

Received 3 October 2022, accepted 4 November 2022, date of publication 14 November 2022, date of current version 28 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3221941

APPLIED RESEARCH

Exploring Machine Learning and Deep Learning Approaches for Multi-Step Forecasting in Municipal Solid Waste Generation

OSHAN MUDANNAYAKE¹, DISNI RATHNAYAKE, JEROME DINAL HERATH, DINUNI K. FERNANDO¹, AND MGNAS FERNANDO, (Member, IEEE)

University of Colombo School of Computing, Colombo 00700, Sri Lanka

Corresponding author: Dinuni K. Fernando (dkf@ucsc.cmb.ac.lk)

This work was supported in part by the University of Colombo School of Computing.

ABSTRACT Municipal Solid Waste (MSW) management enact a significant role in protecting public health and the environment. The main objective of this paper is to explore the utility of using state-of-the-art machine learning and deep learning-based models for predicting future variations in MSW generation for a given geographical region, considering its past waste generation pattern. We consider nine different machine learning and deep-learning models to examine and evaluate their capability in forecasting the daily generated waste amount. In order to have a comprehensive evaluation, we explore the utility of two training and prediction paradigms, a single-model approach and a multi-model ensemble approach. Three Sri Lankan datasets from; Boralesgamuwa, Dehiwala, and Moratuwa, and open-source daily waste datasets from the city of Austin and Ballarat, are considered in this study. Our results show that Austin and Ballarat datasets got lower error percentage values of 8.03% and 8.3% for Linear Regression and Random Forest models respectively. In Sri Lankan datasets, Random Forest model outperformed other potential models in terms of MAPE by 28.02% to 36.89%. In addition, we provide an in-depth discussion on important considerations to make when choosing a model for predicting MSW generation to enhance the study.

INDEX TERMS Waste management, time series, machine learning, deep learning.

I. INTRODUCTION

Daily human activities are directly and indirectly linked to solid waste generation. Globally, there is around 2.01 billion tons of Municipal Solid Waste generated per year, of which at least 33% is not managed in an environmentally safe manner [1]. Poor management and unsafe disposal of solid waste pose a threat to both the environment and human health. The management of MSW faces various challenges related to urbanization, climate change [2] and population growth, which adds complexity and dynamics to the problem. The attentiveness of urban waste requires suitable disposal facilities, infrastructure and transport [3]. Further, main tasks and fundamental causes of managing solid waste are lack of waste sorting, poor waste collection mechanisms and absence of public engagement in waste management.

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu¹.

Moreover, managing solid waste is a crucial phenomena in both developing and developed countries as it directly impact health and hygiene related issues.

Due to rapid urbanization and population growth, annual global waste generation is expected to increase to 3.4 billion tons over the next 30 years, up from 2.01 billion tons in 2016 [4]. Thus, the prediction and analysis of MSW generation can provide scientific decision-making information for the environmental planning of urban areas and overall quantity control to achieve the reduction, resource, and harmlessness of MSW [5].

It is obvious that proper tracking and waste collection mechanisms are needed to quantify and predict waste generation for a sustainable environment. In particular, the ability to forecast the quantity of waste generated in future would alleviate the burden of managing solid waste, where authorities could factor in future predicted variations in solid waste generation into decision making at the present time.

Thereby effectively utilizing resources for waste collection, sorting and other waste management practices. Overall, the ability to accurately estimate future waste generation rates can help motivate gap analysis in existing waste management and pave the way for better strategic planning.

Against this backdrop, in this work we aim to explore the utility of using state-of-the-art machine learning and deep learning-based models for the propose of predicting future variations in solid waste generation for a given geographical region, considering its past waste generation pattern.

The work in [6] and [7] have already provided a study on comparing machine learning and deep learning models for MSW-prediction, however unlike in the study of [6] we consider nine different machine learning and deep-learning models from the simplest Linear Regression model to state-of-the-art deep learning models like Transformers [8] in order to evaluate the suitability of each model with daily solid waste prediction. Additionally, unlike in the study of [7] we consider daily solid waste data from different geographical areas; Sri Lanka (Boralesgamuwa, Dehiwala, Moratuwa), City of Austin in Texas in USA, and City of Ballarat in Australia.

Furthermore, we consider the weekly seasonal patterns especially in Ballarat and Austin datasets and explore a multi-model ensemble approach which specifically gives additional focus to the weekly seasonal patterns that exist within the waste generation of each day of the week.

In this work we evaluated the predictive power of nine forecasting models, five machine learning-based models—Linear regression [9], Auto ARIMA [10], Light GBM [11], Random Forest [12], Prophet [13] and four deep learning-based models—Long short-term memory(LSTM) [14], Temporal Convolutional Network(TCN) [15], Transformer [8], and N-Beats [16]. We considered these models because they are used in many different time series forecasting studies [17], [18], [19], [11], [15], [20], [21], [22], [23], [24], [25]. We explored two trains of through with respect to training models, a single model approach where a single predictive model is trained to predict solid waste generation similar to a typical time series forecasting task, and a multi-model ensemble approach where seven different models of the same type were trained and used separately for each day of the week. We explored these two options due to the seasonal pattern observed in solid waste generation in Austin and Ballarat datasets, and to identify if there would be any increment in predictive power or decrease in resource utilization in terms of training smaller deep learning-based models for the ensemble.

In this work, we compared the prediction ability of these models mentioned above to forecast daily waste amounts for datasets chosen from three geopolitically diverse locations (i.e., Australia, USA and Sri Lanka). We consider five datasets across these regions, a dataset from Ballarat, which is the third largest city in Victoria, Australia, another dataset from Austin, capital of U.S. state of Texas and also three datasets collected from different Municipal and Urban Authorities in Sri Lanka—Dehiwala, Boralesgamuwa, and

Moratuwa. We applied both single-model and multi-model approaches to all nine models we used in this study. The models were evaluated based on the Root Mean Square Error, Mean Absolute Error, Mean Absolute Percentage Error values. In a nutshell our contribution can be summarized as follows,

- We explore the utility of five machine learning-based predictive models and four state-of-the-art deep learning-based forecasting models for the purpose of predicting solid waste generation.
- We compare the predictive capability of these models extensively, across five datasets. Sri Lankan datasets are from three local authorities in Colombo; Boralesgamuwa, Dehiwala, and Moratuwa. Additionally, two open source datasets from Ballarat, Australia and Austin, Texas.
- We explore the utility of two training and prediction paradigms of using these models, a single model approach and a multi-model ensemble approach.
- Finally we provide extensive evaluation by discussing three important points that may be useful for employing models for predicting solid waste generation—1) the seasonality of data 2) choosing between a single-model or multi-model approach and 3) choosing between a machine learning or deep learning-based model.

Organization: The rest of the paper is organized as follows. Section 2 provides related work of solid waste prediction and time series forecasting using machine learning techniques. In Section 3, we explore the problem statement. Section 4 describes the datasets and data pre-processing steps carried out in this work. In Section 5, we discussed the methodologies of the study. Section 6 presents our evaluation for forecasting solid waste generation in this study including experimental setup, experimental results and discussion. In Section 7, we presented the conclusions of the study.

II. RELATED WORK

This section discusses background details of solid waste prediction and time series forecasting using machine learning techniques.

A. SOLID WASTE PREDICTION

Municipal Solid Waste generation is becoming one of the crucial issues with the rapid development around the world [26]. Presently, the global waste generation of 3.3 million tonnes per day is becoming unmanageable, and this amount is expected to rise up to 11 million tonnes per day by 2100 [27]. Accurate forecasting and prediction of waste are very important because the best strategies for waste management and planning are highly dependent on waste quantification [28], [29].

According to various studies, Municipal Solid Waste forecasting methods can be mainly classified into five categories [30]. They are statistical analysis [31]; regression

analysis [32]; material flow analysis [33]; time series analysis [34]; and artificial intelligence [35], [36], [37], [38]. However, each and every model or method have their own merits and demerits comparatively. Among them, the artificial intelligence model has been gaining popularity in the forecasting of the generation of Municipal Solid Waste due to its high flexibility and proven prediction capabilities compared to the other conventional methods, like regression analysis, time series analysis, [30], [37], [38], [39], [40], [41] etc.

