APPLIED RESEARCH

Optimized Operation Management With Predicted Filling Levels of the Litter Bins for a Fleet of Autonomous Urban Service Robots

ANTON POLLAK[ID]**, ABHISHEK GUPTA**[ID]**, (Graduate Student Member, IEEE), AND DIETMAR GÖHLICH**[ID]

Chair of Methods in Product Development and Mechatronics, Technical University of Berlin, 10623 Berlin, Germany

Corresponding author: Abhishek Gupta (abhishek.gupta@tu-berlin.de)

ABSTRACT Autonomous smart waste management services are becoming an essential component of sustainable urbanization. However, the lack of data and insights from current service-providers impedes a reliable transition from labor-intensive to autonomous services. Deploying information gathering devices makes services expensive and resource-demanding. In project MARBLE (Mobile Autonomous RoBot for Litter Emptying) we are currently investigating the implementation of a fleet of service robots. In this framework, we could show that the absence of filling data of litter bins (LBs) hinders the possibility of providing an energy-efficient and time-effective service. Hence, we introduce an approach where machine learning-based predictions for filling levels of LBs, derived from our extensive data gathering, are used to effectively manage the autonomous emptying process. The novel *Simulated Rebalancing* approach in route-planning combined with the Knapsack algorithm ensures efficient service in comparison to the Nearest Neighbor algorithm. A promising 82% filling level prediction accuracy was achieved with the *XGBoost* binary classifier, as compared to the 59% baseline accuracy. Through incorporating the predicted filling level data in the *Simulated Rebalancing* approach, a reduction of 26% in operational time and 31% in energy consumption was achieved for our simulated tests for service-event-area (SEA) James-Simon-Monbijoupark in Berlin with 49 LBs.

INDEX TERMS Knapsack algorithm, litter bin filling-level prediction, machine learning, responsible energy consumption, robot-fleets route planning, smart cities, sustainable waste management, urban service robots, XGBoost.

I. INTRODUCTION

Waste management is amongst the many service areas that are incorporating automation in their various involved processes [1], [2], [3], [4], [5], [6]. Automation is being provided by robots in processes like waste sorting [7], [8], waste collecting, and compacting [9], [10]. Achieving service goals while integrating sustainability goals for the

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro[ID].

development and management of the service by robots is a relatively new, but important field [11]. Responsible energy consumption in automated services can be achieved by providing robots with information for making decisions that prevent unnecessary energy consumption. The information can be either gathered from smart Internet of Things (IoT) devices, or predicted based on the previously gathered data.

The municipality of Berlin is responsible for waste collection and street cleaning in Berlin. Litter bins (LBs)

deployed by the municipality of Berlin in its urban landscape play a very crucial role in the waste management process [12], [13]. Their current LBs emptying process involves a worker with a diesel truck stopping at every LB within their assigned area. Such manual processes lead to higher energy consumption, while exhaust emissions and the strenuous work behind manually emptying the LBs have an impact on the health of the workers. The municipality of Berlin is incorporating several changes to provide solutions for these issues and address all the dimensions for better and sustainable waste management: ecological, economic, social and society as well as quality of life [14].

One of the many possible solutions is testing the feasibility of automating processes and introducing human-robot collaboration in the existing processes with multiple research partners. In cooperation with the municipality of Berlin, the Technical University of Berlin's project MARBLE (**M**obile **A**utonomous **R**o**B**ot for **L**itter **E**mptying) [15], [16] has successfully developed a fully operational prototype of an autonomous robot designed to empty LBs. The main objective of this project is to reduce energy consumption and CO_2 emissions by automating the LB emptying process. Urban service robots lack the experience based knowledge which humans have gathered over the years while performing their jobs [17]. Similarly, MARBLE has no previous knowledge about the filling levels of the LBs. Preceding works have provided assistance in route planning by developing simulated routes for emptying the LBs with the least possible energy consumption [18], [19], [20]. These routes are till now either based on the assumption that the dustbins will be 50% full [18], [20] or that they will be provided with assistance from future smart LBs [19]. But neither is the assumption of 50% reliable, nor converting all the conventional LBs into smart LBs economically and ecologically feasible in the near future.

This research work focuses on firstly gathering the filling levels of all LBs within a defined area over multiple weeks, while also recording potentially relevant dynamic features of the data points such as the date, weather conditions and special events, combining them with static features related to the locations of the bins. Using this data, it is shown that the prediction of filling levels is possible. This can be used to further optimize routes. By omitting LBs from the route filled less than a predefined percentage and scheduling them for the following day, energy consumption can already be reduced significantly. The robot fleet's routes are then optimized further towards minimal energy consumption in order to make them more competitive with the current, traditional work process.

The use case and the research focus are elaborated in Section II and a detailed state-of-the-art comparison is provided in Section III. Section IV provides insight into the scientific approach of the proposed solution. The results are described in Section V. Based on the analysis of the results, a conclusion and outlook related to this research work are outlaid in Section VI.

II. RESEARCH FOCUS

The prototype of the autonomous service robot MARBLE and its workflow have been illustrated in Figure 1. An autonomous platform (AP) equipped with sensors such as Lidar, ultrasonic sensors, GPS, and depth cameras enables autonomous navigation (process ①) towards the litter bins (LB). The robot arm (RA) with the depth camera opens the LB (process ②) after its recognition and detection. The garbage is collected in a waste basket (WB), that can also rotate around its attached position and this way aid in closing the LB (process ③). The rotating movement is used to transfer the garbage (process ④) into the container with a built-in press (Longitudinal press (LP)) and extractor (CPE) for compressing the garbage (process ⑤). Upon reaching the maximum storage capacity of the compressed garbage, it is removed with the help of the extractor (process ⑥).

Due to the limited available energy and compressed garbage storage capacity, a conceptual assistance vehicle (*mothership*) has been incorporated into the service, tasked with collecting the extracted garbage as well as battery swapping. The goal is to assist in the uninterrupted operation of the robots, preventing them from having to return to the garbage depot after the maximum capacity has been reached.

To further enhance the overall operational efficiency, the concept of operating multiple robots in a fleet has been developed [20]. The operational service-event-area (SEA) under consideration for this research work is the James-Simon- and Monbijoupark in Berlin (short Monbijoupark in the following) with 49 LBs. The positions of these have been marked on the park's map with orange circles in Figure 2. The robots do not know how full a LB might be. This can lead to wasted energy if the robots travel to a LB with no garbage to offer, or to a LB with more garbage than their storage capacity.

The easiest and most logical solution to such a problem would be to provide the actual status of the filling levels of the LBs by converting all the conventional LBs into smart LBs [19]. But doing so is costly and resource- (material-, energy- and time-) demanding, especially if it is to be implemented in all of Berlin with its 26,000 dustbins [22]. All the dimensions of sustainability should be addressed for incorporating new autonomous technologies in society, so that their incorporation has a minimal negative impact [11], [23]. The overarching objective of the MARBLE project is to encompass all aspects of sustainability, which have thus far been considered, integrated and their impact has been analyzed in the prototype development phase [24], [25].

The 17 Sustainable Development Goals (SDGs) provide a comprehensive framework for evaluating the impact of emerging technologies like robotics and automation across various dimensions of sustainability [11]. SDG-12 focuses on responsible resource consumption and plays a crucial role in guiding the integration of robotics and automation to promote sustainability [11]. In the context of the MARBLE project and in line with SDG-12, our research focus lies on creating an operational management system. The objective is
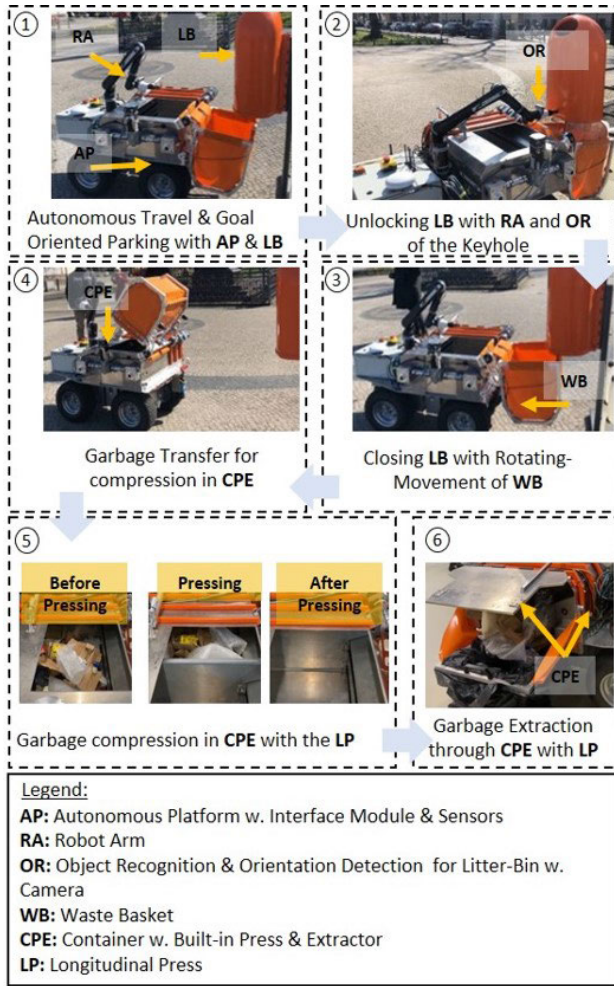
FIGURE 1. MARBLE performing its functionalities during a test at palace Charlottenburg in Berlin.
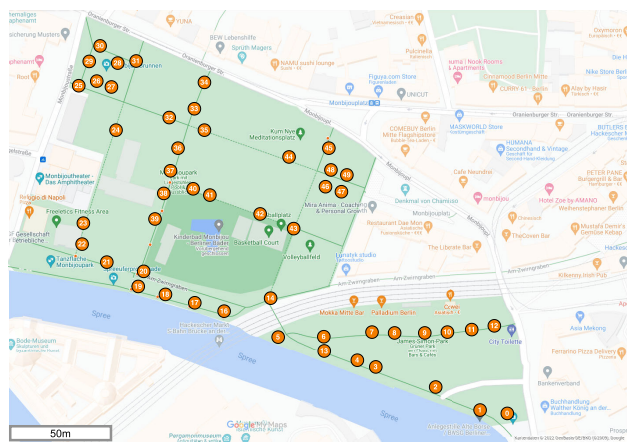

FIGURE 2. Labelled litter bin positions (orange circles) on the map of the James-Simon-Monbijoupark in Berlin (modified according to [21]).

to enable robots to operate with the utmost energy efficiency when providing their services. Instead of the costly and resource-intensive conversion of all bins into smart bins, we are investigating an alternative approach using machine learning to predict LB fill levels. These predictions are then

utilized in route planning to minimize energy consumption as much as possible during service operation. Hence, the research focus of this work is to provide a simulated proof of concept for the question of how the prediction of LB filling levels using machine learning can help to reduce the energy consumption of automated LB emptying in the streets of Berlin.

