomaly points, the final dataset is given as an input to the ensemble-based outlier detection module. This forms a dataset which is similar to the real time application database.

Second theoretical motivation for this research is how to identify which sequential combination of anomaly detection algorithm works better for the given application. In an effort to solve this problem an experimental design with an algorithm module that has the following four supervised and four unsupervised algorithms has been developed. In this study Local Outlier Factor (LOF), Minimum Covariance Determinant (MCD) Estimator, One Class SVM (OCSVM) and Isolation Forest (IF) algorithms are the unsupervised base detectors. Random Forest (RF), XGBoost. AdaBoost and K nearest neighbour (Knn) algorithms are the supervised counterparts.

Figure 2. Proposed workbench design



Unsupervised techniques can be proximity-based approaches, statistical techniques or clustering based methods (Han et al, 2012). In this study, base detector LOF is a proximity-based outlier detection that marks a data object as an outlier based on its distance from its closest neighbours (Domingues et al., 2018). MCD estimator estimates the Mahalanobis distance of an object from the centre of a data cloud to estimate the outliers (Hubert et al, 2018). OCSVM calculates the distance between data objects indirectly using kernel functions. Isolation forest is an unsupervised algorithm that classifies the dataset by isolating the outlier points (Liu et al, 2008). These four techniques are the unsupervised base detectors used here.

Supervised outlier detection techniques model both normal and abnormal data. K nearest neighbour classifier calculates the similarity between a data object and each training data instance. A test data object is classified under the same category of the training data instance to which it is most similar (Liao and Rao, 2002). Random Forest is an ensemble method that generates a number of random decision trees in parallel, final prediction on the outlier point is based on the majority votes obtained from individual decision trees (Angelo et al, 2018). In eXtreme gradient boosting (XGBoost), is a sequential ensemble technique. At the initial stage, each data object has an equal probability of being selected by a decision tree for further analysis. At the sequential stages new trees are added into the results of the previous decision trees which help to reduce the loss function and overall prediction error (Callens et al, 2020). AdaBoost is also a sequential classifier, in which the weights of the training instances are initialised with the same value while modifying these weights at subsequent iterations based on the training error rate. This will help the succeeding learners to deal accurately with those instances that were misclassified previously. Since these hard patterns will have higher probability of being an outlier, Adaboost is a good choice for outlier detection (Xing and Liu, 2019; Ferreira and Figueiredo, 2012). These four algorithms are selected to form base detectors from the category of supervised outlier detection algorithms.

Consensus approaches in the ensemble algorithm tells us how to combine various base detectors. Here also, after selecting the base algorithms, algorithms must be combined to get the final ensemble. Although some of the related works suggest that poor performing detectors need to be removed from the analysis (Rayana et al., 2016), distinguishing the efficient combination is a toughest job for a data scientist. Hence, all algorithm combinations are taken in the current analysis in order to find the highest performing combination by checking all the 64 combinations obtained by combining the same set of algorithms in the first and second stage of the ensemble.

Variance of outlier detection algorithms can be reduced by different instantiations of data or by randomized instantiations of detection algorithms (Aggarwal and Sathe, 2017). To achieve this characteristic in the proposed approach, dataset having varying percentages of local and global outliers are used in this analysis. Experiments are conducted on two different data distributions. First one is a Gaussian distribution and the second one is a random walk model. The Algorithm used in the workbench is explained in the next section.

Overall error of an algorithm is measured as the sum of squared bias value and variance value (Aggarwal and Sathe, 2017). In order to reduce bias of our algorithm design we eliminate outliers generated from the first stage of an outlier detection algorithm and select this refined data for the second stage. In the succeeding stage, the detector works on this refined data, this helps to build an ensemble model which is more robust to outliers.

After designing the ensemble maker in order to solve the second theoretical problem, output of the ensemble module will generate a report. This report module presents the type of normal and anomalous data and the best algorithm for each input combination. By comparing the application dataset with the workbench input, this methodology assists the data scientist in selecting an acceptable ensemble algorithm for their application, depending on the features of the data and anomalies of the given application. The researchers can use the final result to choose the appropriate algorithm combination for their application data. Thus, the proposed workbench design will be helpful to find

the best sequential ensemble outlier detection algorithm for a given application without performing all the experiments again and again.

```
Workbench Design Algorithm
Begin
Input: type of distribution: dist
Variables: data_size, outlier_percentage, outlier_type D: Initial
dataset
Output: evaluation metrics for different outlier combinations
For 'i' in number of classifiers
Call Gen (dist):
        Generate Gaussian (data_size, outlier_percentage,
outlier_type)
        Call Stage1 (D)
Generate_Randomwalk (data_size, outlier_percentage, outlier_type)
        Call Stage1 (D)
Call Stage1 (D):
clf_name: Call:RF, Knn, XGBoost,AdaBoost,IF,LOF,MCD,OCSVM
O1: S1 outliers
For 'j' in in number of classifiers
Input: D1= D- O1
Call S2 (D1):
S2: Call: RF,Knn,XGBoost, AdaBoost,IF,LOF,MCD,OCSVM
O2: Stage 2 outliers
S2 Inliers = D1-O2
Set   pred_outliers =D- S2 Inliers as '1'
pred_output=S2 Inliers+ pred_outliers
target_output=getlabels_inputfile ()
Call Eval: confusion_mat (target Output, pred_output)
outlier_detect_rate, false_alarm_rate
Output: S1 clf_name, S2 clf_name outlier_detect_rate
S1 clf_name, S2 clf_name false_alarm_rate
End
```

Workbench Characterisation Results

Experiments were conducted with the following percentages of outliers, 3%, 5%, 7% and 8%. Selection of these outlier percentages is based on the fact that most of the real time applications contain less than 5% of outliers (Steinbuss and Bohm, 2021; Li et al. 2016; Emmott et al., 2013). Both local outlier points and global outlier points are included in the study. Since most of the applications have a high number of samples, sample size is taken as 10000. Most commonly used outlier detection algorithms and their combinations are taken for analysis. A brute force approach is used selecting the final combination (Zhao et al, 2015). Table 2-6 contains results of all the algorithms combinations as well as the performance of independent algorithms.

Table 2 and Table 3 give the outlier detection rate and false alarm rate of the independent base detectors in Gaussian distribution and Random walk model in the presence of local and global anomalies with varying anomaly frequencies. Table 4 and Table 5 give outlier detection rate and false alarm rate of different detector combinations in Gaussian distribution in the presence of global and local anomalies. Table 6 and Table 7 presents results obtained for random walk models in the presence of global and local outliers. Here, outlier detection rate is the number of outliers that are correctly classified by the detector combination under specified conditions of data and anomalies.

Table 2. Performance of independent outlier detection algorithms: Gaussian distribution

Algorithm	Outlier detection rate								False alarm rate							
	3%	5%	7%	8%	3%	5%	7%	8%	3%	5%	7%	8%	3%	5%	7%	8%
	Global outlier				Local outlier				Global outlier				Local outlier			
Isolation Forest	0.555	0.566	0.643	0.640	0.02	0.08	0.168	0.18	0.0147	0.0242	0.032	0.0313	0.03	0.05	0.069	0.08
One Class SVM	1.0	1.0	1.0	1.0	0.88	0.89	0.87	0.85	0.484	0.474	0.462	0.463	0.48	0.47	0.47	0.46
Local Outlier Factor	0.334	0.383	0.273	0.258	0.017	0.09	0.11	0.16	0.0215	0.033	0.0596	0.058	0.03	0.05	0.07	0.09
Minimum Covariance Estimator	0.581	0.614	0.736	0.701	0.013	0.09	0.177	0.25	0.0139	0.021	0.0251	0.0269	0.03	0.05	0.06	0.08
Random Forest	.434	0.586	0.690	0.67	0.013	0.10	0.18	0.20	0.012	0.0245	0.024	0.021	0.002	0.008	0.02	0.03
XG Boost	0.492	0.540	0.717	0.664	0	0.06	0.11	0.17	0.011	0.0256	0.024	0.0203	0.003	0.007	0.01	0.02
Ada Boost	0.449	0.620	0.855	0.694	0	0.01	0.02	0.09	0.0077	0.0287	0.028	0.0225	0.001	0.001	0.002	0.005
K nearest neighbor classifier	0.405	0.551	0.611	0.634	0.041	0.212	0.31	0.31	0.014	0.0271	0.023	0.0235	0.02	0.04	0.05	0.06

Table 3. Performance of independent outlier detection algorithms: Random walk model

Algorithm	Outlier detection rate								False alarm rate							
	3%	5%	7%	8%	3%	5%	7%	8%	3%	5%	7%	8%	3%	5%	7%	8%
	Global outlier				Local outlier				Global outlier				Local outlier			
Isolation Forest	0.032	0.044	0.076	0.104	0.05	0.05	0.06	0.08	0.031	0.044	0.082	0.088	0.03	0.04	0.06	0.08
One Class SVM	0.464	0.504	0.504	0.501	0.51	0.45	0.47	0.53	0.501	0.499	0.499	0.499	0.49	0.5	0.50	0.49
Local Outlier Factor	0.561	0.547	0.624	0.620	0.09	0.11	0.17	0.17	0.015	0.022	0.037	0.0421	0.029	0.04	0.06	0.07
Minimum Covariance Estimator	0.038	0.084	0.099	0.129	0.02	0.04	0.08	0.08	0.031	0.042	0.080	0.086	0.03	0.04	0.067	0.08
Random Forest	0.035	0.097	0.168	0.2	0.03	0.01	0.05	0.02	0.002	0.002	0.002	0.009	0.008	0.012	0.01	0.02
XG Boost	0.0175	0.083	0.054	0.127	0	0.01	0	0	0	0	0	0.0027	0.001	0.001	0.001	0.002
Ada Boost	0	0.013	0.006	0.024	0	0	0	0	0	0	0.0016	0.00108	0	0.0005	0.0005	0.002
K nearest neighbor classifier	0.140	0.222	0.265	0.333	0.05	0	0.11	0.07	0.008	0.0129	0.0141	0.021	0.025	0.04	0.05	0.06

False alarm rate is the number of normal data objects that are misclassified as outliers. An algorithm with highest detection rate and low false alarm rate is the best detector under specific input conditions.

