**Smart Waste Management System**

**using LSTM**

**Pericharla Raja Ramesh**

Department of MCA

Chaitanya Bharathi Institute of Technology,

Hyderabad, Telangana, India.

**rajaramesh7410@gmail.com**

**Dr. B. Indira**

Department of MCA

Chaitanya Bharathi Institute of Technology,

Hyderabad, Telangana, India.

**bindira\_mca@cbit.ac.in**

**Abstract**

A Smart Waste Management system is an enabling technology addressing the challenges of the waste transportation optimization. It will allow each recycling container reporting its filling level. The advanced functionality of such a system will enable predicting the expected emptying time of a recycling container, the time when the container’s filling level will achieve a certain critical value. Filling level predictions will allow avoiding redundant transportation without violating the overfilling requirement. However, the quality of filling level predictions will determine the efficiency of a Smart Waste Management system. There are several technical challenges for achieving a high-quality prediction. Our analysis of an operating Smart Waste Management system revealed that one of these challenges is a problem of an accurate detection of a container being emptied using the measurements from a sensor mounted on top of a container.

**Introduction**

In order to quickly address the real-world problems, the automated machine learning (Automated Machine Learning) approach has been proposed. This article extends the Auto ML approach with the data-driven methodology applied to industrial problems with existing solutions. Waste transportation and its optimization can significantly increase the positive effects. At the same time, there is a clear requirement that in order to keep recycling stations clean they should be emptied at a right time. It is important to fulfil this requirement in a scenario with several hundreds of recycling stations that are spread over a large geographical area.

**Literature Survey**

The success of machine learning in a broad range of applications has led to an ever-growing demand for machine learning systems that can be used off the shelf by non-experts. To be effective in practice, such systems need to automatically choose a good algorithm and feature preprocessing steps for a new dataset at hand, and also set their respective hyperparameters. Recent work has started to tackle this automated machine learning (AutoML) problem with the help of efficient Bayesian optimization methods. In this work we introduce a robust new AutoML system based on scikit-learn (using 15 classifiers, 14 feature preprocessing methods, and 4 data preprocessing methods, giving rise to a structured hypothesis space with 110 hyperparameters). This system, which we dub auto-sklearn, improves on existing AutoML methods by automatically taking into account past performance on similar datasets, and by constructing ensembles from the models evaluated during the optimization. Our system won the first phase of the ongoing ChaLearn AutoML challenge, and our comprehensive analysis on over 100 diverse datasets shows that it substantially outperforms the previous state of the art in AutoML. We also demonstrate the performance gains due to each of our contributions and derive insights into the effectiveness of the individual components of auto-sklearn**.** [1]

Many different machine learning algorithms exist; taking into account each algorithm's hyperparameters, there is a staggeringly large number of possible alternatives overall. We consider the problem of simultaneously selecting a learning algorithm and setting its hyperparameters, going beyond previous work that addresses these issues in isolation. We show that this problem can be addressed by a fully automated approach, leveraging recent innovations in Bayesian optimization. Specifically, we consider a wide range of feature selection techniques (combining 3 search and 8 evaluator methods) and all classification approaches implemented in WEKA, spanning 2 ensemble methods, 10 meta-methods, 27 base classifiers, and hyperparameter settings for each classifier. On each of 21 popular datasets from the UCI repository, the KDD Cup 09, variants of the MNIST dataset and CIFAR-10, we show classification performance often much better than using standard selection/hyperparameter optimization methods. We hope that our approach will help non-expert users to more effectively identify machine learning algorithms and hyperparameter settings appropriate to their applications, and hence to achieve improved performance. [2]

It explains when waste ceases to be waste and becomes a secondary raw material, and how to distinguish between waste and by-products. The Directive also introduces the "polluter pays principle" and the "extended producer responsibility". The foundation of EU waste management is the five-step “waste hierarchy”, established in the Waste Framework Directive. It establishes an order of preference for managing and disposing of waste. [3]

D. Rutqvist and F. Blomstedt are with the BnearIT AB, Lulea, Sweden. The system aims at optimising the waste management where a particular goal is to predict when a recycling container is going to be full. Each recycling container in the system is equipped with a sensor, which is mounted inside the container. The hardware is equipped with an ultrasonic range sensor, an accelerometer, and a GSM module. The server collects the statistics and performs predictive analytics of the data received from the sensors. Finally, the server delivers the extracted information to a user interface.