III. STATE OF THE ART AND TECHNOLOGY
A. ROUTE PLANNING
In the context of this paper, route planning is defined as the high-level planning of a full service route through an entire SEA. It can be modelled as either a Traveling Salesperson Problem (TSP) or a Vehicle Routing Problem (VRP). The former is a simple touring problem for visiting each node in the SEA exactly once while finding the shortest tour [26]. The latter is a generalization of the TSP with multiple agents available for the task [27]. There are many algorithms attempting to solve these problems. For the MARBLE project, two were implemented. The first one is the Simulated Annealing (SA) algorithm, which involves a metaheuristic, modelled after the cooling process in metallurgy, intended to "guide" a heuristic route optimization algorithm [28]. The algorithm starts with an initial route created using a nearest neighbor or random node order algorithm [29], which is then iteratively optimized (see previous publication [29] for more information). Furthermore, the Knapsack problem (KSP) algorithm was implemented for a single as well as a swarm of MARBLE robots. Apart from their distance to the current position, it takes into account the bins' filling levels in order to optimize the utilization of MARBLE's limited internal waste capacity [30].

B. MACHINE LEARNING
Before performing machine learning (ML), the Exploratory Data Analysis (EDA) is a common method to achieve an initial overview over a dataset and perform simple preliminary analyses. This is often done through graphical visualizations [31]. The field of machine learning (ML) has experienced a significant development in recent years [32]. Shallow learning describes all ML techniques that do not belong to deep learning (i.e., involve deep neural networks) [33]. It encompasses popular traditional ML algorithms, such as *Linear Regressions* or *Decision Trees* as examples for supervised algorithms [34]. Furthermore, ensemble methods were developed to combine predictions of multiple models of a certain algorithm, called *base estimators*, into one final prediction. The aim is to improve generalizability and robustness compared to a single model [35]. The base estimators can either be combined using averaging methods, as in a *Random Forest* [36], or using boosting methods, such as in *Adaptive Boosting (AdaBoost)* [37] and *eXtreme Gradient Boosting (XGBoost)* [38].

C. SMART WASTE MANAGEMENT
The prediction of waste generation using machine learning has been performed before. However, the scopes of these

projects were usually on a larger scale than for this paper, both in terms of time and location, hence not directly applicable for our research focus. An example of these larger scales is the prediction on an annual basis for entire countries. Kulisz et al. performed a prediction of the annual waste production for Poland [39]. The social and economic features chosen for prediction have large-scale character, like population, revenue per capita and employment rate in this case [39]. On a smaller geographical scale, Fan & Fan predicted the Municipal Solid Waste (MSW) generation for the entire city of Shanghai, China [40]. They also dealt with more large-scale features, for example the daily average of tourists as well as the average household income [40]. Islam et al. predicted the quarterly yard waste generation (waste hailing from landscaping and yard maintenance) for the city of Winnipeg, Canada [41]. They used socioeconomic as well as climatic factors.

Several approaches involved the development of an IoT sensor for bins. Uganya et al. performed the prediction of monthly waste generation with a focus on hazardous waste and toxic gases [42]. Yusoff et al. also developed a sensor device which was deployed to a university [43]. While all other approaches used classical discriminative ML methods such as *Linear Regressions* or *Support Vector Machines*, Ahmed et al. applied time-series models to a historical dataset of bin filling levels for the city of Wyndham, Melbourne, Australia, which were obtained using smart sensors [44]. Walk et al. also used a time-series model to produce hourly forecasts. Their application domain was a Reverse Vending Machine used to deposit bottles, which collected data with a sensor located above the bin [45]. Furthermore, Alexopoulos et al. have used a camera-based deep learning model to estimate metal scrap bin filling levels in a copper tube plant [46].

There are also approaches that aimed to combine prediction and route planning. Ferrer & Alba predicted the filling levels of paper bins in a Spanish city through a time-series model and used these predictions for route planning [47]. Camero et al. built a deep learning model on the same paper bin filling level dataset [48]. Vicentini et al. have combined prediction and route planning, but with the purpose of managing household waste [49]. Finally, these predictive models have never been developed in the context of an urban service robot and the unique challenges that come along with it and thus, the predictions have not been applied in their route planning. Combined with the small-scale day-to-day prediction of filling levels for public dustbins, this is the research gap this paper aims to fill.

IV. METHODOLOGY

Figure 3 summarizes the methodology proposed for the MARBLE system. Initially, data on the filling levels in a certain SEA are collected manually and fed into a database. In the database, the data points are augmented with additional information such as weather conditions, special events, the day of the week and relative positions of LBs, as these factors affect the filling levels of LBs. A machine learning algorithm uses this database to predict the filling levels within an SEA. Then, the list of bins for this particular tour is rebalanced based on these predicted filling levels (more on this method in Section IV-D). In combination with information about the available MARBLEs, the mothership, as well as the SEA, the predictions are used to plan the operation using a route planning algorithm. As a result, the planned route has specific operational characteristics, such as the number of MARBLEs required or the estimated energy consumed. The findings obtained from real-world implementations of simulated routes in the future can serve as valuable lessons for enhancing the accuracy of filling level predictions and refining route planning.
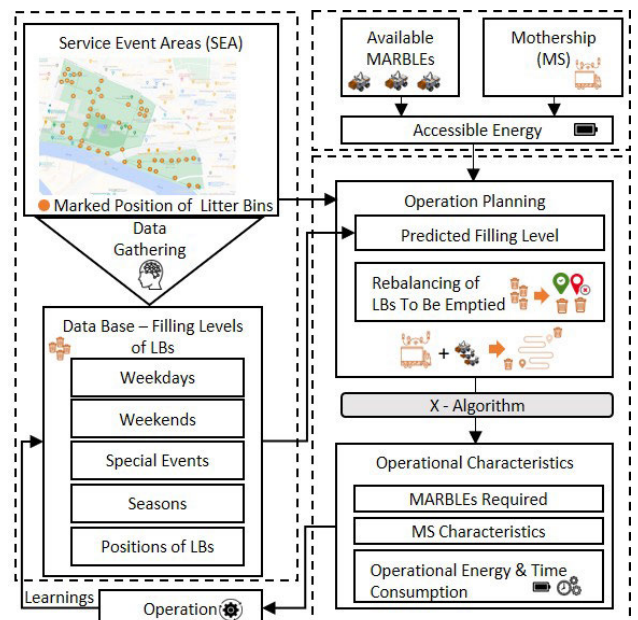
FIGURE 3. Methodology for extracting Operational Characteristics (OC) such as required number of MARBLEs and litter bins to be emptied.

A. DATASET GATHERING

Currently, there is no dataset available for monitoring the filling levels of the LBs. Therefore, we have designed a measuring device for manual filling level assessment. The device employed for measuring is illustrated in Figure 4. It uses an ultrasonic sensor to measure the distance from the top of the LB to the surface of the trash, as well as an *Arduino Uno* microcontroller to evaluate the sensor's measurements and convert them to a filling level percentage. This value is then shown on a small display.

The SEA chosen for this survey was the Monbijoupark in central Berlin, with a total of 49 dustbins. The data was recorded by walking off a fixed route and measuring each LB. This was always done around the same time early in the morning, before the municipality emptied the bins. This approach guaranteed the creation of a dependable dataset, which is essential since MARBLE must be able to efficiently remove waste from the LBs.

(a) CAD model of the measurement device

(b) Physical measurement device
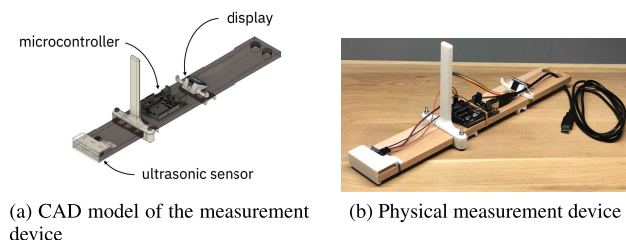
FIGURE 4. CAD model and actual measurement device for filling levels [50].

In addition to the filling levels and the exact timestamp for each data point, the weather data for each day was recorded and processed, and then added to the dataset as a feature. The weather recordings provided by Custom Weather Inc. [51] were used for this purpose [50]. This data has been processed in order to obtain a single score representing the weather quality. The score was constructed by combining two separate scores for temperature and weather condition, and averaging them. Besides these dynamic features, the dataset was also enriched with static features for each LB [50]. These were:

- Number of close neighbors: whether the LB has other LBs nearby (within a radius of 5m)
- Major pathway: whether the LB is located on a major pathway within the park
- Bench: whether the LB is located next to a bench

The dataset features used for the classification and prediction of the filling levels of LBs have been listed in Table 1.

TABLE 1. Features of the original filling level dataset.

Feature	Type	Possible values
Filling level (%)	continuous	$[0, 110]$
Weekday	categorical	$\{0, ..., 6\}$
Number of close neighbors	ordinal	$\mathbb{N} = \mathbb{Z}_+$
Major pathway	boolean	$\{True, False\}$
Has bench	boolean	$\{True, False\}$
Weather rating	continuous	$[0, 1]$

Apart from the standard dataset, a separate dataset for filling levels shortly after special events has been collected by Gao & Pollak. It was recorded on four days around Christmas 2022 and New Year's Eve 2022/23 [50].