Summary of the experimental data used in the analysis is given in Table 8.

Although results given in Tables give detailed information about experiments, end users are interested only in the best combinations for a particular dataset. Hence, graphical representation of top results aids the user for picking out a best ensemble for a particular data and anomaly statistics.

First two graphs in Figure 3 present the comparative analysis of best performing independent detectors with the best performing two stage detectors in terms of outlier detection rate and false alarm rate in case of Gaussian distribution in the presence of global outliers. In the case of Gaussian distribution, two stage algorithms, MCD-MCD, IF-IF combination or IF-MCD combinations have shown higher efficiency when compared with highest performing independent detectors and other

Table 4. Performance of Outlier Ensembles: Outlier detection rate: Gaussian distribution

First Stage	Second Stage															
	Global outliers								Local outliers							
	MCD	LOF	IF	OC SVM	RF	XG Boost	Ada Boost	Knn	MCD	LOF	IF	OC SVM	RF	XG Boost	Ada Boost	Knn
Outlier Percentage: 3%																
MCD	0.90	0.77	0.85	1.0	0.37	0.34	0.17	0.34	0.21	0.21	0.24	0.92	0.04	0.02	0	0.18
LOF	0.83	0.55	0.81	1.0	0.4	0.36	0.36	0.48	0.21	0.19	0.20	0.92	0.05	0.04	0	0.20
IF	0.9	0.70	0.87	1.0	0.34	0.39	0.23	0.42	0.22	0.21	0.20	0.92	0.10	0.10	0	0.21
OCSVM	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.95	0.93	0.94	1.0	0	0	0	0
RF	0.69	0.49	0.64	1.0	0.53	0.55	0.64	0.516	0.07	0.07	0.07	0.91	0.02	0.02	0.012	0.14
XGBoost	0.69	0.49	0.66	1.0	0.36	0.46	0.43	0.48	0.07	0.07	0.06	0.91	0.02	0.05	0	0.10
AdaBoost	0.68	0.49	0.65	1.0	0.39	0.42	0.42	0.40	0.05	0.05	0.05	0.91	0.04	0.04	0	0.21
Knn	0.7	0.49	0.62	1.0	0.47	0.53	0.50	0.43	0.08	0.09	0.09	0.91	0.01	0.02	0.01	0.18
Outlier Percentage: 5%																
MCD	0.97	0.83	0.94	1.0	0.48	0.4	0.45	0.37	0.29	0.25	0.28	0.90	0.06	0.09	0.009	0.18
LOF	0.85	0.50	0.80	1.0	0.58	0.56	0.58	0.57	0.27	0.24	0.25	0.93	0.03	0.05	0	0.20
IF	0.97	0.79	0.97	1.0	0.36	0.34	0.26	0.42	0.29	0.24	0.26	0.92	0.11	0.04	0.02	0.28
OCSVM	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.95	0.94	0.94	1.0	0	0	0	0.1
RF	0.73	0.44	0.67	1.0	0.57	0.55	0.56	0.47	0.09	0.07	0.07	0.91	0.06	0.05	0.03	0.21
XGBoost	0.72	0.42	0.68	1.0	0.58	0.6	0.63	0.5	0.10	0.09	0.09	0.91	0.10	0.07	0.002	0.18
AdaBoost	0.73	0.44	0.70	1.0	0.62	0.61	0.57	0.61	0.08	0.06	0.06	0.90	0.05	0.07	0.06	0.21
Knn	0.73	0.44	0.64	1.0	0.58	0.60	0.62	0.55	0.12	0.11	0.10	0.91	0.11	0.07	0.03	0.21
Outlier Percentage: 7%																
MCD	1.0	0.88	0.99	1.0	0.61	0.52	0.61	0.44	0.38	0.33	0.36	0.90	0.10	0.08	0.009	0.23
LOF	0.87	0.40	0.80	1.0	0.73	0.74	0.86	0.62	0.34	0.26	0.33	0.92	0.08	0.06	0	0.24
IF	1.0	0.78	0.97	1.0	0.60	0.50	0.43	0.64	0.37	0.30	0.36	0.87	0.11	0.08	0.008	0.18
OCSVM	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.94	0.90	0.95	1.0	0	0	0	0.11
RF	0.84	0.47	0.77	1.0	0.78	0.73	0.84	0.67	0.16	0.15	0.13	0.86	0.09	0.07	0.01	0.20
XGBoost	0.84	0.45	0.77	1.0	0.7	0.76	0.77	0.61	0.15	0.13	0.14	0.86	0.07	0.07	0.007	0.16
AdaBoost	0.85	0.48	0.78	1.0	0.8	0.8	0.93	0.61	0.14	0.10	0.115	0.85	0.17	0.12	0.007	0.27
Knn	0.82	0.76	0.73	1.0	0.70	0.72	0.76	0.57	0.18	0.16	0.14	0.87	0.04	0.08	0.02	0.23
Outlier Percentage: 8%																
MCD	1.0	0.87	0.99	1.0	0.47	0.52	0.45	0.52	0.49	0.40	0.41	0.92	0.17	0.14	0.01	0.25
LOF	0.88	0.39	0.81	1.0	0.77	0.76	0.91	0.71	0.43	0.30	0.44	0.91	0.12	0.13	0.02	0.24
IF	1.0	0.75	0.99	1.0	0.81	0.75	0.77	0.62	0.47	0.36	0.45	0.92	0.21	0.18	0.008	0.25
OCSVM	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.96	0.90	0.95	1.0	0	0.11	0	0.17
RF	0.88	0.43	0.78	1.0	0.75	0.79	0.85	0.70	0.22	0.18	0.19	0.88	0.12	0.13	0.04	0.25
XGBoost	0.88	0.42	0.81	1.0	0.79	0.78	0.88	0.73	0.21	0.18	0.16	0.88	0.13	0.12	0.03	0.30
AdaBoost	0.88	0.44	0.82	1.0	0.74	0.76	0.81	0.63	0.19	0.14	0.18	0.88	0.16	0.14	0.03	0.28
Knn	0.87	0.39	0.76	1.0	0.83	0.81	0.89	0.67	0.24	0.19	0.23	0.89	0.16	0.13	0.02	0.21

sequential ensembles. Although One Class SVM (OCSVM) shows higher detection rate, their false alarm rate is higher which makes it unsuitable for sensitive applications (Raman et al., 2017), hence we have not included OCSVM in the graphical presentation. Thus, MCD-MCD ensembles, IF-IF ensembles and IF - MCD combinations are the best performing algorithms in Gaussian distribution with global outliers.

