Regularized Noise based GRU Model to Forecast Solid Waste Generation in the Urban Region

[1]Rashmi G, [2]Dr. S Sathish Kumar

[1]Research Scholar, Visvesvaraya Technological University, Belagavi, India
[2]Professor & HoD, RNS Institute of Technology, Bangalore, India
[1]rashmi4mahendrakar@gmail.com, [2]s.sathishkumar@rnsit.ac.in

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Today's Solid Waste Management System needs better strategies to perform effective operations. The speed at which waste is being generated in urban regions is directly affecting the surrounding environment impacting the health of living beings. Advancements in deep learning has made time series forecasting very efficient making better predictions. Recurrent Neural Network (RNN) is a deep learning approach which helps in analyzing and forecasting time series data by holding the complication of sequence dependencies. In this study, a RNN model, Regularized Noise based Gated Recurrent Unit (RNGRU) has been proposed to analyze and predict solid waste generation in Australia. The dataset is collected from www. oecd.org which has waste data 20780 records from 1990 to 2018. Various input features such as Region name, Year, Waste generated in tonnes and technical indicators such as Moving Average (MA), Exponential Moving Average (EMA), Moving Average Convergence/Divergence (MACD), Relative Strength Index (RSI), On Balance Volume (OBV), Momentum (MTM), Daily waste variation are used to train the model. The proposed RNGRU model is compared with LSTM model and the best model is chosen based on the performance metrics, Mean Absolute Error(MAE), Mean Squared Error (MSE), and Root Mean Squared Error(RMSE). The outcome of the experiments shows that the RNGRU model is best for prediction with low error rates, MAE value 0.0147, MSE value 0.0010 and RMSE value 0.0900 respectively.

Keywords: Artificial Neural Network, Recurrent Neural Network, Machine Learning, Deep Learning, Gated Recurrent Unit, Long Short Term Memory

1. Introduction

Management of solid waste [3] in a smart city has become a serious issue in today's life. It is a result of increasing population and industrialization. Solid waste comprises of food waste, sanitary waste, construction and demolition waste, industrialization waste, commercial waste etc. These waste includes plastics, vegetable peels, papers, metals, glasses, batteries, paints etc.

There are numerous elements contributing to the growth of solid waste such as living standards, food habits, various commercial activities, construction and demolition waste [4], population etc. The human greediness, irresponsibility in the public, the way wastes are being generated, accumulated and handled is contributing to waste generation affecting public health polluting the living environment. This can be prevented and existing waste management [12] can be improved only by the correct forecasting of solid waste growth [5]. It is very much essential to understand the nature, composition and amount of waste is being generated which helps in better planning of the waste management system [8]. Predictive models play very important role in handling such challenges by taking care of insufficient and uncertain data.

The contributions of the proposed work are:

1. Proposed RGNRU to forecast solid waste generation in smart cities.
2. Comparison of LSTM model with proposed technique.
3. Best model is chosen based on the performance metrics MAE, MSE and RMSE.

2. Organization of the template

The organization of the paper is as follows: Section 3 explains about related work. Section 4 explains the general approach used to conduct this research work. Section 5 explains deep learning techniques. Section 6 discusses the Experimental Evaluation. The conclusion is discussed in section 7.

3. Related work

Various research experts collected historical data and predicted the rate of future solid waste generation [2]. Since prediction depends on several factors [6] such as literacy, age group, employment, type of residence, geography, climate and can't be done straight forward, suitable modelling approaches are needed. To analyse the relationship between various factors or variables, few earlier research scholars employed regression and time series models. Ghinea et al. [7] employed these models to predict waste generation in Lasi, Romania. The input variables used for predicting waste generation are population, life expectancy, volume of waste generated and number of inhabitants.

Noori et al. [13] used Artificial Neural Network and multivariate linear regression models for short term prediction in Tehran. The input features used for prediction are weekly waste generated data in tonnes and number of trucks used to carry the waste per week. The experiments revealed that ANN performs better than multivariate linear regression model. Antanasijevic et al. [1] used ANN model to forecast waste growth in Bulgaria and Serbia revealing good prediction.

K.A. Kolekar et al. [9] presented a review of various models employed by various researchers to forecast municipal waste growth using socio-economic, demographic data and identified several factors which help in choosing the critical design options in mathematical modelling.

