A

Mini Project

On

GEO TRACKING OF WASTE AND TRIGGERING ALERTS AND MAPPING AREAS WITH HIGH WASTE INDEX

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

S.SAI DEEPA REDDY (207R1A0555)

S. KUSHAL DEEP (207R1A0553)

P. HRUDAY REDDY (207R1A0518)

Under the Guidance of

Mr. SVSV PRASAD SANABOINA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"GEO TRACKING OF WASTE AND TRIGGERING ALERTS AND MAPPING AREAS WITH HIGH WASTE INDEX"** being submitted by **S. SAI DEEPA REDDY (207R1A0555), S. KUSHAL DEEP (207R1A0553) & P. HRUDAY REDDY (207R1A0518)** partial fulfilment of the requirement for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, record of bonafide work carried out by them under our guidance and supervision during year 2023-2024.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mr. SVSV PRASAD SANABOINA
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful competition of this project.

We take this opportunity to Express my profound gratitude and deep regard To my guide **Mr. SVSV. Prasad Sanaboina**, Assistant professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, Help and guidance given by him cell carriers a long way in the journey of life on which we are about to embark.

We also take the opportunity to express a deep sense of gratitude to the project review committee (PRC) **G.Vinesh Shanker, Dr. J. Narasimharao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing of this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. we also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank Our family, for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

S. SAI DEEPA REDDY (207R1A0555)

S. KUSHAL DEEP (207R1A0553)

P. HRUDAY REDDY (207R1A0518)

ABSTRACT

This article presents the use of automated machine learning for solving a practical problem of a real-life Smart Waste Management system. In particular, the focus of the article is on the problem of detection of an emptying of a recycling container using sensor measurements. Numerous data-driven methods for solving the problem were investigated in a realistic setting where most of the events were not actually emptying.

The investigated methods included the existing manually engineered model and its modification as well as conventional machine learning algorithms. The use of machine learning allowed for improving the classification accuracy and recall of the existing manually engineered model.

This solution used a convolutional neural network classifier on a set of features based on the filling level at different given time spans. Finally, compared to the baseline existing manually engineered model, the best-performing solution also improved the quality of forecasts for the emptying time of recycling containers.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled “Geo Tracking Of Waste And Triggering Alerts And Mapping Areas With High Waste Index”. Develop a waste tracking system using machine learning and data analytics techniques to assist sanitation workers and users in making informed decisions regarding waste management. The system will leverage historical and real-time data to provide accurate data on waste collection and management.

1.2 PROJECT PURPOSE

The objective of this system is to ensure proper waste management and provide a detailed analysis of the waste generated in a community. We collect data from the sensor and build a model that is designed to predict the future values of the "Fill Percentage" of waste bins based on historical data. We are building a Machine Learning architecture that makes predictions based on historical waste fill percentage data. By using the above predictions, we can identify the high waste production area and perform the necessary actions to control it.

1.3 PROJECT FEATURES

The main features of this project are that this model classifies the It address the problem of learning hierarchical representation with a single algorithm or a few algorithms and Waste management models typically use a variety of features, or input variables, to make accurate predictions. The choice of features can vary depending on the various types of waste, region, and available data sources. There are different deep learning approaches like convolutional Neural Network(CNN) .

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is an important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays the important role of an interrogator and dwells deep into the workings of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "What must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The problem that the Indian Waste management sector is facing is the integration of technology to bring the desired outputs. With the advent of new technologies and the growth of population, the amount of waste generated has also increased rapidly. In order to manage the waste an advanced algorithm is proposed to perform accurate prediction and handle inconsistent trends in waste production various machine learning algorithms like CNN etc. can be applied to get a pattern. It will complement the Waste disposal system in India and together augment the ease of living for The urban and rural areas. In the past, many researchers have applied machine learning techniques to make this process efficient in the country.

2.2 EXISTING SYSTEM

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. However, it is possible to combine ultrasonic measurements and vibration strength scores in a more complex model with several thresholds. This idea was used to build the existing manually engineered model, which is presented in detail it was manually engineered using expert knowledge of the domain. It is also known that the detection performance of the existing manually engineered model is a bottleneck for improving the emptying time predictions. Therefore, we propose using the data-driven approach for improving the quality of the emptying detection part of the system. It should be noted that the acquisition of a dataset used in this article was done as a part of the implementation work in this study.