Graph 3 and 4 of Figure 3 shows the performance of independent and sequential ensembles in random walk model with global anomalies embedded into it. Graph 3 presents outlier detection rate

Table 5. Performance of Outlier Ensembles: False alarm rate: Gaussian distribution

First Stage	Second Stage								Second Stage							
	Global Outlier								Local Outlier							
	MCD	LOF	IF	OCSVM	RF	XG Boost	Ada Boost	Knn	MCD	LOF	IF	OCSVM	RF	XG Boost	Ada Boost	Knn
Outlier Percentage: 3%																
MCD	0.04	0.05	0.04	0.50	0.007	0.007	0.004	0.01	0.07	0.07	0.07	0.50	0.008	0.008	0.0005	0.02
LOF	0.05	0.06	0.05	0.50	0.01	0.01	0.007	0.01	0.07	0.07	0.07	0.50	0.007	0.007	0.007	0.03
IF	0.04	0.05	0.04	0.50	0.009	0.01	0.007	0.01	0.07	0.07	0.07	0.50	0.004	0.002	0.006	0.03
OCSVM	0.50	0.50	0.50	0.74	0.486	0.486	0.486	0.486	0.50	0.50	0.50	0.74	0	0.001	0.002	0.004
RF	0.02	0.03	0.02	0.48	0.013	0.01	0.02	0.015	0.04	0.04	0.04	0.48	0.007	0.007	0.0005	0.02
XGBoost	0.02	0.03	0.02	0.48	0.01	0.01	0.003	0.01	0.04	0.04	0.04	0.48	0.009	0.007	0.001	0.03
AdaBoost	0.02	0.03	0.02	0.48	0.01	0.01	0.01	0.01	0.03	0.03	0.03	0.48	0.007	0.007	0.0005	0.03
Knn	0.02	0.03	0.02	0.48	0.01	0.01	0.01	0.01	0.04	0.04	0.04	0.48	0.009	0.004	0.0015	0.02
Outlier Percentage: 5%																
MCD	0.06	0.06	0.06	0.50	0.008	0.006	0.009	0.009	0.08	0.09	0.08	0.50	0.007	0.01	0.003	0.04
LOF	0.06	0.08	0.06	0.50	0.01	0.01	0.01	0.01	0.08	0.09	0.09	0.50	0.01	0.01	0.002	0.04
IF	0.06	0.06	0.06	0.50	0.009	0.008	0.007	0.007	0.08	0.09	0.08	0.50	0.01	0.005	0.002	0.03
OCSVM	0.50	0.50	0.50	0.74	0.486	0.486	0.486	0.486	0.50	0.50	0.50	0.73	0.001	0.001	0.03	0.005
RF	0.02	0.04	0.03	0.47	0.01	0.01	0.01	0.01	0.05	0.05	0.05	0.48	0.01	0.009	0.003	0.04
XGBoost	0.02	0.04	0.02	0.47	0.01	0.01	0.02	0.01	0.05	0.05	0.05	0.48	0.01	0.01	0	0.04
AdaBoost	0.02	0.04	0.02	0.47	0.02	0.02	0.01	0.02	0.04	0.05	0.05	0.47	0.006	0.01	0.01	0.04
Knn	0.02	0.04	0.03	0.47	0.01	0.01	0.01	0.01	0.05	0.05	0.05	0.48	0.03	0.007	0.004	0.03
Outlier Percentage: 7%																
MCD	0.08	0.09	0.08	0.50	0.008	0.009	0.009	0.008	0.12	0.12	0.12	0.50	0.02	0.02	0.002	0.05
LOF	0.09	0.13	0.09	0.50	0.02	0.02	0.02	0.02	0.12	0.13	0.12	0.50	0.02	0.01	0.001	0.05
IF	0.08	0.10	0.08	0.50	0.01	0.01	0.01	0.03	0.12	0.13	0.12	0.47	0.02	0.008	0.0005	0.05
OCSVM	0.50	0.48	0.50	0.48	0.48	0.48	0.48	0.48	0.50	0.50	0.50	0.73	0.002	0.005	0.005	0.01
RF	0.03	0.06	0.04	0.46	0.02	0.02	0.03	0.02	0.07	0.07	0.07	0.47	0.01	0.01	0.002	0.04
XGBoost	0.03	0.06	0.04	0.46	0.01	0.01	0.02	0.01	0.07	0.07	0.07	0.47	0.02	0.02	0.005	0.06
AdaBoost	0.03	0.06	0.04	0.47	0.02	0.02	0.03	0.03	0.06	0.07	0.07	0.47	0.02	0.003	0.007	0.06
Knn	0.03	0.06	0.04	0.46	0.02	0.02	0.02	0.01	0.08	0.08	0.08	0.48	0.02	0.01	0.004	0.05
Outlier Percentage: 8%																
MCD	0.10	0.11	0.10	0.50	0.004	0.005	0.003	0.005	0.14	0.14	0.14	0.51	0.02	0.01	0	0.05
LOF	0.11	0.16	0.12	0.50	0.03	0.03	0.03	0.02	0.14	0.15	0.14	0.51	0.02	0.02	0.002	0.05
IF	0.10	0.12	0.10	0.50	0.01	0.01	0.01	0.01	0.14	0.15	0.14	0.51	0.02	0.02	0.002	0.06
OCSVM	0.50	0.48	0.50	0.48	0.48	0.48	0.48	0.48	0.50	0.51	0.50	0.72	0.002	0.002	0.002	0.008
RF	0.04	0.08	0.05	0.46	0.02	0.02	0.02	0.02	0.08	0.08	0.08	0.47	0.02	0.02	0.005	0.06
XGBoost	0.04	0.08	0.04	0.46	0.02	0.02	0.02	0.02	0.08	0.08	0.08	0.46	0.02	0.01	0.003	0.05
AdaBoost	0.04	0.08	0.02	0.46	0.02	0.02	0.02	0.02	0.07	0.08	0.08	0.46	0.02	0.02	0.002	0.06
Knn	0.04	0.08	0.05	0.46	0.02	0.02	0.03	0.02	0.09	0.09	0.09	0.47	0.03	0.02	0.003	0.04

and Graph 4 depicts the false alarm rate in different combinations. In a random walk model, combining LOF algorithm with other algorithms will give maximum performance.

Figure 4 gives a performance comparison of the best single stage algorithm with ensemble algorithms in the presence of local outliers. Graph 1 and Graph 2 of Figure 4 aids the user to identify the best performing algorithm in Gaussian distribution. Graph 3 and Graph 4 presents the best algorithms in the random walk model. In Gaussian distribution, MCD-MCD, IF-MCD and IF-IF combinations have higher detection rate. Although, single stage OCSVM gives higher detection in a random walk

Table 6. Performance of Outlier Ensembles: Outlier detection rate: Random walk model

First Stage	Second Stage								Second Stage							
	Global outliers								Local Outliers							
	MCD	LOF	IF	OCSVM	RF	XG Boost	Ada Boost	Knn	MCD	LOF	IF	OCSVM	RF	XG Boost	Ada Boost	Knn
Outlier Percentage: 3%																
MCD	0.06	0.53	0.08	0.44	0.03	0.05	0.01	0.11	0.05	0.27	0.07	0.52	0.01	0.01	0.01	0.01
LOF	0.53	0.55	0.53	0.74	0	0	0	0.06	0.25	0.26	0.28	0.64	0	0	0	0.02
IF	0.08	0.53	0.08	0.43	0.03	0.01	0	0.24	0.07	0.27	0.06	0.52	0	0	0	0.02
OCSVM	0.42	0.72	0.43	0.69	0.02	0.05	0	0.21	0.51	0.63	0.50	0.72	0	0	0	0
RF	0.04	0.52	0.05	0.42	0.08	0.05	0	0.15	0.03	0.24	0.05	0.50	0	0	0	0.02
XGBoost	0.04	0.50	0.05	0.42	0.07	0.01	0	0.20	0.03	0.24	0.05	0.50	0	0	0	0.01
AdaBoost	0.03	0.51	0.04	0.42	0.08	0	0	0.1	0.03	0.25	0.04	0.50	0	0	0	0
Knn	0.07	0.52	0.08	0.44	0.06	0.03	0	0.07	0.03	0.25	0.05	0.50	0	0	0.019	0
Outlier Percentage: 5%																
MCD	0.12	0.56	0.15	0.53	0.17	0.12	0.04	0.13	0.09	0.33	0.13	0.54	0.06	0.03	0.02	0.11
LOF	0.57	0.60	0.58	0.76	0	0	0	0.04	0.34	0.33	0.35	0.67	0	0	0	0.05
IF	0.13	0.57	0.19	0.54	0.23	0.14	0.03	0.22	0.13	0.36	0.15	0.55	0.01	0	0	0.08
OCSVM	0.53	0.75	0.57	0.72	0.1	0.08	0.02	0.16	0.53	0.66	0.55	0.74	0	0	0	0.08
RF	0.07	0.55	0.10	0.54	0.10	0.06	0.04	0.18	0.05	0.32	0.08	0.52	0.05	0.03	0.02	0.07
XGBoost	0.06	0.55	0.10	0.53	0.17	0.10	0.05	0.16	0.05	0.31	0.09	0.52	0.03	0.01	0.01	0.10
AdaBoost	0.05	0.55	0.07	0.53	0.14	0.07	0.02	0.19	0.05	0.31	0.07	0.52	0.02	0.01	0.009	0.09
Knn	0.09	0.55	0.10	0.55	0.17	0.09	0.03	0.22	0.06	0.32	0.09	0.52	0.07	0.03	0.02	0.07
Outlier Percentage: 7%																
MCD	0.27	0.64	0.38	0.60	0.28	0.28	0.13	0.23	0.22	0.41	0.25	0.53	0.08	0.04	0.02	0.20
LOF	0.63	0.67	0.63	0.82	0.01	0	0	0.03	0.42	0.43	0.42	0.68	0.02	0	0	0.05
IF	0.29	0.64	0.38	0.62	0.18	0.15	0.04	0.18	0.22	0.42	0.24	0.55	0.07	0.03	0.01	0.11
OCSVM	0.65	0.82	0.70	0.76	0.29	0.35	0.21	0.29	0.58	0.67	0.60	0.74	0.04	0.04	0.03	0.18
RF	0.15	0.61	0.23	0.57	0.23	0.26	0.12	0.26	0.12	0.37	0.15	0.51	0.04	0.03	0.01	0.15
XGBoost	0.15	0.61	0.23	0.57	0.22	0.23	0.11	0.23	0.11	0.37	0.16	0.51	0.09	0.04	0.03	0.12
AdaBoost	0.13	0.61	0.22	0.56	0.27	0.26	0.14	0.26	0.10	0.37	0.15	0.50	0.12	0.10	0.05	0.16
Knn	0.17	0.62	0.25	0.58	0.27	0.26	0.09	0.22	0.13	0.38	0.16	0.52	0.05	0.02	0.01	0.11
Outlier Percentage: 8%																
MCD	0.18	0.66	0.25	0.57	0.24	0.21	0.11	0.35	0.15	0.48	0.15	0.54	0.07	0.02	0	0.16
LOF	0.66	0.68	0.66	0.81	0	0	0	0.07	0.48	0.46	0.47	0.73	0	0	0.01	0.06
IF	0.21	0.66	0.30	0.57	0.26	0.21	0.08	0.25	0.19	0.46	0.18	0.54	0.07	0.06	0	0.16
OCSVM	0.56	0.81	0.63	0.75	0.23	0.26	0	0.41	0.57	0.72	0.55	0.75	0.11	0.04	0	0.21
RF	0.14	0.64	0.18	0.53	0.21	0.15	0.05	0.30	0.11	0.43	0.11	0.52	0.07	0.006	0	0.13
XGBoost	0.13	0.63	0.18	0.52	0.24	0.17	0.08	0.34	0.09	0.43	0.09	0.51	0.06	0.02	0	0.16
AdaBoost	0.115	0.63	0.16	0.51	0.24	0.19	0.06	0.29	0.09	0.43	0.09	0.51	0.08	0.03	0.006	0.18
Knn	0.15	0.64	0.23	0.54	0.22	0.17	0.09	0.25	0.13	0.44	0.12	0.54	0.10	0.02	0.01	0.15

