Airport Restroom Cleanliness Prediction Using Real Time User Feedback Data

Kilian Ros
Department of Computer Science
University of Twente
Enschede, The Netherlands
kilianros@hotmail.com

Elena Mocanu
Department of Computer Science
University of Twente
Enschede, The Netherlands
e.mocanu@utwente.nl

Christin Seifert
Department of Computer Science
University of Twente
Enschede, The Netherlands
c.seifert@utwente.nl

***Abstract*—Large airports aim to offer a maximized experience to its passengers. A main contributor to customer experience is the cleanliness of restrooms, which is measured by feedback devices installed in restrooms at airports. This paper reviews to what extent real-time feedback data and classification techniques can be useful in practice to predict the cleanliness of restrooms. Within this topic, different class definitions of clean and unclean are introduced and a distinction is made between a combined prediction model that includes the entire environment and restroom-specific prediction models that focus only on a single restroom. The dataset is imbalanced and visualizations show that there is class overlap. To overcame these limitations various sampling methods with two different encoding mechanisms are investigated. Sampling methods do not improve the performance of the combined prediction model but do improve the performance of some of the restroom-specific prediction models, especially those with a high class imbalance. The major cause of the unsatisfying performance is not class imbalance, but the data ambiguity that leads to class overlap. To obtain prediction models that are useful in practice, we provide recommendations regarding the dataset and how this should be enriched with features that are capable of distinguishing the two classes more clearly.**

***Index Terms*—machine learning restroom cleanliness airport smiley boxes feedback**

I. Introduction

Being an important international airport, Amsterdam Airport Schiphol is always looking for ways to improve their services and offer a maximized airport experience to its passengers. One of the main contributors to the overall passenger satisfaction is the cleaning of restrooms located all over the airport. To monitor the user satisfaction of restrooms, most restrooms are equipped with so-called smiley boxes. These devices have three buttons which allow the user to rate the cleanliness of the restroom using a green, orange or red smiley, corresponding to good, average and bad respectively. To prevent undesirable user behaviour from affecting the data, the device does not register multiple votes that are cast right after each other.

Fig. 1. Smiley box device used to collect user feedback data.

With this technology in place, real time user feedback data is obtained and cleaning contractors are expected to utilize this data to improve on the current situation. The main objective that the airport has given them is to increase the overall percentage of green votes by a certain percentage.

In order to increase the overall percentage of green votes, the number of non-green votes, orange and red, has to be reduced. The assumption is that cleaning activities at the right moments will improve the cleanliness and prevent users from rating the cleanliness with a non-green vote.

Preliminary analysis of the data already identified two weak spots in the scheduling of cleaners. The first one is that the earliest shift of cleaners start their workday at six in the morning while there is an observed peak of bad votes between five and six. The second one is that all cleaners take their lunch break at the same time. During this lunch break, there is also an observed peak in the number of bad votes. These flaws can easily be exploited to increase the overall percentage of green votes. Minor changes to the way of organizing cleaners and cleaning tasks can quickly yield benefits with very little effort. Although manual analysis of the data can certainly contribute to the increase of user satisfaction, it is non-adaptive to the dynamic environment of the airport and it is a very tedious exercise.

Because of this, we aim for a more automated solution that is capable of anticipating to changes in the dynamic environment, through accurate prediction of non-green votes. This can contribute greatly to increasing the percentage of green votes because it allows cleaning contractors to prevent non-green votes by cleaning the restrooms at the right time. An accurate prediction model could be implemented to dynamically adjust current cleaning schedules and redirect cleaners to restrooms where cleanliness is most likely to become poor.

With the number of bad votes being a continuous number, a logical first step would be to approach this problem using regression techniques. Nevertheless, for application relevance the call to action, which redirects a cleaner, is more important than predicting the exact number of bad votes. Because of this, we decided to approach the problem as a binary classification problem. This means that the decision of when to redirect a cleaner depends on how the two classes, clean and unclean, are defined.

The purpose of this paper is to study the potential of real time feedback data and classification methods to develop a model that is useful in practice and contributes to the increase of user satisfaction at Amsterdam Airport Schiphol. To achieve this goal, the dataset is analyzed and several useful features are extracted. Multiple classification algorithms are applied to find the best solutions for different settings of the problem. The practical usefulness of these settings is then evaluated using the expertise of senior personnel.

The contribution of this paper is the exploration of using classification techniques in combination with a novel, real-world dataset that represents a very dynamic and subjective environment. This paper reviews to what extent real-time feedback data and classification techniques can be useful in practice to predict the cleanliness of restrooms. Within this topic, different class definitions of clean and unclean are studied and a distinction is made between a combined prediction model that includes the entire environment and restroom-specific prediction models that focus only on a single restroom.

The remainder of this paper is organized as follows: Section II discusses related work that is connected to this particular dataset. Section III analyses the dataset and Section IV outlines the research method. Section V presents the results. Section VI discusses limitations as well as recommendations, and finally, Section VII draws the conclusions from the results.

II. Related Work

The dataset used in this study appears to be quite novel in the research area of machine learning. To our best knowledge, there is no other work that uses real time user feedback data, or other subjective data generated by humans, to predict the cleanliness of rooms. Despite the fact that this kind of data is rather uncommon in literature, it does have characteristics that are widely studied in the field of machine learning, such as the imbalanced learning problem, class overlap and dimensionality reduction.

A. Real-time Customer Feedback Processing

The dataset used in this study is generated by smiley boxes located in restrooms. Restroom users press a red, orange or green smiley to express their satisfaction about the cleanliness of a restroom. Where our data is generated on a three-point scale, other customer satisfaction systems collect feedback in different ways or on different scales. The patent of Canora describes a feedback system that uses a five-point scale to measure customer satisfaction regarding a certain question [1]. Another patent of Bossemeyer and Connolly describes a feedback system where users can provide feedback using their voice [2]. Although this way of collecting feedback is qualitative instead of quantitative, they suggest a data mining tool to identify trends in the collected feedback.