2.2.1 DISADVANTAGES OF THE EXISTING SYSTEM

The following are the disadvantages of the existing system:

- The problem is accurate detection of a container being emptied
- Low Maintenance
- Less Accuracy
- Low Efficiency
- High Cost

2.3 PROPOSED SYSTEM

The objective of this system is to ensure proper waste management and provide a detailed analysis of the waste generated in a community. We collect data from the sensor and build a model that is designed to predict the future values of the “Fill Percentage” of waste bins based on historical data. We are building a Machine Learning architecture that makes predictions based on historical waste fill percentage data. By using the above predictions, we can identify the high waste production area and perform the necessary actions to control it.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Improves the Efficiency of waste management system
- Reduce overflowing of Dustbins
- High Accuracy
- High Maintenance
- Reduce Pollution

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three considerations involved in the feasibility analysis:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the system's hardware components. The following are some hardware requirements.

- Processor : Intel i3 (or) Higher
- RAM : 4GB (or) Higher
- HDD : 500 GB (or) Higher

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specify the logical characteristics of each interface and software components of the system. The following are some software requirements:

- Operatin System : Microsoft Windows
- Python : PyCharm 2021.1.

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

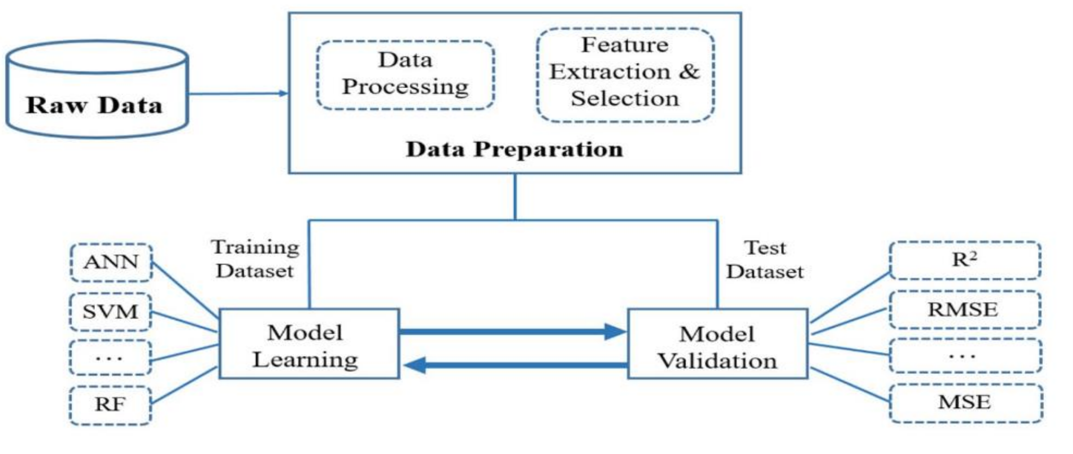


Figure 3.1: Project Architecture of Geo tracking of waste and triggering alerts and mapping areas with high waste index.

3.2 DESCRIPTION:

Waste transportation greatly affects both aspects and its optimisation 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 non-trivial to fulfil this requirement in a scenario with several hundreds of recycling stations (each with several containers) that are spread over a large geographical area.

3.3 USE CASE DIAGRAM

A use-case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. The roles of the actors in the system can be depicted.