Miyuru Kannangara et al. [10] employed Decision trees and Neural Network techniques to predict waste generation in Canada. They collected a dataset from 220 municipalities in the province of Ontario, Canada and considered social, economic and demographic parameters while developing models. Results revealed that these models are fit for good predictions.

Soni U et al. [13] used different models such as ANN, ANFIS, DWT -ANN, DWT-ANFIS, GA-ANN and GA-ANFIS to forecast the waste generation in India. They used Delhi, India dataset to train and analyze these models. The best model has been chosen based on the performance metrics such as RMSE, R^2 and IA. The hybrid model of genetic algorithm and ANN were concluded to be the best models with lesser RMSE and higher IA and R^2.

Solano Meza JK [14] presented an analysis of three ANN models, Decision tree, SVM and simple RNN used to predict solid waste growth in urban city of Bogota. The experiment revealed that SVM is the most appropriate model.

Kulisz et al. [11] employed ANN for proving the forecasts concerning to the amount of rain fall in Poland. The employed model used several explanatory variables to reveal the effect of social, economic and demographic features on the quantity of waste produced. MSE and R metrics have been used to measure the productivity of the models. The experimental results revealed that ANN is effective in forecasting the waste generation and is a cost effective approach.

Yinghao Chu et al. [15] proposed a multilayer hybrid deep learning automated technique to categorize the waste thrown by public in the urban region. This research work used camera to arrest waste images and sensors to identify the required features. Multilayer Perceptron (MLP) technique is employed to classify recyclable waste from other wastes. The experimental results prove that hybrid deep learning technique could achieve more than 90% accuracy.

4. Work Flow of the Current Research

Time series forecasting is becoming very strong due to the advancements in Deep learning techniques. It uses time as an input variable and changes the variable analyzed over a period to foresee future observations. The objective of time series forecasting is to predict future using past data. Technical indicators help in understanding how much previous data is needed to perform computations to suit the future needs. Figure 1. shows the step by step procedure carried out in the current research work. It deals with four stages. Namely, Data Collection, Data Preparation, Model Creation and Performance Evaluation.

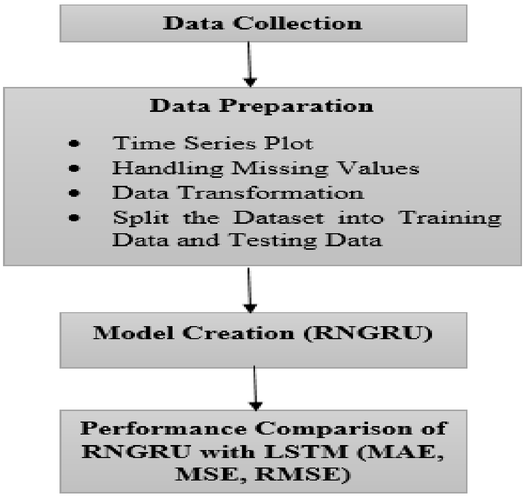
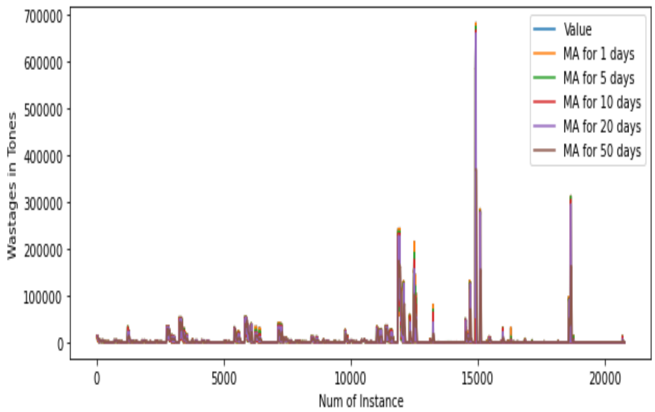


Figure 1: Work flow of the Research

4.1 Data Collection: Raw dataset to carry out this study is collected from www.data.oecd.org. It contains 20000 records, 80% of the records is used in training phase and 20% of the records in testing phase.

4.2 Data Preparation: In this stage, collected raw data is analysed, checked and modified to extract only the required features to develop the models. Following activities are performed on the raw data to segregate irrelevant features from the relevant features.