B. Class Imbalance

Imbalanced datasets are very common in real-world domains and applications such as healthcareand creditcard fraud detection [3], [4]. Garcia and He [5] divide the problem into two categories: between-class imbalance and within-class imbalance. In a binary classification problem, between-class imbalance means that one class occurs more often than the other. Within-class imbalance is concerned with the distribution of representative data for subconcepts that exist within a certain class. Class imbalance can exist in different orders (e.g. imbalances of 1:100, 1:1.000 and 1:10.000) and the effect on learning performance can effectively be mitigated using several approaches such as sampling methods, cost-sensitive methods and learning methods designed specifically for imbalanced problems.

Sampling methods use data modification techniques to create a balanced class distribution. The most simple methods are probably random oversampling and undersampling, which copies minority samples and deletes majority samples at random in order to create an equal class balance. A more sophisticated undersampling method is called informed undersampling, to which for example EasyEnsemble [6] belongs. Another promising method is sampling with synthetic data generation. Two techniques that implement this are SMOTE [7], which combines minority oversampling and majority undersampling, and ADASYN [8], which uses synthetic sampling in an adaptive manner. SMOTE is also often used in combination with data cleaning techniques such as Tomek links and the edited nearest neighbor rule (ENN) [9]. The goal of these techniques is to remove class overlap that is introduced when sampling methods are applied. By removing some overlapping samples, clusters in the training data can be separated more clearly, which might lead to better defined rules and improved performance.

Studies have shown that a balanced dataset improves overall classification performance compared to the original imbalanced dataset [10]. Garcia and He [5] state that for most imbalanced datasets, applying sampling methods indeed improves classifier accuracy.

Where sampling methods try to obtain more balance between classes, cost-sensitive learning methods try to counteract the negative effects of class imbalance by assigning different misclassification costs, or weights, to the classes [11]. Basic implementations of cost-sensitive learning simply apply misclassification costs to the dataset as weights that can be initialized when constructing a model. More advanced implementations apply cost-minimizing techniques to ensemble methods that integrate standard learning algorithms to develop cost-sensitive classifiers. Although these methods can significantly improve the performance, they require that the costs of misclassification for the classes are known. Very often this is not the case and there is only an intuition that one class should be more expensive than the other class [12].

C. Class Overlap

Although many of the works mentioned above assume class imbalance to be the cause of performance loss, Prati et al. [13] notice that in some cases learning algorithms perform good on imbalanced datasets and therefore class imbalance cannot

directly be correlated to the loss of performance. Their work suggests that the problem is not directly caused by class imbalance, but is also related to the degree of overlapping among the classes. Class overlap occurs when two data samples are nearly or completely identical in terms of their features but belong to different classes. Figure 2 depicts a simple example.

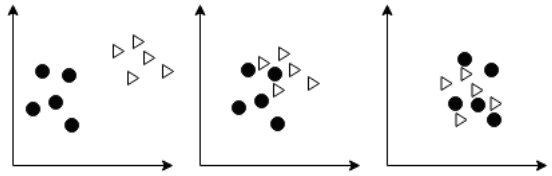

Fig. 2. Simple Example of Class Overlap Between Two Classes. Left: No Overlap, Middle: Minor Overlap, Right: Major Overlap

The dataset used in this paper contains both class imbalance as well as class overlap. A possible solution is provided by Batista et al. [9], who conclude that general oversampling and SMOTE-based methods are very effective when dealing with highly imbalanced and overlapping data. Results show that these methods are able to achieve similar performance compared to a naturally balanced distribution. Additionally, they state that the SMOTE technique with ENN data cleaning seems to be especially suitable when there is a high degree of class overlap. These suggested sampling methods will be included in the grid search and the performance of these methods is evaluated.

D. Dimensionality Reduction

Very often real-world datasets have a large number of features leading to a high dimensional data space that is hard to visualize. Without clear visualizations of data, a problem can be very hard to comprehend and eventually solve. A useful method to overcome this problem is dimensionality reduction. As stated in the dimensionality reduction techniques survey of Sorzano et al. [14], Principal Component Analysis (PCA) is probably the best known and most widely used technique.

According to Abdi and Williams [15], the main goal of PCA is to extract the important information from the data and express this as new features called principal components. These components are obtained as linear combinations of the original features. The first principal component is required to have the largest possible variance, and therefore explain most of the variance within the dataset. The second component is constraint to be orthogonal to the first one and should also have the largest possible variance without violating the constraint. Other components are computed likewise.

III. Data Analysis

A. Data Description

The dataset used in this study contains ten weeks of real time user feedback data in the period ranging from Monday march 11th till Sunday may 19th 2019. The airport consists of many areas with restrooms such as boulevards, lounges, baggage reclaim halls and piers. At every pier, there are multiple gates that are being used for arrivals and departures of flights.

Let $\mathcal{R}$ denote the set of restrooms, such that $R_i \in \mathcal{R}, \forall i \in \mathbb{N}$ representing the index of the restrooms analyzed. The total received votes per restroom, V_i, is a sum over all green (G), yellow (Y), and red (R) votes in a specific interval of time Δt. So far, measuring the perceived cleanliness in service environments [16] is not standardized yet. Therein, based on the prediction capabilities of votes, the objective is to increase the overall percentage of green votes. Thus, in order to treat the problem as a binary classification problem, we work under two assumption:

A 1: For all $R_i \in \mathcal{R}, \forall i \in \mathbb{N}$ in a specific time interval, the green votes refer to a clean restroom.

A 2: For all $R_i \in \mathcal{R}, \forall i \in \mathbb{N}$ in a specific time interval, the non-green votes refer to an unclean restroom, such that a bad vote $B_i = Y_i + R_i$.

B. Data Acquisition

All the restrooms are equipped with smiley boxes to collect data. The restrooms consist of multiple toilets and in male restrooms also urinals

Together, these restrooms received a total of 88,517 votes, of which approximately 65% are green, during the specified time period of ten weeks. This leads with an average of 37 votes per day.