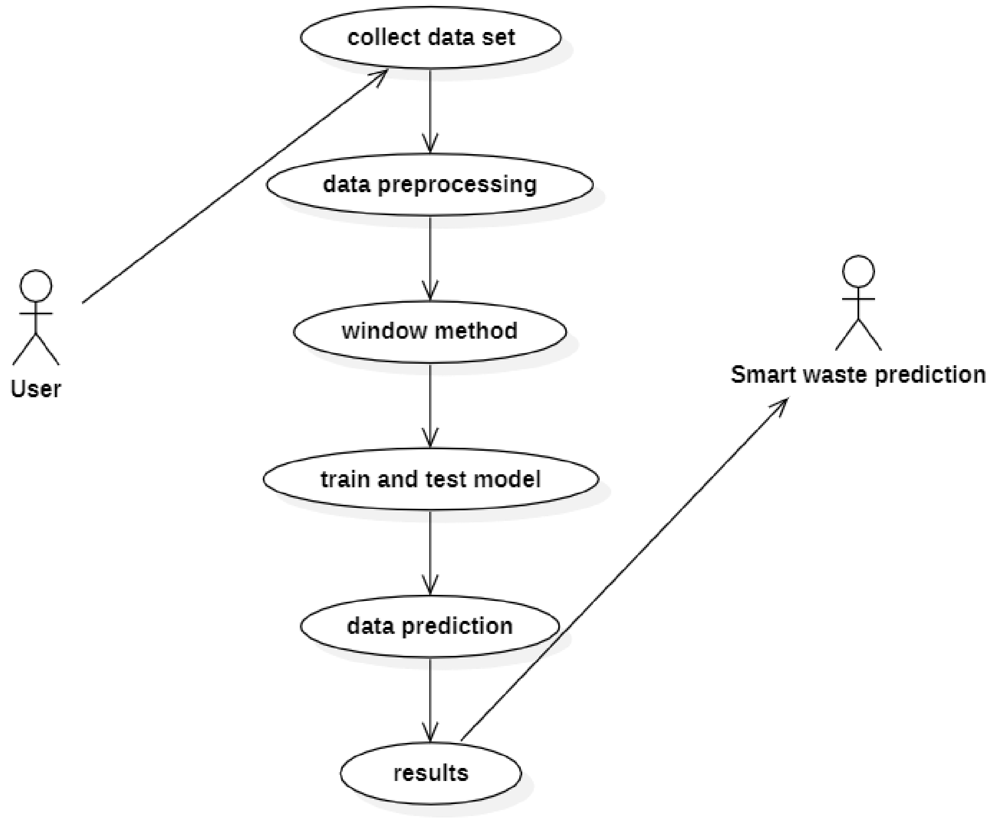


Figure 3.2: Use Case Diagram for Geo tracking of waste and triggering alerts and mapping areas with high waste index

3.4 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

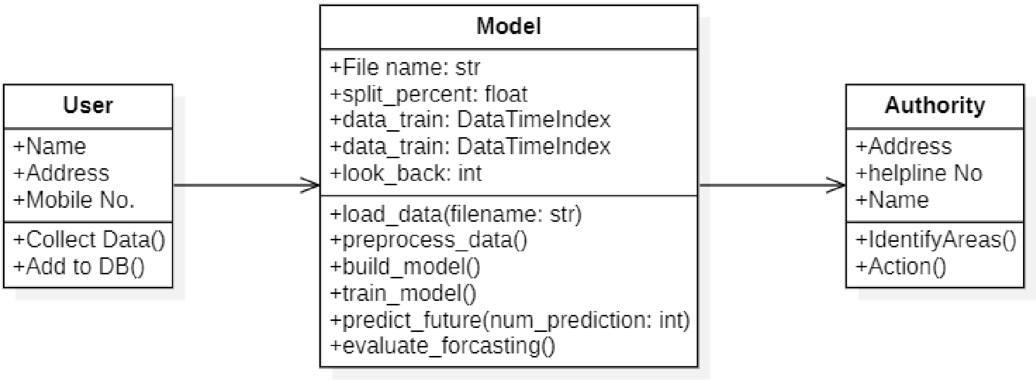


Figure 3.3 Class Diagram for Geo tracking of waste and triggering alerts and mapping areas with high waste index