Time series plot: used to analyse the data using visualisation tools. Technical Indicators here are used to analyse the collected historical dataset and predict future performance. Technical indicators such as MA (see Figure 2), EMA (see Figure 3), MACD (see Figure 4), OBV (see Figure 5), RSI (see Figure 6), Momentum (see Figure 7), Daily waste variation (see Figure 8) are used to train the model.



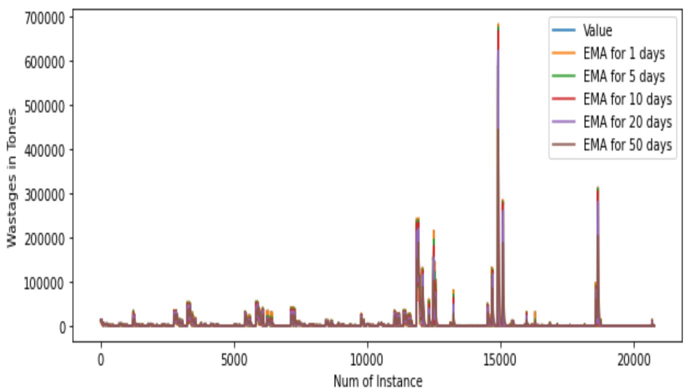
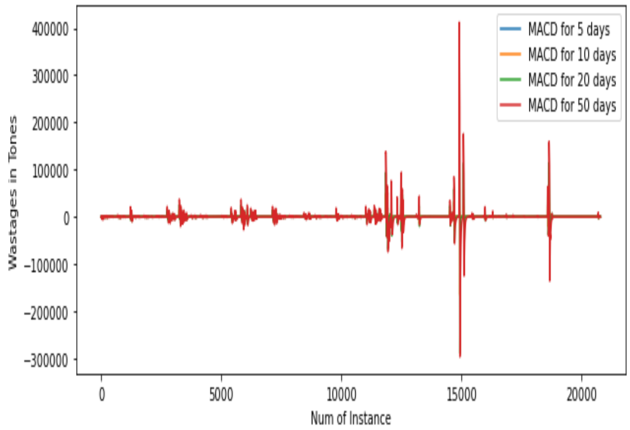


Figure 2: Moving Average Technical Analysis   Figure 3: Exponential Moving Average Technical Analysis



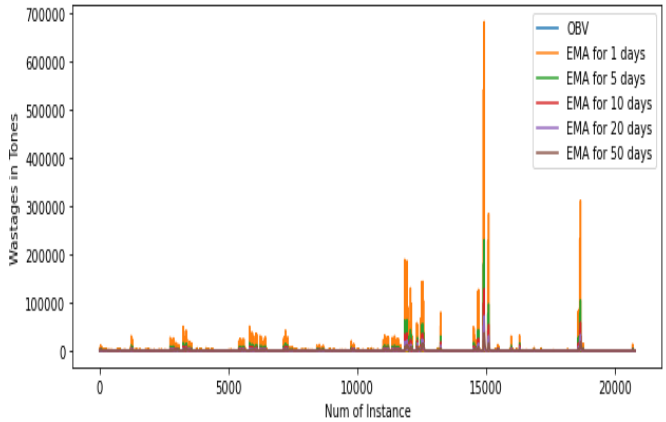
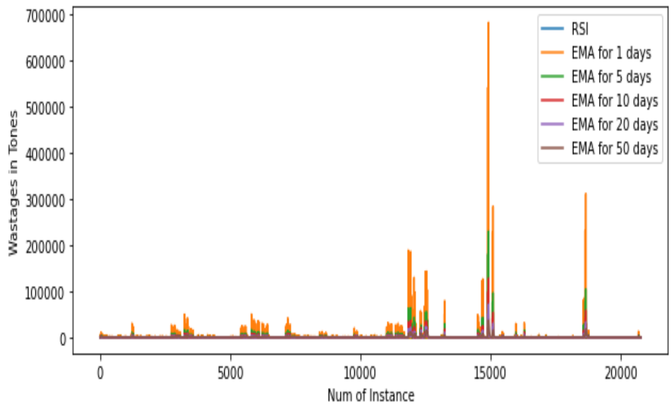