B. MACHINE LEARNING FOR TIME SERIES FORECASTING

Statistical time series modeling is widely used in many prediction and forecasting tasks [13], [42]. Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA, which is a generalization of ARMA) models are widely used to fit to the time series data either to better recognize the data or to forecast data in the series. [43] has analyzed, compared and selected the best time series model to forecast solid waste generation for the next years in the city of Arusha in Tanzania among these two models ARMA and ARIMA, and Exponential Smoothing models. The result showed that ARIMA(1, 1, 1) outperformed ARMA model in terms of MAPE, MAD and RMSE measures. [44] also studied the best time series model to forecast the amount of solid waste generation in city of Tehran. Monthly amount of solid waste data collected by the city authorities from year 2009 to 2014 was used in the study. The result showed that ARIMA(2, 1, 0) outperformed other ARIMA models like ARIMA(0,1,1), ARIMA(1,1,1) to forecast the solid waste generation for the coming years. [45] developed a suitable ARIMA model, on the basis of different statistical parameters, in order to forecast healthcare waste quantity from the hospitals of Garhwal region of Uttarakhand, India. [46] aimed in selecting and evaluating several methods like regression (Life Cycle Assessment of Integrated Waste Management (LCA-IWM) (Available at <http://www.lca-iwm.net>) and time series modeling methods (ARIMA and Seasonal Exponential Smoothing (SES)) for Municipal Solid Waste forecasting in a medium-scaled Eastern European city of Kaunas, Lithuania, with respect to affluence-related and seasonal impacts in the study. For the time series analysis, the combination of ARIMA and SES techniques were found to be the most accurate. [47] used ARIMA model in order to explore the dynamics of solid waste generation and also forecasted monthly solid waste generation in Kumasi Metropolitan Area of Ghana. The analysis indicated that ARIMA(1, 1, 1) was the best model for forecasting solid waste generation in Kumasi Metropolitan Area. [48] conducted a study to evaluate the performance of various statistical modeling methods in order to forecast medical waste generation of Istanbul, in Turkey. ARIMA(0,1,2), showed a best prediction performance compared to Support Vector Regression, and Grey Modeling (1,1) in the annual medical waste generation from 2018 to 2023. In the study of [47] monthly solid waste generation data from year 2005 to 2010 was used

while [48] used historical waste data from 1995 to 2017. In both studies, ARIMA showed the best forecasting performance.

In several recent studies, Artificial Neural Network (ANN) was trained and tested to model waste generation. [49] predicted solid waste generation rates using ANN and Multiple Linear Regression (MLR) in Fars region of Iran. [50] used ANN model to predict industrial solid waste generation and then compared the value with the results obtained from an ANFIS (Adaptive Neuro-Fuzzy Inference System) model. [38] compared six ANN and ANFIS based models to evaluate and determine the effectiveness in Municipal Solid Waste forecasting. According to the results obtained, GA-ANN (i.e. [38] used genetic algorithm techniques to determine the optimal biases and the weights of the ANN, instead of using the back-propagation optimization.) was found to be the most accurate model among the six models. [40] analyzed and compared ANN and ARMA to predict the weekly amounts of solid waste generated by individuals in fourteen households in the residential area of Kator in Juba city. According to the literature, many studies have successfully applied ANNs in the time series analysis and forecasting of solid waste.

There are many other machine learning models were used for time series forecasting. LSTM is a common candidate in time series forecasts, in many recent studies [22], [23], [24], [25]. The LSTM model is the elegant recurrent neural network variant, which uses the purpose-built LSTM memory cells to represent the long-term dependencies in time series data [51]. [52] aimed at the temporal variation of MSW generation in their study, and a LSTM neural network consisting of LSTM layers and a dropout layer was established and optimized for forecasting MSW generation. To have better illustrate of the accuracy and reliability of the LSTM neural network, an ARIMA model and a conventional ANN model was used to forecast Municipal Solid Waste. Results proved that LSTM neural network's superior capability in forecasting solid waste.

[53] have focused on a comparative study to discern the performance of the ANN model compared to the conventional regression approach for forecasting the mean monthly total ozone concentration over Arosa, Switzerland. Also, [54] proposed a hybrid model that combines a linear regression model and deep belief network model for the prediction of time series data. [12] have mentioned that random forest time series modeling provides enhanced predictive ability over existing time series models for the prediction of infectious disease outbreaks.

Reference [11] showed in their experiments on multiple public datasets, that LightGBM speeds up the training process of conventional gradient boosting decision tree by up to over 20 times while achieving almost the same accuracy. Reference [55] show that the robustness of the LightGBM model is better than the other methods like Gradient Boosting Decision Tree algorithm, in their study of cryptocurrency price trend forecasting.

Some recent studies of time series forecasting used state-of-the-art machine learning models like Facebook Prophet, TCN which have outperformed statistical time series modeling methods like SARIMA, ARIMA, etc [17], [18], [15], [19], [20], [21].

III. PROBLEM STATEMENT

In this work, we explore the feasibility of state-of-the-art machine learning and deep learning-based predictive models for the purpose of accurately predicting the daily amount of solid waste generated within a designated geographical authority. The ability to accurately predict solid waste generation multiple days into the future would be helpful for authorities to maximize landfill diversion and better utilize resources to help manage proper disposal and logistics [33], [56], thereby effectively reducing waste management costs and increasing operational efficiency.

We model this problem as a uni-variate time series forecasting task, where the objective at time T is to predict the daily amount of solid waste for k days into the future (i.e., $\hat{Y}_T = [\hat{y}_{T+1}, \hat{y}_{T+2}, \dots, \hat{y}_{T+(k-1)}, \hat{y}_{T+k}]$) based on the amount of daily solid waste generated in the past n days (i.e., $X_T = [x_{T-n}, x_{T-(n-1)}, \dots, x_{T-1}, x_T]$). In our formulation we denote \hat{Y}_T as the predictions made by the model and $Y_T = [y_{T+1}, y_{T+2}, \dots, y_{T+(k-1)}, y_{T+k}]$ as the actual solid waste amounts generated between the T^{th} day and $(T+k)^{\text{th}}$ day.

IV. DATASETS AND DATA PREPROCESSING

In this section, we describe the datasets and data preprocessing steps carried out in this work.

A. DATASETS

We utilized five different datasets of daily waste collected in different cities (shown in Table 1). This includes two open source datasets from Ballarat, Australia and Austin, Texas. Ballarat is primarily a residential area, along with significant industrial, commercial and rural areas. It is a city in the Central Highlands of Victoria, Australia. Austin is the most sub urban major metro in Texas, United States with a strong economy. Additionally, we also utilized datasets from three distinct local authorities in Sri Lanka, which is a developing/emerging country with a lower-middle income economy.

Sri Lankan datasets are from three local authorities in Colombo, Sri Lanka. Colombo is the commercial capital and the largest city of Sri Lanka in terms of population. The urban area of Colombo extends well beyond the boundaries of a single local authority, encompassing other municipal and urban councils. In this study, we used the daily collected waste amounts from the Boralessgamuwa Urban Council, Dehiwala Mount Lavinia Municipal Council and Moratuwa Urban Council. We found that the Sri Lankan datasets contained many missing values due to the irregular waste collection and reporting of the data by the waste collection authorities. The reason to present three data sets from Sri Lanka as one geographical region is that variations of the results of Boralessgamuwa, Moratuwa, and Dehiwala are slightly similar and

these three regions are geographically not varied and located in same proximity.

1) BALLARAT, AUSTRALIA DATASET

Ballarat, Australia Municipal Solid Waste dataset [57] contains the daily statistics of garbage collection in the City of Ballarat. It includes date(July 2000 - March 2015), number of garbage bins collected, tonnes of waste collected, and area of collection. For our study, we have extracted the tonnes of waste collected per day.

2) AUSTIN, TEXAS DATASET

Austin, Texas Municipal Solid Waste dataset [58] contains waste collection information based on several variables. It includes the daily Municipal Solid Waste amount(in tonnes) from January 2003 to July 2021 in city of Austin, Texas. In this study we have extracted only the Report Date and Load Weight variables. Then we calculated the summation of Load Weight per Report Date.

3) BORALESGAMUWA URBAN COUNCIL DATASET

Boralessgamuwa Urban Council's Municipal Solid Waste is the first Sri Lankan region dataset consisting of 2541 data points. It contains the daily Municipal Solid Waste amount (in kg) from January 2012 to December 2018. Municipal Solid Waste amount in the Boralessgamuwa Urban Council area varies from 3741.3kg to 89580kg from the year 2012 to 2018.

