An Experimental Analysis of Fraud Detection Methods in Enterprise Telecommunication Data using Unsupervised Outlier Ensembles

Georgios Kaiafas*, Christian Hammerschmidt*, Radu State*,
Cu D Nguyen†, Thorsten Ries†, Mohamed Ourdane†
*SnT, University of Luxembourg
firstname.lastname@uni.lu
†POST Luxembourg
firstname.lastname@post.lu

***Abstract*—This work uses outlier ensembles to detect fraudulent calls in telephone communication logs made on the network of POST Luxembourg. Outlier detection on high-dimensional data is challenging and developing an approach which is robust enough is of paramount importance to automatically identify unexpected events. For use in real-world business applications it is important to obtain a robust detection method, i.e. a method that can perform well on different types of data, to ensure that the method will not impact that business in unexpected ways. Many factors affect the robustness of an outlier detection approach and this experimental analysis exposes these factors in the context of outlier ensembles using feature bagging. Real-world problems demand knowledge about possible candidate approaches that address the problem, and decide for the best performing method using a train-test split of labeled data. In the unsupervised setup the knowledge about performance is missing during the learning phase thus is difficult to decide during that phase. Hence, in this setup it is important to know about how the performance is affected before the learning phase. Hence, this analysis demonstrates that despite the collective power of outlier ensembles they are still affected by i) data normalization schemes, ii) combination functions iii) outlier detection algorithms.**

I. Introduction

Outlier detection is the process of identifying those observations that deviate substantially from the remaining data. In particular, identifying outliers in high-dimensional data can provide important insights into many real-world applications, e.g., detection of frauds, sensor failures, or outlying gene expressions. Choosing an unsupervised method such as outlier detection over supervised or semi-supervised approaches is influenced by the availability of labels [4].

In unsupervised approaches, every algorithm is based on a model making specific assumptions on the nature of outliers. Hence, every model is specialized for different characteristics of observations and fits only to some aspects of the total ground truth. Outlier ensembles overcome the subjectivity of each model by integrating various different outlier detection results to build more robust detectors. The authors in [20] point out the challenges related to developing outlier ensembles. The two major challenges are:
i) How to deal with accuracy? Since there is nothing known about the accuracy of outlier detectors during the learning phase, evaluation is difficult.
ii) How to assess diversity? Diversity is the most important ingredient for ensemble construction; homogeneous methods cannot benefit enough from parameters optimization ensembling approach [20].

Telecommunication frauds are malicious usage and/or exploitation of telephone connections for criminal purposes. These can range from finance gain for fraudsters to damaging public reputation of enterprises. More often than not, they cause substantial financial loses for victims. Here, we focus on unauthorized calls to international premium-rate numbers. These calls can cause large costs within short time frames. Quick and reliable detection and mitigation of fraudulent calls is therefore extremely important.

Here, we deal with the detection of fraudulent private phone exchange (PBX) phone calls made on the network of the largest provider in Luxembourg, POST Luxembourg. We identify fraudulent calls by developing outlier ensembles on high-dimensional call record data. Compared to the supervised learning setup where the ground truth is used to tune classifiers or regressors, the ground truth is not available in the unsupervised setup to tune the outlier detection algorithm. In the following we present the core elements used for the construction of an outlier ensemble based on the literature of outlier ensembles [1], [20]:

- **Data normalization** as a preprocessing step to scale each attribute to [0,1] or N(0,1). The authors in [2] show that outlier detection techniques on normalized datasets perform better compared to unnormalized.
- **Subspace outlier detection** to avoid irrelevant attributes and learn diverse models
- **Normalization of outlier scores** to make comparable the scores from heterogeneous outlier detection algorithms
- **Combination functions** for the selected outlier scores.

At each step it is challenging to select the best performing method without leveraging the ground truth. Throughout this

978-3-903176-15-7

work, we will focus on the unsupervised scenario, with no further information about the ground truth.

In our analysis we are motivated to experimentally investigate the effect of developing outlier ensembles with competitive methods for each of the aforementioned core elements. Hence, we utilize four **normalization** schemes, we apply feature bagging [7] as the **subspace outlier detection** technique, we **normalize the outlier scores** by employing the Z-score method and finally, we use average and maximum to **combine** outlier score vectors.