D. Kleyko is with the Department of Computer Science, Electrical and Space Engineering, Lulea University of Technology. Recall that the goal of a Smart Waste Management system is to predict emptying time, i.e., the time when a recycling container will be full enough to be emptied. In the considered system, it is defined that a recycling container should be emptied when its filling level reaches 90.0 %. The statistics gathered from the live deployments have shown that the filling rate mostly follows either a line or a simple polynomial function. Therefore, the system can predict the filling level by fitting a regression model to the measurements reported by the ultrasonic sensor. The fitted regression model is used to extrapolate the filling level in the near future. Thus, given the regression model it is possible to estimate the emptying time. A crucial prerequisite for this approach to function is that the regression model is built using only ultrasonic measurements obtained after the last emptying. It is clear that the fitted regression model does not have any predictive power since it fitted the model to two filling cycles. Note also that the other regression models fitted to a single filling cycle closely followed the sensory measurements (solid line). Therefore, this problem was identified as one of the key technical problems for the continued development of the considered Smart Waste Management system. The main measure, which should be taken to avoid such problematic situations, is the accurate detection of container emptying. Moreover, it is desirable to achieve the accurate emptying detection without complicating the sensor, i.e., by using measurements only from the ultrasonic sensor and accelerometer.

A simple solution to emptying detection would be a single threshold-based model where the values measured by either the ultrasonic sensor or accelerometer exceeded the threshold would lead to detection. There are, however, practical limitations for the use of such model. Due to the physical characteristics of the ultrasonic sensor, objects other than the actual waste level could be measured. For example, recycling containers usually have supporting structures or other parts related to the emptying mechanism, which interfere with ultrasonic pulses and, thus, create a false echo. Due to the false echo, the filling level will never reach zero even when the recycling container is actually empty. This fact invalidates the idea of measuring the absence of the waste for emptying detection. In the case of the accelerometer, since recycling containers are emptied by lifting them with a crane this event should trigger a single distinct vibration sample. However, in reality an emptying is not the only event, which generates vibration samples. Extra vibrations are often registered when waste is thrown in a recycling container. Thus, the use of accelerometer measurements could lead to many false detections. The above arguments explain why it is not feasible to use a simple threshold model for the accurate detection of emptying with either the ultrasonic range sensor or the accelerometer.

The existing manually engineered model to the emptying detection uses a set of static rules. These rules are applied every second hour to check for new vibration strength score (denoted as Vstr). The existing manually engineered model uses only the vibration strength score and the ultrasonic measurements (filling level) directly preceding and succeeding the vibration strength score in time. This makes the rules susceptible to minor measurement errors in the near proximity of an emptying. The filling level before a vibration strength score is denoted as F L1 and the filling level after is denoted as F L2. The filling level change, ∆F L, is calculated as ∆F L = F L2 − F L1. In order for a vibration strength score to be considered to be an emptying, Vstr must exceed six, i.e., more than five interrupts must have been registered. Next, the filling level must drop more than 10 % (i.e., ∆F L < −10 %) and the filling level after (i.e., F L2) must be below 10 %. Only in this case, the emptying will be detected otherwise the event will be treated as a non-emptying.

**Related Work and Methodology**

There are several technical challenges for achieving a high-quality prediction. So, we propose a data-driven approach for improving the quality of the emptying detection part of the system. Also considering the statistical methods with ML Algorithms. Statistical data is used in ML Algorithms for analysis. Previously they have used many conventional algorithms like Artificial Neural Networks (ANN), k-Nearest Neighbors (KNN), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Convolutional Neural Network (CNN).

As Time Stamp is an important parameter for this work, LSTM (Long Short-Term Memory) may perform very well as it is used for processing time series data. That means LSTM process entire sequences of data without treating each point in the sequence independently, but rather, retaining useful information about previous data in the sequence to help with the processing of new data points. LSTM dependent on three things:
▹ The current long-term memory of the network — known as the cell state
▹ The output at the previous point in time — known as the previous hidden state
▹ The input data at the current time step

LSTM consists of 3 gates: forget gate, input gate and output gate.