model it has a high false alarm rate. Hence, a data scientist can select LOF combinations for their risky applications in Gaussian distribution in the presence of local anomalies.

In the random walk model, single stage OCSVM outperforms other detector combinations with the penalty of high false alarm rate. However, combinations of LOF also give reasonable performance in this case. Users can select either OCSVM or LOF combinations based on sensitivity of applications.

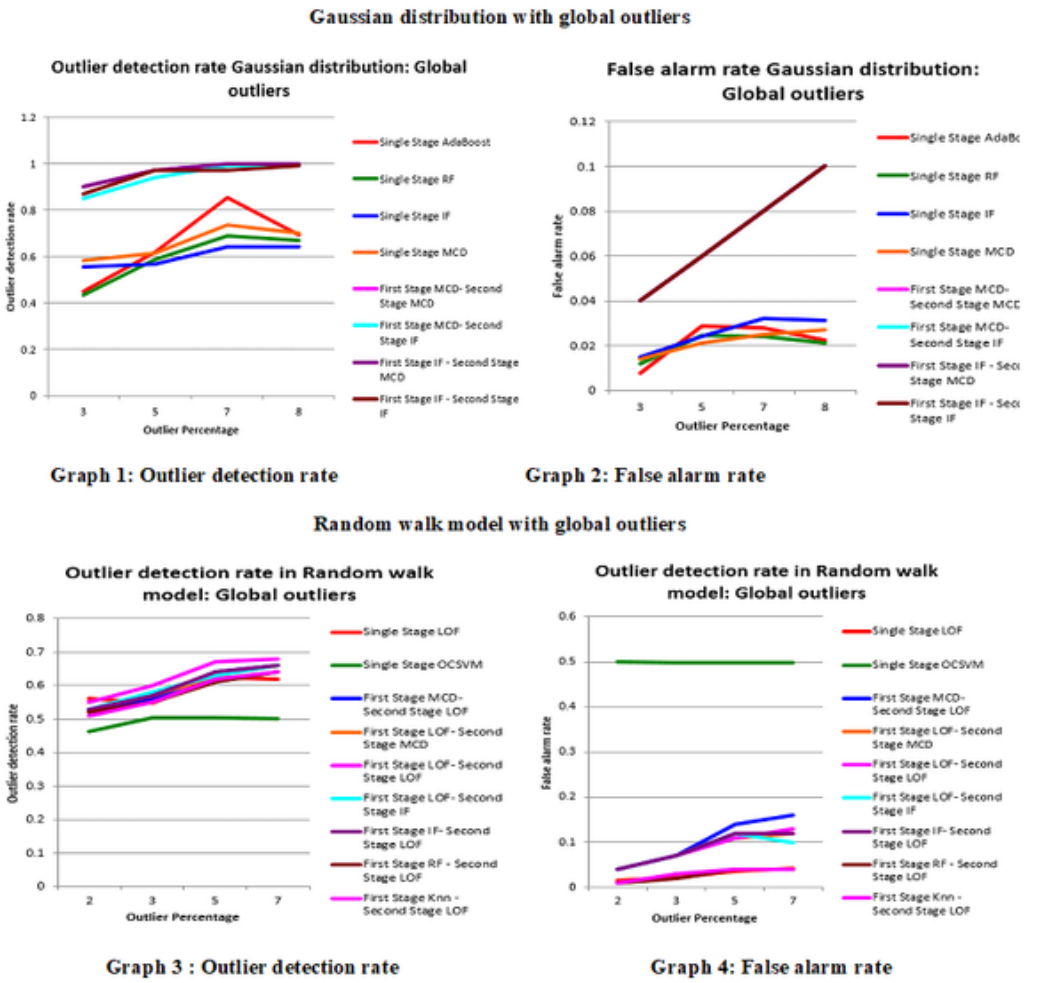
Table 7. Performance of Outlier Ensembles: False alarm rate: Random walk model

First Stage	Second Stage															
	Global Outliers								Local Outliers							
	MCD	LOF	IF	OCSVM	RF	XG Boost	Ada Boost	Knn	MCD	LOF	IF	OCSVM	RF	XG Boost	Ada Boost	Knn
Outlier Percentage: 3%																
MCD	0.06	0.04	0.06	0.51	0.001	0	0.005	0.006	0.05	0.04	0.05	0.51	0.003	0.0005	0	0.01
LOF	0.04	0.04	0.04	0.50	0.0005	0	0.0005	0.005	0.04	0.04	0.04	0.64	0.003	0	0.005	0.01
IF	0.06	0.04	0.06	0.51	0.0005	0	0	0.003	0.05	0.04	0.05	0.51	0.001	0	0	0.012
OCSVM	0.51	0.50	0.51	0.75	0.003	0.002	0	0.007	0.51	0.50	0.51	0.75	0.001	0	0	0.01
RF	0.03	0.01	0.03	0.50	0.003	0	0	0.01	0.02	0.02	0.02	0.50	0.002	0	0	0.01
XGBoost	0.03	0.01	0.03	0.50	0.002	0.001	0	0.008	0.02	0.02	0.02	0.49	0.002	0	0	0.01
AdaBoost	0.03	0.01	0.03	0.50	0.001	0.001	0.0005	0.01	0.02	0.02	0.02	0.49	0.001	0	0	0.01
Knn	0.03	0.01	0.03	0.50	0.001	0	0	0.003	0.02	0.02	0.02	0.50	0.003	0	0	0.01
Outlier Percentage: 5%																
MCD	0.10	0.07	0.09	0.52	0.002	0.001	0.001	0.009	0.10	0.08	0.09	0.52	0.007	0.001	0	0.02
LOF	0.07	0.07	0.07	0.51	0.003	0	0.0005	0.008	0.08	0.08	0.08	0.51	0.006	0	0	0.02
IF	0.10	0.07	0.09	0.52	0.001	0	0	0.01	0.09	0.08	0.09	0.52	0.007	0.002	0	0.02
OCSVM	0.52	0.51	0.52	0.75	0.004	0.004	0.001	0.008	0.52	0.51	0.52	0.75	0.01	0.005	0	0.01
RF	0.05	0.02	0.05	0.49	0.0005	0	0	0.01	0.05	0.03	0.05	0.49	0.004	0.001	0.005	0.01
XGBoost	0.05	0.02	0.05	0.49	0.004	0.0005	0.0005	0.013	0.05	0.03	0.04	0.49	0.006	0.001	0	0.02
AdaBoost	0.05	0.02	0.05	0.49	0.002	0.0005	0.0005	0.01	0.05	0.03	0.05	0.49	0.005	0.001	0.0005	0.02
Knn	0.05	0.03	0.05	0.49	0.002	0	0.001	0.009	0.05	0.04	0.05	0.50	0.006	0.001	0.0005	0.01
Outlier Percentage: 7%																
MCD	0.14	0.11	0.13	0.53	0.01	0.001	0.0005	0.01	0.14	0.13	0.14	0.53	0.01	0.002	0.0005	0.02
LOF	0.11	0.11	0.11	0.51	0.004	0.0005	0.0005	0.01	0.13	0.13	0.13	0.52	0.01	0.001	0.002	0.03
IF	0.14	0.11	0.13	0.53	0.01	0.001	0.0005	0.01	0.14	0.13	0.14	0.53	0.02	0.002	0	0.03
OCSVM	0.53	0.51	0.52	0.74	0.01	0.003	0.003	0.008	0.53	0.52	0.53	0.74	0.01	0.005	0.002	0.02
RF	0.08	0.04	0.07	0.49	0.01	0.002	0	0.01	0.08	0.06	0.08	0.50	0.02	0.004	0	0.03
XGBoost	0.07	0.04	0.07	0.49	0.009	0.001	0.0005	0.01	0.08	0.05	0.07	0.50	0.02	0.003	0	0.03
AdaBoost	0.07	0.03	0.07	0.49	0.004	0.0005	0.0005	0.01	0.07	0.05	0.07	0.49	0.02	0.002	0	0.02
Knn	0.08	0.04	0.07	0.49	0.007	0.003	0.0005	0.01	0.08	0.06	0.08	0.50	0.01	0.001	0	0.02
Outlier Percentage: 8%																
MCD	0.16	0.12	0.16	0.54	0.007	0.002	0.0005	0.01	0.17	0.14	0.17	0.54	0.01	0.001	0.001	0.03
LOF	0.12	0.12	0.12	0.52	0.006	0	0	0.01	0.14	0.14	0.14	0.52	0.01	0.005	0.001	0.02
IF	0.16	0.12	0.15	0.54	0.01	0.005	0	0.02	0.16	0.14	0.16	0.54	0.01	0.002	0.001	0.03
OCSVM	0.54	0.52	0.53	0.74	0.008	0.003	0.05	0.02	0.54	0.52	0.54	0.74	0.007	0.003	0	0.03
RF	0.08	0.04	0.08	0.49	0.009	0.001	0	0.09	0.08	0.06	0.08	0.49	0.01	0.002	0	0.03
XGBoost	0.08	0.04	0.08	0.49	0.007	0.002	0.001	0.01	0.08	0.05	0.08	0.49	0.009	0.002	0	0.03
AdaBoost	0.08	0.04	0.08	0.49	0.003	0	0.0005	0.015	0.08	0.05	0.08	0.49	0.01	0.001	0	0.03
Knn	0.09	0.04	0.08	0.50	0.004	0	0	0.019	0.09	0.06	0.09	0.50	0.01	0.002	0	0.02