4) DEHIWALA MOUNT LAVINIA MUNICIPAL COUNCIL DATASET

Dehiwala Mount Lavinia Municipal is the second Sri Lankan Municipal Solid Waste dataset consisting of 2534 data points. It includes the daily Municipal Solid Waste amount(in kg) from January 2012 to December 2018. Municipal Solid Waste amount in the Dehiwala Mount Lavinia Municipal Council area varies from 3600kg to 39487kg from the year 2012 to 2018.

5) MORATUWA URBAN COUNCIL DATASET

Moratuwa Urban Council Municipal Solid Waste is the third Sri Lankan dataset which contains 1376 data points. It includes the daily Municipal Solid Waste amount(in kg) from January 2012 to December 2018. Municipal Solid Waste amount varies from 590kg to 51544kg from the year 2014 to 2018 in the Moratuwa Urban Council area.

B. DATA PREPROCESSING

This section present the data preprocessing steps. We utilized a machine learning pipeline consisting of three main preprocessing steps. First, we removed the outliers. Then we completed the datasets for a specific period by imputing the missing values. We carry out data imputation by filling in missing values with estimated values based on available data [59], [60], [61]. Finally, we split the data taking 70% as the training data and the remaining 30% as the testing data.

TABLE 1. Dataset details. SD refers to the Standard Deviation of daily solid waste generated.

	Ballarat	Austin	Boralesgamuwa	Dehiwala	Moratuwa
Country/State	Australia	Texas, USA	Sri Lanka	Sri Lanka	Sri Lanka
From	2001	2005	2012	2012	2015
To	2011	2019	2018	2018	2018
After preprocessing					
Chosen start date	2001-01-01	2005-01-01	2012-01-01	2012-01-01	2015-01-01
Chosen end date	2011-12-31	2019-12-31	2018-12-31	2015-12-31	2018-12-31
Number of available values	3991	5452	2417	1433	1308
Number of missing values	26	26	140	28	153
Number of removed outliers	21	29	5	8	9
Number of imputed values	47	55	145	36	162
Percentage of imputed values	1.16%	1.01%	6.01%	2.53%	12.47%
Mean daily weight (kg)	70174	1403398	26932	140641	86297
SD daily weight (kg)	9649	34124	3828	30227	14729

TABLE 2. Grid search values of XGBoost used for data imputation.

Parameter	Values list
Number of lag features	1, 2, 3, 4, 5, 6, 7, 14, 21, 28, 35, 42, 49, 56, 30, 60, 90, 120, 150, 180, 360
n_estimators	10, 100, 1000
max_depth	3, 4, 5, 6, 7, 8, 9
subsample	0.5, 0.6, 0.7, 0.8, 0.9
eta	0.001, 0.01, 0.1, 0.2, 0.3

In the following section we discuss the data imputation steps in more detail.

1) IDENTIFYING AND DEALING WITH THE MISSING VALUES

We found that training a machine learning model on existing data is the best way to impute missing values in this study [62], [63]. We opted for a supervised learning approach with lag features to use the available data from the entire dataset to train a model and impute the missing values instead of utilizing a time series model such as ARIMA which could only use the previous values to impute a particular missing value. We believe that the supervised learning approach is much superior to the alternative for datasets like Moratuwa where more than 10% of the values had to be imputed. We selected the XGBoost model [64] for this task, where the predictions made by the model for a corresponding missing data instance was used to fill in that sequential position in the time series. The XGBoost algorithm can identify a best way to combine the individual variable context information with those about variables efficiently. We chose a grid search to determine the parameters and the number of lag features of the model that could best fit the existing data. The range of each parameter of the grid search is depicted in Table 2. We also tried a initial attempt for imputing values through an ARIMA model by treating each dataset as a series of values. However, this made imputing missing values that appeared early in the series difficult as only the data before the missing value could be used to train the model.

After the grid search, we chose two models for each dataset based on the Root Mean Square Error of the models in order to satisfy the following conditions.

- *Model I* - The model with the lowest RMSE value was chosen as the main imputation model when sufficient data for the lag features preceding the current missing point in the series is present, this would result in better imputation due to the presence of optimal number of lag features.
- *Model II* - There may be cases where the first missing value in the dataset does not have sufficient data preceding it to create the lag features of *Model I*. Then a second model (i.e., *Model II*) was chosen for these scenarios which requires fewer lag features than the number of available values used for *Model I*.

The chosen hyperparameters for imputing data in each dataset are listed in Table 3.

V. METHODOLOGY

In this work we explore two machine learning-based prediction paradigms, a single-model approach and a multi-model [65], [66]. Additionally, we explore the utility of five machine learning-based time series prediction models and three state-of-the-art deep learning-based time series forecasting models. In this section, we explain our rationale behind exploring these two approaches and an overview of the models utilized in this work.

A. IMPLEMENTATION APPROACHES

We consider two different approaches in utilizing machine learning and deep learning-based models for this predictive task, a single-model approach and a multi-model approach.

1) SINGLE-MODEL APPROACH

A single predictive model is trained to predict solid waste generation—similar to a typical time series predictive task. The entire dataset is split into two sets, a *train set* and a *test set*, where both sets comprise of a continues stream of data. A given model is trained on the *train set* and performance is compared using the *test set*.

2) MULTI-MODEL ENSEMBLE APPROACH

During our data analysis, we found clear weekly seasonal patterns as shown in examples for the Ballarat and Austin

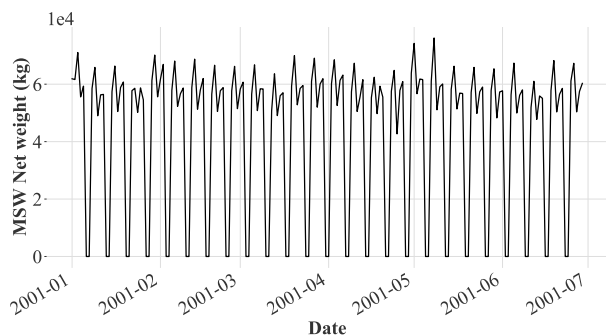


FIGURE 1. Weekly pattern of the daily waste data in city of ballarat.

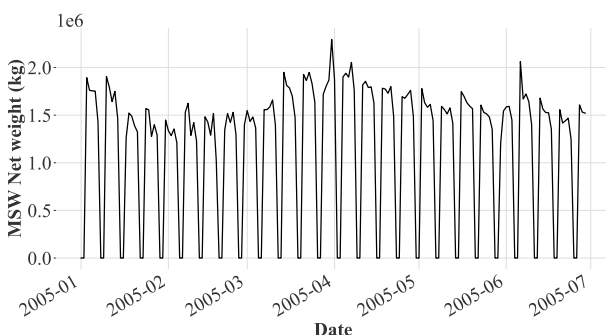


FIGURE 2. Weekly pattern of the daily waste data in city of austin.

shown in Figures 1 and 2 respectively. As shown in these figures in both these datasets, we observe comparatively lower values for the weekends where as values for weekdays follow a tentative weekly pattern.

This observation prompted us to explore a multi-model approach for this predictive task. Instead of predicting future solid waste generation using a single model, we trained seven models of similar architecture to predict the waste generation of each day in the week. The distinction is that we consider each separate day in a week as a different time series by grouping past solid waste generation values for a given day into its own series.

Here, we first extract the data belonging to each day in the original dataset as a separate series and split each of the series according to the original 70% : 30% ratio. At the end of the prediction task, all the predicted series of different days of the

week were combined together to form one single prediction. The main purpose of this approach was to investigate whether it was possible to achieve better performance through modeling each day separately in contrast to using a single model to encompass all the data in a dataset.

B. MACHINE LEARNING AND DEEP LEARNING MODELS

This section presents a brief overview of all the machine learning and deep learning models explored in this study. In total, we consider five machine learning models (i.e., Linear Regression [9], Auto ARIMA [10], Light GBM [11], Random Forest [12] and Prophet [13] and four state-of-the-art deep learning-based time series prediction models—LSTM [14], TCN [15] Transformers [8] and N-Beats [16] in this work.

1) LINEAR REGRESSION

Linear regression takes a linear approach to model the relationship between a dependent variable and one or more independent variables. Linear regression attempts to estimate a straight line that best fits the given data and the equation of that line gives the regression equation. Using one explanatory variable for regression is called simple linear regression, which we use as a baseline model to forecast solid waste generation. Simple linear regression is commonly used in time series forecasting and also in financial analysis. Multiple Linear Regression (MLR) is when several explanatory variables are used for the regression. In this work, we consider a multiple linear regression model as a forecasting model with some of the target series’ lag features which are variables in regression that contains data from earlier time steps. We have empirically chosen these lag values after tuning the linear regression model specifically for each dataset.