B. DATASET PREPROCESSING

The machine learning process was repeatedly performed using different combinations of preprocessing techniques, which are shown in Figure 6. Firstly, the outliers have been removed based on the distribution for each LB using the interquartile rule (see e.g., [52] for more information). The total number of samples was 894 before outlier removal and 846 after. This means that 94.6% of the data points remain in the dataset.

Furthermore, outlier removal has been performed before the train-test split. Since the dataset is comparatively small, the identification of outliers can already be difficult. This

problem would be exacerbated by separating the testing set from the training set, and thus only having a subset of the data to identify outliers. The columns *LB ID* and *weekday* need to be encoded since they contain numerical data (integers from 0 to 49 and 0 to 6 respectively), whose ordinal character is unfavorable since it can be misinterpreted by an ML algorithm. When one-hot encoding a dataset, multicollinearity is introduced, which can cause the ML performance to degrade and affects the model's interpretability [53], [54]. To mitigate this, one of the superfluous features was removed. The *filling level* feature to be predicted is numerical and continuous. Therefore, regression algorithms, whose output is a continuous value, can be applied. However, classification algorithms should also be explored. In order to do that, the *filling level* feature needed to be discretized into categories. Five different rounding functions were explored, including a *low-level* binary function ($f \leq 15\%$ and $f > 15\%$) and a *normal* function with four classes that each span a range of 25%. For the train-test split, a stratified 80%/20% split was employed. This ratio was chosen because it is commonly used for data science projects; furthermore, there is no fixed rule according to which the ratio can be determined [55]. Figure 5 shows the class distributions of the entire dataset and the subsets for two of the rounding functions.

Furthermore, three different dimensionality reduction techniques were implemented. Firstly, a univariate feature selection was performed using the `f_regression` score function from `scikit-learn` [56] for the regression dataset, and the χ^2 test for the classification dataset. The number of features to select was set at 10, based on plotting all features' scores in descending order and manually choosing a cutoff point. Secondly, a Principal Component Analysis (PCA) was performed. It is technically not a feature selection method because instead of selecting a subset of features, new features are created based on the original ones (see for example this discussion: [57]). It still serves the same purpose as feature selection methods and was therefore implemented as well. The number of PCs was chosen based on the cumulative explained variance curve, choosing a value close to its inflection point [58].

Thirdly, a feature importance metric was used. Each feature in a dataset can be assigned an importance based on how relevant it is for an ML algorithm to make a prediction of the output variable [59]. In this case, an `ExtraTreesClassifier` implemented in `scikit-learn` [36] was fitted onto the dataset, using the *Gini* criterion. Since this feature selection method is performed using a classifier, it is only applicable to the classification dataset. The Synthetic Minority Oversampling TEchnique (SMOTE) was first described by Chawla et al. in 2002 [60]. It is a method used to combat imbalance of classes in datasets and is therefore only applicable to classification problems.

Through SMOTE, all classes occur the same number of times, therefore rectifying the dataset's bias towards one particular class. The original paper only suggests that applying SMOTE in combination with undersampling of the
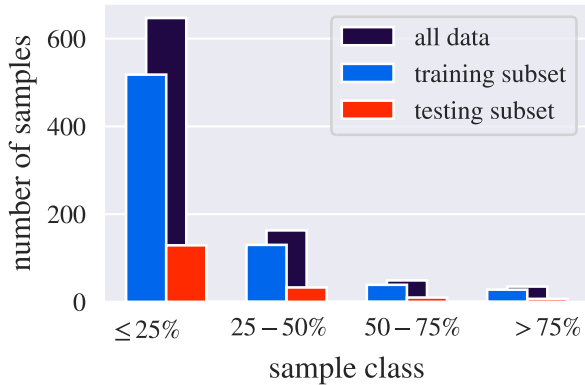
(a) Distributions for *normal* rounding function
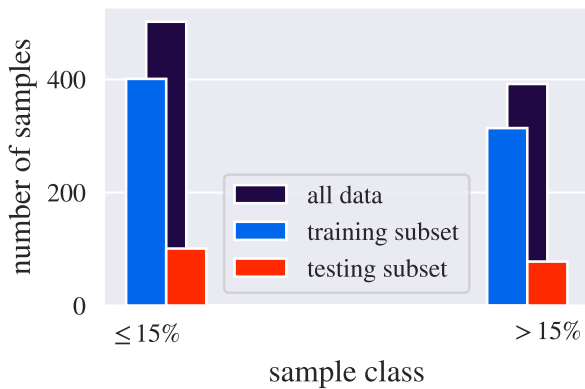
(b) Distributions for *low-level* (15%) rounding function

FIGURE 5. Sample distributions for training and testing subsets.

majority class can lead to a better performance than only applying undersampling. However, as the dataset is already quite small, the undersampling of the majority class has been omitted. Experiments using a combination of SMOTE and the undersampling method Edited Nearest Neighbors (ENN), which is implemented in the Python library `imbalanced-learn` as `SMOTEENN` [61], showed that undersampling actually worsened performance in this case. Therefore, it has not been pursued in the following. Figure 6 summarizes all preprocessing techniques used in this work.
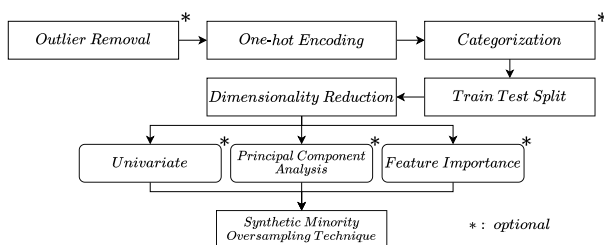

FIGURE 6. Flowchart of the applied preprocessing techniques.

C. MACHINE LEARNING FOR FILLING LEVEL PREDICTION

For both prediction types (regression and classification), multiple models have been tested in order to achieve a broad

overview. 12 models were implemented in total. Additionally, a baseline model was established for both cases to be able to compare the models' performances accurately.

For the regression, the baseline model always predicts the average filling level of all data points in the training subset. Furthermore, the implemented models were:

- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor (Ensemble)
- XGBoost Regressor (Ensemble)

For the classification task, the baseline model predicts the mode (most frequently occurring class) of the training data distribution. This baseline was compared against the following models:

- Decision Tree Classifier
- Random Forest Classifier (Ensemble)
- Random Forest Classifier (cost-sensitive learning) (Ensemble)
- Support Vector Machine
- Gaussian Naive Bayes
- XGBoost Classifier (Ensemble)
- AdaBoost Classifier (Ensemble)
- Neural Network Classifier

These models were selected because they were the most commonly used ones for general ML tasks. A wide range of models was chosen to provide an accurate comparison of current ML methods and their performance on the dataset. The Ensemble methods used were a Random Forest as the most prominent representative for averaging methods, as well as *AdaBoost* and *XGBoost* as the most common ones for boosting methods. There are ML models for specific use cases, such as time-series models (see for example [62]). However, this work firstly aimed to utilize the classical, general-purpose ML algorithms listed here. Other models could be explored in the future, as described in Section VI. The setup of this ML project involved a two-fold grid search in order to evaluate the best possible combination of parameters. The first grid search dimension concerned the preprocessing parameters, since there are many possibilities of combining them (see Figure 6). In order to find the combination resulting in the best performance, a grid was set up and many different combinations explored. This included the application of outlier removal, a feature selection method, SMOTE and a choice of rounding type for the classification task. The second grid search dimension was the hyperparameter optimization for each particular model. A randomized parameter optimization technique was employed, implemented in `scikit-learn`'s `RandomizedSearchCV` class [63]. Appropriate distributions for each model's hyperparameters were researched and passed to the search algorithm. The cross validation schema reduces the risk of random choice distorting the sample distribution and helps in obtaining reliable results [64], since the size of the dataset is relatively small. For all regression models, the Root Mean Square Error (RMSE) has been selected as the primary evaluation metric, mainly because it has the same unit as the

target variable (in this case unitless %) [65]. This makes it easy to interpret. For the classification models, accuracy was used as the primary evaluation metric because for this specific use case, no clear preference towards precision or recall could be determined. In order to not be misled by high accuracy scores in an imbalanced classification task (as is the case for the *normal* rounding function, see Figure 5), it was important to establish sensible baselines, as described previously.

D. SIMULATED REBALANCING BASED ROUTE PLANNING

In total, three weeks of data were collected by Gao & Pollak. In order to evaluate the route planning on this data in a sensible way, the three weeks were summarized into one representative "simulated week". This was achieved by collecting all data points for a particular LB ID and weekday combination, and then taking the average value for the respective entry in the simulated week.

At this point, it is useful to utilize an intermediate result introduced later on in Section V-A: As clear from Figure 9, the bins are usually relatively empty, with an average level of 21.74%. This information can be leveraged to further optimize the route planning: Before employing the algorithm, the simulated week can be manipulated. To reduce the number of bins per route, every LB along a route that is filled below a certain percentage can be excluded for that particular day. Then, the amount of waste that remains inside the LB can be added to the filling level of the consecutive day. As soon as the set threshold is reached, it will be included on the route and emptied as normal. This way, the route and the robot's press utilization can be optimized. The threshold chosen for this project was 25%, as it is above the main peak of the filling level distribution. In addition, 25% was deemed acceptable as a remaining filling level, especially considering that the waste amount added on top of that during an additional day will be in a similar range in most cases. Ferrer & Alba used a threshold of 80% filling level [47], however, their application domain is domestic paper waste, which arguably has higher predictability since it is less likely to be influenced by special events in the area boosting waste disposal. Furthermore, frequent emptying is not as necessary for paper waste as it is for other organic waste disposed in public LBs. Figure 7 shows a visualization of this rebalancing technique for LB number 49. On Tuesday, Friday and Sunday, the filling level stays below the threshold and is thus rebalanced to the following day. In the case of Sunday, the filling level is rolled over to Monday in order to make the simulated week as representative as possible. The implementation can be seen in Algorithm 1.