Table 8. Summary of Experimental data used in the analysis

Type of Regular Dataset	Type of outlier	Frequency of outliers
Gaussian Distribution	Local outlier	3%,5%,7%,8%
Gaussian Distribution	Global outlier	3%,5%,7%,8%
Random walk model	Local outlier	3%,5%,7%,8%
Random walk model	Global outlier	3%,5%,7%,8%

Figure 3. Comparison of independent and sequential outlier ensembles for different datasets in the presence of global outliers



CASE STUDY

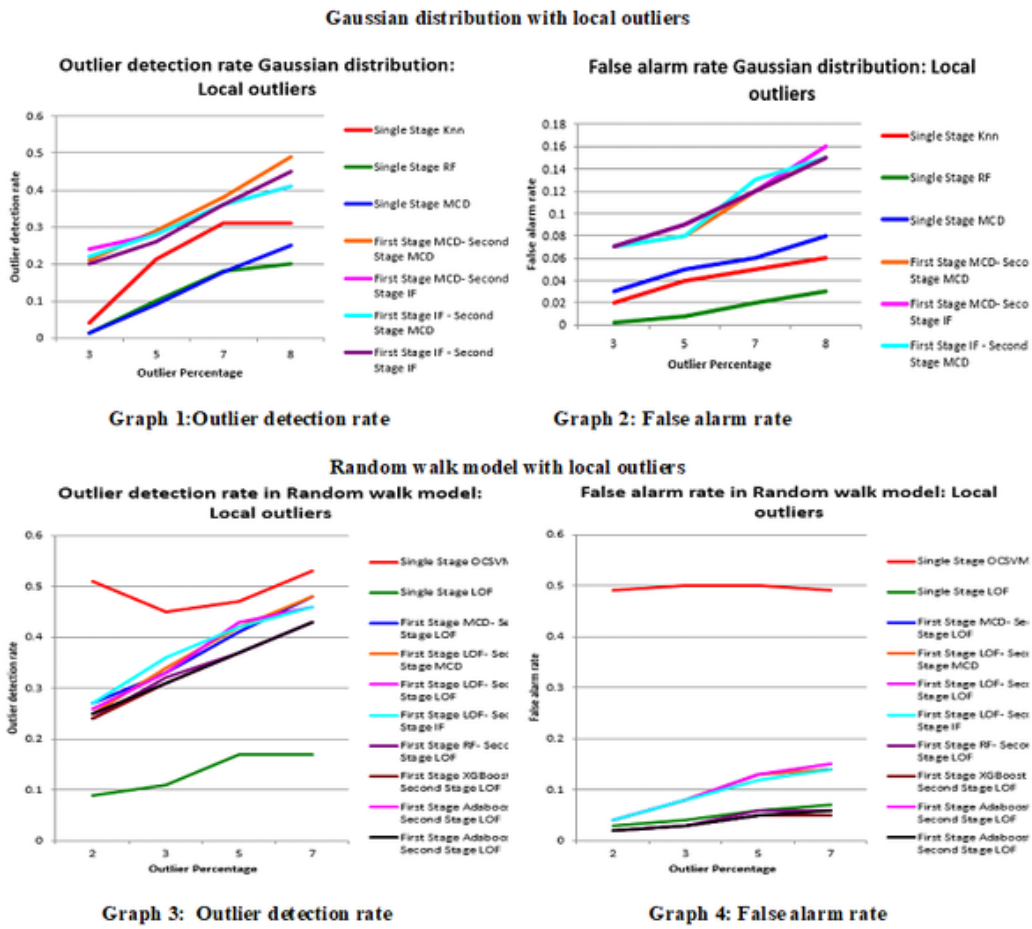
In this section two case studies are conducted. First part discusses study on a realistic dataset and the second part explains real time case study.

Case Study Based on Realistic Data

The proposed workbench is used to identify the best algorithm combination for a dataset that follows Gaussian distribution or random walk model with specific anomaly statistics. To demonstrate this methodology on a realistic dataset, a case study is conducted using yahoo anomaly benchmark data.

Flow chart for getting recommendation on which sequential outlier ensemble is better for the given application is given in Figure 5 This flowchart aids the user to select the best sequential outlier ensemble for their specific application. Because the workbench is set up to operate with two different sorts of regular instances, two case studies with the same data distribution are used in this paper.

Figure 4. Comparison of independent and sequential outlier ensembles for different datasets in the presence of local outliers



Gaussian Distribution

A data scientist must identify the data and anomaly statistics once it has been retrieved from the application database. The type of data distribution is determined using a Chi square test. In this case, the data follows a Gaussian distribution. It is found that the current dataset has 0.55% of global outliers which lies away from 3*Std. Dev. Performance of the sequential outlier ensembles along with its characteristics is given in Table 9.

The proposed work bench also recommends two stage algorithms MCD-MCD, IF-IF combination or IF-MCD combination for a dataset that follows Gaussian distribution. Case analysis assures that a data scientist can use the output from the workbench's report module if the user's individual application follows a Gaussian distribution.

Random Walk Model

In order to validate the accuracy of workbench design for a dataset which follows a random walk model, another set of data is taken from Yahoo anomaly benchmarks. Augmented Dickey-Fuller test is used to confirm that the dataset follows a random walk model. The selected data contains 0.7% of global outliers which lies at a distance higher than 3*Std.dev. Table 10 presents the best algorithms for the current case study using random walk model.

Figure 5. Flow chart for getting recommendation on sequential ensemble algorithm for the given application

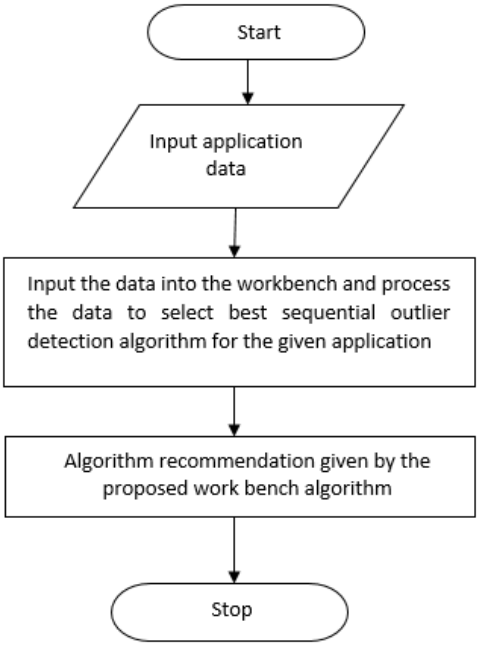


Table 9. Top performing algorithms in yahoo dataset: Gaussian distribution

Rank	Name of the algorithm	Evaluation metrics
1	IF-MCD, IF-IF,MCD-MCD,MCD-IF	Outlier detection rate:1.0 False alarm rate:0.0006

Table 10. Top performing algorithms in yahoo dataset: Random walk model

Rank	Name of the algorithm	Evaluation metrics
1	MCD-LOF, LOF-MCD, LOF-IF, IF-LOF,RF-IF, Knn-IF, XGBoost-IF, XGBoost-MCD	Outlier detection rate:1.0 False alarm rate:0.0001

Workbench experiments claim that LOF combinations give better detection rate than independent algorithms or other outlier ensembles. In the case analysis, along with LOF combinations IF combinations and XGBoost-MCD combinations also give better results. This minor discrepancy could be related to the fact that these global anomalies are more easily detectable due to their greater distance from the 3 * Std.dev.