3.5 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

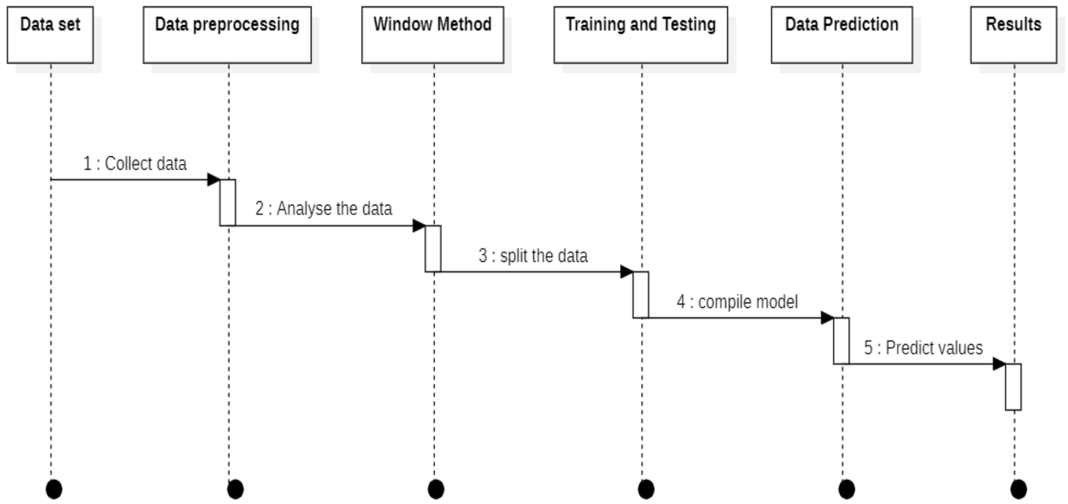


Figure 3.4 Sequence Diagram for Geo tracking of waste and triggering alerts and mapping areas with high waste index.

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

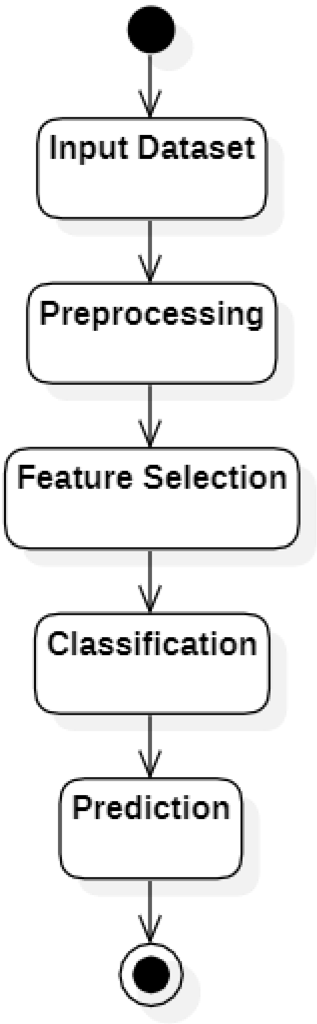


Figure 3.5 Activity Diagram for Geo tracking of waste and triggering alerts and mapping areas with high waste index.

4. IMPLEMENTATION

4.1 SAMPLE CODE:

```
import pandas as pd
import numpy as np
import keras
import tensorflow as tf
import matplotlib.pyplot as plt
from keras.preprocessing.sequence import TimeseriesGenerator
filename = 'D:/PYTHON/RAGURAM/smart/bin.csv' #load CSV file
df = pd.read_csv(filename)
print(df.info())
df['eventDate'] = pd.to_datetime(df['eventDate']) #convert to datetime(time series'd)
df.set_axis(df['eventDate'], inplace=True) #making date as index
df.drop(columns=['id', 'type', 'Battery', 'updateState', 'Distance'], inplace=True) #drop
columns not required for our modeling
df
plt.figure(figsize=(16,8))
df['FillPercentage'].plot()
plt.xlabel('Date')
plt.ylabel('Fill Percentage')
plt.show()
fill_data = df['FillPercentage'].values
fill_data = fill_data.reshape((-1,1))
split_percent = 0.80 #split data as 80% training set and 20% test set
split = int(split_percent*len(fill_data))
fill_train = fill_data[:split]
fill_test = fill_data[split:]
date_train = df['eventDate'][:split]
```

```
date_test = df['eventDate'][split:]

print(len(fill_train))

print(len(fill_test))

look_back = 15

train_generator = TimeseriesGenerator(fill_train, fill_train, length=look_back,
batch_size=20)

test_generator = TimeseriesGenerator(fill_test, fill_test, length=look_back,
batch_size=1)

from keras.layers import Dense,RepeatVector

from keras.layers import Flatten

from keras.layers import TimeDistributed

from keras.layers.convolutional import Conv1D

from keras.layers.convolutional import MaxPooling1D

from keras.layers import Dense

from keras.layers import Dropout

from keras.layers import LSTM

from keras.models import Sequential

model = Sequential()

model.add(Conv1D(filters=128, kernel_size=2, activation='relu',
input_shape=(look_back,1)))

model.add((MaxPooling1D(pool_size=2)))

model.add(TimeDistributed(Flatten()))

model.add(LSTM(50, activation='relu'))

model.add(Dense(1))

model.compile(loss='mse', optimizer='adam') #model compilation

model.summary() #model summary

num_epochs = 500

model.fit_generator(train_generator, epochs=num_epochs, verbose=1)

import plotly.graph_objects as go #library for visualization
```