Using the results of this rebalancing technique, the route planning was executed with different algorithms. For the distances between the dustbins, the road distance between each pair of bins was calculated using the Google Maps API, polling it for the walking distance between the exact coordinates of the bins (see [66] for more information). For bins located directly adjacent of each other, the API returned a distance of 0m. In that case, the Haversine distance between
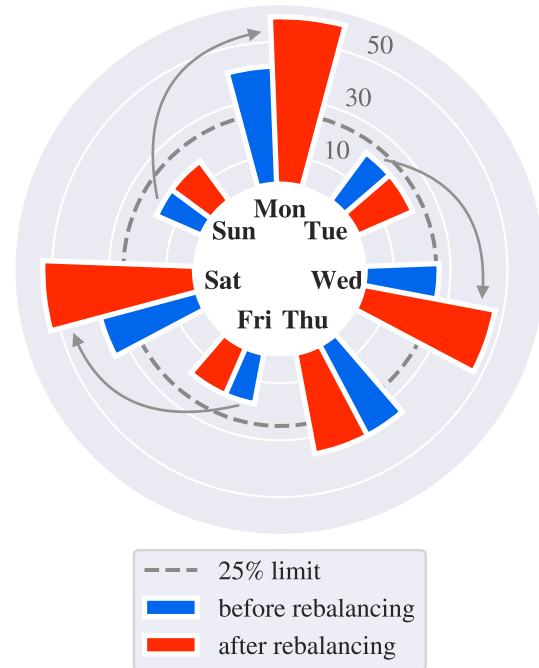

FIGURE 7. Rebalancing example for litter bin ID 49 (radial axis ≙ filling level (%)).

Algorithm 1 Rebalancing Implementation

Data: dataframe: containing filling level data for each
       bin
       weekdays
       bin IDs
for *weekday in weekdays* **do**
    for *ID in bin IDs* **do**
        if *dataframe(weekday, ID) ≤ 25* **then**
            if *weekday ≠ Sunday* **then**
                dataframe(weekday+1, ID) +=
                 dataframe(weekday, ID)
            else
                dataframe(Monday, ID) +=
                 dataframe(weekday, ID)
            end
        end
    end
end

---

the two coordinates was used instead (see [67] for more information).

Using these filling levels and distances, the Knapsack algorithm that was previously implemented for the MAR-BLE project was employed for route planning [19]. Its implementation for a single MARBLE agent can be seen in Algorithm 2. A specific value function is used to create the `values` matrix by combining distance and filling level information. The Knapsack algorithm then iteratively chooses the next waypoint based on this matrix. In order to be able to compare the Knapsack algorithm to a more naive one,

Algorithm 2 Knapsack Implementation

Data: *unvisited*
  distances
  levels: bin filling levels
  bin IDs
  MARBLECapacity
▷ Building the values matrix
values ← *matrix*(*len*(*bin IDs*), *len*(*bin IDs*))
for *ID1 in bin IDs* **do**
  for *ID2 in ID1* **do**
   values[*ID1*, *ID2*] ←
   valueFunction(*ID1,ID2*)
  end
 end
visited ← *unvisited*[0]
unvisited ← *unvisited*[1 :]
▷ Knapsack algorithm loop
while *unvisited* ≠ ∅ **do**
  visited ← *visited* +
  knapsack(*MARBLECapacity*, *levels*, *values*[*visited*[−1], :
  ], *unvisited*)
  unvisited ← *unvisited* − *visited*[−1]
 end
Function valueFunction(*ID1,ID2*):
  return $\frac{levels[ID1]+levels[ID2]}{sum(levels)distances[ID1,ID2]}$
End Function
Function
 knapsack(*MARBLECapacity*, *weights*, *values*, *unvisited*):
  K ← *matrix*(*len*(*weights*), *len*(*unvisited*))
  for *i in len*(*unvisited*) **do**
   for *w in len*(*weights*) **do**
    if *i = 0 or w = 0* **then**
     K[*i*, *w*] ← 0
    else if *weights*[*i* − 1] ≤ *w* **then**
     K[*i*, *w*] = *max*(*levels*[*i* − 1] + *K*[*i* −
    1, *w* − *weights*[*i* − 1]], *K*[*i* − 1, *w*])
    else
     K[*i*, *w*] = *K*[*i* − 1, *w*]
    end
   end
  end
  return *K*[*n, W*]
End Function

(a) Knapsack algorithm without rebalancing

(b) Knapsack algorithm with rebalancing for Monday of simulated week

FIGURE 8. Planned example routes using the Knapsack algorithm.

the Nearest Neighbor (NN) algorithm was implemented as well. Because this algorithm does not make use of smart LB information, two simple periodical emptying schedules are used, depending on the assumption of average filling levels. When making the optimistic assumption that every LB is at most 50% full, one MARBLE can empty four dustbins before its internal press needs to be emptied, due to its capacity of two full LB loads [68]. However, this would lead to the robot failing if the sum of the four consecutive LB filling levels exceeds 200%. Therefore, a pessimistic assumption has been implemented as well. In this case, each LB is assumed to be 100% full and emptying has to be performed after every second dustbin. Figures 8a and 8b show routes planned using the Knapsack algorithm, based on the full route and based on the rebalanced route for Monday of the simulated week.

It is clearly visible how the reduced number of stops declutters the planned route and has the potential to make the robot work more efficiently. Finally, the Knapsack algorithm

was also implemented for more than one agent. Therefore, the case of three robots simultaneously emptying the bins was tested as well. The way the Knapsack algorithm is implemented for this case firstly allocates route points to the first MARBLE agent until it has been assigned n/m bins, with n being the total number of bins along the route and m the number of MARBLE agents. Then, the algorithm proceeds to the next agent until all agents have been allocated their share of bins (in case the division is not even, rounding is applied).

The routes were then evaluated with respect to energy consumption and time taken. Regarding energy for the municipality baseline, this meant calculating the Diesel trip fuel consumption (based on 13.36l/100km [69] for the Mercedes-Benz Sprinter truck primarily used for these tasks). For the MARBLE case, the energy consumption included expenditure by the electric mothership as well as the robot itself, comprised of a constant component (load by computer, sensors, etc.) and dynamic components (locomotion and emptying process). The calculation of the time required an estimated average speed for both the municipality Sprinter tours and the MARBLE tours. The Sprinter tour speed was chosen based on an average walking speed of around 5km/h (or 3.085mph ≈ 3.1mph), averaged for females and males aged between 20 and 59 [70], which approximately represent BSR's workforce. This is the basis for the estimation, since the worker needs to walk to the LB, then from the LB to the rear of the vehicle to offload the collected trash, and finally back to the driver's seat for each emptying. The remaining time will be spent on moving around the park using the vehicle, where the maximum speed is limited, as well as

on operation time at each LB when the worker empties it. These two other factors are estimated to approximately cancel each other out, so that the average speed stays at approx. 5km/h. The MARBLE tour speed of 3.24km/h is based on the robot platform's maximum speed [71], as well as estimated duration for the processes of LB opening/closing, waste compression, waste offloading and battery swapping of 120s, 300s, 30s and 240s, respectively.

V. RESULTS AND ANALYSIS

A. EXPLORATORY DATA ANALYSIS

For the filling levels, different plots can be used to visualize their distribution (see Figures 9 and 10).
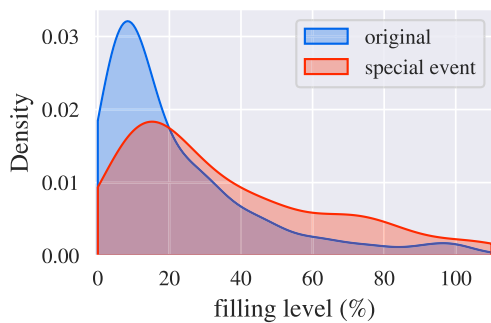
FIGURE 9. Distribution of filling levels (original vs. special event).

The majority of the filling levels can be found at the lower end, with a peak around 10%. It should be noted that the distributions extend past 100% filling level because some dustbins were occasionally overfilled. The average filling level for the original dataset is 21.74%. For the special event dataset, the average filling level increased significantly to around 35% (see Figure 10).
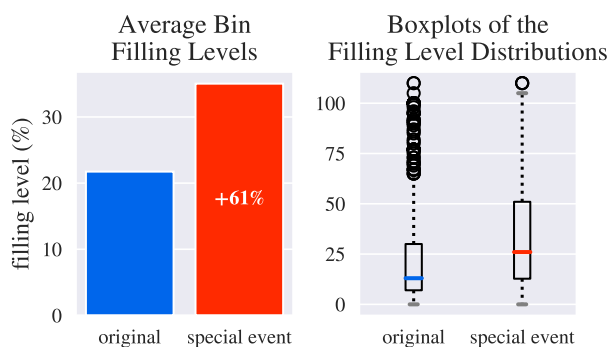
FIGURE 10. Characteristics of the normal and special event distributions in comparison.

The right skewed distributions in both datasets (see Figure 9) lead to class imbalance, as visible in Figure 5. It is more pronounced the more fine-grained the discretization into categories is. This property is inherent to the filling level feature and thus difficult and even undesirable to combat at the data collection stage in this application, as it would produce a biased representation of the real world conditions. However, the *Simulated Rebalancing* method relies on this

skewed real world distribution towards lower filling levels for its success. Methods to mitigate this uneven distribution at the ML stage are described in Section IV-B. All recorded values for each LB ID and their distribution in the original dataset can be visualized using a boxplot (see Figure 11).

A tendency for each LB can be observed. This indicates that there are explicit differences in how much a certain LB is frequented. In addition to the LB ID as the parameter varied, the same boxplots can be created for the weekdays (see Figure 12). Similarly, tendencies for particular days can be observed, for example elevated filling levels on Mondays.