Thus, the workbench definitely helps a data scientist to select the best performing algorithm for a particular distribution. These two case studies act as a guideline to select a particular sequential outlier ensemble algorithm for a given application.

Case Study Based on Real-Time Data

To evaluate the proposed methodology, an experimental study was conducted by using the real time dataset obtained from Advanced Centre for Atmospheric Radar Research (ACARR) located at Cochin

University of Science and Technology, Kerala, India. ACARR dedicated for studies in measurements pertaining to atmospheric variations. The data obtained from the centre contains time series data for different periods of the year. Here, event rain is considered as outlier. Pre-processing of the dataset is done by selecting only those variables that affect the occurrence of the event rain. In order to construct an outlier detection model, the study used the variables temperature, humidity, wind speed, pressure, and radiation. Rainfall is measured in mili meter in the ACARR centre and if rainfall value is greater than 0 then those observations are categorised under outlier class and the rest of the data points are labelled as inliers. Current study tries to find out occurrence of rain in the month of February in the year 2021 and in winter season. Winter season contains data analysis for three months December 2020, January 2021 and February 2021. In the month of February 2021, the event rain happened at a rate of 2.3% and during the winter season, rain events took place at a rate of 3.08%. Hence, analysis is conducted by considering percentage of anomalies as 2.3% and 3.08%. Inter seasonal analysis is not considered in the current analysis as those data may contain seasonality factor which may affect the accuracy of the model.

Analysis is conducted by using both independent and sequential algorithms and performance of both types of algorithms is presented visually using Figure 6 and Figure 7. Current analysis shows the performance of both independent and sequential ensemble-based outlier detection algorithms in the rain event detection for a single month and during the winter season.

Based on the proposed workbench design, if the type of the application data is known, report obtained from the work bench will suggest the best algorithm for the given application. The given dataset follows a random walk model. From figures 6 and 7 it is clear that sequential outlier ensembles have a greater detection rate and a reduced false alarm rate in the given real time data. Based on the workbench design and if the dataset follows random walk model, combinations of LOF and IF gives

Figure 6. Performance of single stage and sequential ensemble algorithms in detecting event rain: Single month

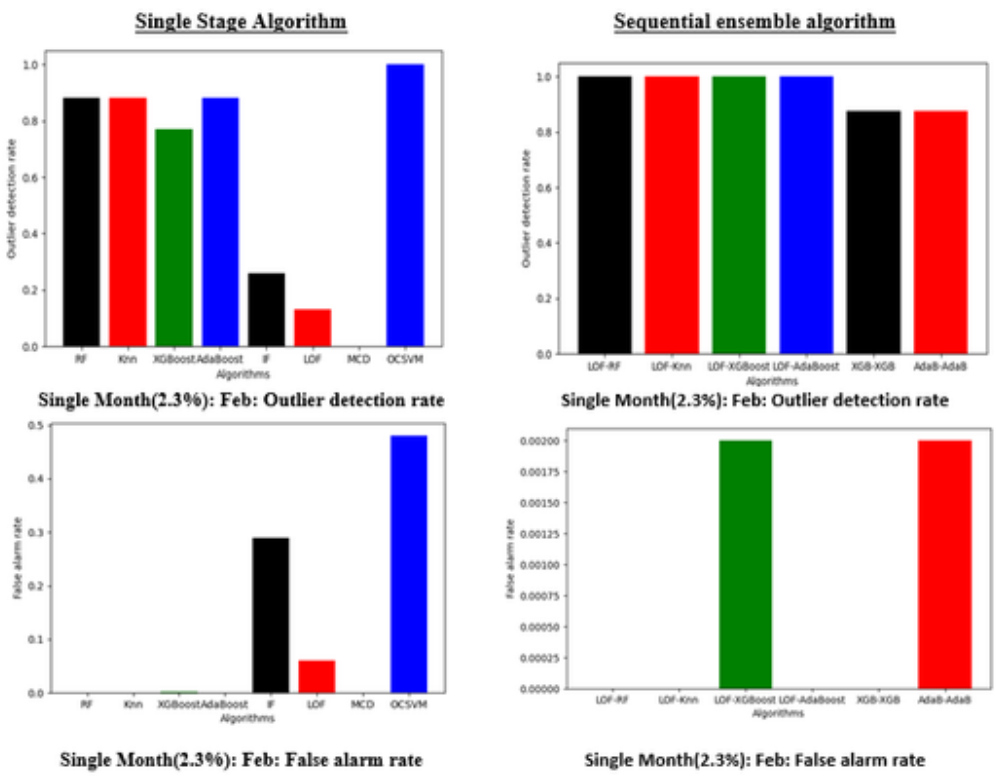
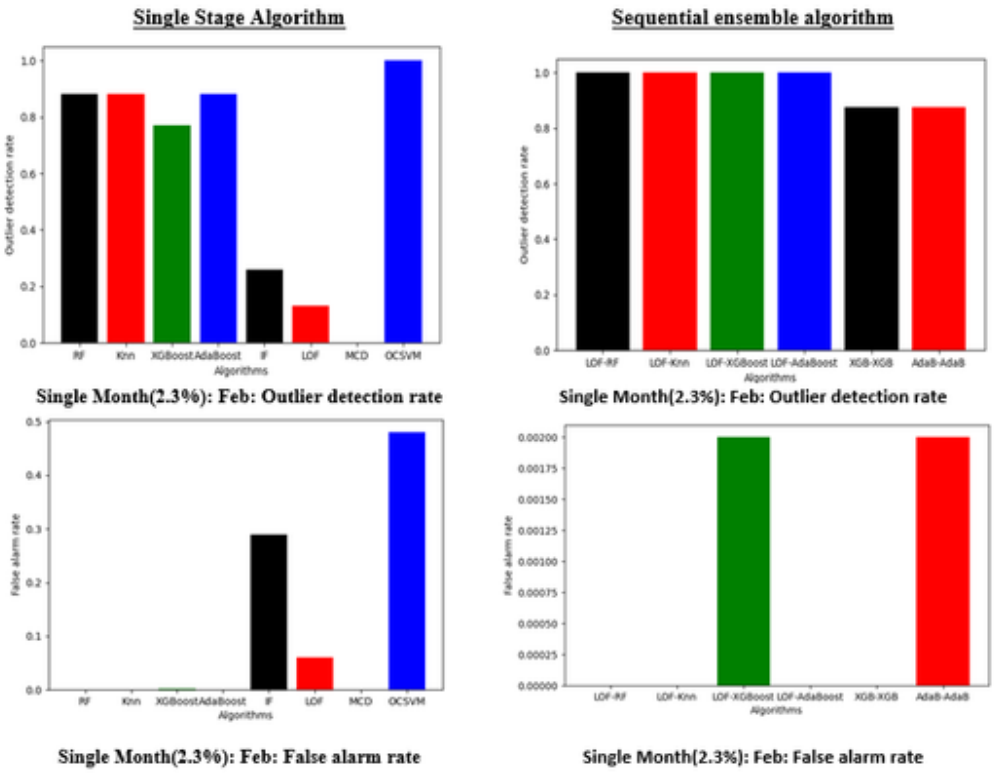


Figure 7. Performance of single stage and sequential ensemble algorithms in detecting event rain: Winter Season



best performance. While performing analysis on a single month data, sequential combinations of LOF algorithms expressed higher detection rate with lower false alarm rates. Analysis on the winter season also reveals that LOF or IF algorithms combined with supervised algorithms perform better than their independent counterparts. This proves the validity of the proposed algorithm.

DISCUSSION

Carcillo et al. (2019) proposed an outlier ensemble approach that combines supervised and unsupervised learning techniques to perform credit card fraud detection. Inspired from this approach and using the theoretical framework of sequential outlier ensemble proposed by Agarwal and Sathe (2017), we created a workbench that combines supervised and unsupervised outlier detection techniques whose efficiency is based on the underlying data and anomaly statistics.