When looking at Fig. 3, we observe a steady increase in the number of votes until week 16 and then a decrease until week 20. This is probably caused by the increase in passenger volume of almost five hundred thousand (8%) comparing March (weeks 11, 12 and 13) and April (weeks 14, 15, 16 and 17). We also note that week 18 is a holiday week in the Netherlands, but this week shows no notable differences compared to other weeks.

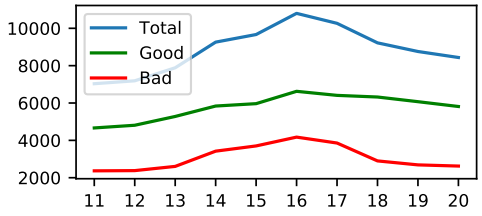

Fig. 3. Weekly number of votes recorded with the smiley-boxes.

C. Data Preparation

The continuous values of votes are discretized, using a time step of 30 minutes. This results in a dataset that includes 34 toilets monitored over 70 days with 30 minutes resolution, adding up to 114,240 datapoints. Each datapoint is the response from a smiley box, aggregated over a 30 minute time interval. A datapoint consists of the number of green, orange and red votes and all additional features as listed in I. We

split this dataset into 80% for training the models, 10% for validation and hyperparameter optimization to select the best model and 10% for testing the selected model. The splitting is done in chronological order, so week 19 is used for validation and week 20 is used for testing. Fig. 3 shows us that the validation and test weeks show no significant differences in the number of votes, which is good because otherwise, it might have a substantial influence on the results.

D. Feature Generation

In order to increase the learning capabilities of the methods, we augment the database using a sliding time window. Furthermore, in order to distinguish between restrooms, two different encoders were used to encode the restroom number: Rank-based encoding and one-hot encoding. The rank of a restroom is based on the number of bad votes in the training set, which is the first eight weeks. The restroom which has received the highest number of bad votes is ranked 33 and the restroom with the lowest number of bad votes is ranked 0. One-hot encoding creates a binary feature for all restroom numbers and sets all values to 0 except for the corresponding restroom number, which is set to 1. Next to the restroom encoding, other restroom related features that are included are the surface of a restroom, the number of toilets in a restroom and the gender of a restroom. Table I lists all the features. During preliminary analysis, other features have been created and tested, such as flight details and manually obtained cleaning times of restrooms. None of these however did improve the results

TABLE I
FEATURE NAMES, DESCRIPTIONS AND CATEGORIES

#	Name	Description	Category
		Target Feature	
1	B_t	Number of bad votes at t	Numerical
		Number of Bad Votes	
2	B_{t-1}	One time interval earlier	Numerical
3	B_{t-2}	Two time intervals earlier	Numerical
4	B_{t-3}	Three time intervals earlier	Numerical
5	B_{d-1}	One day earlier	Numerical
		Number of Votes	
6	V_{t-1}	One time interval earlier	Numerical
7	V_{t-2}	Two time intervals earlier	Numerical
8	V_{t-3}	Three time intervals earlier	Numerical
		Time Related Features	
9	$d[\#]$	Day, $d \in \{0, 6\}$	Ordinal
10	$I[\#]$	Time interval, $I \in \{0, 47\}$	Ordinal
		Restroom Related Features	
11	Surface	Surface of the entire restroom	Numerical
12	Toilet [#]	Number of toilets in restroom	Numerical
13	Gender	Male or female restroom	Categorical
14	Rank	Rank of the restroom	Ordinal

Figure 4 shows the Pearson correlation heatmap of all available features. To construct this heatmap, a time window

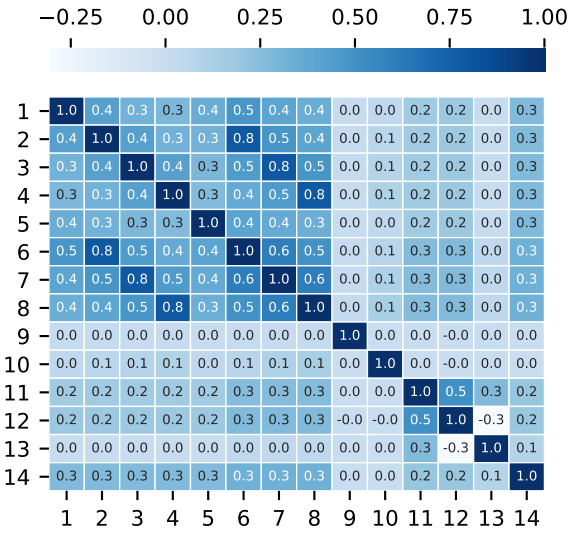

Fig. 4. Pearson correlation heatmap using rank-based restroom encoding with time window size 3. Feature numbers correspond to Table I.

of 3 was chosen and the restrooms are distinguished using the rank-based method. Most important is the first row which shows the correlation coefficients between the number of bad votes and all of the above-mentioned features. We observe the strongest correlations with the number of votes received in previous time intervals, followed by the number of bad votes received in previous time intervals. For both the number of votes and the number of bad votes received in previous time intervals we see that the correlation coefficient decreases as the time difference increases. It stands out that B_{d-1} has a stronger correlation than B_{t-2} with B_t. This suggests that looking at the previous day would be better than increasing the time window larger than one. It is also remarkable that the day of the week, the interval of the day and the gender of a restroom are not correlated at all to the number of bad votes. Furthermore the surface, the number of toilets and the rank of a restroom show only low correlation coefficients.

The dataset is not particularly high dimensional, ranging from 9 dimensions to 103 for time windows 1 and 48 respectively, with 48 being the largest used time window in this study. The dimensionality can be increased by 33 if the one-hot encoding method is used instead of rank-based restroom encoding. Because of this number of dimensions, model training times are expected to be reasonable and all features are included.