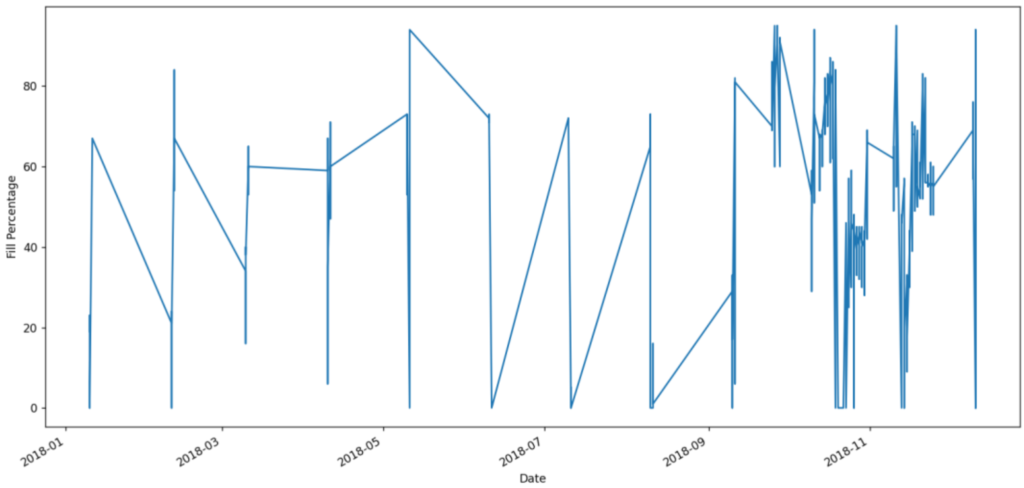
```
prediction = model.predict_generator(test_generator) #forecast/predict for test data
fill_train = fill_train.reshape((-1))
fill_test = fill_test.reshape((-1))
prediction = prediction.reshape((-1))
trace1 = go.Scatter(
    x = date_train,
    y = fill_train,
    mode = 'lines',
    name = 'Fill Percentage'
)
trace2 = go.Scatter(
    x = date_test,
    y = prediction,
    mode = 'lines',
    name = 'Forecasts'
)
trace3 = go.Scatter(
    x = date_test,
    y = fill_test,
    mode='lines',
    name = 'Ground Truth'
)
layout = go.Layout(
    title = "Waste Fill Levels",
    xaxis = {'title' : "Date"},
    yaxis = {'title' : "Fill Percentage"}
)
fig = go.Figure(data=[trace1, trace2, trace3], layout=layout)
fig.show()
```

```
plt.close()
fill_data = fill_data.reshape((-1))
def predict(num_prediction, model):
    prediction_list = fill_data[-look_back:]
    for _ in range(num_prediction):
        x = prediction_list[-look_back:]
        x = x.reshape((1, look_back, 1))
        out = model.predict(x)[0][0]
        prediction_list = np.append(prediction_list, out)
    prediction_list = prediction_list[look_back-1:]
    return prediction_list
def predict_dates(num_prediction):
    last_date = df['eventDate'].values[-1]
    prediction_dates = pd.date_range(last_date, periods=num_prediction+1).tolist()
    return prediction_dates
num_prediction = 30
forecast = predict(num_prediction, model)
forecast_dates = predict_dates(num_prediction)
future_forecast = np.append(prediction,forecast)
trace1 = go.Scatter(
    x = date_test,
    y = prediction,
    mode = 'lines',
    name = 'Test Forecast'
)
trace2 = go.Scatter(
    x = forecast_dates,
    y = forecast,
    mode='lines',
```

```
    name = 'Future Forecast'
)
layout = go.Layout(
    title = "Waste Fill Levels",
    xaxis = {'title' : "eventDate"},
    yaxis = {'title' : "Fill Percentage"}
)
fig = go.Figure(data=[trace1, trace2], layout=layout)
fig.show()
plt.close()
plt.figure(figsize = (15,10))
plt.plot(fill_test)
plt.plot(prediction)
plt.title('Actual vs Prediction plot')
plt.ylabel('waste fill levels')
plt.xlabel('eventDate')
plt.legend(['actual', 'prediction'], loc='upper left')
plt.show()
fig, ax = plt.subplots(1, 2,figsize=(11,4))
ax[0].set_title('predicted one')
ax[0].plot(prediction)
ax[1].set_title('real one')
ax[1].plot(fill_test)
plt.show()
def create_dataset(dataset, look_back=15):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back):
        a = dataset[i:(i+look_back)]
```

```
        dataX.append(a)
        dataY.append(dataset[i + look_back])
    return np.array(dataX), np.array(dataY)
testD = create_dataset(fill_test, 15)
def forecast_accuracy(forecast, actual):
    mape = np.mean(np.abs(forecast - actual)/np.abs(actual))  # MAPE
    me = np.mean(forecast - actual)             # ME
    mae = np.mean(np.abs(forecast - actual))    # MAE
    mpe = np.mean((forecast - actual)/actual)   # MPE
    rmse = np.mean((forecast - actual)**2)**.5  # RMSE
    corr = np.corrcoef(forecast, actual)[0,1]   # corr
    mins = np.amin(np.hstack([forecast[:,None],
                              actual[:,None]]), axis=1)
    maxs = np.amax(np.hstack([forecast[:,None],
                              actual[:,None]]), axis=1)
    return({'mape':mape, 'me':me, 'mae': mae,
            'mpe': mpe, 'rmse':rmse,
            'corr':corr})
forecast_accuracy(testD[1], prediction)
```