Figure 4: MACD Technical Analysis

Figure 5: OBV Technical Analysis



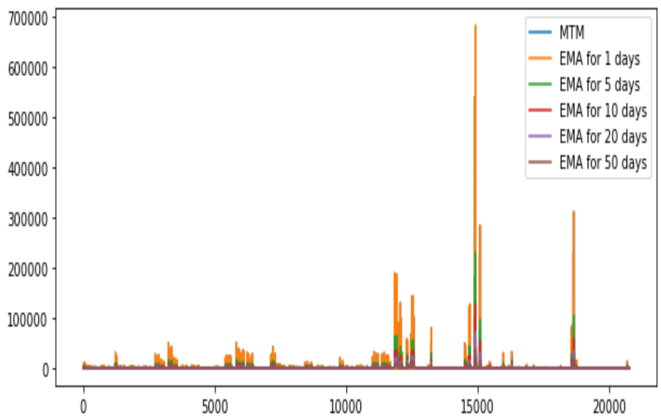


Figure 6: RSI Technical Analysis

Figure 7: Momentum

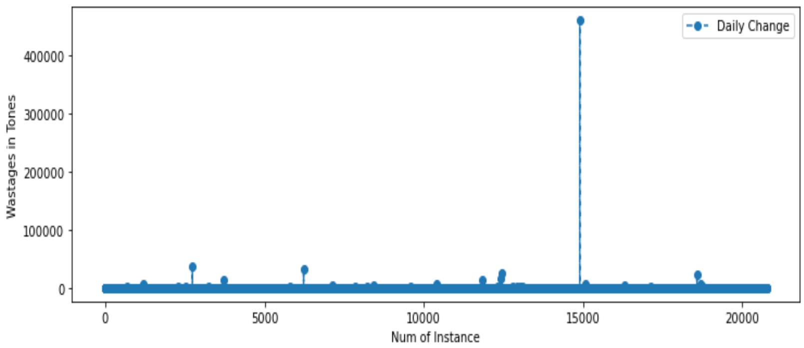


Figure 8. Daily change in solid waste

Handled missing values: this is done by removing irrelevant rows and columns.

Data Transformation: Data collected contains categorical as well as numerical data and hence label encoding is performed on the categorical data to achieve uniform data distribution.

- Divided the dataset into training data and testing data: Final data is divided into training data (80%)and testing data (20%).
- Create input: Data obtained in the previous phase is used as an input for the models discussed in the next section.

4.3 Models Creation: Proposed RNGRU and LSTM models are trained using training data tested later using testing data.

4.4 Evaluation of Model's performance: The models are compared and finest model is selected based on the metrics, MAE, MSE and RMSE.

5. Deep Learning Techniques

5.1 Long Short Term Memory (LSTM) Model

RNNs are well suited for making predictions but they suffer from the problem of short term memory wherein they find hard to carry the info from the former steps to later ones if a sequence of data is long enough. To overcome this, LSTM and GRU were created as a solution using a mechanism of gates. LSTM uses the concept of gates to control the flow of information. Figure 9. Shows the structure of LSTM having following components.

1. **Cell State (Memory Cell):** takes care of remembering and forgetting information.
2. **Gates:** determine whether to keep the information or discard it. It uses sigmoid functions and squishes the values between 0 and 1 which helps in forget or update gates. Numeric 0 indicates value to be forgotten and 1 indicates value to be retained.

Forget Gate: It determines what info to retain or discard. It takes the info from the current hidden state and info from the prior hidden state and passes them to the sigmoid function which outcomes in either 0 or 1.

Input Gate: It is used to update memory cell. It uses earlier hidden state and present information into sigmoid which results in either 1 or value between 0 and 1.0. Numeric 1 means keep the information and value between 0 to 1.0 means forget the information. Then, tanh function is used to regulate the network resulting values between -1 and 1. Then, the output is multiplied with sigmoid output which decides which info is to be retained. This enough information available is enough to compute the cell state which gets multiplied by the forget vector followed by pointwise addition which updates and gives new cell state.

Output Gate: It decides what should be the following hidden state. Hidden state holds info on former inputs which help in forecasting.