Rules established by pioneers (Steinbuss and Böhm, 2020; Emmott et al., 2015) have been successfully used by us for the generation of dataset. Two datasets that follow Gaussian distribution and random walk model with various types and percentages of anomalies are used in this study. Using this workbench, the performance characteristics of the ensemble with collected data type with collected outliers have been tested and ensemble performance is characterised. Our methodology helps users to select an appropriate ensemble based on the data and anomaly statistics involved in the application dataset. This can be done by comparing the input of the workbench experiments with selected application data. In the current analysis, certain sequential combinations have better performance in a particular distribution than the rest. For example, MCD-MCD, IF-MCD,IF-IF combinations show better performance when the underlying distribution is Gaussian and LOF combinations produce

better results if the data follows a random walk model. Thus, output of the workbench helps the data scientist to select a sequential outlier ensemble algorithm for their specific application. Users of the workbench does not need to perform any further analysis for the selection of a specific outlier ensemble algorithm for their application, instead they can directly use the results of workbench.

In order to demonstrate the recommended approach for a given application, a case analysis is conducted which gives a correct guideline for users on how to use the workbench results for their specific application. A flow chart describing the various steps to be followed is given in figure 5. By using this flowchart two studies are conducted in realistic data and two studies are conducted in a real time dataset. First example illustrates working on data that follows Gaussian distribution. In this case, the same outlier ensemble algorithms as the workbench recommendations show higher efficiencies. Second study ensures that case analysis for random walk models also gives similar results as obtained from workbench. These two, case analysis aids the user to utilize the proposed system for a specific application. These studies successfully demonstrated the use of recommended workbench design with Yahoo dataset which is a realistic dataset. Second case study is conducted on a real time dataset. In this case, data follows a random walk model. Similar to realistic data, in real time dataset also, case study analysis gives the same results as those obtained from the workbench recommendation report.

This research work agrees with Carcillo et al. (2019) yet, application of this research more general. Hence, this study acts as a recommendation system which helps a user to select a suitable outlier ensemble algorithm for their application irrespective of the application domain.

CONCLUSION

Outlier ensembles are developed by researchers for specific applications such as fraud detection which intend to improve the efficiency of the anomaly detection process. However, a general framework which is applicable over multiple domains is not available. Hence, this research work presents a workbench design for identifying better ensemble-based anomaly detection algorithms for a given application. Sequential outlier ensemble algorithm is utilised in this research work for achieving this objective. Results obtained from this study can be used in multiple applications by using the knowledge of underlying data and anomalies. The recommendation has been successfully demonstrated using one realistic dataset and one real time dataset. The proposed system makes a significant contribution in the area of ensemble-based outlier detection by creating a generalised workbench design which can be used in multiple application domains. Results obtained from the report in the workbench act as guideline for the data scientist to select the best ensemble algorithm for their application without conducting any further analysis. Contributions explained in Table 1 makes the proposed system outstanding from the set of existing anomaly detection algorithms.

Workbench designed in this analysis employs two data types for regular instances, Gaussian distribution and random walk model. This can be extended to more data types which help the user to apply this framework for more applications. Behaviour of anomaly detection algorithms in the presence of 2-7% of outliers is analysed in this paper. However, more analysis can be conducted using different anomaly percentages to get a better understanding of algorithms at various anomaly statistics.

CONFLICT OF INTEREST

The authors of this publication declare there is no conflict of interest

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Aggarwal, C. C., & Sathe, S. (2015). Theoretical Foundations and Algorithms for Outlier Ensembles. *SIGKDD Explorations*, *17*(1), 24–47. doi:10.1145/2830544.2830549

Aggarwal, C., & Sathe, S. (2017). *Outlier Ensembles*. Springer., doi:10.1007/978-3-319-54765-7

Angelo, P., Resende, A., & Drummond, A. C. (2018). A Survey of Random Forest Based Methods for Intrusion Detection Systems. *ACM Computing Surveys*, *51*(3), 48. doi:10.1145/3178582

Blest, D. (2002). Theory & Methods: Rank Correlation — an Alternative Measure. *Australian & New Zealand Journal of Statistics*, *42*(1), 101–111. doi:10.1111/1467-842X.00110

Callens, A., Morichon, D., Abadie, S., Delpey, M., & Liquet, B. (2020). Using Random Forest and Gradient boosting trees to improve wave forecast at a specific location. *Applied Ocean Research*, *104*, 102339. doi:10.1016/j.apor.2020.102339

Campos, G. O., Zimek, A., & Meira, W. (2018). *An Unsupervised Boosting Strategy for Outlier Detection Ensembles, Advances in Knowledge Discovery and Data Mining. PAKDD, 10937*. Springer., doi:10.1007/978-3-319-93034-3_45

Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2019). Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection. *Information Sciences*, *557*, 317–331. doi:10.1016/j.ins.2019.05.042

Chiang, A., David, E., Lee, Y.-J., Leshem, G., & Yeh, Y.-R. (2017). A study on anomaly detection ensembles. *Journal of Applied Logic*, *21*, 1–13. doi:10.1016/j.jal.2016.12.002

Domingues, R., Filippone, M., & Zouaoui, J. (2017). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, *74*, 406–421. doi:10.1016/j.patcog.2017.09.037

Domingues, R., Filippone, M., Michiardi, P., & Zouaoui, J. (2018). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, *74*, pp 406-421. 10.1016/j.patcog.2017.09.037

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*(2), 241–258. doi:10.1007/s11704-019-8208-z

Emmott, A. F., Das, S., Dietterich, T., Fern, A., & Wong, W. K. (2013). Systematic Construction of Anomaly Detection Benchmarks from Real Data. *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, 16-21. .10.1145/2500853.2500858

Emmott, A., Das, S., Dietterich, T. G., Fern, A., & Wong, W. (2015). A Meta-Analysis of the Anomaly Detection Problem. arXiv: Artificial Intelligence.

Ferreira, A. J., & Figueiredo, M. A. T. (2012). *Boosting Algorithms: A Review of Methods, Theory, and Applications. Ensemble Machine Learning*. Springer. doi:10.1007/978-1-4419-9326-7_2

Fujita, H., Matsukawa, T., & Suzuki, E. (2020). Detecting outliers with one-class selective transfer machine. *Knowledge and Information Systems*, *62*(5), 1781–1818. doi:10.1007/s10115-019-01407-5

Gao, J., & Tan, P.-N. (2006). Converting Output Scores from Outlier Detection Algorithms into Probability Estimates. *6th IEEE International Conference on Data Mining (ICDM 2006)*. 212-221. doi:10.1109/ICDM.2006.43

Ghosh, P., Bardhan, M., Chowdhury, N. R., & Phadikar, S. (2017). IDS Using Reinforcement Learning Automata for Preserving Security in Cloud Environment. [IJISMD]. *International Journal of Information System Modeling and Design*, *8*(4), 21–37. doi:10.4018/IJISMD.2017100102

Gokcesu, K., Neyshabouri, M. M., Gokcesu, H., & Kozat, S. S. (2019). Sequential Outlier Detection Based on Incremental Decision Trees. *IEEE Transactions on Signal Processing*, *67*(4), 993–1005. doi:10.1109/TSP.2018.2887406

Han, J. K. & Pei M. J. (2012), Data mining (3rd edn). Morgan Kaufmann, Waltham.

Hodge, V., & Austin, J. (2004). A Survey of Outlier Detection Methodologies (2004). *Artificial Intelligence Review*, *22*(2), 85–126. doi:10.1023/B:AIRE.0000045502.10941.a9

Hubert, M., Debruyne, M., & Rousseeuw, P. (2018). Minimum Covariance Determinant and Extensions. *Wiley Interdisciplinary Reviews: Computational Statistics*, *10*(3), e1421. doi:10.1002/wics.1421

Kiani, R., Keshavarzi, A., & Bohlouli, M. (2020). Detection of Thin Boundaries between Different Types of Anomalies in Outlier Detection Using Enhanced Neural Networks. *Applied Artificial Intelligence*, *34*(5), 345–377. doi:10.1080/08839514.2020.1722933

Kumar, R., & Vassilvitskii, S. (2010), Generalized distances between rankings. In *Proceedings of the 19th International Conference on World Wide Web*, ACM, 571–580. doi:10.1145/1772690.1772749

Lazarevic, A. & Kumar, V. (2015), Feature Bagging for Outlier Detection, *ACM KDD Conference, ACM SIGKDD Explorations*, *17*(1), 157-166.

Lazarevic, A., & Kumar, V. (2005). Feature bagging for outlier detection. *In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* [ACM.]. *KDD: Proceedings / International Conference on Knowledge Discovery & Data Mining. International Conference on Knowledge Discovery & Data Mining*, *05*, 157–166.