B. ANALYSIS OF PREDICTED FILLING LEVELS

Figure 13 shows the results of the regression model grid search. The baseline has an RMSE of 20.96%. The best-performing model is an *XGBoost* regressor on a dataset in which the outliers were removed, but no feature selection was performed. Its RMSE is 15.535%, a 25.9% improvement from the baseline. Furthermore, four out of the top five models are decision tree-based. The results for the classification task with the *normal* rounding function can be seen in Figure 14. The baseline predictor achieves an accuracy of 75.9%. This value is comparatively high due to the imbalanced dataset (as visible from Figure 5). The best-performing predictor, an *XGBoost classifier*, achieves an accuracy of 80.6%. In addition to the *normal* rounding function, the performance of the algorithms applied to the dataset categorized using the *low-level* (15%) rounding function can be seen in Figure 15. The baseline predictor has a significantly lower performance, with an accuracy of 58.8%. This is due to the fact that the distribution of the two classes is more balanced in this case. The best model is an *XGBoost classifier* as well, with an accuracy of 81%. The tie between the *XGBoost* and *Random Forest* classifiers was broken using the models' F1 scores. Table 2 shows the entire classification report for this best performing model. The majority of the best-performing models in these two cases is also decision tree-based.

TABLE 2. Classification report for the XGBoost classifier.

	Precision	Recall	F1 score	Support
$\leq 15\%$	0.84	0.82	0.83	101
$> 15\%$	0.78	0.79	0.78	78
accuracy	0.81	0.81	0.81	0.81
macro avg	0.81	0.81	0.81	179
weighted avg	0.81	0.81	0.81	179

In addition to the shallow learning techniques, the performance of a Deep Neural Network (DNN) was tested as well.

However, the DNN classifier rapidly overfitted on the training set, even with a very simple structure, and was therefore not considered for further analysis. The superiority of tree-based models over neural networks on tabular data is a common observation in the field of ML. Grinsztajn et al. performed an extensive survey on this issue [72].
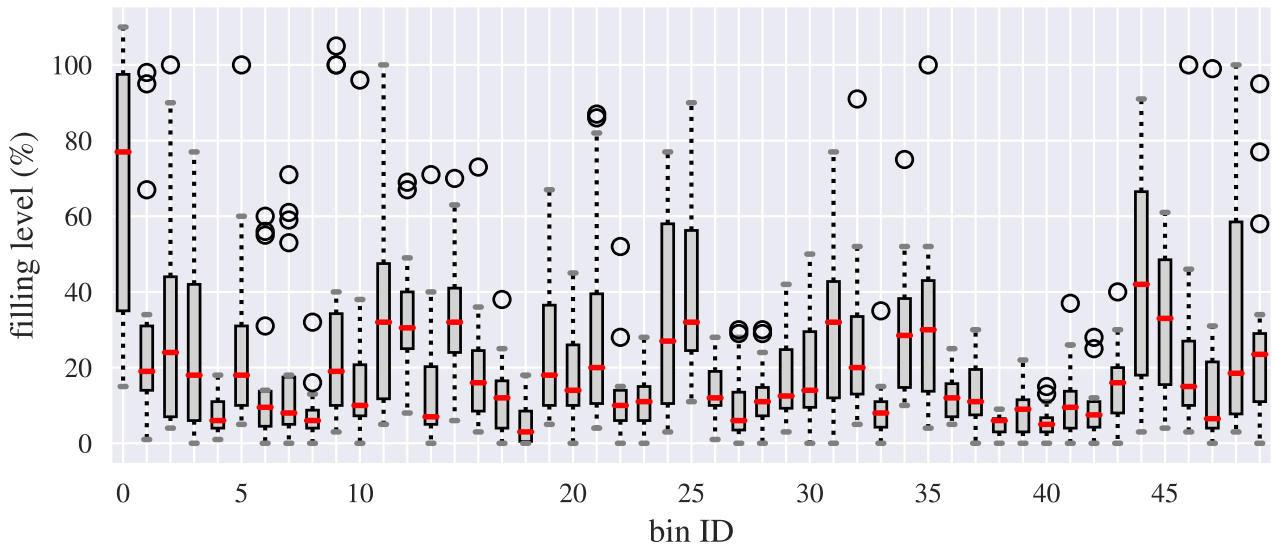
FIGURE 11. Boxplots for the recorded filling levels of each litter bin (skips in x-axis labels caused by missing IDs due to irregular numbering).
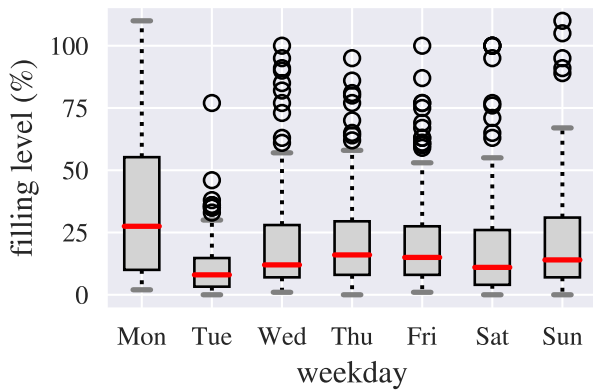

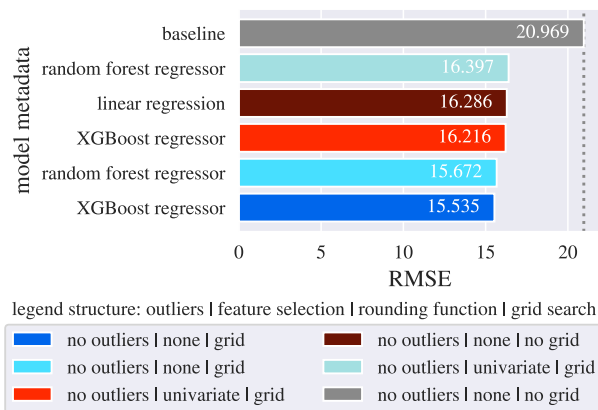**FIGURE 12.** Boxplots for the recorded filling levels of each weekday.

FIGURE 14. Accuracy of the 7 best-performing classification models (*normal* rounding function).

FIGURE 13. RMSE of the 5 best-performing regression models and baseline.

FIGURE 15. Accuracy of the 7 best-performing classification models (*low-level* (15%) rounding function).

C. ROUTE PLANNING PERFORMANCE EVALUATION

Figure 16 shows how the amount of low-level bins per weekday is reduced using *Rebalancing*. The average LB utilization in the park is raised and thus, fewer bins are below the threshold. The number of low-level bins is reduced by as much as 53%.
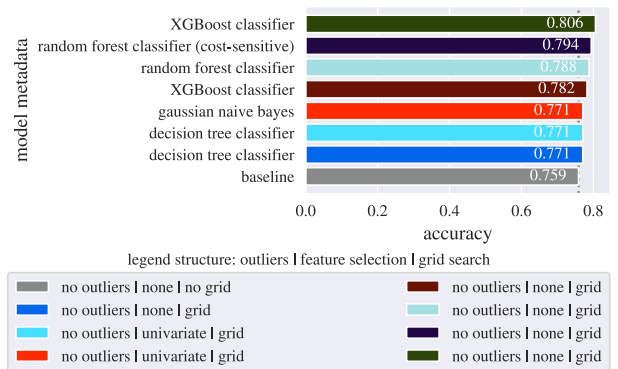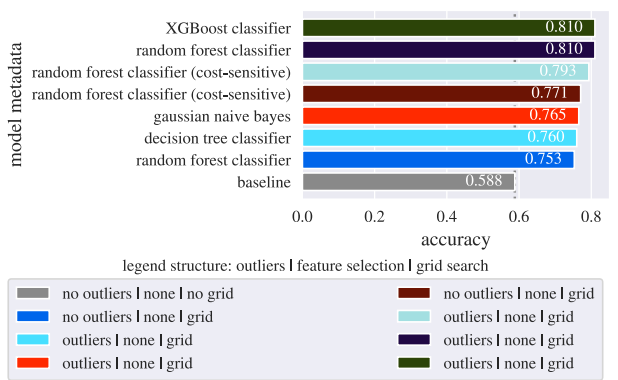
The results for route planning with respect to energy using the different algorithms can be seen in Figure 17. The share of the mothership's energy consumption is shown at each bar. The results with respect to time are depicted in Figure 18.

Furthermore, Figure 19 shows all route planning solutions for these representative weeks with respect to energy and
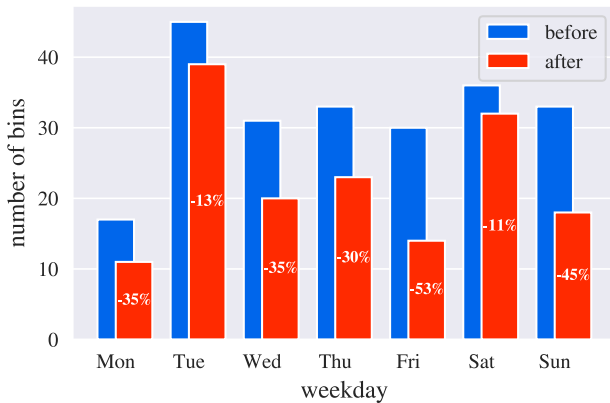
FIGURE 16. Number of low-level bins (≤ 25%) when applying *Rebalancing* (with reduction in number of low-level bins in percent).
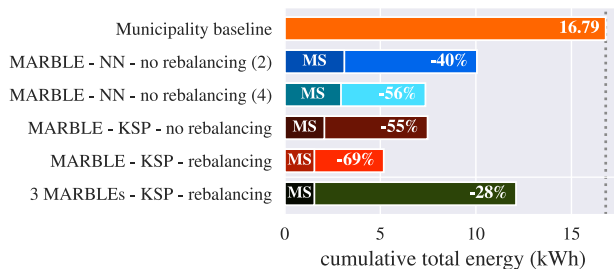

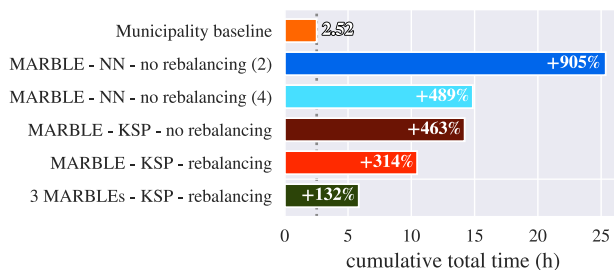**FIGURE 17.** Cumulative total energy over entire week.