E. Numerical Feature Scaling

Because some machine learning models are sensitive to feature scaling, we use two basic functions to scale numerical features: Normalization and standardization. Normalization scales each value in a feature vector within the range [0, 1]

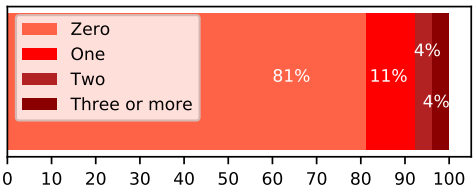

Fig. 5. Distribution of the number of bad votes.

TABLE II
DATASET OVERVIEW. SHOWING NUMBER OF FEATURES (FEAT) AND CLASS RATIOS FOR DIFFERENT SIZES OF THE SLIDING WINDOW (WS) AND CLASS DEFINITIONS.

Definition	ws	feat.	classes	points	ratio
Strict	1	9	2 (Clean, Unclean)	114.240	6:1
Lenient	1	9	2 (Clean, Unclean)	114.240	16:1
Strict	24	55	2 (Clean, Unclean)	114.240	6:1
Lenient	24	55	2 (Clean, Unclean)	114.240	16:1
Strict	48	103	2 (Clean, Unclean)	114.240	6:1
Lenient	48	103	2 (Clean, Unclean)	114.240	16:1

and standardization scales each value in a feature vector such that the mean is zero and the variance is one.

F. Class Definition

The two defined classes for the binary classification problem at hand are: clean and unclean, which refers to the state of a restroom as observed by the users. We acknowledge the fact that user observations are subjective and sometimes do not correspond to the actual state of a restroom, this will further be discussed in Section VI.

Fig. 5 shows the distribution of the number of bad votes per time interval of thirty minutes. It immediately becomes clear that we are dealing with an imbalanced dataset, regardless of how we define the two classes. The reason that more than 81% of the time intervals receive no bad votes at all is twofold. First of all, during the night there are much fewer passengers at the airport than during the day. This results in considerably fewer votes during the night, and most of the time no votes at all. Secondly, there are large differences between the number of visitors per restroom, which in turn influences the probability of receiving votes. Smaller restrooms that are not often visited have many intervals at which there are no votes at all.

1) Strict Class Definition: The most obvious class definition would be to define zero bad votes received as clean, and one or more bad votes received as unclean. This would result in a clean:unclean balance ratio of around 6:1. When discussing this class definition with senior personnel and decision makers, it became clear that this would not be ideal because it would mean that in practice cleaners could be redirected to another restroom for only a single bad vote. This is considered too costly for the benefit that it could yield.

2) Lenient Class Definition: A logically following and slightly different class definition would be to define zero or one bad vote received as clean, and two or more bad votes received as unclean. Resulting in a clean:unclean balance ratio of around 16:1. According to senior personnel and decision makers, this would make more sense because it doubles the possible benefit compared to the other class definition. Also, it is more unlikely that an observation of the class unclean is noise since multiple bad votes are received instead of a single bad vote.

We expect a certain trade-off between classification model performance and practical usability when defining the two classes. Therefore we include both above-mentioned class definitions in the research and compare the results to study the effect of different class definitions on classifier performance. A summary of the dataset for different class definitions and sliding window sizes is presented in Table II.

G. Model Definition

Next to the distinction between two class definitions, there is also a distinction in the type of prediction model. The first type is a combined prediction model that includes features of all the restrooms and makes predictions for all the restrooms. The second type is a restroom-specific prediction model that focuses only on a single restroom. The choice to also study restroom-specific prediction models is based on the fact that there is a lot of variation between the restrooms and we assume that this will lead to differences in model performance. Fig. 6 shows the occurrences of unclean samples per restroom in the case of a lenient class definition. The red lines indicate how often an unclean sample occurs on average per day, on two different levels. We observe that restrooms 60 Male and 60 Female are responsible for many of the unclean occurrences, roughly twice as much as the runner-up, restrooms 43. Next to that, we see that almost half of the restrooms do not even have one unclean sample per day on average. If the number of unclean occurrences decreases, the class imbalance logically increases. This raises the question of whether restrooms to the far right of the plot are even worth considering when the objective is to improve user satisfaction.

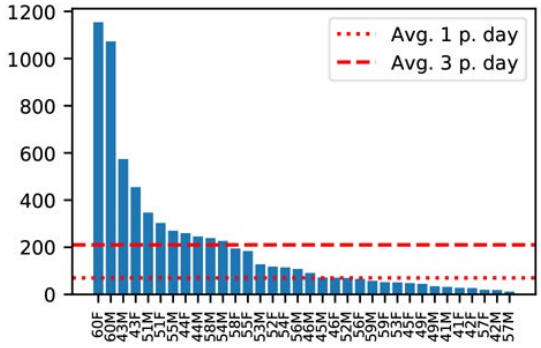

Fig. 6. Number of unclean samples per restroom.

IV. RESEARCH METHOD

In order to study the usefulness of real time feedback data and classification techniques to distinguish and predict clean and unclean restrooms in practice, we are searching for the

best performing classifiers on different settings of the problem. This section explains the experimental setup and metrics used for performance evaluation.

A. Classifiers

Classifiers have been constructed for the combined prediction model type as well as for the restroom-specific prediction model type. This is also done for the two different class definitions, strict and lenient. For the restroom-specific prediction models, Random Forrest (RF), Support Vector Machine (SVM), AdaBoost (AB) and K-Nearest Neighbors (KNN) algorithms have been applied. For the combined prediction models RF, AB and KNN were applied. The selection of algorithms was based on the results obtained during a preliminary classification experiment. Algorithms that were also included in this preliminary experiment but not selected do to their poor performance on this dataset were: Decision Tree, EasyEnsemble, RUSBoost, Complement Naïve Bayes and Multilayer Perceptron. To identify the best model settings for each classification algorithm, exhaustive grid search was performed using a validation set.