2) AUTO ARIMA

ARIMA [67] (Autoregressive Integrated Moving Average) is a time series forecasting model that operates with three parameters, ARIMA (p, d, q), where; p is the number of autoregressive terms which refers to past values used to predict the next value, d is the number of nonseasonal differences to eliminate the seasonality of time series data, and q is the number of lagged forecast errors in the prediction equation

TABLE 3. Hyperparameters of the models that were used to impute the missing data.

		Ballarat	Austin	Boralesgamuwa	Dehiwala	Moratuwa
Model I	Number of lag features	42	360	150	120	49
	n_estimators	1000	10	100	1000	10
	max_depth	7	3	3	8	9
	subsample	0.6	0.9	0.9	0.9	0.7
	eta	0.01	0.3	0.1	0.1	0.3
Model II	Number of lag features	14	150	150	90	49
	n_estimators	1000	1000	100	1000	10
	max_depth	4	6	3	7	9
	subsample	0.6	0.8	0.9	0.6	0.7
	eta	0.01	0.1	0.1	0.01	0.3

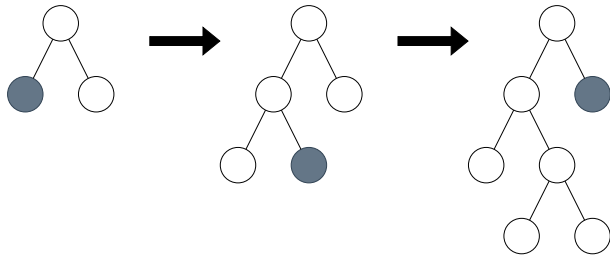


FIGURE 3. Leaf-wise tree growth in LightGBM.

used to define the number of past forecast errors used to predict future values. When training an ARIMA model, statistical techniques are used to generate these p , d , and q values by performing the differencing to eliminate the non-stationary nature of data and plotting the autocorrelation function and the partial autocorrelation function graphs. In Auto ARIMA, the model itself generates the optimal p , d , and q values that would fit the dataset in order to provide the best predictions.

In this study Auto ARIMA model is considered and implemented as a thin wrapper around *pmdarima* library, which provides functionality similar to R's *auto.arima*. The Auto ARIMA model supports the same parameters as the *pmdarima* AutoARIMA model.¹

3) LIGHT GBM

Light Gradient Boosting Machine [11] also known as Light GBM is a gradient boosting framework that uses tree-based learning algorithms. Light GBM shows leaf-wise tree growth. Since it is based on decision tree algorithms, it divides the tree by leaf with the best fit, while other boosting algorithms divide the tree by depth or level rather than by leaf. The leaf-wise algorithm can reduce more losses than the level-wise algorithm used in other gradient boosting methods and therefore gives much better precision which can rarely be achieved by any of the existing boosting algorithms. In addition, Light GBM is very fast in training.

In our work, we consider a LightGBM implementation of the Gradient Boosted Trees algorithm as a univariate forecasting model with lag features.

4) RANDOM FOREST

Random Forest is a type of ensemble machine learning algorithm. It can be used for both classification and regression problems while playing as an extension of bootstrap aggregation of decision trees.

Random Forests are mostly used for classification problems and predictive regression modeling with structured data sets. However, they can also be used for time series forecasting, although this requires that the time series first be turned into a supervised learning problem. It also requires evaluating the model using walk-forward validation, as evaluating the

model using k-fold cross validation would result in optimistically biased results.

In this study, we use random forest regression as a forecasting model for prediction solid waste generation. It also uses lag features in order to obtain a forecast. Our Random Forest implementation is a wrapper around the *RandomForestRegressor* in *sklearn* as [68], [69].

5) PROPHET

Prophet is an open source time series forecasting framework based on the idea of using decomposable models, developed by Facebook [13]. Unlike the previous models, Prophet supports the inclusion of the impact of custom seasonality and holidays. Prophet works with decomposable time series containing three components; trend, seasonality and holidays [70].

The equation of the Prophet is given by, $y(t) = g(t) + s(t) + h(t) + e(t)$ where, $g(t)$ refers to trend, $s(t)$ refers to seasonality, $h(t)$ refers to effects of holidays to the forecast, $e(t)$ refers to the unconditional changes that is scenario specific which is also called the error term and, $y(t)$ is the forecast.

Prophet is designed to have intuitive settings that can be adjusted without knowing the details of the underlying model. The approach of modeling seasonality as an additive component is same as exponential smoothing [71]. Multiplicative seasonality, where the seasonal effect is a factor that multiplies $g(t)$, can be achieved via a logarithmic transformation. Prophet only uses time as a regressor but possibly several linear and non-linear functions of time as components.

We use a wrapper around Prophet implementation in our experiments. We have only added the optional argument of holidays for the datasets of Ballarat and Austin, as the Sri Lankan calendar has not yet been made available in the library.

6) LSTM

The Long Short-Term Memory (LSTM) [72] is an improved Recurrent Neural Network (RNN) based model that has shown promising results with respect to learning long and short-term relationships in time series data. LSTMs overcome two major obstacles that RNN's have had to deal with, which are vanishing gradients and exploding gradients. LSTM fixes this by having a gated structure. LSTMs allow RNNs to remember input over a long time period. This is because LSTMs hold information in a memory. In addition to handling long term dependencies, LSTM retain short term information. The LSTM able to read, write and delete information from its memory via this gated mechanism.

This memory can be thought of as a gated cell, where gated means that the cell decides whether to keep or erase information (i.e. whether to open the gates or not) depending on the importance attached to the information. The importance is assigned via weights, which the algorithm learns. It means that it learns which information is important and which is not over the time.

¹<https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.AutoARIMA.html>

In an LSTM, there are three gates: input, forgetting and output gate. These gates determine whether or not to let in a new input (input gate), suppress information because it is not important (forgotten gate) or let them assign the output to the current time step (output gate).

The gates of an LSTM are analog in the form of sigmoids, which means that they go from zero to one. The fact that they are analog allows them to do backpropagation. The problem of vanishing gradients are solved with LSTM because it keeps the gradients abrupt enough, which keeps relatively short training and high accuracy.

7) TEMPORAL CONVOLUTIONAL NETWORKS

Temporal Convolutional Network (TCN) [15] is a specialized deep learning architecture designed for time series tasks. TCN is able to extract long-term patterns using dilated causal convolutions and residual blocks, which may also be more computationally efficient. This convolution increases the receptive field of the neural network without resorting to pooling operations, so there is no loss of resolution [73]. TCN satisfies two main principles: the network outlet has the same length as the input sequence, similar to LSTM networks; and they prevent information leakage from future to the past using causal convolutions [74].

A common approach to increase the receptive field of the network is to concatenate several blocks of TCNs. But this leads to deeper architectures with many more parameters, which tends to complicate the learning process. For this reason, residual connections have been proposed by [75] to improve performance in very deep architectures and consist of adding the input of a TCN block to its output.

These characteristics make TCN a much more appropriate deep learning architecture for complicated time series problems. The main advantage of TCNs is that, similar to RNNs, they can handle variable-length inputs by dragging or sliding the one-dimensional causal convolutional kernel. Additionally, TCNs are more memory-efficient than recurrent networks due to their shared convolution architecture that allows them to process long sequences in a parallel way. But in RNNs, input sequences are processed sequentially, resulting in higher computation time. In addition, TCNs are trained with the standard backpropagation algorithm, thus it avoids the gradient issues of the backpropagation-through-time algorithm used in RNN [76]. The TCN architecture used in this study is an implementation of a dilated TCN used for forecasting, inspired from the experiments done in the work of [74].

8) TRANSFORMERS

Transformers are the state-of-the-art deep learning model that is commonly used for natural language processing (NLP) tasks. Transformers can also be used for time series forecasting tasks as well.

Transformers are an encoder-decoder architecture. Its main feature is known as a multi-head attention mechanism, which is able to establish intra-dependencies in the input vector

and in the output vector known as auto-attention, as well as inter-dependencies between input and output vectors known as encoder-decoder attention. The multi-head attention mechanism is highly parallelizable when used with GPUs.

Unlike other sequence-aligned deep learning models, Transformer do not process data in an orderly fashion. Instead, it processes the entire data sequence and uses a self-attention mechanism to learn dependencies in the sequence. Therefore, Transformer-based models are generic frameworks that have the ability to model the complex dynamics of time series data that are difficult for sequence models.