FIGURE 18. Cumulative total time over entire week.

time in one plot. The municipality baseline tour with a Diesel vehicle requires by far the most energy. This is due to the low efficiency of Diesel engines, which is around 45% [73]. The MARBLE NN baselines and the KSP algorithm without *Rebalancing* already manage around a 50% reduction compared to the municipality baseline. The KSP algorithm with *Rebalancing* achieves a 69% reduction. Finally, the use of three MARBLEs in a swarm is also shown in the plot. Using three robots increases the energy consumption to only a 28% reduction against the baseline, however the proportion of energy the mothership consumes stays the same. Combining the *Simulated Rebalancing* technique with the KSP algorithm reduces its energy consumption by 30.67%. Furthermore, this combination achieves a 48% reduction over the pessimistic MARBLE NN baseline.

Regarding the total time for an entire week (Figure 18), the municipality baseline is quickest, with only around 2h 31min in total, which is equivalent to approx. 22min per day. The duration is increased more than nine-fold when using the
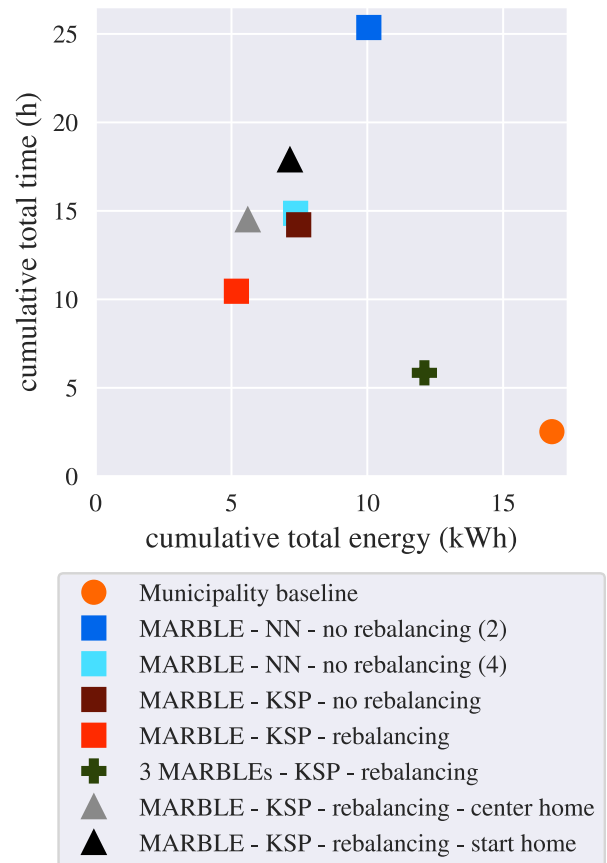

FIGURE 19. Energy and time for all solutions.

pessimistic MARBLE NN algorithm. However, the KSP algorithm combined with *Simulated Rebalancing* brings this value down to around 4.6 times the municipality value, or approx. 1.5h per day. This combination yields a reduction of 26.42% compared to the KSP algorithm alone, and a 59% reduction compared to the pessimistic MARBLE NN algorithm. Finally, the same algorithm with 3 robots is even more competitive with the baseline and only has a 132% increase in time.

In addition to all the baseline models mentioned previously, two more complement Figure 19. The current MARBLE concept always implies a mothership. For these two models, the mothership has been replaced by a home base concept, which fulfills the same tasks, but is immobile. Therefore, more energy is expended for return trips to the home base, but the base itself does not consume any energy for locomotion. The "center home" model uses LB ID 42 in the center of the park as the home base, while "start home" uses the starting point (LB ID 0) at the edge of the park (see the map in Figure 2). The choice of the center node has been optimized for lowest energy expenditure. The "center home" model compares relatively well with the MARBLE KSP rebalancing model regarding energy, but requires close to 50% more time. Meanwhile, the "start home" solution performs significantly worse in both dimensions.

D. REAL-WORLD VALIDATION

A real-world validation for the week between 30th October and 5th November 2023 was conducted. For that, the filling levels for each day were collected in the same way as for the original dataset, as described in Section IV-A. The week showed substantially elevated filling levels compared to the original dataset, at 35.68% versus 21.74%. The prediction performance of the binary *XGBoost* classifier trained on the main dataset (see Table 2) was evaluated on this new dataset, achieving an accuracy of 72.45%. This slightly lower performance is likely due to unknown factors such as special events (Halloween) and multiple unprecedented weather changes, which presumably also caused the elevated filling levels. Still, the classifier is able to maintain a high accuracy despite this. Then, *Simulated Rebalancing* and the route planning algorithms were applied (similar to the routes in Figure 8) and the routes were evaluated. The results are shown in Table 3 and are generally consistent with the previous results. The Table opposes the route planning results for the simulated week, as described in Section IV-D, and this newly collected week. It can be observed that, when applied to the KSP algorithm, *SR* works slightly less effectively, which is likely caused by the significantly higher average filling level. Similarly, this means that the advantage over the MARBLE NN baseline is reduced, since there is less potential to avoid unnecessary emptyings that the algorithm can leverage, because the bins are generally fuller. However, with 33.60% and 48.74% in energy and time, respectively, the savings are still substantial.

TABLE 3. Operational efficiency and prediction accuracy comparison of real-world validation.

Compared Scenario - MARBLE KSP with SR	Original Dataset	Tested Week
Energy consumption reduction against:		
MARBLE KSP no SR	30.67%	21.79%
MARBLE NN no SR	48.47%	33.60%
Time reduction against:		
MARBLE KSP no SR	26.42%	20.47%
MARBLE NN no SR	58.76%	48.74%
Prediction accuracy		
Best *low-level* (15%) classification accuracy	81.00%	72.45%

E. COMPARISON WITH LITERATURE

To evaluate this paper's results in the context of related works, it is sensible to conduct a short comparison with results from selected literature that is most closely related. For the prediction of filling levels, accuracies like 77.5% to 95% for the work by Alexopoulos et al. [46] are similar to our results for the classification task. However, it should be noted that their approach used a total of 10 classes versus 4 or 2 for this work, and they used a direct way of measuring filling levels through a deep neural network trained on image data of the bin, while our approach only relied on external features (weather, bin ID, etc.). References [45], [47], and [48] all only use regression models and evaluate them using the

Mean Absolute Error (MAE), which cannot be compared with the RMSE used in this work [74]. Furthermore, with daily/hourly item returns, [45] also uses an entirely different domain. However, these publications also show that waste prediction on a smaller scale using machine learning is possible. Regarding the route planning optimization, Ferrier & Alba were able to shorten the refuse collection routes by 33.2% over the traditional process [47], consequently achieving approximately the same time and energy reduction rates. This corresponds to our reduction rates of 48% and 59% in the energy and time domains over the NN baseline.

F. IMPLICATIONS

The elimination of unnecessary emptying of LBs based on the predicted filling levels proves to be an effective strategy, ensuring that the simulated fleet of MARBLEs utilizes its resources in an efficient and responsible way. Moreover, it has the potential to prevent costly investments in IoT sensors inside the LBs. Both implications support the objective of incorporating the sustainable development goals 11 (Sustainable cities and communities) and 12 (responsible consumption and production) [75] in urban waste management and robotics. The approach can also be applied for other SEAs or for combining various SEAs with a varying number of robots in a fleet. This work serves as a proof of concept that a service robot's efficiency can be improved through the employment of informed ML-based planning strategies. The strategy is likely to be transferable to other urban service robots such as for street sweeping, household waste collection and other maintenance applications. However, it will have to be further refined for a potential deployment.

VI. CONCLUSION AND OUTLOOK
A. SUMMARY

This research work developed a simulated data-driven route planning solution for optimizing the operational efficiency of a fleet of mobile robots in emptying the 49 bins in a park in Berlin. Initial data collection over 3 weeks highlighted distinct variations in LB filling levels based on LB ID and weekday, among other factors. While devices for measuring and communicating filling levels of bins have been developed and are currently available, challenges such as vandalism, high acquisition costs as well as operational and maintenance costs persist, which emphasizes the necessity to explore alternative solutions. Using machine learning techniques, we created predictive models to estimate the filling levels of the bins. The *XGBoost Regressor* achieved an impressive 15.54% Root Mean Square Error, a significant improvement over the initial 20.97%. This highlights the potential for accurate predictions based on collected data. Additionally, the *XGBoost Classifier* showed an 81% accuracy in binary categorization, surpassing the baseline accuracy of 58.8%. This underlines the possibility of optimizing route planning without the need for converting all conventional bins into smart LBs.

We introduced *Simulated Rebalancing* for efficient route planning, rescheduling bins below 25% capacity for consecutive emptying upon reaching this threshold. Combining it with the *Knapsack* algorithm reduced energy consumption by 31% and operational time by 26%. Compared to the *Nearest Neighbor* baseline, our KSP-*Simulated Rebalancing* method showed impressive improvements, reducing energy and time by 48% and 59%, respectively. Applied to a separately collected dataset, these methods still proved to be effective. The binary *XGBoost* classifier achieved an accuracy of 72.45%, while the KSP-*Simulated Rebalancing* method resulted in energy and time savings of 34% and 49%, respectively, over the NN baseline. The overall approach improved the efficiency of the bin-emptying process by responsibly utilizing the limited garbage storage capacity of the robots, emptying only those bins that require it and reducing the trips to the mothership.