B. Baselines

To evaluate the performance of the best classifier models, they are compared to three baselines. The first baseline is the prior probability (PP) baseline that uses the prior probabilities to make a prediction. In other words, in case of a strict class definition, it predicts clean in 81% of the observations. The second one is the average bad vote (ABV) baseline that uses the average number of bad votes for a given time interval to make a prediction. The third one is the daily average bad vote (DABV) baseline which is nearly the same as the ABV baseline but next to time interval it also takes the day of the week into account.

C. Sampling Techniques

Related work has pointed out that sampling techniques might offer a solution when working with an imbalanced dataset or class overlap. To evaluate whether these techniques indeed improve the performance on this particular dataset, we include them in the exhaustive grid search. The included sampling techniques are Random Undersampling (RUS), Random Oversampling (ROS), Adaptive Synthetic Oversampling (ADASYN), Synthetic Minority Over-sampling Technique (SMOTE), SMOTE with data cleaning using Tomek links (SMOTE + Tomek) and SMOTE with data cleaning using Edited Nearest Neighbours (SMOTE + ENN).

D. Evaluation Metrics

In an imbalanced learning scenario, the traditional accuracy metric turns out to be quite ineffective for evaluating the performance of a classifier [5]. Therefore there is a need for a different kind of evaluation metric. Reasoning from a practical point of view, we want to predict unclean as accurately as possible and after that as many as possible. This means that for the class unclean, precision is more important than recall. High precision is important because redirecting a cleaner is a costly intervention, so the model needs a high degree of certainty about the prediction. A high recall is less important because there are only a few cleaners to respond to an alert that is caused by an unclean prediction. If there are too many alerts in a short time period, the cleaners will not be able to adequately respond to all of them. Because of this, the main metric that we focus on is the F-Beta score of the class unclean with a beta of 0.5. This F0.5-score means that precision is twice as important as recall in calculating the weighted harmonic mean of both. Next to the F0.5-score of the class unclean, we also present the corresponding precision and recall scores.

V. Results

A. Visual Insights

In order to visualize the data, PCA is performed to obtain the first two principal components and plot them in two-dimensional space. Figures 7 and 8 show how different time windows, data rescaling methods and restroom encodings lead to different 2D PCA visualizations of the data. The visualizations are obtained using all restrooms and a strict class definition. We observe clear visualization differences between different rescaling methods and also between different time windows. Between one-hot encoded restrooms in Fig. 7 and rank-based encoded restrooms in Fig. 8 we only see very small, negligible, differences when looking at the normalization and standardization plots. Only the plots without data rescaling, indicated by *none*, show noticeable but no significant differences. From this, we conclude that different time windows and rescaling methods will probably lead to different classifier performance, where different restroom encodings will probably not. Although we see some concentrated unclean (blue) datapoints in some of the plots, they are still intertwined with many clean (red) datapoints, which indicates major class overlap. By looking at the visualizations, one would say that the cases without data rescaling and time windows 24 and 48 and standardization with time window 24 would be best separable, but the results of section V-B will show that they are not. This confirms that even those, by eye quite separable, cases have a large degree of class overlap.

Fig. 9 shows the effect of the different class definitions using 2D PCA visualization. Logically we see less unclean (blue) points in the right plot because for this class definition one bad vote is also considered as clean (red). The lenient class definition is basically a subset of the strict class definition. The hypothesis is that lenient class definition reduces class overlap compared to a strict class definition. This could be the case if for example lower located unclean (blue) points would turn into clean (red) points while upper located unclean (blue) points would remain when switching from a strict class definition to a lenient class definition. From this figure, we conclude that a different class definition does not significantly reduce class overlap, but that classifier performance is likely to be different because of different balance ratios.

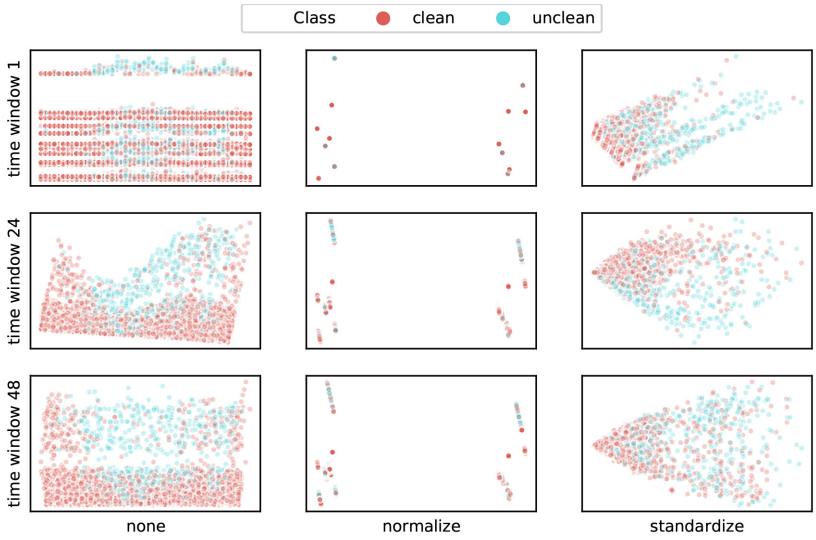

Fig. 7. 2D PCA Visualization for different time windows and rescaling for OHE restrooms with strict class definition.

Fig. 8. 2D PCA Visualization for different time windows and rescaling for rank-based restrooms with strict class definition.

B. Combined Prediction Model

1) Strict Class Definition: Table III lists the performance results for a combined prediction model that includes all restrooms. In this table, we see that the two more informed baseline classifiers, ABV and DABV, perform much better than the naive baseline, PP.

When comparing the F0.5 scores of the RF, AB and KNN algorithms, we see that all three perform slightly better than the baselines and that AB and KNN perform roughly the same. The main difference is that KNN has better precision, where AB has better recall. Because precision is deemed more important than recall, we select KNN as the best model. The best performance was found with ranked restroom encoding, no data rescaling and a time window of one. We observe approximately the same results when using the first two principal components as input features and can therefore give a representative visualization of the performance using decision regions of the model. Fig. 10 shows the 2D PCA visualization of the whole dataset as well as the decision regions as used by the KNN model. Visual inspection shows that the model is capable of classifying many of the unclean datapoints in the upper part of the plot, but not capable of classifying unclean datapoints that are located more towards the center of the plot, leading to a low recall. Although the model correctly classifies many unclean datapoints in the upper part, it also misclassifies a lot of clean datapoints within this region, leading to an unsatisfying precision.