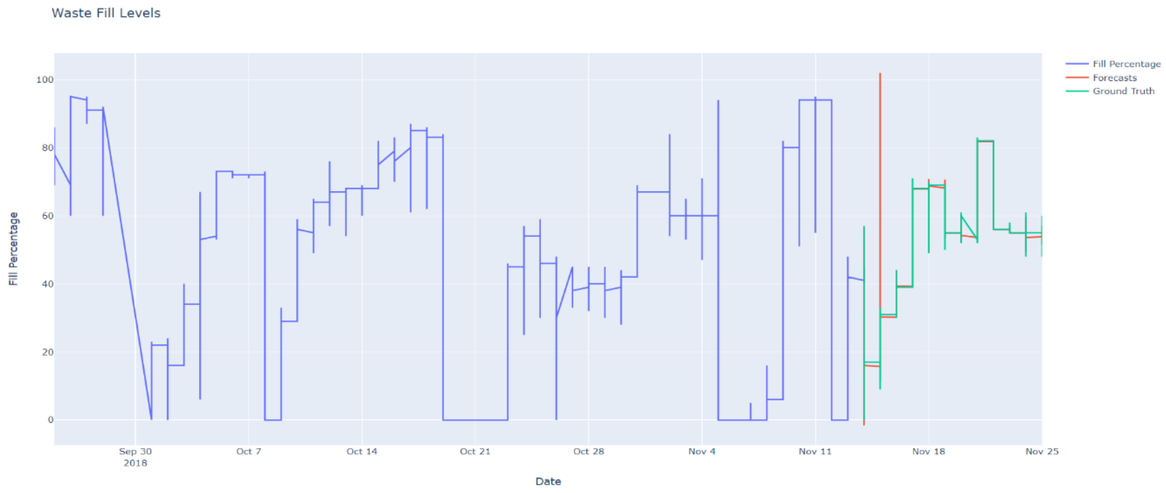
5.SCREENSHOTS



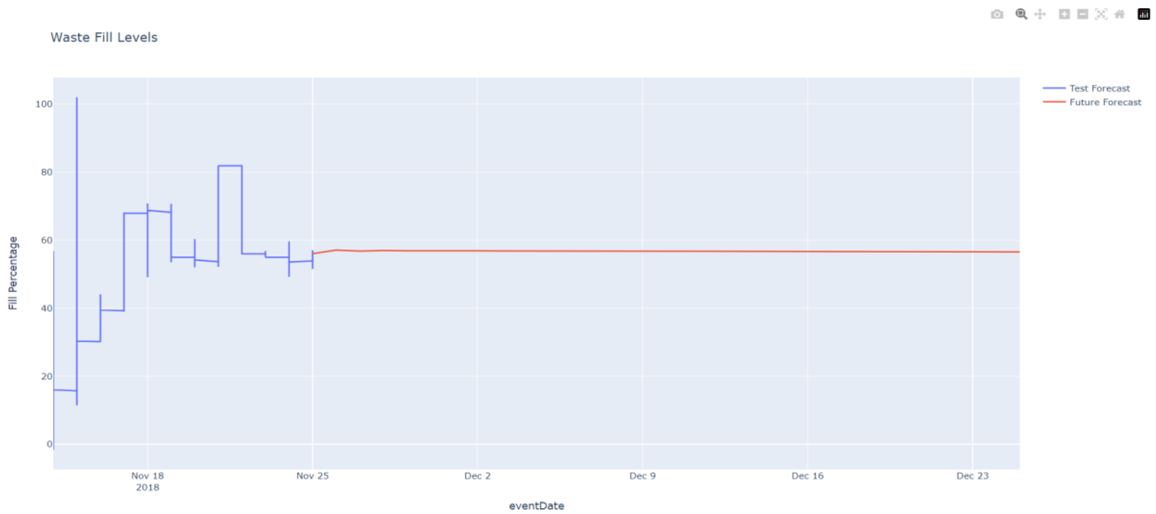
Screenshot 5.1 Representing The Dataset

```
Epoch 481/500
672/672 [==============================] - 4s 6ms/step - loss: 10.1703
Epoch 482/500
672/672 [==============================] - 4s 6ms/step - loss: 9.9172
Epoch 483/500
672/672 [==============================] - 4s 6ms/step - loss: 10.0141
Epoch 484/500
672/672 [==============================] - 4s 6ms/step - loss: 10.3377
Epoch 485/500
672/672 [==============================] - 4s 6ms/step - loss: 10.3947
Epoch 486/500
672/672 [==============================] - 5s 7ms/step - loss: 10.1261
Epoch 487/500
672/672 [==============================] - 5s 7ms/step - loss: 9.9964
Epoch 488/500
672/672 [==============================] - 4s 6ms/step - loss: 10.1201
Epoch 489/500
672/672 [==============================] - 4s 7ms/step - loss: 10.1389
Epoch 490/500
672/672 [==============================] - 5s 7ms/step - loss: 10.3013
Epoch 491/500
672/672 [==============================] - 5s 7ms/step - loss: 10.3567
Epoch 492/500
672/672 [==============================] - 4s 6ms/step - loss: 10.0315
Epoch 493/500
672/672 [==============================] - 4s 6ms/step - loss: 10.3921
Epoch 494/500
672/672 [==============================] - 3s 4ms/step - loss: 10.1955
Epoch 495/500
672/672 [==============================] - 5s 6ms/step - loss: 10.2887
Epoch 496/500
672/672 [==============================] - 5s 7ms/step - loss: 10.1346
Epoch 497/500
672/672 [==============================] - 5s 7ms/step - loss: 10.0731
Epoch 498/500
672/672 [==============================] - 4s 6ms/step - loss: 10.2509
Epoch 499/500
672/672 [==============================] - 3s 4ms/step - loss: 9.8039
Epoch 500/500
672/672 [==============================] - 3s 4ms/step - loss: 9.9887
```