B. LIMITATIONS AND OUTLOOK

Further focusing on autonomous LB emptying and waste management, we suggest the following improvements and extensions. A larger dataset will surely improve the predictive models. In addition, optimizing the synergy between prediction and route planning by using other algorithms (for example ant colony optimization) with the integration of *Simulated Rebalancing* can provide different energy and time efficient solutions. Currently, *Simulated Rebalancing* and the subsequent route planning is only implemented for one SEA, the Monbijoupark in central Berlin. Therefore, incorporating *Simulated Rebalancing* into more SEAs is suggested as an interesting field of research, as well as exceptional circumstances such as malfunctions of robots and their effects on the overall service. Also, the route planning was only conducted with either one or three MARBLE robots. Varying this number further will provide more insight into its resulting influence on the routes. Furthermore, optimizing the positions of the LBs based on the filling patterns will not just reduce unnecessary energy consumption of the robots, but also improve efficiency for municipalities by avoiding unnecessary LBs as well as combating overfilled LBs through installing more if required. Regarding the EDA, only univariate correlations were considered. Evaluating the multivariate correlations will yield more insight into what actually drives the filling levels. Furthermore, several other methods could be applied to the ML problem at hand. Examples include SMOTE for Regression [76] to further improve the regression results, cyclical feature encoding to capture the cyclical character of the time features (also over larger timespans such as seasons), Auto ML to explore more models, unsupervised ML, as well as the implementation of specialized DNNs for tabular data (see [77] for a comprehensive overview). Apart from these techniques, time-series models could also be considered as a more use case-specific ML technique. To address unexpected LB overfilling in the context of *Simulated Rebalancing*, the implementation of a public reporting system is recommended. Furthermore, investigating the interplay between advanced regression models and route planning can lead to improved overall operational efficiency. Conducting sensitivity analyses is crucial to anticipate and manage potential issues. This approach holds relevance not only in waste management, but also in services like freight collection, where disparities between communicated and actual volumes can significantly impact operational logistics and pricing strategies. Finally, given the limited space for trucks in fast-growing, densely populated areas, they could simply not be feasible anymore at a certain point. Thus, it is important to consider the replacement of trucks by robots while taking into account their impact in ecological, economic, and social dimensions equally.

Due to the complicated nature of autonomous services, which encompass tasks such as navigation and interaction with the surroundings, a careful and holistic development process as a smart product service system is crucial. However, further elaboration on these areas would be outside the scope of this paper.
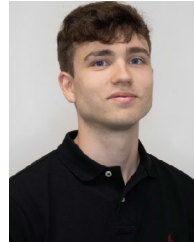
REFERENCES

[1] G. Gomathy, P. Kalaiselvi, D. Selvaraj, D. Dhinakaran, and D. A. Kumar, "Automatic waste management based on IoT using a wireless sensor network," in *Proc. Int. Conf. Edge Comput. Appl. (ICECAA)*, Oct. 2022, pp. 629–634.

[2] N. H. Kamarudin, A. A. A. Rahim, N. E. Abdullah, I. S. A. Halim, and S. L. M. Hassan, "Development of automatic waste segregator with monitoring system," in *Proc. 4th Int. Conf. Inf. Technol., Inf. Syst. Electr. Eng. (ICITISEE)*, Nov. 2019, pp. 190–195.

[3] R. Anitha, R. Maruthi, and S. Sudha, "Automated segregation and microbial degradation of plastic wastes: A greener solution to waste management problems," *Global Transitions*, vol. 3, no. 1, pp. 100–103, Jun. 2022.

[4] D. Rutqvist, D. Kleyko, and F. Blomstedt, "An automated machine learning approach for smart waste management systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 384–392, Jan. 2020.

[5] A. Medehal, A. Annaluru, S. Bandyopadhyay, and T. S. Chandar, "Automated smart garbage monitoring system with optimal route generation for collection," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Sep. 2020, pp. 1–7.

[6] A. M. F. Durrani, A. U. Rehman, A. Farooq, J. A. Meo, and M. T. Sadiq, "An automated waste control management system (AWCMS) by using Arduino," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Feb. 2019, pp. 1–6.

[7] ZenRobotics. (2023). *ZenRobotics*. [Online]. Available: https://www.terex.com/zenrobotics

[8] AMP Robotics. (2023). *Recycling Reimagined*. [Online]. Available: https://www.amprobotics.com/

[9] Bigbelly. (2023). *Public Space Waste Solutions*. [Online]. Available: https://bigbelly.com/solutions

[10] ENVAC. (2023). *Discover the Automated Waste Collection System by Envac*. [Online]. Available: https://www.envacgroup.com/how-it-works/

[11] V. Mai, B. Vanderborght, T. Haidegger, A. Khamis, N. Bhargava, D. B. O. Boesl, K. Gabriels, A. Jacobs, A. Moon, R. Murphy, Y. Nakauchi, E. Prestes, B. Rao R., R. Vinuesa, and C.-M. Morch, "The role of robotics in achieving the united nations sustainable development goals—The experts' meeting at the 2021 IEEE/RSJ IROS workshop [industry activities]," *IEEE Robot. Autom. Mag.*, vol. 29, no. 1, pp. 92–107, Mar. 2022.

[12] B. Gallenkemper, K. Gellenbeck, J. Breer, R. Reuter, M. Balhar, N. Portz, and R. Burger, "Handlungsempfehlung Zur Optimierung Der Kommunalen Stadtreinigung," *Stadt und Gemeinde INTERAKTIV*, vol. 1, pp. 1–28, Feb. 2007.

[13] Berliner Stadtreinigungsbetriebe. (2021). *Stattliche Leistung: Mit vollem Einsatz für ein Sauberes Berlin, 2021*. Accessed: May 30, 2023. [Online]. Available: https://www.bsr.de/assets/downloads/bsr_0067_017_Stattliche_Leistung_FL_105x210_barrierefrei_WEB.pdf

[14] Berliner Stadtreinigungsbetriebe. *Nachhaltigkeitsberichte der Berliner Stadtreinigung*. Accessed: Nov. 6, 2023. [Online]. Available: https://www.bsr.de/nachhaltig-wirtschaften-22371.php

[15] Technical University of Berlin. (2023). *Mobile Autonomous RoBot for Litter Emptying (MARBLE)*. [Online]. Available: https://www.tu.berlin/en/mpm/research/projects/murmel

[16] D. Göhlich, A. M. Syré, M. J. van der Schoor, D. Jefferies, A. Grahle, and L. Heide, "Design methodologies for sustainable mobility systems," in *Design Methodology for Future Products: Data Driven, Agile and Flexible*. Cham, Switzerland: Springer, 2022, pp. 123–144.

[17] J. Hautala, "Can robots possess knowledge? Rethinking the DIK(W) pyramid through the lens of employees of an automotive factory," *Humanities Social Sci. Commun.*, vol. 8, no. 1, pp. 1–10, Sep. 2021.

[18] J. Bräutigam, A. Gupta, and D. Göhlich, "Simulated annealing-based energy efficient route planning for urban service robots," in *Proc. 26th Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2022, pp. 294–299.

[19] A. Gupta, P. Kremer, S. Park, and D. Göhlich, "Intelligent route planning for autonomous service robots using communicating smart dustbins," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2022, pp. 1202–1207.

[20] A. Gupta, M. J. van der Schoor, J. Bräutigam, V. B. Justo, T. F. Umland, and D. Göhlich, "Autonomous service robots for urban waste management–multiagent route planning and cooperative operation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8972–8979, Oct. 2022.

[21] Google Maps. *Monbijoupark/James-Simon-Park Survey Route*. Accessed: Jul. 20, 2023. [Online]. Available: https://www.google.com/maps/@52.5228997,13.3973866,16.71z

[22] Berliner Stadtreinigungsbetriebe. *StraßEnreinigung in Berlin*. Accessed: Jul. 16, 2023. [Online]. Available: https://www.bsr.de/strassenreinigung-20471.php

[23] S. Guenat et al., "Meeting sustainable development goals via robotics and autonomous systems," *Nature Commun.*, vol. 13, no. 1, p. 3559, 2022.

[24] J. L. Kohl, M. J. van der Schoor, A. M. Syré, and D. Göhlich, "Social sustainability in the development of service robots," in *Proc. Design Soc., DESIGN Conf.*, vol. 1. Cambridge, U.K.: Cambridge Univ. Press, 2020, pp. 1949–1958.

[25] M. J. van der Schoor and D. Göhlich, "Integrating sustainability in the design process of urban service robots," *Frontiers Robot. AI*, vol. 10, 2023, Art. no. 1250697.

[26] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall Series in Artificial Intelligence), 3rd ed. Boston, MA, USA: Pearson, 2016.

[27] *Answer to 'What is the Real Difference Between Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP)*. Accessed: May 4, 2023. [Online]. Available: https://cstheory.stackexchange.com/a/41087

[28] F. Glover and K. Sörensen, "Metaheuristics," *Scholarpedia*, vol. 10, no. 4, p. 6532, Apr. 2015. [Online]. Available: http://www.scholarpedia.org/article/Metaheuristics

[29] J. Bräutigam, A. Gupta, and D. Göhlich, "Simulated annealing-based energy efficient route planning for urban service robots," in *Proc. 26th Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2022, pp. 294–299.

[30] V. B. Justo, "Real life scenario based sensor data monitoring of MURMEL and assisting infrastructure for efficient dynamic task allocation," Methods Product Develop. Mechtron., Berlin, Germany, Tech. Rep., Sep. 2022.

[31] IBM. *What is Exploratory Data Analysis? | IBM*. Accessed: Jun. 10, 2023. [Online]. Available: https://www.ibm.com/topics/exploratory-data-analysis

[32] L. Sage. *Machine Learning's Rise, Applications, and Challenges*. Accessed: Jul. 15, 2023. [Online]. Available: https://venturebeat.com/ai/machine-learnings-rise-applications-and-challenges/

[33] NBRO. *Answer to 'What is the Difference Between Deep Learning and Shallow Learning?* Accessed: Jul. 17, 2023. [Online]. Available: https://ai.stackexchange.com/a/21227

[34] IBM. *What is Supervised Learning? | IBM*. Accessed: Nov. 7, 2023. [Online]. Available: https://www.ibm.com/topics/supervised-learning

[35] Scikit-Learn Developers. (2023). *1.11. Ensemble Methods*. Accessed: Mar. 27, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html

[36] Scikit-Learn Developers. (2023). *Sklearn.ensemble.ExtraTreesClassifier*. Accessed: Mar. 14, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[37] Scikit-Learn Developers. (2023). *Sklearn.ensemble.AdaBoostClassifier*. Accessed: Apr. 27, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[38] NVIDIA Corporation. (Accessed: Jul. 16, 2023). *What is XGBoost?* [Online]. Available: https://www.nvidia.com/en-us/glossary/data-science/xgboost/

[39] M. Kulisz and J. Kujawska, "Prediction of municipal waste generation in Poland using neural network modeling," *Sustainability*, vol. 12, no. 23, p. 10088, Dec. 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/23/10088

[40] Z. Fan and Y. Fan, "Research on prediction of municipal solid waste production based on grey relation analysis and grey prediction model," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 300, no. 3, Jul. 2019, Art. no. 032070, doi: 10.1088/1755-1315/300/3/032070.