Reference [77] developed a novel time series forecasting approach based on Transformer architecture by [78]. Reference [77] mentioned that the approach works by using self-attention mechanisms to learn complex and dynamics patterns from time series data. In our study, we used an implementation of Transformers architecture based on the study of [78].

9) N - BEATS

Reference [16] proposed N-BEATS: Neural Basis Expansion Analysis, a deep neural architecture designed to solve the univariate times series point forecasting problem using deep learning. N-BEATS is known as a pure deep learning architecture in time series forecasting. This model is constructed using backward and forward residual links and a deep stack of fully-connected layers for interpretable time series forecasting.

The design of N-BEATS is based on a few key principles. First, the basic architecture should be simple and generic, but expressive. Secondly, the architecture should not rely on feature engineering or input scaling specific to time series. Finally, the architecture must be extensible to make its outputs interpretable by humans.

Reference [16] showed that this architecture is general, flexible and it outperforms other models on a wide range of time series forecasting problems. Reference [16] demonstrated the state-of-the-art performance in both generic and interpretable configurations.

This helped in validating two important hypotheses:

- 1) The generic deep learning approach performs well on heterogeneous univariate time series forecasting problems without using time series domain knowledge,
- 2) It is possible to further coerce a deep learning model to force it to break down its predictions into distinct human-interpretable outputs.

In addition, [16] demonstrated that deep learning models can be trained over multiple time series in a multitasking fashion, successfully transferring and sharing individual learnings.

Reference [79] presented their work N-BEATS-RNN, which is an extended version of ensemble of deep learning networks for time series forecasting, N-BEATS. They applied the state-of-the-art neural architecture search, based on a

TABLE 4. Single-model architectures used for each dataset.

Model	Parameter	Dataset				
		Boralesgamuwa	Dehiwala	Moratuwa	Ballarat	Austin
Linear Regression	n_lags	180	150	150	360	360
Random Forest	n_lags	150	210	90	6	360
	n_estimators	10	10	10	100	100
	max_depth	8	8	6	8	10
Light GBM	n_lags	210	180	90	90	330
LSTM	input_chunk_length	21	21	1	360	360
	n_epochs	400	400	400	400	400
	hidden_dim	32	32	32	32	32
	n_rnn_layers	1	1	1	1	1
	training_length	330	330	330	330	330
	dropout	0.2	0.2	0.2	0.2	0.2
	optimizer_kwargs	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001
TCN	input_chunk_length	3	21	3	360	330
	output_chunk_length	1	1	1	1	1
	n_epochs	500	500	500	500	500
	dilation_base	2	20	2	359	329
	kernel_size	2	20	2	359	329
	num_filters	10	10	10	10	10
	dropout	0.2	0.2	0.2	0.2	0.2
	optimizer_kwargs	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001
	weight_norm	True	True	True	True	True
Transformer	input_chunk_length	3	14	3	360	270
	output_chunk_length	1	1	1	1	1
	n_epochs	500	500	500	500	200
	d_model	16	16	16	16	16
	nhead	8	8	8	8	8
	num_encoder_layers	2	2	2	2	2
	num_decoder_layers	2	2	2	2	2
	dim_feedforward	128	128	128	128	128
	dropout	0.2	0.2	0.2	0.2	0.2
optimizer_kwargs	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	
N-BEATS	input_chunk_length	300	28	210	90	360
	output_chunk_length	1	1	1	1	1
	n_epochs	200	200	200	200	200
	num_stacks	10	10	10	10	10
	num_blocks	1	1	1	1	1
	num_layers	4	4	4	4	4
	layer_widths	512	512	512	512	512
	optimizer_kwargs	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001
	generic_architecture	True	True	True	True	True

fast and efficient weight-sharing search, as a solution for an ideal Recurrent Neural Network architecture to be added to N-BEATS. In our study, we used the univariate version of the implementation of N-BEATS architecture, as outlined in the study of [16].

VI. EVALUATION

In this section, we present our evaluation for forecasting solid waste generation by machine learning and deep learning models. Additionally, we provide a discussion on important considerations on choosing an appropriate forecasting model.

A. EXPERIMENTAL SETUP

In our experiments we explored the prediction power of eight forecasting models across two prediction paradigms, a single-model approach and a multi-model approach. All experiments in both single-model and multi-model ensemble approach considers a multi-step prediction of the last 30% values in each dataset. All tests were run on a 12-core Ryzen 3900 machine with a base clock speed of 3.1GHz and 32GB

RAM. The models were trained on a Nvidia RTX 2070 Super GPU with 8 GB GDDR6 VRAM. The implementations for the machine learning and deep learning models were carried out using Python and Darts [80], a machine learning library for Python with a focus on time series forecasting.

Machine learning models were tuned with an exhaustive grid search heuristic. Deep learning models were manually tuned to the best of our ability. The parameters of the best models for the single-model approach and multi-model approach are in Table 4 and Table 5, respectively. We haven't included the parameters for the Prophet and Auto ARIMA in these tables as they were chosen automatically by the respective algorithms.

1) EVALUATION METRICS

We used three metrics for the evaluation of the models during this study— Root Mean Square Error, Mean Absolute Error, and Mean Absolute Percentage Error. Let y_{ij} be the i^{th} test sample for the j^{th} prediction step where $i \in [1, k]$, and \hat{y}_{ij} be

TABLE 5. Multi-model architectures used for each dataset.

Model	Parameter	Dataset				
		Boralesgamuwa	Dehiwala	Moratuwa	Ballarat	Austin
Linear Regression	n_lags	16	24	2	48	48
Random Forest	n_lags	2	7	6	1	6
	n_estimators	1000	1000	100	100	10
	max_depth	4	8	10	4	4
Light GBM	n_lags	1	7	1	2	48
LSTM	input_chunk_length	5	8	12	5	2
	n_epochs	200	200	200	200	200
	hidden_dim	32	32	32	32	32
	n_rnn_layers	1	1	1	1	1
	training_length	48	48	48	10	4
	dropout	0.2	0.2	0.2	0.2	0.2
	optimizer_kwargs	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001
TCN	input_chunk_length	5	24	5	48	48
	output_chunk_length	1	1	1	1	1
	n_epochs	1000	1000	1000	1000	1000
	dilation_base	10	10	10	10	10
	kernel_size	4	23	4	47	47
	num_filters	10	10	10	10	10
	dropout	0.2	0.2	0.2	0.2	0.2
	optimizer_kwargs	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001
	weight_norm	True	True	True	True	True
	Transformer	input_chunk_length	3	48	3	36
output_chunk_length		1	1	1	1	1
n_epochs		500	500	500	500	500
d_model		16	16	16	16	16
nhead		8	8	8	8	8
num_encoder_layers		2	2	2	2	2
num_decoder_layers		2	2	2	2	2
dim_feedforward		128	128	128	128	128
dropout		0.2	0.2	0.2	0.2	0.2
optimizer_kwargs		'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001
N-BEATS	input_chunk_length	4	7	4	48	4
	output_chunk_length	1	1	1	1	1
	n_epochs	100	100	100	100	200
	num_stacks	10	10	10	5	10
	num_blocks	1	1	1	1	1
	num_layers	4	4	4	2	4
	layer_widths	128	128	128	128	128
	optimizer_kwargs	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001	'lr': 0.0001
	generic_architecture	True	True	True	True	True

the predicted value of y_{ij} and k is the number of test samples. The Root Mean Square Error, Mean Absolute Error and Mean Absolute Percentage Error are given by Equations 1, 2 and 3 respectively.