The analysis conducted here helps us identifying selection criteria for robust ensemble unsupervised methods. Ultimately, this analysis is an important and mandatory step towards the automation of hybrid supervised learning approaches guided by outlier ensembles without the involvement of domain experts in the modelling phase.

In the remainder of this paper, we will first have a look at the research area of unsupervised outlier ensembles in Section II. Afterwards, in Section III, we will describe the PBX data sample produced on POST's network. The methodology followed in this work and the comparative evaluation of different normalization schemes, algorithms and combining functions will be presented in Section IV & V respectively. Finally, we close with conclusions and future improvements in Section VI.

II. Preliminaries & Related Work

Outlier detection algorithms rely on subjective assumptions of the underlying generative process to model the nature of outliers. This subjectivity affects the obtained outlier results; some parts of the data may be modeled well, whereas other parts of the data may not. Furthermore, outlier detection algorithms may sometimes be effective on a given data set, but may not be effective on other data sets. We call this effect data-centric robustness. Also, they may be extremely sensitive to the choice of parameters, an effect called parameter-centric robustness. Without robustness, it is difficult for a company to trust a method to reliably solve a business problem with varying data. Individual models fit only to some aspects of the whole truth [1]. Hence, regarding these dependencies ensemble analysis addresses the above limitations of individual outlier detection algorithms and increases the robustness of the data mining process.

High-dimensional data pose special challenges to outlier detection algorithms; irrelevant attributes are concealing relevant information. The authors in [22] highlight that in high-dimensional spaces only subsets of relevant attributes provide the meaningful information; the residual attributes are irrelevant for the outlier detection task. Therefore, it is efficient to identify outliers from appropriate subspaces i.e. [5], [11], [12], [15], [18]

III. Dataset Description

This work uses a dataset which consists of telephone communication logs. Due to GDPR rules, aggregation by time windows (10 minutes) and anonymization is applied before the experimental analysis. Henceforth, D is our high-dimensional dataset which includes information related to calling number, calling time, calling duration, number of called parties, total calling cost and destination countries of the called parties. Such data are mainly used for billing purposes, but we leverage the dataset for fraud detection and mitigation. Table I presents the fields and notation of D.

TABLE I
Notation

Field	Notation
Average number of distinct calls previous 3h	AvgDc
Average calling times previous 3h	AvgCount
Average cost previous 3h	AvgCost
Number of distinct calls	Dc
Destination countries	Countries
Calling number	ANumber
Number of calls	Count
Call duration	Duration
Call cost	Cost
Time	Time

Our dataset D consists of 64000 data points with the 10 columns listed in Table I. The dataset has been manually labeled by experts from the provider. Fraudulent calls have been confirmed as fraudulent. Non-fraudulent calls are not necessarily non-fraudulent, but may contain previously unnoticed types of fraudulent calls. All call activities were made across a time span of 1 month. There are 40930 unique calling numbers (ANumber) in D. Only 0.04% of them have at least 1 fraud calling activity. In the rest of this paper we will refer to a *calling number* which has at least 1 fraud calling activity as an *fraud calling number*. Conversely, we will refer to a *calling number* which does not have any fraud calling activity as an *normal calling number*. Furthermore, one fraudulent number can have multiple fraudulent calls. The overall percentage of fraud calling activities is 0.57% which makes D significantly imbalanced. In outlier detection literature the outlierness percentage of the most used datasets [14] varies significantly between 0.03% and 32% whereas the number of data points varies between 129 and 567479 data points.

IV. Methodology

In this section we are giving details about the methodological part of our analysis and we are guided by a primary question: "What is the effect of preprocessing steps on the performance of the developed ensemble variants?"

A. Feature Engineering

In the unsupervised setup we do not leverage the ground truth to find the best performing set of features thus business knowledge is significantly important to extract composite features that will not be noisy. As a result, the set of those features described below, is the result of using POST's domain knowledge aiming to expose more informative patterns:

- Month, Day, Weekday, Hour
- Continent percentage: What fraction of *countries* called belong to Europe, Africa, Asia, etc.?