[41] M. R. Islam, G. Kabir, K. T. W. Ng, and S. M. Ali, "Yard waste prediction from estimated municipal solid waste using the grey theory to achieve a zero-waste strategy," *Environ. Sci. Pollut. Res.*, vol. 29, no. 31, pp. 46859–46874, Jul. 2022.

[42] G. Uganya, D. Rajalakshmi, Y. Teekaraman, R. Kuppusamy, and A. Radhakrishnan, "A novel strategy for waste prediction using machine learning algorithm with IoT based intelligent waste management system," *Wireless Commun. Mobile Comput.*, vol. 2022, Feb. 2022, Art. no. e2063372. [Online]. Available: https://www.hindawi.com/journals/wcmc/2022/2063372/

[43] S. H. Yusoff, U. N. K. A. Din, H. Mansor, N. S. Midi, and S. A. Zaini, "Design of smart waste bin and prediction algorithm for waste management in household area," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, pp. 748–758, Nov. 2018. [Online]. Available: https://ijeecs.iaescore.com/index.php/IJEECS/article/view/14548

[44] S. Ahmed, S. Mubarak, J. T. Du, and S. Wibowo, "Forecasting the status of municipal waste in smart bins using deep learning," *Int. J. Environ. Res. Public Health*, vol. 19, no. 24, p. 16798, Jan. 2022. [Online]. Available: https://www.mdpi.com/1660-4601/19/24/16798

[45] J. Walk, R. Hirt, N. Kühl, and E. R. Hersløv, "Half-empty or half-full? A hybrid approach to predict recycling behavior of consumers to increase reverse vending machine uptime," Mar. 2020, *arXiv:2003.13304*.

[46] K. Alexopoulos, P. Catti, G. Kanellopoulos, N. Nikolakis, A. Blatsiotis, K. Christodoulopoulos, A. Kaimenopoulos, and E. Ziata, "Deep learning for estimating the fill-level of industrial waste containers of metal scrap: A case study of a copper tube plant," *Appl. Sci.*, vol. 13, no. 4, p. 2575, Jan. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/4/2575

[47] J. Ferrer and E. Alba, "BIN-CT: Urban waste collection based on predicting the container fill level," *Biosystems*, vol. 186, Dec. 2019, Art. no. 103962. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303264718301333

[48] A. Camero, J. Toutouh, J. Ferrer, and E. Alba, "Waste generation prediction under uncertainty in smart cities through deep neuroevolution," *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 93, pp. 128–138, Dec. 2019.

[49] F. Vicentini, A. Giusti, A. Rovetta, X. Fan, Q. He, M. Zhu, and B. Liu, "Sensorized waste collection container for content estimation and collection optimization," *Waste Manage.*, vol. 29, no. 5, pp. 1467–1472, May 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0956053X08003735

[50] Y. Gao and A. Pollak, "Data collection and synthetic data generation for dustbin filling levels in the MURMEL project," Methods Product Develop. Mechtron., Tech. Univ. Berlin, Berlin, Germany, Project Rep., Apr. 2023.

[51] CustomWeather. *Syndicated Content for Complete Global Weather Coverage*. Accessed: Jul. 13, 2023. [Online]. Available: https://customweather.com/

[52] L. Sullivan and W. W. LaMorte. (May 2016). *InterQuartile Range (IQR)*. Accessed: Jun. 16, 2023. [Online]. Available: https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_summarizingdata/bs704_summarizingdata7.html

[53] K. Mahto. *One-Hot-Encoding, Multicollinearity and the Dummy Variable Trap*. Accessed: Jul. 14, 2023. [Online]. Available: https://towardsdatascience.com/one-hot-encoding-multicollinearity-and-the-dummy-variable-trap-b5840be3c41a

[54] D. Kumar. *Introduction to Data Preprocessing in Machine Learning*. Accessed: Jul. 17, 2023. [Online]. Available: https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d

[55] J. Brownlee. *Train-Test Split for Evaluating Machine Learning Algorithms*. Accessed: Jul. 12, 2023. [Online]. Available: https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

[56] Scikit-Learn Developers. (2023). *Sklearn.Feature_Selection.F_Regression*. Accessed: Nov. 8, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html

[57] Michael. *Using Principal Component Analysis (PCA) for Feature Selection*. Accessed: Aug. 10, 2023. [Online]. Available: https://stats.stackexchange.com/q/27300

[58] Foobar. (Mar. 2022). *How to Automatically Choose the Number of Components for PCA?* Accessed: Mar. 28, 2023. [Online]. Available: https://stats.stackexchange.com/q/396002

[59] T. Shin. (Feb. 2021). *Understanding Feature Importance and How to Implement it in Python*. Accessed: Mar. 14, 2023. [Online]. Available: https://towardsdatascience.com/understanding-feature-importance-and-how-to-implement-it-in-python-ff0287b20285

[60] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[61] The Imbalanced-Learn Developers. (2022). *SMOTEENN —Version 0.10.1*. Accessed: May 17, 2023. [Online]. Available: https://imbalanced-learn.org/stable/references/generated/imblearn.combine.SMOTEENN.html

[62] Google LLC. (2023). *Time Series Forecasting | TensorFlow Core*. Accessed: Oct. 31, 2023. [Online]. Available: https://www.tensorflow.org/tutorials/structured_data/time_series

[63] Scikit-Learn Developers. (2023). *3.2. Tuning the Hyper-Parameters of an Estimator*. Accessed: Mar. 27, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/grid_search.html

[64] Scikit-Learn Developers. (2023). *3.1. Cross-Validation: Evaluating Estimator Performance*. Accessed: Mar. 27, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

[65] J. Brownlee. (Jan. 2021). *Regression Metrics for Machine Learning*. Accessed: Apr. 14, 2023. [Online]. Available: https://machinelearningmastery.com/regression-metrics-for-machine-learning/

[66] Google LLC. *Directions API Overview*. Accessed: Jul. 12, 2023. [Online]. Available: https://developers.google.com/maps/documentation/directions/overview

[67] Scikit-Learn Developers. (2023). *Sklearn.metrics.pairwise.haversine_distances*. Accessed: Jul. 14, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.haversine_distances.html

[68] A. Gupta, P. Kremer, S. Park, and D. Göhlich, "Intelligent route planning for autonomous service robots using communicating smart dustbins," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 1202–1207.

[69] Spritmonitor. *Diesel Consumption: 13,36 L/100km—Mercedes-Benz, Sprinter, 513 CDI*. Accessed: Jun. 16, 2023. [Online]. Available: https://www.spritmonitor.de/en/detail/654866.html

[70] J. Fletcher. *Average Walking Speed: Comparisons by Age, Sex, and Walking for Health*. Accessed: May 10, 2023. [Online]. Available: https://www.medicalnewstoday.com/articles/average-walking-speed

[71] M. L. Urbanek, "Konzeptentwicklung eines Energiemodells und einer Leistungsverteilung für den autonomen Serviceroboter MURMEL," Bachelor's thesis, Methods Product Develop. Mechtron., Technical Univ. Berlin, Berlin, Germany, Feb. 2021.

[72] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?" Jul. 2022, *arXiv:2207.08815*.

[73] TÜV Nord Group. (2023). *Wirkungsgrade von Elektroautos | TÜV NORD Mobilität*. Accessed: Jun. 12, 2023. [Online]. Available: https://www.tuev-nord.de/de/privatkunden/verkehr/auto-motorrad-caravan/elektromobilitaet/wirkungsgrad/

[74] J. Christensen. (Jan. 2013). *Answer to 'Mean Absolute Error OR Root Mean Squared Error'?* Accessed: Oct. 31, 2023. [Online]. Available: https://stats.stackexchange.com/a/48268

[75] United Nations. (2023). *THE 17 GOALS | Sustainable Development*. Accessed: Oct. 30, 2023. [Online]. Available: https://sdgs.un.org/goals

[76] L. Torgo, R. Ribeiro, B. Pfahringer, and P. Branco, "Smote for regression," *Prog. Artif. Intell.*, vol. 8154, pp. 378–389, Sep. 2013.

[77] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–21, Jun. 2022.

ANTON POLLAK is currently pursuing the degree in mechanical engineering with the Technical University of Berlin. As part of the degree, he spent a semester abroad at The University of Melbourne, Australia. He is a Student Assistant with the Department Methods of Product Development and Mechatronics, Technical University of Berlin, in the framework of the research projects MARBLE and zeroCUTS II. His research interests include robotics, machine learning, and life cycle analyses in product development.

ABHISHEK GUPTA (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from the Bharati Vidyapeeth College of Engineering, Pune, India, in 2015, and the M.S. degree in energy efficient technical systems from TH Brandenburg an der Havel, Germany, in 2019. He is currently pursuing the Ph.D. degree in the operation management of a fleet of urban service robots with the Technical University of Berlin, with a focus on responsible energy consumption during the operational phase. Since October 2019, he has been a Research Associate with the Technical University of Berlin, for the project MARBLE. His research interests include the development of multiple-robot systems, operation management of autonomous systems, and integrating various aspects of sustainability in the design and operational phases of autonomous products.

DIETMAR GÖHLICH received the degree in mechanical engineering from the Technical University of Berlin and the Ph.D. degree from the Georgia Institute of Technology, Atlanta. Since 2010, he has been leading the Chair Product Development Methods and Mechatronics and the Director of the Institute for Machine Design and System Technologies, TU Berlin. Since 2015, he has been the Chairperson of the Research Campus Mobility2Grid, Berlin, bringing together 36 partners from industry and academia, with a focus on new sustainable mobility solutions for metropolises. Previously, he held a senior management position in the passenger car development with Daimler AG. His research interests include the development of new sustainable concepts for electric mobility, autonomous service robots, and the field of innovative methods in the product development process.

・・・