A remarkable result is that despite the findings of Batista et al. [9], who conclude that SMOTE-based methods are very effective when dealing with highly imbalanced and overlapping data, we observe severe performance decrease

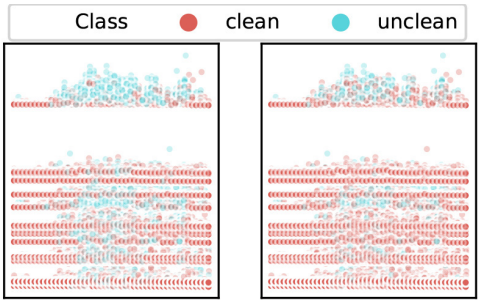

Fig. 9. 2D PCA Visualization differences between class definitions strict and lenient for OHE restrooms without data rescaling and time window 1.

TABLE III
PERFORMANCE METRICS FOR CLASS *Unclean* FOR COMBINED PREDICTION MODEL WITH STRICT CLASS DEFINITION

	Train			Test		
	F0.5	Prec	Rec	F0.5	Prec	Rec
PP	0.14	0.14	0.14	0.14	0.14	0.14
ABV	0.47	0.74	0.19	0.45	0.70	0.19
DABV	0.51	0.75	0.22	0.44	0.64	0.20
RF	0.91	1.00	0.68	0.47	0.58	0.27
AB	0.56	0.63	0.39	0.50	0.56	0.35
KNN	**0.57**	**0.72**	**0.31**	**0.49**	**0.62**	**0.27**
KNN PCA	0.54	0.70	0.28	0.49	0.64	0.26
KNN SMOTE+ENN	0.92	0.92	0.92	0.38	0.34	0.79

TABLE IV
PERFORMANCE METRICS FOR CLASS *Unclean* FOR COMBINED PREDICTION MODEL WITH LENIENT CLASS DEFINITION

	Train			Test		
	F0.5	Prec	Rec	F0.5	Prec	Rec
PP	0.06	0.06	0.06	0.05	0.05	0.06
ABV	0.49	0.55	0.34	0.41	0.44	0.33
DABV	0.52	0.56	0.39	0.37	0.38	0.33
RF	0.60	0.77	0.31	0.41	0.51	0.22
AB	**0.52**	**0.67**	**0.28**	**0.42**	**0.52**	**0.23**
AB PCA	0.48	0.66	0.23	0.41	0.51	0.22
KNN	0.47	0.71	0.20	0.42	0.56	0.21
KNN PCA	0.10	0.56	0.02	0.01	0.08	0.00
AB SMOTE+Tomek	0.93	0.93	0.93	0.23	0.19	0.77

when implementing sampling methods. The best performing sampling method is SMOTE + ENN, which decreases the F0.5 score of the KNN model from 0.49 to 0.38 and precision from 0.62 to 0.34. We do observe a large increase in recall, which means that the sampling caused an expansion of the unclean decision region.

2) Lenient Class Definition: Table IV lists the performance results for the lenient class definition. With this class definition, we see that the classification algorithms are hardly capable of outperforming the ABV baseline in terms of F0.5 score. They do perform better on precision but do this at the expense of a lower recall score. When comparing the three classification algorithms, we see a similar performance and would select KNN as the best algorithm based on the precision score. But when using the first two principal components as input features, we obtain completely different results. This is caused by the fact that the best KNN performance was found with OHE restroom encoding, data normalization and

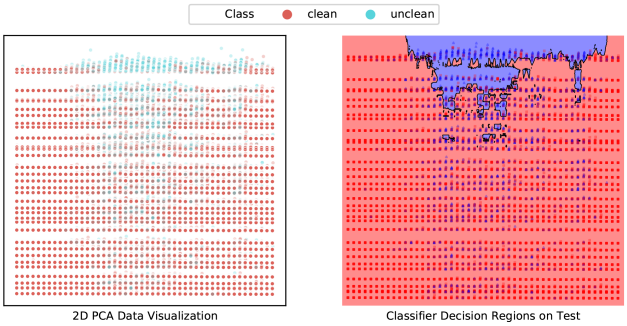

Fig. 10. K-Nearest Neighbors PCA Visualization with strict class definition, ranked restroom encoding, no data rescaling and time window one. (left) all datapoints, and (right) Test set datapoints and KNN PCA model decision regions.

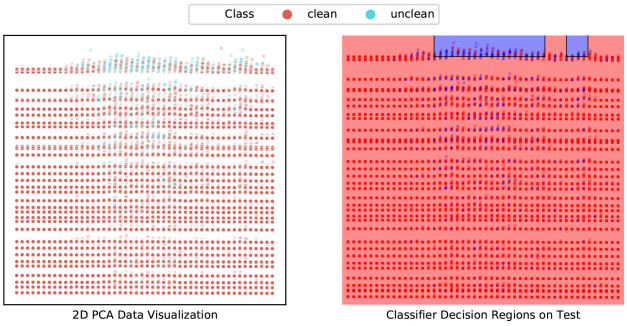

Fig. 11. AdaBoost PCA Visualization with lenient class definition, ranked restroom encoding, no data rescaling and time window one. (left) all datapoints, and (right) Test set datapoints and AB PCA model decision regions.

a time window of 24. The plot in the center of Fig. 7 shows us that this configuration leads to a situation where there is almost no difference between the two classes. In combination with more clean than unclean datapoints, caused by class imbalance, this leads to poor performance when using the first two principal components as input features. Therefore we select the AdaBoost algorithm, which performed best with ranked restroom encoding, no data rescaling and a time window of one, to visualize the classifier decision regions. Fig. 11 shows this visualization. Inspection of the decision regions clarifies the low precision and even lower recall, the model can only classify a minority of the unclean datapoints while simultaneously misclassifying clean datapoints.