- What is the difference of continent percentage between 2 calling events from the same *ANumber*?
- Duration difference of two consecutive calls (Calculated by *ANumber* and *ANumber* & *Dc*)
- How much time passed between two calls? (Calculated by *ANumber*)
- How many times called distinct calling numbers
- Duration per distinct call
- Cost for each Continent
- Average cost per call
- Average cost per destination calling number
- Count of call for each Continent
- Call duration duration by Continent

The above composite features have been calculated ignoring the time variable. Hence, in order to reveal patterns related to time we calculated lag features which are described below. Those features are used when time series problems are transformed into machine learning problems.

- Rolling Average of *Duration* for the last 3 calls made by each *ANumber*
- Rolling Average of *Duration* for the last 3 calls made by each *ANumber* per hour of the day
- Rolling Average of *Cost* for the last 3 calls made by each *ANumber*
- Rolling Average of *Cost* for the last 3 calls made by each *ANumber* per hour of the day
- Rolling Average of *Dc* for the last 3 calls made by each *ANumber*
- Rolling Average of *Dc* for the last 3 calls made by each *ANumber* per hour of the day

In the rest of this paper, S is our high-dimensional dataset which is consisted of the attributes in D, plus all the above handcrafted features.

B. Outlier Ensembles

Developing an outlier ensemble approach is challenging when it comes to decide what will be the best performing methods or components. In this section we construct variants of bagging outlier ensembles which contain the following four components.

*1) **Data normalization**:* One of the main preprocessing steps for many statistical learning tasks is normalizing the data. The authors in [2] show that outlier detection methods on normalized datasets perform better compared to unnormalized datasets. We normalize all the numerical variables in S dataset based on the following schemes.

1) Minimum and maximum normalization (Min-Max)
   Each column x is transformed to $\frac{x-min(x)}{max(x)-min(x)}$ where $min(x)$ and $max(x)$ are the minimum and maximum values of x respectively
2) Mean and standard deviation normalization (Mean-SD)
   Each column x is transformed to $\frac{x-mean(x)}{sd(x}$ where $mean(x)$ and $sd(x)$ are the mean and standard deviation values of x respectively
3) Median and the IQR normalization (Median-IQR)
   Each column x is transformed to $\frac{x-median(x)}{IQR(x}$ where $median(x)$ and $IQR(x)$ are the median and the interquartile range of x respectively
4) Median and median absolute deviation normalization (Median-MAD)
   $MAD(x) = median(\mid x - median(x) \mid)$ and each column x is transformed to $\frac{x-median(x)}{MAD(x}$

As a result, we end up with four variants of the S dataset where S_1 is the S dataset normalized based on Min-Max formula, S_2 on Mean-SD, S_3 on Median-IQR and S_4 on Median-IQR.

*2) **Subspace Outlier Detection**:* In this work, we employ feature bagging [7] to discover relevant subspaces. In feature bagging, an outlier detection algorithm is applied to various random lower dimensional projections, i.e. using only a subset of the available features. At each projected space outlier scores are produced. Then, all the outlier scores are combined to produce the final results. In the rest of this paper we will refer to an outlier detection algorithm as *detector* or *base detector*

*3) **Normalization of Outlier Scores**:* Different outlier detectors may often report outputs on different numeric scales. Therefore, before combining the outlier scores we have to apply normalization. Otherwise, some algorithms might dominate in the combination score. In addition, it is important to convert minimization scores to maximization ones and vice versa.

The authors in [1] suggest that using Z-scores turns out to be quite effective in many settings. When a detector produces smaller scores as indicators of greater outlierness the negative of the Z-value is used in the case. In this work, we also normalize the outlier scores by utilizing the Z-score scheme.

*4) **Combination functions**:* In this section, we are describing at which cases we apply combination functions to build the ensembles and which are those functions.

We apply combination functions to unify the outlier scores of different executions of one detector type obtained by using feature bagging and different parameter values. In addition to feature bagging, we employ the same detector with different parameter values at each lower dimension projection. Once we combine the scores of a detector, which are generated as we described previously, we end up with the desired outlier ensemble.

The combination functions that we are using are the *mean of scores* and the *maximum of scores*. The authors in [1] explain the benefits of using the mean and maximum as combinations functions.