Li, L., Yan, J., Wang, H., & Jin, Y. (2020), Anomaly Detection of Time Series with Smoothness-Inducing Sequential Variational Auto-Encoder. IEEE Transactions on Neural Networks and Learning Systems, doi:10.1109/TNNLS.2020.2980749

Liao, Y., & Vemuri, R. (2002). Use of K-Nearest Neighbor classifier for intrusion detection. *Computers & Security*, *21*(5), 439–448. doi:10.1016/S0167-4048(02)00514-X

Liu, B., Xiao, Y., Yu, P. S., Hao, Z., & Cao, L. (2014). An Efficient Approach for Outlier Detection with Imperfect Data Labels, in IEEE Transactions on Knowledge and Data Engineering, 26(7), 1602-1616. doi:10.1109/TKDE.2013.108

Liu, T., Ting, K. M., & Zhou, Z. Isolation Forest (2008), *Eighth IEEE International Conference on Data Mining*, Pisa, 413-422, doi:10.1109/ICDM.2008.17

Mohotti, W. A., & Nayak, R. (2020). Efficient Outlier Detection in Text Corpus Using Rare Frequency and Ranking. *ACM Transactions on Knowledge Discovery from Data*, *14*(6), 7. Advance online publication. doi:10.1145/3399712

Muller, E. Schiffer, M., & Seidl, T. (2011), Statistical selection of relevant ¨subspace projections for outlier ranking. *In IEEE 27th International Conference on Data Engineering,* 434–445.

Pasillas-Díaz, J. & Ratté, S. (2016). Bagged Subspaces for Unsupervised Outlier Detection: FBSO. *Computational Intelligence,* 33.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Raman, M. R. G., Somu, N., Kirthivasan, K., Liscano, R., & Sriram, V. S. S. (2017). An Efficient Intrusion Detection System based on Hypergraph- Genetic Algorithm for Parameter Optimization and Feature Selection in Support Vector Machine. *Knowledge-Based Systems*, *134*, 1–12. doi:10.1016/j.knosys.2017.07.005

Rayana, S., Akoglu, L (2016), Less is more: building selective anomaly ensembles. *ACM TKDD 10*(4), 42:1–42:33.

Rayana, S., Zhong, W., & Akoglu, L. (2016): Sequential ensemble learning for outlier detection: a bias-variance perspective. In *Proceedings of ICDM*, 1167–1172. doi:10.1109/ICDM.2016.0154

Reunanen, N., Räty, T., & Lintonen, T. (2020). Automatic optimization of outlier detection ensembles using a limited number of outlier examples. *International Journal of Data Science and Analytics*, *10*(4), 377–394. doi:10.1007/s41060-020-00222-4

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery*, *8*(4), e1249. doi:10.1002/widm.1249

Schubert, E., Wojdanowski, R., Zimek, A., & Kriegel, H.-P. (2012) On evaluation of outlier rankings and outlier scores. *Proceedings of the Twelfth SIAM International Conference on Data Mining,* SIAM / Omnipress, 1047–1058.

Sharma, M., & Gupta, A. K. (2021). An Algorithm for Target Detection, Identification, Tracking and Estimation of Motion for Passive Homing Missile Autopilot Guidance. Mobile Radio Communications and 5G Networks. Lecture Notes in Networks and Systems, 140. Springer. doi:10.1007/978-981-15-7130-5_5

Sinha, V. K., Gupta, A. K., & Mahajan, M. (2018). Detecting fake iris in iris bio-metric system. *Digital Investigation*, *25*, 97–104. doi:10.1016/j.diin.2018.03.002

Smolyakov, D., Sviridenko, N., Ishimtsev, V., Burikov, E., & Burnaev, E. (2019). Learning Ensembles of Anomaly Detectors on Synthetic Data. Advances in Neural Networks. Lecture Notes in Computer Science, 11555. Springer. doi:10.1007/978-3-030-22808-8_30

Steinbuss, G., & Böhm, K. (2021). Benchmarking Unsupervised Outlier Detection with Realistic Synthetic Data. *ACM Transactions on Knowledge Discovery from Data*, *15*(4), 65. Advance online publication. doi:10.1145/3441453

Tang, B., & He, H. (2017). A local density-based approach for outlier detection. *Neurocomputing*, *241*, 171–180. doi:10.1016/j.neucom.2017.02.039

Van Nguyen, T. T. Nguyen & Q. U. Nguyen, Sequential ensemble method for unsupervised anomaly detection, *International Conference on Knowledge and Systems Engineering (KSE)*, Hue, 71-76. doi:10.1109/KSE.2017.8119437

Veeramachaneni, K., Arnaldo, I., Korrapati, V., Bassias, C., & Li, K. (2016). Training a big data machine to defend. *IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, IEEE, 49–54. doi:10.1109/BigDataSecurity-HPSC-IDS.2016.79

Vinayakumar, R., Soman, K., & Poornachandran, P. (2017). Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). [IJISMD]. *International Journal of Information System Modeling and Design*, *8*(3), 43–63. doi:10.4018/IJISMD.2017070103

Wang, B., & Mao, Z. (2019). Outlier detection based on a dynamic ensemble model: Applied to process monitoring. *Information Fusion*, *51*, 244–258. doi:10.1016/j.inffus.2019.02.006

Wang, B., & Mao, Z. (2020). A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule. *Information Fusion*, *63*, 30–40. doi:10.1016/j.inffus.2020.05.001

Wang, B., & Mao, Z. (2020). Detecting outliers in industrial systems using a hybrid ensemble scheme'. *Neural Computing & Applications*, *32*(12), 8047–8063. doi:10.1007/s00521-019-04307-5

Wang, Z., Yuan, G., Pei, H., Zhang, Y., & Liu, X. (2020). Unsupervised learning trajectory anomaly detection algorithm based on deep representation. *International Journal of Distributed Sensor Networks*, *16*(12). doi:10.1177/1550147720971504

Xing, H. J. & Liu, W. T. (2019), Robust AdaBoost based ensemble of one-class support vector machines. *Information Fusion*. 10.1016/j.inffus.2019.08.002

Xu, K., Xia, M., Mu, X., Wang, Y., & Cao, N. (2019). Ensemble Lens: Ensemble-based Visual Exploration of Anomaly Detection Algorithms with Multidimensional Data. IEEE Transactions on Visualization and Computer Graphics, 25(1) 109-119, doi:10.1109/TVCG.2018.2864825

Yahaya, S. W., Lotfi, A., & Mahmud, M. (2019). A Consensus Novelty Detection Ensemble Approach for Anomaly Detection in Activities of Daily Living. *Applied Soft Computing*, *83*, 105613. doi:10.1016/j.asoc.2019.105613

Zhang, J., Li, Z., Nai, K., Gu, Yu., & Sallam, A. (2019). DELR: A double-level ensemble learning method for unsupervised anomaly detection. *Knowledge-Based Systems*, *181*, 104783. doi:10.1016/j.knosys.2019.05.026

Zhang, W., Yang, D., Zhang, S., & Rosas, A. (2020). A novel multi-stage ensemble model with enhanced outlier adaptation for credit scoring. *Expert Systems with Applications*, *165*, 113872. doi:10.1016/j.eswa.2020.113872

Zhao, Y. & Hryniewicki & Maciej. (2018). DCSO: Dynamic Combination of Detector Scores for Outlier Ensembles. Zhao, Yue & Hryniewicki, Maciej. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining* ACM, doi:10.13140/RG.2.2.11165.77288

Zhao, Z., Mehrotra, K. G., & Mohan, C. K. (2015). Ensemble Algorithms for Unsupervised Anomaly Detection. In M. Ali, Y. Kwon, C. H. Lee, J. Kim, & Y. Kim (Eds.), Lecture Notes in Computer Science: Vol. 9101. *Current Approaches in Applied Artificial Intelligence. IEA/AIE 2015*. Springer. doi:10.1007/978-3-319-19066-2_50

Zimek, A., & Filzmoser, P. (2018). There and Back Again: Outlier Detection between Statistical Reasoning and Data Mining Algorithms. *Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery*, *8*(6), e1280. doi:10.1002/widm.1280

Zimek, A., Campello, R. J. G. B., & Sander, J. (2014). Ensembles for unsupervised outlier detection: Challenges and research questions. *SIGKDD Explorations*, *15*(1), 11–22. doi:10.1145/2594473.2594476

Zimek, A., Gaudet, M., Campello, R. J. G. B., & Sander, J. (2013), Subsampling for efficient and effective unsupervised outlier detection ensembles. *In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 428–436.

Divya D is a Research Scholar in the Division of IT, School of Engineering at Cochin University of Science and Technology, Kerala, India. She earned her undergraduate and post graduate degree from Mahatma Gandhi University, Kerala, India. Her research areas include Data mining, Anomaly detection, Machine learning and Artificial Neural Networks.

Dr. M. Bhasi holds a Ph.D. and Masters in Industrial Engineering and Management from the Indian Institute of Technology, Kharagpur. He is currently Senior Professor at the School of Management Studies, Cochin University of Science and Technology where he supervises students in Faculties of Management, Technology and Engineering. He has been the Dean and Director of the School and has also been Dean of Management Studies at two other Universities in Kerala. He serves as Independent Director in Public sector Organisations. He is also currently the Director of the Inter-University Centre for IPR studies in CUSAT. His areas of research and teaching interest are, Information Systems and Innovation & Entrepreneurship, Supply Chain and Operations Management.

Dr. Santosh Kumar M.B is an Associate Professor in the Division of IT, School of Engineering at Cochin University of Science and Technology, Kerala,India where he supervises students in Faculties of Engineering. His areas of interest includes Agricultural data mining and machine learning