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^k (\hat{y}_{ij} - y_{ij})^2}{k}} \quad (1)$$

$$MAE_j = \frac{\sum_{i=1}^k |\hat{y}_{ij} - y_{ij}|}{k} \quad (2)$$

$$MAPE_j = \frac{1}{k} \sum_{i=1}^k \frac{|\hat{y}_{ij} - y_{ij}|}{y_{ij}} \quad (3)$$

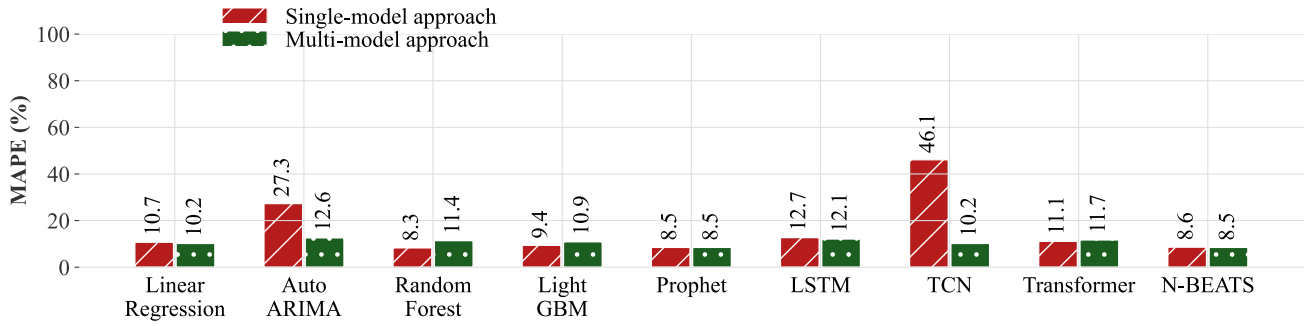
B. EXPERIMENTAL RESULTS

This section will describe the results for the testing phase of all the experiments conducted during our study. Figures 4a, 4b, and 4c correspond to the Mean Absolute Percentage Error values for the best models for Ballarat, Austin

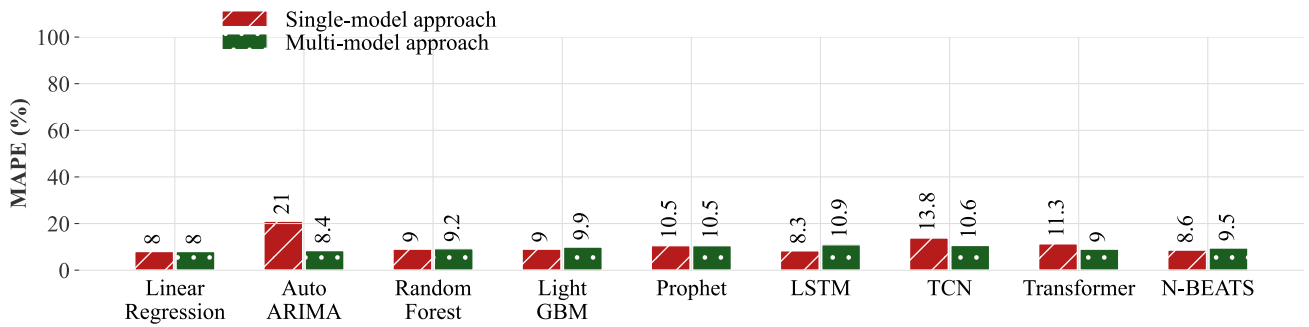
and Sri Lankan datasets respectively. Table 6, Table 7 and Table 8 contains the performance of each model on Ballarat, Austin and Sri Lankan datasets respectively. The tables contain the Root Mean Square Error, Mean Absolute Error and Mean Absolute Percentage Error values of the best model in each experiment.

Figure 4a and Table 6 shows the results for Ballarat dataset for machine learning and deep learning models trained using the single-model approach and the multi-model ensemble approach, respectively. As shown in Figure 4a, the Random Forest model and the N-BEATS model show the best performance for single-model and multi-model training approaches for the Ballarat dataset as 8.3% and 8.47%. Overall, the single-model Random Forest is the most successful model in predicting the waste generation patterns in the Ballarat dataset with an average improvement of 3.85% against the rest of the machine learning models and an average improvement of 6.82% against the deep learning-based models.

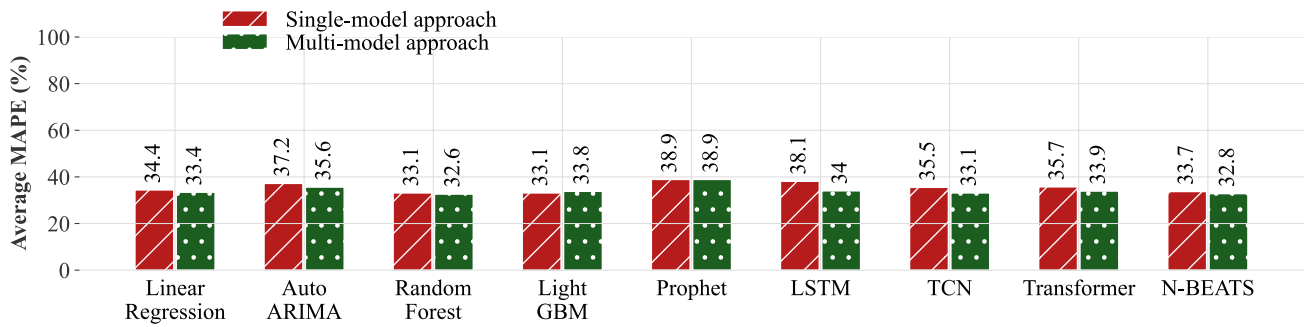
Prophet also shows strong results for the Ballarat dataset (i.e., 8.47% in MAPE). For the Ballarat dataset, Auto ARIMA



(a) Ballarat dataset



(b) Austin dataset



(c) Sri Lankan datasets (average)

FIGURE 4. MAPE of the chosen models with different datasets.

and TCN seems to have shown a significant improvement of over 15% reduction of MAPE with the multi-model approach. Performance of single-model and multi-model approaches average around MAPE of 15.85% and 10.67% across all model types. Therefore the multi-model approach has worked better for the Ballarat dataset.

We see similar variations in MAPE results for Austin dataset (shown in Table 7 and Figure 4b) where Linear Regression has obtained the best performance for the Austin dataset in all scenarios. The error for the results in this dataset are comparable to the Ballarat dataset. Here, on average the best Linear regression model outperforms all other machine learning models by 4.37% and other deep learning models by 2.46% when considering the single model approach. As for the multi-model approach Linear regression

is 1.49% greater than machine learning models and 1.96% greater than deep learning models. Auto ARIMA's results have significantly improved by over 12% with the multi-model approach. LSTM and N-BEATS has shown a decrease in performance in multi-model approach in comparison to the single-model training. The single-model approach shows an average MAPE of 11.07% while the multi-model approach performed slightly better at an average MAPE of 9.56%.

Random Forest model shows the best performance for all Sri Lankan datasets with an average of 32.86% in MAPE except for the single-model training mode in Boralessgamuwa dataset. Light GBM shows the best single-model performance for the Boralessgamuwa dataset (i.e., 28.84% in MAPE). It is apparent that the Random Forest has been most

TABLE 6. Forecast error of the models for the Ballarat dataset.

Type	Model	Ballarat		
		RMSE	MAE	MAPE
Single-model	Linear Regression	10487.26	8267.73	10.68%
	Auto ARIMA	23578.65	20940.20	27.29%
	Random Forest	8046.01	6188.04	8.30%
	Light GBM	9143.33	7119.98	9.39%
	Prophet	8222.38	6428.33	8.47%
	LSTM	12154.72	9733.43	12.71%
	TCN	38310.66	35065.34	46.14%
	Transformer	10493.07	8475.70	11.08%
	N-BEATS	8321.65	6434.21	8.56%
	Multi-model	Linear Regression	9898.02	7835.63
Auto ARIMA		11927.42	9791.61	12.59%
Random Forest		11200.84	8869.95	11.38%
Light GBM		10604.58	8369.83	10.90%
Prophet		8222.38	6428.33	8.47%
LSTM		11459.74	9287.42	12.10%
TCN		10003.99	7834.69	10.20%
Transformer		11166.82	8963.56	11.66%
N-BEATS		8124.68	6268.66	8.47%

TABLE 7. Forecast error of the models for the Austin dataset.