C. Assessing Diversity

The goal of our strategy to select detectors is to obtain great diversity among the produced outlier scores from each detector. The benefit of diverse outlier scores that are generated by multiple detectors is the restriction of the space of where the true result most probably lies. Our work takes the findings of [16], [20] into consideration for selecting the detectors in order to generate scores that are dissimilar.

We increase the diversity by using different parameters of the same detector, i.e. different values of k for the neighborhood size. In addition, we select detectors that fall into families that learn dissimilar results. Finally, feature bagging [7] is an unstable technique which aids to produce very uncorrelated results and, thus, to improve ensembles [21]

D. Detectors

The strategy for selecting the detectors and perform our experimental analysis discussed on IV-C section. In principle, however, one could choose any detector to perform a similar analysis as long as the strategy is the same. Hence, the following algorithms will be the base detectors of each ensemble variant.

i) **Kernel Density** based detector, **KDEOS** [17], computes a kernel density estimation over a user-given range of k-nearest neighbors. The gaussian kernel is used for estimation.

ii) **Local Outlier Probabilities**, **LoOP** [6], detector computes a local density based on probabilistic set distance for observations, with one parameter the k-nearest neighbors. The density is compared to the density of the respective nearest neighbors, resulting in the local outlier probability.

ii) **iForest** [8] detects anomalies in a tree ensemble fashion. It isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. We skip further mathematical details of each detector due to lack of space.

E. The Pipeline of constructing Bagging Ensembles

In this section we are describing the steps followed to construct the outlier ensembles used for experimental analysis. First, we **normalize** data based on the discussed formulas to obtain four datasets S_i and we apply feature bagging as the **subspace outlier detection** technique. Then, we **normalize the outlier scores** by employing the Z-score method and finally, we use the average and maximum functions to **combine** outlier score vectors and construct the outlier ensembles.

Therefore, we construct eight outlier ensembles for each of the three detectors (KDEOS, LoOP, iForest). A detector is employed on each S_i and feature bagging is performed to discover subspaces and induce diversity. Then we unify outlier scores by using maximum as the combining function. As a result, we end up with four outlier ensembles; one on each S_i. In addition, the same steps are followed but average is the combining function.

In the rest of this work, we will refer to an outlier ensemble constructed on a S_i, having as detector one of the KDEOS, LoOP, iForest and unifying outlier scores with maximum or average combining function as *Bagging Ensembler*. Our experimental analysis is composed of 24 in total *Bagging Ensemblers*

V. Experimental Analysis

A. Setup

The experiments were performed using R and Python. We used R for the feature engineering part with the *data.table* package [3] and implemented KDEOS and LoOP detectors by using the *DDoutlier* package [9]. We used the iForest detector implemented in Python's scikit-learn library [13].

All four high-dimensional datasets S_i contain 64000 events and consist of 91 numerical attributes. Since the construction of each *Bagging Ensembler* is independent to the rest of the *Bagging Ensemblers* we developed them in parallel.

B. Results

In our analysis, the widely used area under the ROC curve *(AUC)* measure is used to evaluate the outlier detectors. Additionally, we use the precision measure based on our intuition of the outlierness percentage in the dataset. The first 400 events with the greatest outlier score are used as threshold to calculate the precision which is denoted as *P@400*.

In Figures 1– 4 we present the results of all the bagging executions of the detectors discussed in IV-D which are applied on each S_i dataset. These data sets are described in detail in Sect. IV-B1. More specifically, box plots of AUC performances for the executions of the detectors for all S_i are shown. Each box plot summarizes 100 different executions of a base detector created by different parameter values and random projections to lower dimensions.

In Figure 5 we present the results of all the possible *Bagging Ensemblers* that we defined in IV-E. To unify the outlier scores produced by the feature bagging technique, we use average and maximum as the combination functions.

LoOP detector performs slightly better than random guess and iForest shows the highest values of the AUC. iForest managed to detect all the *fraud calling numbers*.

In addition, Tables II, III show the standard deviation of AUC for *Bagging Ensemblers* with two different kinds of granularity. Table II shows that the AUC of KDEOS detector with the maximum combination function has the largest standard deviation. Also, the same detector with the average combination function has the second highest standard deviation. Overall, using either maximum or average combination function across all data normalization schemes, KDEOS has the largest deviation. Hence, KDEOS detector is the affected the most by data normalization approaches.