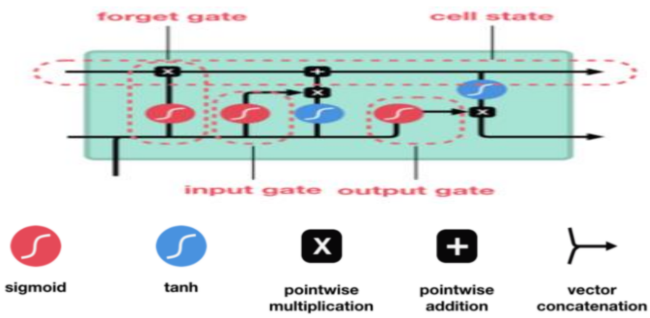


Figure 9. Structure of LSTM

Pseudo code: LSTM

1. ***Input:*** *PreProcessed Wastage Data.*
2. ***Output:*** *Predicted Wastages.*

3. *Intialize the Network Layers [LSTM, Softmax, Dense] and set the Parameters*
4. *batch_size ← 10*
5. *train_ratio ← 0.7*
6. *epochs ← 10*
7. *runs ← 20*
8. *Normilize_data← [data]*
9. *L← Softmax Activation*
10. *regularizers ← [None, L]*
11. *units ← [64, 128, 256]*
12. train, test ← split_data (Normilize_data, train_ratio)
13. for xi in X do

a. for regularizer in regularizers do

i. for unit_number in units do

ii. i ← 0

iii. results ← [ ]

iv. while i < runs do

1. i ← i+ +
2. model ← build_model (xi, regularizer, unit_number)
3. model.train(train)
4. mse ← model.predict (test)
5. results.append (mse)

14. mean ← get_mean (results)

5.2 Proposed Regularized Noise based GRU (RNGRU)

Gated Recurrent Unit (GRU) is a new generation of RNNs pretty similar to LSTM. It eliminated cell state and hidden states of LSTM. It has only

two gates shown in Figure 10. Update Gate, which behaves similar to forget and input gate of LSTM. It determines which information to retain or discard as well as what novel information to be added. Then comes, Reset Gate, it decides how much of previous information to disremember. It can be trained fast compared to LSTMs.

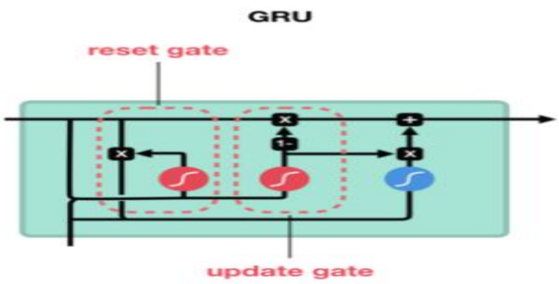


Figure 10. Structure of GRU

Regularization in GRU

Regularization is a process where in learning algorithm is modified to reduce its generalized errors. It addresses the problem of overfitting where the model learns too much from the dataset but cannot make reliable predictions yet. It improves the overall performance of the model through different regularization methods. Following are some of the regularization methods available.

Drop Out: Turns off few neurons in every iteration with probability P.

Early Stopping: deals with how many iterations can be run before the model begins to overfit.

Weight constraint: scales weights to a predefined threshold.

Noise: introduces stochastic noise into training process.

In the proposed RNGRU, regularization method Gaussian noise is used to improve the performance by adding noise to Gaussian noise layer from Keras. Gaussian noise is a statistical noise having Probability Density

Function(PDF) equal to that of the normal distribution. PDF of a Gaussian Random Variable z is given by eq. (1).

$$PG(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (1)$$

Where z represents the grey level, μ the mean value and σ standard deviation

RNGRU addresses the overfitting problem of models by adding noise which increases the size of our dataset. A random noise is added to each training sample which makes model less capable to learn from training samples. Thus, neural network learns more general features and has lower generalization errors.

Pseudo code: Proposed RNGRU

1. ***Input:*** *PreProcessed Wastage Data.*
2. ***Output:*** *Predicted Wastages.*

3. *Intialize the Network Layers [GRU, GaussianNoise, Softmax,Dense] and set the Parameters*
4. *batch_size ← 10*
5. *train_ratio ← 0.7*
6. *epochs ← 10*
7. *runs ← 20*
8. *Normilize_data← [data]*
9. *L1← GaussianNoise*
10. *L2← Softmax Activation*
11. *regularizers ← [None, L1, L2]*
12. *units ← [64, 128, 256]*
13. *train, test ← split_data (Normilize_data, train_ratio)*
14. *for dropout in dropouts do*

    a. *for regularizer in regularizers do*

    i. *for unit_number in units do*

    ii. *i ← 0*

    iii. *results ← [ ]*

    iv. *while i < runs do*

    1. *i ← i+ +*
    2. *model ← build_model (dropout, regularizer, unit_number)*
    3. *model.train(train)*
    4. *mse ← model.predict (test)*
    5. *results.append (mse)*