C. Restroom-specific Prediction Model

To train and evaluate restroom-specific prediction models, six different restrooms were selected. This selection is based on the number of unclean sample occurrences as shown in Fig. 6. Restrooms 60 male and female have a high number of unclean samples, restrooms 57 male and female have a low number of unclean samples and restrooms 46 male and female are somewhere in the middle. Using a single restroom to train a model means that the size of the dataset is reduced to 3.360 datapoints, obtained by 70 days times 48 intervals.

1) Strict Class Definition: Table V shows the best baseline and the best classification algorithm for every selected

TABLE V
PERFORMANCE METRICS FOR CLASS *Unclean* FOR RESTROOM-SPECIFIC PREDICTION MODELS WITH STRICT CLASS DEFINITION

		Train			Test		
Restroom	Model	F0.5	Pre	Rec	F0.5	Pre	Rec
60 Male	ABV	0.73	0.76	0.63	0.73	0.75	0.65
60 Male	AB	0.78	0.77	0.84	0.73	0.70	0.89
60 Female	ABV	0.75	0.75	0.74	0.76	0.75	0.78
60 Female	KNN	0.82	0.83	0.79	0.74	0.79	0.61
46 Male	PP	0.12	0.11	0.12	0.14	0.15	0.13
46 Male	RF ROS	0.97	0.97	0.99	0.34	0.35	0.31
46 Female	PP	0.10	0.10	0.09	0.00	0.00	0.00
46 Female	AB SMOTE +Tomek	0.90	0.92	0.81	0.31	0.31	0.33
57 Male	PP	0.00	0.00	0.00	0.00	0.00	0.00
57 Male	AB SMOTE +Tomek	0.95	0.97	0.91	0.00	0.00	0.00
57 Female	PP	0.01	0.01	0.01	0.00	0.00	0.00
57 Female	SVM Adasyn	0.90	0.98	0.66	0.12	0.14	0.08

TABLE VI
PERFORMANCE METRICS FOR CLASS *Unclean* FOR RESTROOM-SPECIFIC PREDICTION MODELS WITH LENIENT CLASS DEFINITION

		Train			Test		
Restroom	Model	F0.5	Pre	Rec	F0.5	Pre	Rec
60 Male	ABV	0.61	0.59	0.72	0.56	0.53	0.73
60 Male	AB	0.68	0.69	0.68	0.58	0.58	0.56
60 Female	ABV	0.63	0.60	0.81	0.54	0.49	0.84
60 Female	KNN	0.73	0.76	0.61	0.58	0.59	0.57
46 Male	PP	0.03	0.03	0.03	0.00	0.00	0.00
46 Male	SVM	0.53	1.00	0.18	0.31	0.33	0.23
46 Female	DABV	0.06	0.12	0.02	0.00	0.00	0.00
46 Female	KNN RUS	0.69	0.69	0.67	0.10	0.09	0.14
57 Male	PP	0.00	0.00	0.00	0.00	0.00	0.00
57 Male	RF SMOTE	1.00	1.00	1.00	0.00	0.00	0.00
57 Female	PP	0.00	0.00	0.00	0.00	0.00	0.00
57 Female	RF ROS	1.00	1.00	1.00	0.62	1.00	0.25

restroom. The results show that the best models of restrooms 60 male and female perform equal to the best performing baselines whereas the other restrooms best models outperform their best baselines. Next to that, we observe that except for restrooms 60, the exhaustive grid search designated a model using a sampling method to be the best model. This is in contrast to the results of the combined prediction model, where sampling methods decrease performance. We believe that the reason for improved performance using sampling methods with some restroom-specific prediction models is the greater class imbalance of the corresponding datasets. For example, the unclean:clean balance ratio of restroom 57 male is 1:37. It is also worth mentioning that, when comparing the results on train and test data, the models with sampling methods

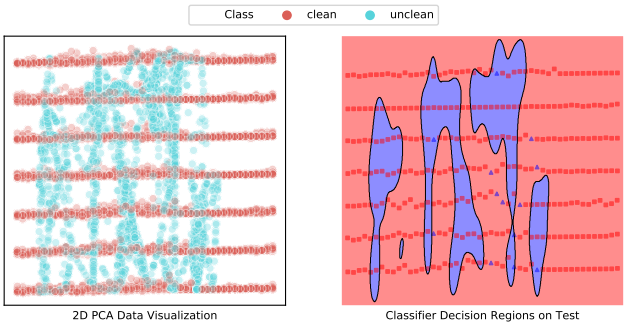

Fig. 12. Restroom 57 female: SVM-PCA visualization with strict class definition, no data rescaling and time window 24. (left) all datapoints, and (right) test set datapoints and SVM PCA model decision regions.

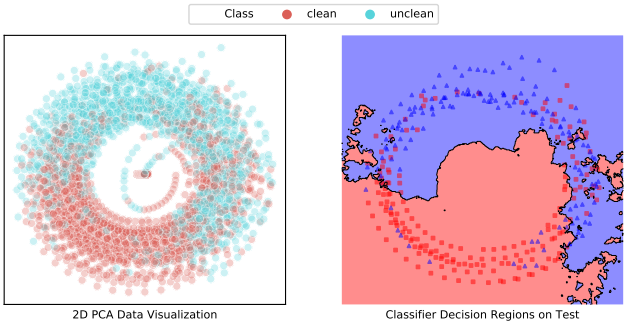

Fig. 13. Restroom 60 Female: K-NN-PCA visualization with strict class definition, data standardization and time window 48. (left) all datapoints, and (right) test set datapoints and KNN PCA model decision regions.

seem to be overfitting on the training data. This results in poor performance on the unseen data of the test set, of which restroom 57 female is an example. The dataset and decision regions of this restroom SVM model are plotted in Fig. 12. On the left side of the figure, we see that the Adasyn sampling method has created a lot of synthetic unclean (blue) datapoints, to which the model has overfitted. This is visualized by the decision regions depicted on the right side of the figure, which show that the model is hardly capable of classifying the new unclean samples of the test set. Fig. 13 shows data and decision regions of the best performing restroom-specific prediction model, 60 female KNN. This figure shows that in some parts the classes can be reasonably separated and that the model does quite a good job in classifying new, unseen datapoints.