**Step**: **1**

* **Forget** **Gate**
* Here we will decide which bits of the cell state are useful given both the previous hidden state and new input data.
* Previous hidden state and the new input data are fed into a Sigmoid neural network.
* This network is trained so that, close to 0 when input is deemed irrelevant and closer to 1 when relevant.
* Output is sent up to pointwise multiplication so the irrelevant data is multiplied with 0 and relevant with 1.
* In summary, the forget gate decides which pieces of the long-term memory should now be forgotten.

**Step**: **2**

* **Input** **Gate**
* The goal of this step is to determine what new information should be added to the networks long-term memory.
* This step involves the **new memory network(***tanh***)** and the **input gate(***sigmoid***)**. (Both are NN)
* The both inputs previous hidden sate and input data are fed to above NN.
* The **tanh activated neural network** which has learned how to combine the previous hidden state and new input data to generate a ‘new memory update vector’.
* It doesn’t actually check if the new input data is even worth remembering.
* So, **sigmoid activated network** which acts as a filter, identifying which components of the ‘new memory vector’ are worth retaining.
* The output from these NN are sent to pointwise multiplication and added with previous state.

**Step**: **3**

* **Output** **Gate**
	+ The output gate decides the new hidden state, it takes newly updated cell state, previous hidden state and input data.
	+ Previous hidden state and new input data are set to the sigmoid NN.
	+ Newly updated cell state are fed to tanh non-NN.
	+ The above two points are sent to pointwise multiplication to get the new Output Hidden State.

**Final Step**

* Finally, we need to convert the hidden state to output.
* We actually need to apply a **linear layer** as the very last step in the LSTM process for only once.

**RESULT ANALYSIS**

The predicted results from LSTM model showed a greater accuracy as compared to other algorithms.







**CONCLUSION**

We don’t know inaccuracy detection of emptying affects filling level predictions. This model can be further optimized like by combinations of different algorithms or by use of some different kind of parameters. We cannot predict emptying time during festivals, elections, etc. because for that we need separate analysis. Odor sensor can be implemented to detect the odor from the waste.

**References**

[1] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, “Efficient and Robust Automated Machine Learning,” in Advances in Neural Information Processing Systems 28, 2015, pp. 2962–2970.

 [2] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, “AutoWEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms,” in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 847–855.

[3] European Commission, “Waste,” http://ec.europa.eu/environment/waste/, [Online; accessed 23-May-2018].

[4] T. Anagnostopoulos, A. Zaslavsky, K. Kolomvatsos, A. Medvedev, P. Amirian, J. Morley, and S. Hadjieftymiades, “Challenges and Opportunities of Waste Management in IoT-Enabled Smart Cities: A Survey,” IEEE Transactions on Sustainable Computing, vol. 2, no. 3, pp. 275– 289, 2017.

[5] “Bigbelly - Smart City Solutions,” http://bigbelly.com/, [Online; accessed 27-May-2018]. [Online]. Available: <http://bigbelly.com/>

[6] “Ecube Labs - Smart waste management solution,” https://www.ecubelabs.com/, [Online; accessed 27-May-2018]. [Online]. Available: <https://www.ecubelabs.com/>

[7] “Enevo - Waste and Recycling Services Right-Sized for You,” https://enevo.com/, [Online; accessed 27-May-2018]. [Online]. Available: <https://enevo.com/>

[8] “Sensoneo - Smart Waste Management solution,” [Online; accessed 27-May-2018]. [Online]. Available: <https://sensoneo.com>

[9] “Onlab - Onsense,” http://www.onlab.com.tr/en/onsense eng/, [Online; accessed 27-May-2018]. [Online]. Available: [http://www.onlab.com.tr/en/onsense eng/](http://www.onlab.com.tr/en/onsense%20eng/)

[10] “Smart Waste — Citibrain: Smart Integrated Solutions for Smart Cities,” http://www.citibrain.com/en/solutions/smartwaste/, [Online; accessed 27-May-2018]. [Online]. Available: http://www.citibrain.com/en/solutions/smart-waste/