15. *mean ← get_mean (results)*

6. Experimental Evaluation

The models discussed in the paper are implemented in Python using Anaconda Jupyter with keras and Tensor flow. The dataset used in the study contains 20780 records. The training phase of the models used 80% of the records and testing phase of the models used 20% of the records. Figure 11 and 12 shows changes in the performance of the models over all 10 epochs with batch size 10. An epoch indicates the number of passes accomplished by the models on the whole dataset. One epoch is one complete flow of data forward and backward in the neural network. Number of units represents the number of neurons in the hidden layer of the models. Drop out function is used to make the model robust to modifications. The following metrics MAE, MSE and RMSE are used to choose the model with best performance. Performance analysis of the models is tabulated in Table 1. Figure 13 and Figure 14 shows the LSTM model loss and model performance respectively. Figure 15 and Figure 16 shows the RNGRU model loss and model performance respectively.

- ***Mean Absolute Error (MAE):*** This metric measures the average absolute distance between the expected and actual values. It is computed using the eq. (2).

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |x_i - \hat{x}_i| \quad (2)$$

Where x_i is the actual outcome and $\hat{x}_i$ is the expected outcome

- ***Mean Squared Error(MSE):*** This is used to compute the average squared error between the expected and the actual values. It is computed using the eq. (3).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - x\hat{}_i)^2 \qquad (3)$$

Where x_i is the actual outcome and $x\hat{}_i$ is the expected outcome

- ***Root Mean Squared Error(RMSE):*** It is square root of MSE. It is computed using the eq. (4)

$$RMSE = \sqrt{MSE} \qquad (4)$$

Table 1: Performance Analysis of LSTM and RNGRU

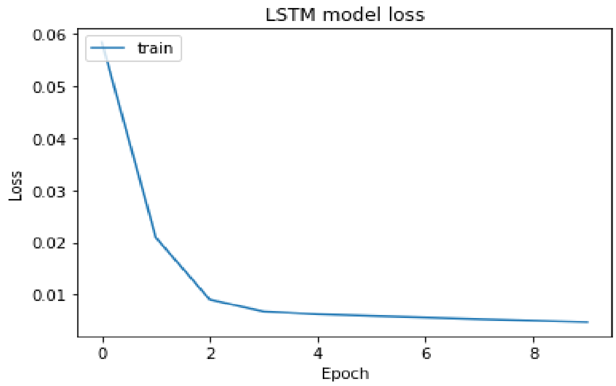
ML Model	MAE	MSE	RMSE
LSTM	0.0231	0.0047	0.1900
RNGRU	0.0147	0.0010	0.0900