C. Discussion

One key benefit of outlier ensembles is their ability to take advantage of diversity between individual executions of base detectors in order to construct better detectors. Especially well-performed subspace techniques induce diversity in the resulting models and make the ensemble perform better. In our analysis KDEOS detector has no execution that performs better than 0.7 AUC as it is shown in figure 1. However, figure 5 presents that *Bagging Ensembler* with the average combination function produces AUC values higher than higher than 0.8.

Furthermore, constructing the *Bagging Ensembler* of LoOP detector has the least improvement compared to its individual executions. LoOP detector performs the best when it is applied on S_3 dataset (Median-MAD) and using the average combination function.

iForest is the only detector that is not affected at all by data normalization schemes and shows steadily AUC values close to 1.0. This detector is an ensemble by its nature compared to KDEOS and LoOP and that is the major reason why it is accurate and robust.

In Figure 6 we present the Precision (P@N) results of all the *Bagging Ensemblers* and the combination of all the *Bagging Ensemblers* using the average and maximum combination function. In addition, in the same figure it is shown that the combination of all the *Bagging Ensemblers* either with average or maximum function improves the performance of each individual *Bagging Ensemblar* except of iForest; the best performing algorithm. Furthermore, the average combination function outperforms the maximum combination function when all the *Bagging Ensemblers* at the three out of four S_i except S_1 Min-Max dataset.

Fig. 1. AUC Performance of KDEOS, LoOP and iForest on data normalized by Mean-SD while performing Feature Bagging

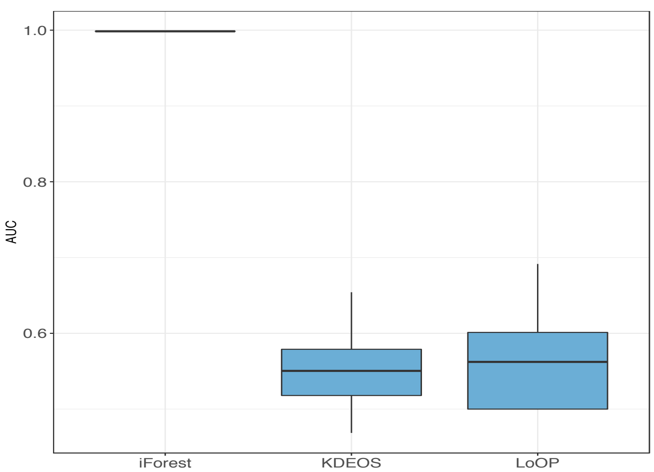

Fig. 2. AUC Performance of KDEOS, LoOP and iForest on data normalized by Min-Max while performing Feature Bagging

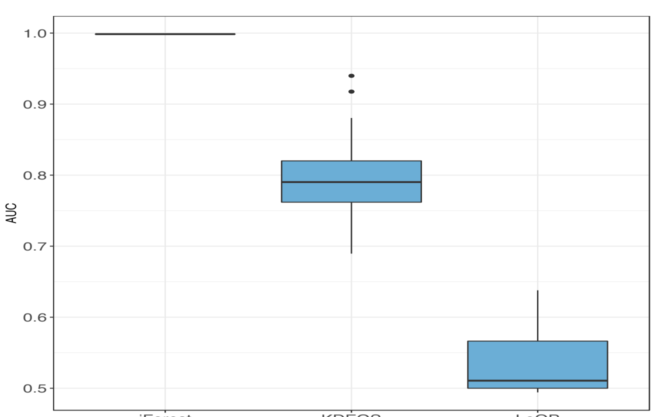

Fig. 3. AUC Performance of KDEOS, LoOP and iForest on data normalized by Median-MAD while performing Feature Bagging