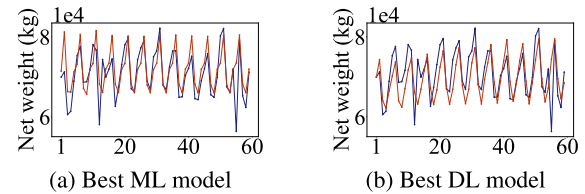
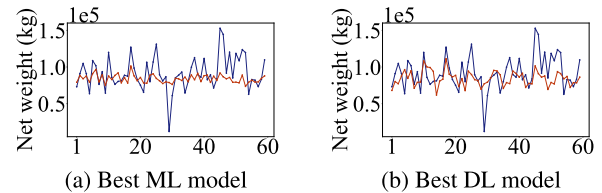
Type	Model	Austin		
		RMSE	MAE	MAPE
Single-model	Linear Regression	201503.02	148864.26	8.03%
	Auto ARIMA	530227.57	426701.85	21.04%
	Random Forest	228542.31	172243.11	9.00%
	Light GBM	222154.08	170821.98	9.05%
	Prophet	236063.73	193601.36	10.51%
	LSTM	211217.66	158671.09	8.30%
	TCN	308054.58	248936.14	13.84%
	Transformer	269734.26	212123.85	11.25%
	N-BEATS	215728.82	163139.54	8.56%
	Multi-model	Linear Regression	201086.41	151262.21
Auto ARIMA		202830.61	156852.36	8.44%
Random Forest		222717.93	171967.60	9.22%
Light GBM		240025.04	185629.54	9.90%
Prophet		236063.73	193601.36	10.51%
LSTM		270133.18	211464.93	10.91%
TCN		247075.30	195148.92	10.56%
Transformer		210849.84	164941.38	8.95%
N-BEATS		224236.34	176008.62	9.53%

successful in capturing the patterns in the Sri Lankan datasets, which didn't have visible seasonal patterns.

In most cases, training according to the multi-model approach shows and improvement in predictive performance. On average, deep learning models haven't shown a significant improvement in prediction for the chosen datasets over machine learning models. Random forests have been able to achieve comparable results as the best deep learning model after a grid search.

C. DISCUSSION

In this section we further discuss three points to consider when choosing a predictive model for forecasting solid waste generation, namely 1) seasonality in data 2) the modeling approach: single-model vs multi-model and 3) choosing between a machine learning vs a deep learning model.

**FIGURE 5.** Actual vs predictions for ballarat.**FIGURE 6.** Actual vs predictions for moratuwa-sri lanka.

1) EFFECT OF SEASONALITY IN DATA

Overall we observe a lower average predictive error for Ballarat and Austin of 13.26% and 10.31% across all models, irrespective of the approach used (i.e., single or multi-model). In contrast there is higher error for the Sri Lankan regional datasets (i.e., average of 25.64%). We attribute this disparity in predictive performance to the seasonal patterns observed in Austin and Ballarat.

Figures 5 shows actual vs predictions made for Ballarat dataset for both best machine learning and deep learning models. Similarly, Figure 6 shows the actual vs predictions made for the Moratuwa-Sri Lankan dataset. Both the figures show the predictions for best performing machine learning and deep learning model for each dataset considering both single-model or the multi-model approach. Based on Figure 5 it is clear that predictions made by the models tend to capture variations more accurately in the presence of seasonality. In contrast as shown in Figure 6, in a dataset that doesn't have seasonality, both the machine learning (Fig 5.a) and deep learning (Fig 5.b) models have a harder time in learning temporal patterns. This is a common trend we see in all other non-seasonal data (i.e., Boralessgamuwa and Dehiwala-Sri Lankan data).

2) SINGLE-MODEL OR MULTI-MODEL APPROACH

The seasonality in the Ballarat and the Austin datasets prompted us to explore the utility of exploring a single-model and a multi-model approach, where the multi-model approach specifically gives additional focus to the weekly seasonal patterns that exist within the waste generation of each day of the week. Our initial assumption was that the multi-model approach would perform better against data with clear seasonal patterns. Table 10 shows the average RMSE, MAE and MAPE error considering all the datasets against a single-model and multi-model approach. Overall, we observe that the multi-model approach reported slightly better performance than the single-model approach (Table 10). However, while the predictive performance itself shows slight differences, the biggest difference in choosing to use either of these

TABLE 8. Forecast error of the models for the Sri Lankan datasets.

Type	Model	Boralesgamuwa			Moratuwa			Dehiwala		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Single-model	Linear Regression	7649.25	5871.78	29.92%	24182.97	18603.94	36.96%	40396.04	29926.84	36.35%
	Auto ARIMA	8151.27	6336.56	31.86%	24427.86	18800.01	38.44%	45323.69	32637.62	41.42%
	Random Forest	7695.17	5781.23	29.54%	25568.08	20011.34	36.69%	38454.14	28822.03	33.11%
	Light GBM	7677.51	5884.13	28.84%	25077.71	19614.25	36.89%	38480.52	28450.88	33.49%
	Prophet	9266.42	7314.15	40.13%	24081.80	18065.11	39.24%	40503.47	28401.74	37.30%
	LSTM	9386.57	7241.04	38.43%	24648.59	19039.41	37.98%	43536.40	36169.52	37.97%
	TCN	8546.49	6847.26	31.42%	25901.39	20241.18	37.23%	45447.95	38963.44	37.95%
	Transformer	8272.64	6518.48	31.47%	24993.83	19372.83	37.61%	43242.67	35559.90	38.10%
	N-BEATS	7606.08	5836.69	29.08%	28863.96	22343.01	37.87%	39264.17	28930.99	34.17%
Multi-model	Linear Regression	7562.20	5745.12	28.62%	24886.28	19555.34	37.90%	39096.37	29953.30	33.80%
	Auto ARIMA	7810.96	5949.68	31.82%	24456.92	18992.51	38.13%	41005.50	27897.12	36.78%
	Random Forest	7699.28	5850.69	28.02%	24569.97	19312.16	36.86%	39116.22	29992.33	32.96%
	Light GBM	8009.02	6146.41	29.92%	26987.72	21537.43	37.71%	39720.90	30233.13	33.67%
	Prophet	9266.42	7314.15	40.13%	24081.80	18065.11	39.24%	40503.47	28401.74	37.30%
	LSTM	7708.24	5826.33	29.23%	24242.05	18504.77	38.31%	38825.70	28824.24	34.44%
	TCN	8044.85	6199.91	28.36%	26241.96	20880.14	37.60%	39929.13	31123.45	33.43%
	Transformer	7706.91	5827.10	29.70%	24243.74	18694.85	37.82%	38715.77	28796.07	34.05%
	N-BEATS	7792.79	5989.28	28.04%	24793.54	19563.35	37.19%	39150.50	30002.43	33.26%

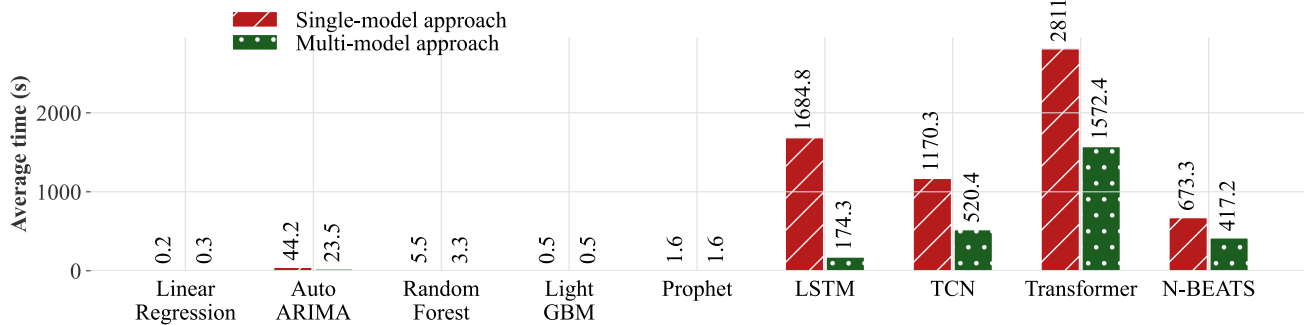


FIGURE 7. Average training time for the models across all the datasets.

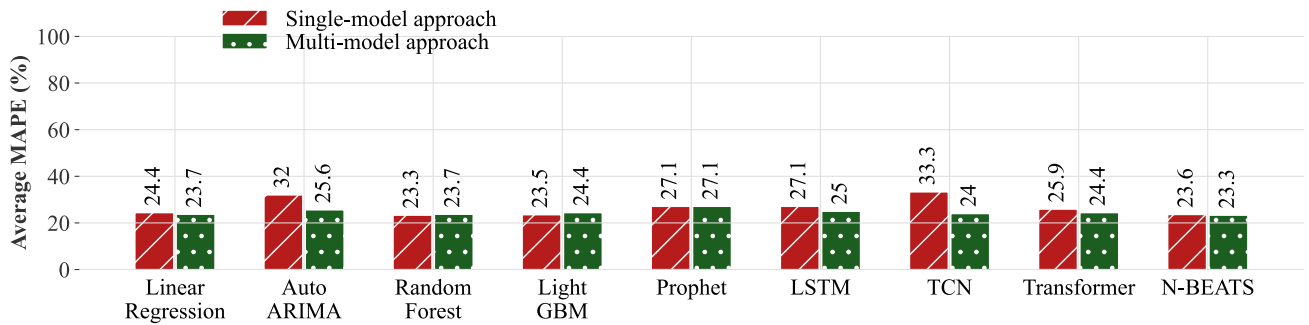


FIGURE 8. Average MAPE of the models across all the datasets.