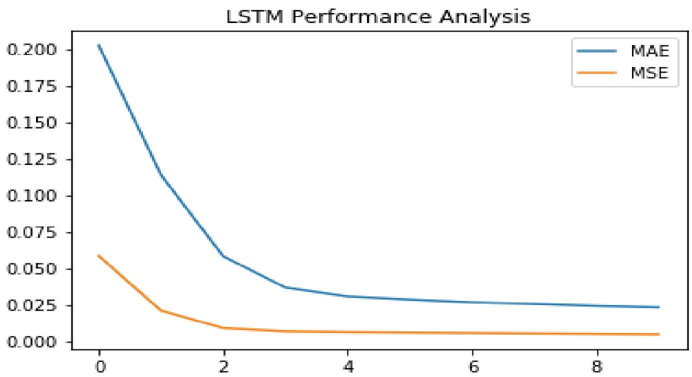
```
Epoch 1/10
 - 2s - loss: 0.0584 - mse: 0.0584 - mae: 0.2024
Epoch 2/10
 - 2s - loss: 0.0209 - mse: 0.0209 - mae: 0.1138
Epoch 3/10
 - 2s - loss: 0.0090 - mse: 0.0090 - mae: 0.0582
Epoch 4/10
 - 2s - loss: 0.0067 - mse: 0.0067 - mae: 0.0367
Epoch 5/10
 - 3s - loss: 0.0062 - mse: 0.0062 - mae: 0.0307
Epoch 6/10
 - 3s - loss: 0.0059 - mse: 0.0059 - mae: 0.0285
Epoch 7/10
 - 2s - loss: 0.0056 - mse: 0.0056 - mae: 0.0265
Epoch 8/10
 - 2s - loss: 0.0053 - mse: 0.0053 - mae: 0.0255
Epoch 9/10
 - 2s - loss: 0.0050 - mse: 0.0050 - mae: 0.0241
Epoch 10/10
 - 2s - loss: 0.0047 - mse: 0.0047 - mae: 0.0231
```

```
Epoch 1/10
 - 12s - loss: 0.0380 - mse: 0.0380 - mae: 0.1166
Epoch 2/10
 - 10s - loss: 0.0018 - mse: 0.0018 - mae: 0.0232
Epoch 3/10
 - 10s - loss: 0.0012 - mse: 0.0012 - mae: 0.0173
Epoch 4/10
 - 10s - loss: 0.0011 - mse: 0.0011 - mae: 0.0159
Epoch 5/10
 - 9s - loss: 0.0010 - mse: 0.0010 - mae: 0.0148
Epoch 6/10
 - 9s - loss: 0.0010 - mse: 0.0010 - mae: 0.0150
Epoch 7/10
 - 10s - loss: 0.0010 - mse: 0.0010 - mae: 0.0147
Epoch 8/10
 - 9s - loss: 0.0010 - mse: 0.0010 - mae: 0.0147
Epoch 9/10
 - 10s - loss: 9.9603e-04 - mse: 9.9603e-04 - mae: 0.0145
Epoch 10/10
 - 10s - loss: 9.5915e-04 - mse: 9.5915e-04 - mae: 0.0137
```

Figure 11: Epoch-LSTM

Figure 12: Epoch-RNGRU





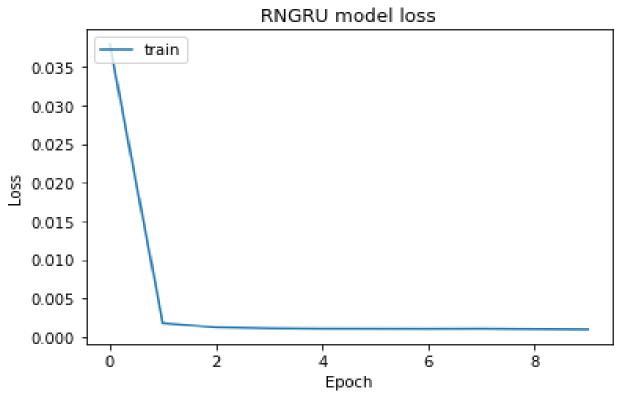


Figure 13: LSTM Model Loss

Figure 14: LSTM Performance Analysis

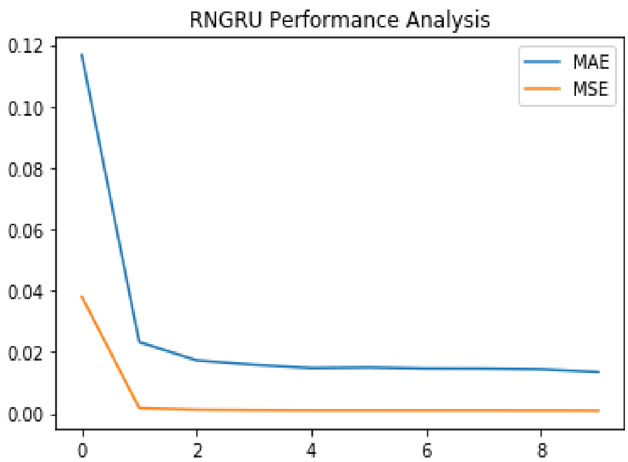


Figure 15: RNGRU Model Loss

Figure 16: Performance Analysis of RNGRU

7. Conclusion

It is very important to have the right model to predict solid waste growth in smart cities which will help the existing waste management system to plan and operate efficiently. The proposed RNGRU model showed low error rates with MAE value 0.0147, MSE value 0.0010 and RMSE value 0.0900 compared to LSTM model and proved to be the suitable model to predict solid waste growth in urban region. The models discussed are also tested with datasets of varying sizes and obtained better outcomes.