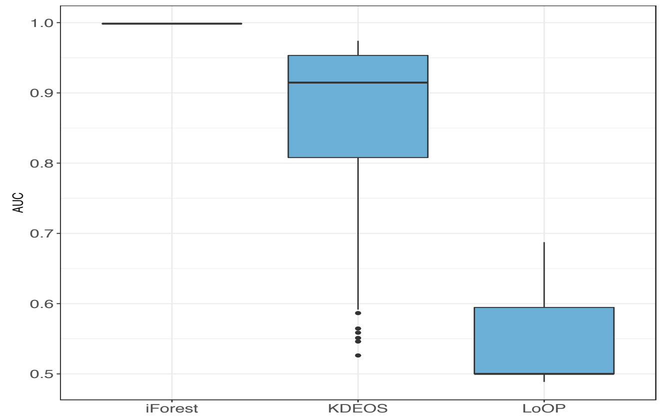

Fig. 4. AUC Performance of KDEOS, LoOP and iForest on data normalized by Median-IQR while performing Feature Bagging

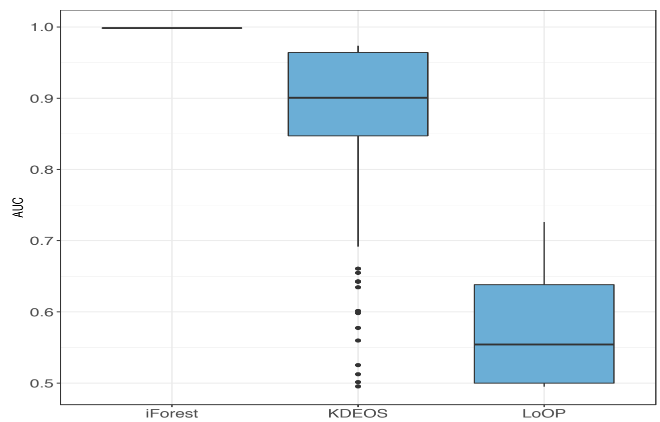

Fig. 5. AUC Performance of all the *Bagging Ensemblers*

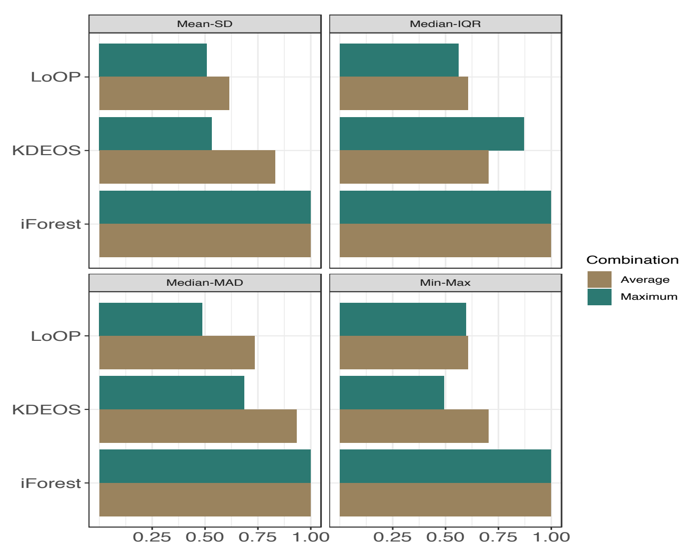

Fig. 6. Precision (P@400) of all the *Bagging Ensemblers* and their combination. On the left, the Maximum combination function is used for iForest, KDEOS, LoOP, and, the ultimate combination of all Bagging Ensemblers. On the right the Average combination function is used.