2) Lenient Class Definition: Table VI again shows the best baseline and the best classification algorithm for every selected restroom, but for the lenient class definition. We see that for restrooms 60 and 46, the best models outperform the best baselines, if only by a little. Next to that, we again observe that for the restrooms with less unclean samples, the best model is one that uses a sampling technique, which is again overfitting the training data. The result of restroom 57 female is suspicious because the best model performs very different from the best baseline result. Inspection of the model shows that there are only four unclean samples in the test set of which one is classified as unclean. Every other sample is classified

as clean, resulting in a recall of 0.25 and a precision of 1.00. From the visual inspection of the model decision regions, we conclude that this correct classification was a coincidence.

VI. Discussion and Recommendations

In the case of users voting for the cleanliness of a restroom, class overlap means that under similar circumstances people tend to vote differently. We think that this has three main causes. Firstly, people have a different perception of cleanliness. A toilet that one person reviews as clean, might be considered unclean by another person. This is the subjective nature of the data that we are working with and it will always be present. Secondly, a restroom has multiple toilets and not every person that casts a vote visits the same toilet. One person might visit a clean toilet while another visits an unclean toilet in the same restroom, resulting in two contradicting votes. A solution to this would be to place a smiley box in every separate toilet, asking people to rate the cleanliness of that particular toilet instead of the whole restroom. This will definitely improve the practical usability because it will reduce the number of contradicting votes per time interval. Especially for restrooms that do not receive a large number of votes, it will be advantageous because it will point out an unclean toilet faster. The third cause is the lack of representative data. The overlapping classes mean that the current data is not capable of separating the unclean samples from the clean samples. This can be improved by creating or searching for more meaningful features that are capable of distinguishing the two classes. Two possible meaningful features that directly come to mind are the actual number of visitors per restroom and the exact cleaning time of a restroom. The number of visitors could prove useful because not every visitor casts a vote and therefore at this moment we do not exactly know how busy a restroom is. The exact cleaning time of a restroom could improve the performance because at this moment we do not exactly know when a restroom was cleaned, while it certainly has an impact on the cleanliness of a restroom.

VII. Conclusions

Data visualizations of the combined prediction model with all restrooms show that there is a certain amount of class overlap present in the data. It turns out that the class imbalance is not a major problem because the decision regions of the best models show that datapoints of the class unclean are correctly classified. The problem is that within this region there are also many datapoints of the class clean that are being misclassified, which negatively affects the precision of the class unclean. While combined prediction models outperform the baselines, the precision is still too low for practical application. Using a strict class definition instead of the lenient class definition increased the performance. The best performing combined prediction model is the kNN algorithm with a F0.5 score of 0.49 is obtained with a corresponding precision of 0.62 and a recall of 0.27, and sampling does not improve results.

Further investigations of prediction models trained separately for each restroom, shows that the restrooms with the most unclean samples, i.e., the minority class, perform significantly better than the combined prediction model, but do not outperform simple informed baselines (in terms of F0.5 score) and sampling improves performance for restrooms with highly unbalanced data.

To conclude, the performance of combined prediction models outperform combined baselines. For the restroom-specific prediction models only restrooms 60 male and female perform very good. To further increase the prediction accuracy our main recommendation is to perform future research on the data ambiguity problem that leads to class overlap and incorporate additional data sources, such as passenger flows and flight schedules into the prediction models.

References

[1] D. Canora, "System and method for distributed and real-time collection of customer satisfaction feedback. patent no. us8231047b2," 2008.

[2] D. C. J. R. W. Bossemeyer, "Customer feedback acquisition and processing system. patent no. us7058625b2," 1999.

[3] R. B. Rao, S. Krishnan, and R. S. Niculescu, "Data mining for improved cardiac care," *SIGKDD Explor. Newsl.*, vol. 8, no. 1, pp. 3–10, Jun. 2006. [Online]. Available: http://doi.acm.org/10.1145/1147234.1147236

[4] A. L. P. S. J. S. P. K. Chan, W. Fan, "The semantic web and its languages," *IEEE Intelligent Systems*, vol. 14, no. 06, pp. 67–73, nov 2000.

[5] E. A. Garcia and H. He, "Learning from imbalanced data," *IEEE Transactions on Knowledge & Data Engineering*, vol. 21, no. 09, pp. 1263–1284, sep 2009.

[6] X. Liu, J. Wu, and Z. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, April 2009.

[7] L. O. H. W. P. K. N. V. Chawla, K. W. Bowyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321 – 357, jun 2002.

[8] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, June 2008, pp. 1322–1328.

[9] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, Jun. 2004. [Online]. Available: http://doi.acm.org/10.1145/1007730.1007735

[10] G. M. Weiss and F. Provost, "The effect of class distribution on classifier learning: An empirical study," *Technical Report ML-TR-44, Department of Computer Science, Rutgers University*, aug 2001.

[11] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 973–978. [Online]. Available: http://dl.acm.org/citation.cfm?id=1642194.1642224

[12] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *Proc. Int. Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II*, 2003.

[13] R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard, "Class imbalances versus class overlapping: An analysis of a learning system behavior," in *MICAI 2004: Advances in Artificial Intelligence*, R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 312–321.

[14] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," *arXiv e-prints*, p. arXiv:1403.2877, Mar 2014.

[15] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.101

[16] M. Vos, M. Galetzka, M. Mobach, M. van Hagen, and A. Pruyn, "Measuring perceived cleanliness in service environments: Scale development and validation," *International journal of hospitality management*, vol. 83, pp. 11–18, 4 2019.