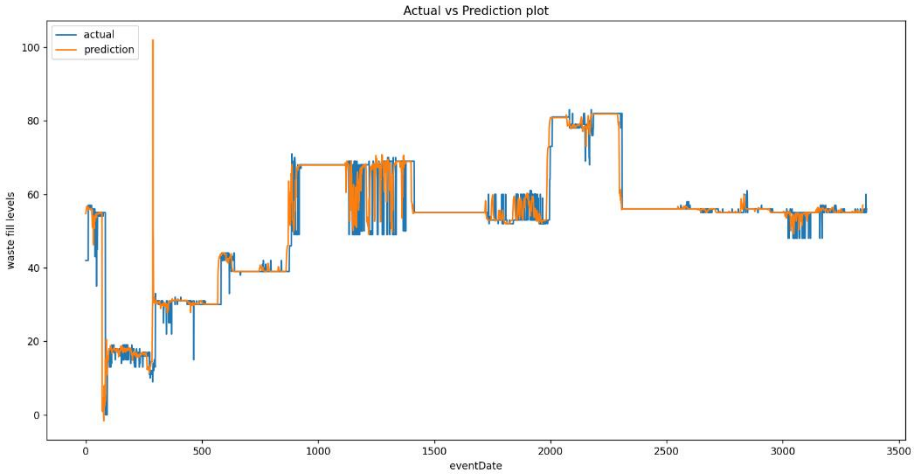
Screenshot 5.2 Training The Model



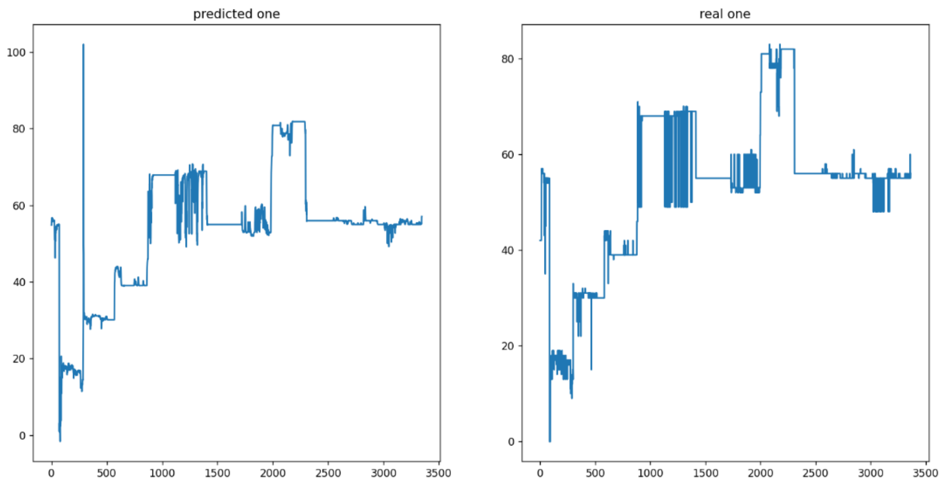
Screenshot 5.3 Waste Index Values



Screenshot 5.4 Waste Index Prediction



Screenshot 5.5 Actual Vs Prediction



Screenshot 5.6 Comparison of real data with predicted data

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING:

Integration tests are designed to test integrated softwareompone cnts to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components

6.3.3 FUNCTIONAL TEST:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	:identified classes of valid input must be accepted.
Invalid Input	:identified classes of invalid input must be rejected.
Functions	:identified functions must be exercised.
Output	:identified classes of application outputs must be exercised.
Systems/Procedures	:interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test case Id	Test case name	Purpose	Input	Output
1	Waste management prediction	To predict the waste generated	The user gives the input in the form of a data set.	An output is the date of the waste index predicted by the algorithm.
2	Waste management prediction	To predict the waste generated	The user gives the input in the form of data set.	An output is the data of the waste index predicted by the algorithm.

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

This article discusses the application of automated machine learning in industrial informatics, specifically for accurately detecting when a recycling container is empty using sensor data. The article outlines a four-step iterative data-driven methodology, which involves assessing the existing solution, optimizing it with collected data, applying machine learning algorithms, and exploring additional features through feature engineering. However, the study has limitations, including not quantifying the impact of inaccurate emptying detection on filling level predictions, assuming the availability of specific data, and not considering computational complexity. The investigation examined various solutions, including manually engineered models, optimized versions, conventional machine learning algorithms, and extended feature versions.

7.2 FUTURE SCOPE

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. The results demonstrated that the detections of the best performing solution improved the emptying time predictions by 14:2 percent compared to the detections of the existing manually engineered model. It is concluded that the best performing solution is applicable for a practical deployment in the production system if the requirement on the availability of the measurements preceding and succeeding the point of interest by twelve hours is relaxed. The achieved results emphasise the importance of the data driven engineering for the design of Smart Waste Management systems.

8. BIBLIOGRAPHY

8.1 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, "Auto- WEKA: 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

8.2 GITHUB LINK

https://github.com/DeepaSrinivas/Project