Acknowledgement

The authors would like to thank Dr. G S Nagaraja, Professor, Department of CSE, RV College of Engineering, Bangalore and friends Dr. Manjula Gururaj, Mrs. Priyanka, and Mr. Mohammed Ameenulla for their valued direction in finishing this research paper**.**

References

A. Antanasijevic D, Pocajt V, Popovic I, Redzic N, Ristic M (2013) The forecasting of municipal waste generation using artificial neural networks and sustainability indicators. Sustain Science 8:37-46.

B. Abdoli M A, Nezhad M F, Sede R S, Behboudian S (2011) Long term forecasting of solid waste generation by the artificial neural networks. Environmental Progress and Sustainable Energy 31 (4):628-636.

C. Beigl, Peter & Lebersorger, S. & Salhofer, Stefan. (2008). Modelling municipal solid waste generation: A review. Waste management (New York, N.Y.). 28. 200-14. 10.1016/j.wasman.2006.12.011.

D. Coskuner, Gulnur, Majeed S Jassim, Metin Zontul, and Seda Karateke. “Application of Artificial Intelligence Neural Network Modeling to Predict the Generation of Domestic, Commercial and Construction Wastes.” Waste Management & Research, (June 2020).

E. Dissanayaka, D.M.S.H. & Vasanthapriyan, Shanmuganathan. (2019). Forecast Municipal Solid Waste Generation in Sri Lanka. 210-215. 10.1109/ICAC49085.2019.9103421.

F. Grazhdani, Dorina. (2015). Assessing the variables affecting on the rate of solid waste generation and recycling: An empirical analysis in Prespa Park. Waste management (New York, N.Y.). 48. 10.1016/j.wasman.2015.09.028.

G. Ghinea, Cristina & Dragoi, Elena & Elena-Diana, Comanita & Gavrilescu, Marius & Câmpean, Teofil & Curteanu, Silvia & Gavrilescu, Maria. (2016). Forecasting municipal solid waste generation using prognostic tools and regression analysis. Journal of Environmental Management. 182. 80-93. 10.1016/j.jenvman.2016.07.026.

H. Hanandeh, Ali & Abbasi, Mariam. (2016). Forecasting municipal solid waste generation using artificial intelligence modelling approaches. Waste Management. 56. 10.1016/j.wasman.2016.05.018.

I. K.A. Kolekar, T. Hazra, S.N. Chakrabarty, A Review on Prediction of Municipal Solid Waste Generation Models, Procedia Environmental Sciences, Volume 35, 2016, Pages 238-244, ISSN 1878-0296, https://doi.org/10.1016/j.proenv.2016.07.087.

J. Kannangara, Miyuru & Dua, Rahul & Ahmadi, Leila & Bensebaa, Farid. (2017). Modeling and prediction of regional municipal solid waste generation and diversion in Canada using machine learning approaches. Waste Management. 74. 10.1016/j.wasman.2017.11.057., Waste Management, Volume 74, 2018, Pages 3-15, ISSN 0956-053X.

K. Kulisz, Monika; Kujawska, Justyna. (2020). "Prediction of Municipal Waste Generation in Poland Using Neural Network Modeling" *Sustainability* 12, no. 23: 10088.

L. Maya Chavan, T . R. Pattanshetti, “Survey on Municipal Waste Collection Management in Smart City”, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 01, Jan-2018, p-ISSN: 2395-0072 e-ISSN: 2395-0056.

M. Soni, U., Roy, A., Verma, A. *et al.* Forecasting municipal solid waste generation using artificial intelligence models—a case study in India. *SN Appl. Sci.* 1, 162 (2019).

N. Solano Meza JK, Orjuela Yepes D, Rodrigo-Ilarri J, Cassiraga E. Predictive analysis of urban waste generation for the city of Bogotá, Colombia, through the implementation of decision trees-based machine learning, support vector machines and artificial neural networks. Heliyon. 2019 Nov;5(11) e02810. doi:10.1016/j.heliyon.2019.e02810. PMID: 31763474; PMCID: PMC6861577.

O. Yinghao Chu, Chen Huang, Xiaodan Xie, Bohai Tan, Shyam Kamal, Xiaogang Xiong, "Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling", *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 5060857, 9 pages, 2018.