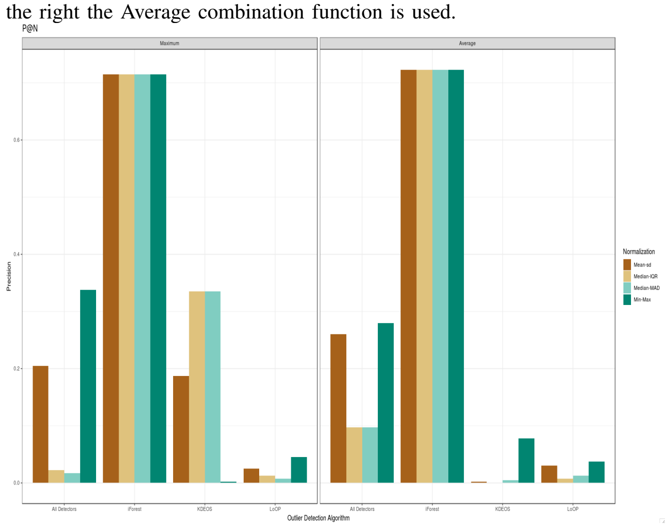

TABLE II
STANDARD DEVIATION OF AUC MEASURE ACROSS ALL S_i FOR EACH BAGGING ENSEMBLER

Detector	Combination Function	Std.
iForest	Maximum	0.0001
KDEOS	Maximum	**0.1718**
LoOP	Maximum	0.0494
iForest	Average	0.0001
KDEOS	Average	**0.1116**
LoOP	Average	0.0626

TABLE III
STANDARD DEVIATION OF BAGGING ENSEMBLERS ACROSS ALL S_i

Detector	Std.
iForest	0.0001
KDEOS	**0.1555**
LoOP	0.0755

VI. CONCLUSIONS AND FUTURE WORK

Constructing outlier ensembles on high-dimensional data is challenging and this paper highlights the difficulty in selecting the best core components of an outlier ensemble pipeline. Addressing a real-world problem with unsupervised techniques requires overcoming these challenges to obtain both robust and accurate predictions. Researchers often develop novel unsupervised methods in artificial environments using toy data sets and therefore do not need to analyze the sensitivity of their approach. In contrast, problems encountered by companies need to address the problem of results varying significantly in order to deploy a robust and reliable solution based on these methods.

Our future work aims to take advantage of the limited knowledge of fraud activities made on the network of POST Luxembourg to develop imbalanced supervised learning approaches guided by outlier detection algorithms. [10], [19]

ACKNOWLEDGMENT

The authors would like to thank POST Luxembourg and FNR (AFR-PPP Project ID 11824564).

REFERENCES

[1] C. C. Aggarwal and S. Sathe. *Outlier ensembles: An introduction*. Springer, 2017.
[2] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4):891–927, 2016.
[3] M. Dowle and A. Srinivasan. *data.table: Extension of 'data.frame'*, 2018. R package version 1.11.8.
[4] M. Goldstein and S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.
[5] F. Keller, E. Muller, and K. Bohm. Hics: high contrast subspaces for density-based outlier ranking. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1037–1048. IEEE, 2012.
[6] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Loop: local outlier probabilities. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1649–1652. ACM, 2009.
[7] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166. ACM, 2005.
[8] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.
[9] J. Madsen. Ddoutlier v0. 1.0.
[10] B. Micenková, B. McWilliams, and I. Assent. Learning representations for outlier detection on a budget. *arXiv preprint arXiv:1507.08104*, 2015.
[11] E. Muller, I. Assent, P. Iglesias, Y. Mulle, and K. Bohm. Outlier ranking via subspace analysis in multiple views of the data. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 529–538. IEEE, 2012.
[12] J. R. Pasillas-Díaz and S. Ratté. Bagged subspaces for unsupervised outlier detection. *Computational Intelligence*, 33(3):507–523, 2017.
[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
[14] S. Rayana. Odds library. *Stony Brook,-2016. NY: Stony Brook University, Department of Computer Science. URL: http://odds. cs. stonybrook. edu (, 2017)*, 2016.
[15] S. Sathe and C. C. Aggarwal. Subspace histograms for outlier detection in linear time. *Knowledge and Information Systems*, pages 1–25, 2018.
[16] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel. On evaluation of outlier rankings and outlier scores. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 1047–1058. SIAM, 2012.
[17] E. Schubert, A. Zimek, and H.-P. Kriegel. Generalized outlier detection with flexible kernel density estimates. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 542–550. SIAM, 2014.
[18] B. van Stein, M. van Leeuwen, and T. Bäck. Local subspace-based outlier detection using global neighbourhoods. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 1136–1142. IEEE, 2016.
[19] Y. Zhao and M. K. Hryniewicki. Xgbod: improving supervised outlier detection with unsupervised representation learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.
[20] A. Zimek, R. J. Campello, and J. Sander. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *Acm Sigkdd Explorations Newsletter*, 15(1):11–22, 2014.
[21] A. Zimek, M. Gaudet, R. J. Campello, and J. Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 428–436. ACM, 2013.
[22] A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.