TABLE 9. Average performance of machine learning vs deep learning models.

Training method	RMSE	MAE	MAPE	Training time (s)	Predicting time (s)
Single-model	71747.34	56118.70	26.70%	710.15	1.82
Multi-model	62217.67	48348.99	24.59%	301.50	24.13

TABLE 10. Average performance of single-model vs multi-model training methods.

Training paradigm	RMSE	MAE	MAPE	Training time (s)	Predicting time (s)
Machine learning	67574.97	52457.47	25.44%	23.23	15.23
Deep learning	66051.49	51882.44	25.96%	1264.18	9.44

approaches comes with the effort and resources in training these models.

Figure 7 shows the average training time for each of the best performing models across all datasets. The average

training time of single-model and multi-model approaches were 710.15 seconds and 301.5 seconds. And the average MAPE values varied as 26.7% and 24.59%. Therefore the multi-model approach has taken less time to train the models with better performance. It takes significantly less time to train the deep learning models using the multi-model approach than the single-model approach. This is because a set of smaller models trained according to the multi-model approach were able to capture the pattern better than one large model that was trained using the conventional single-model approach. This directly benefits in reducing the computational cost of the training phase. Therefore, for situations with higher constraints in time, we believe a multi-model approach may be more suitable as it obtains similar predictive performance while requiring less training time, specifically for instances where deep learning models are used.

3) MACHINE LEARNING OR DEEP LEARNING

Our experiments considered both machine learning and deep learning models in order to determine comparatively what types of models may be more suitable. Figure 8 shows the average MAPE of each of the models considering all the datasets. While the deep learning models do show a slightly better reduction in error, it is marginal amounting to less than a decrease of 5% in MAPE. Meaning that there seems to be less utility in using deep neural network architectures for this forecasting task (i.e., summarized in Table 9).

In addition, the deep learning models take a significantly longer time in training than the machine learning models. As shown in Table 9 on average, the deep learning models have taken 54× times more time to train than the machine learning models. All the machine learning models were trained on the CPU, whereas the deep learning models were trained on a dedicated GPU. The machine learning models were trained with relatively less computational cost than the deep learning models, adding to the utility of using machine learning-based models.

We consider datasets spanning different time periods. The longest spanning dataset Austin contains the records of 14 years. The smallest dataset is the Moratuwa dataset which spans only for 3 years. For both these datasets, the performance of the best machine learning model and the best deep learning model varies by less than 2% MAPE. Therefore we can conclude that the length of the dataset is not a significant factor contributing to the design choice of selecting a machine learning or deep learning model. It seems like both model types work equally well for long-term (i.e., more than 10 years) or shorter-term (i.e., around 2-3 years) time periods of data. While it can be argued that a deep learning model might be able to predict the waste data generation of a much larger dataset with better accuracy, it may not be realistic to assume that the same waste generation patterns could exist for a time period beyond 15 years due to changing in urban populations, waste management policies that could be implemented within such a long period of time.

We have used a grid search method to train the machine learning models. The deep learning models were trained manually with extreme care. Therefore the machine learning models have presented a greater advantage in being able to be trained in environments with less supervision and skilled personal than deep learning models.

While machine learning models on average have slightly outperformed the deep learning models, deep learning models have outperformed machine learning models on several occasions. However the performance improvement of the deep learning models have come at a much greater computational cost. Therefore it is apparent that the machine learning models are specifically well suited for developing regions such as Sri Lanka where there exist limitations in modeling waste data such as lack of computational power and skilled personals.

VII. CONCLUSION

In this paper, we investigated how well the machine learning and the state-of-the-art deep learning models are able to forecast daily waste amount of five different geographical areas. We compared the performance of nine different machine learning and deep learning models across all the five datasets. In our study, we observe comparable results in both machine learning and deep learning models, while machine learning models on average have slightly outperformed the deep learning models. However deep learning models have taken more computational power during the training phase. Therefore we can conclude that machine learning models are sufficient for forecasting municipal solid waste in a given geographical location. Furthermore the training time has been reduced by using the multi-model training paradigm. Also results show that it also contributed to a slight increase of performance as well.

ACKNOWLEDGMENT

The authors would like to thank Sri Lankan Authorities, Boralessgamuwa Urban Council, Dehiwala Mount Lavinia Municipal Council, and Moratuwa Urban Council for providing data for this study. They thank their colleagues from the University of Colombo who provided insight and expertise that greatly assisted the research.

REFERENCES

- [1] H. S. Salem, Y. Yihdego, and M. Y. Pudza, "Scientific and technical perspectives of waste-to-energy conversion," *J. Nature Sci. Sustain. Technol.*, vol. 15, no. 3, pp. 193–217, 2021.
- [2] K. Raab, G. Tolotti, and R. Wagner, "Challenges in solid waste management: Insights into the disposal behavior of suburban consumers in Guatemala city," *Frontiers Sustain. Cities*, vol. 3, 2021.
- [3] S. Ulgiati and A. Zucaro, "Challenges in urban metabolism: Sustainability and well-being in cities," *Frontiers Sustain. Cities*, vol. 1, p. 1, May 2019.
- [4] S. K. Ghosh, *Circular Economy: Global Perspective*. Singapore: Springer, 2020.
- [5] 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.
- [6] H.-N. Guo, S.-B. Wu, Y.-J. Tian, J. Zhang, and H.-T. Liu, "Application of machine learning methods for the prediction of organic solid waste treat-

- ment and recycling processes: A review," *Bioresour. Technol.*, vol. 319, Jan. 2021, Art. no. 124114.
- [7] X. C. Nguyen, T. T. H. Nguyen, D. D. La, G. Kumar, E. R. Rene, D. D. Nguyen, S. W. Chang, W. J. Chung, X. H. Nguyen, and V. K. Nguyen, "Development of machine learning-based models to forecast solid waste generation in residential areas: A case study from Vietnam," *Resour. Conservation Recycling*, vol. 167, Apr. 2021, Art. no. 105381.
 - [8] T. Wolf et al., "HuggingFace's transformers: State-of-the-art natural language processing," 2019, *arXiv:1910.03771*.
 - [9] J. M. Stanton, "Galton, Pearson, and the peas: A brief history of linear regression for statistics instructors," *J. Statist. Educ.*, vol. 9, no. 3, pp. 1–13, Jan. 2001.
 - [10] H. Matsila and P. Bokoro, "Load forecasting using statistical time series model in a medium voltage distribution network," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 4974–4979.
 - [11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3146–3154.
 - [12] M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz, "Comparison of ARIMA and random forest time series models for prediction of avian influenza H5N1 outbreaks," *BMC Bioinf.*, vol. 15, no. 1, pp. 1–9, Dec. 2014.
 - [13] S. J. Taylor and B. Letham, "Forecasting at scale," *Amer. Statistician*, vol. 72, no. 1, pp. 37–45, 2018.
 - [14] Y. Yu, X. Si, C. Hu, and Z. Jianxun, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.
 - [15] P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu, "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," *Soft Comput.*, vol. 24, no. 21, pp. 16453–16482, Nov. 2020.
 - [16] B. N. Oreshkin, D. Carпов, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," 2019, *arXiv:1905.10437*.
 - [17] K. K. R. Samal, K. S. Babu, S. K. Das, and A. Acharya, "Time series based air pollution forecasting using SARIMA and prophet model," in *Proc. Int. Conf. Inf. Technol. Comput. Commun.*, 2019, pp. 80–85.
 - [18] C. B. A. Satrio, W. Darmawan, B. U. Nadia, and N. Hanafiah, "Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET," *Proc. Comput. Sci.*, vol. 179, pp. 524–532, 2021.
 - [19] P. Lara-Benitez, M. Carranza-García, J. M. Luna-Romera, and J. C. Riquelme, "Temporal convolutional networks applied to energy-related time series forecasting," *Appl. Sci.*, vol. 10, no. 7, p. 2322, Mar. 2020.
 - [20] W.-H. Lin, P. Wang, K.-M. Chao, H.-C. Lin, Z.-Y. Yang, and Y.-H. Lai, "Wind power forecasting with deep learning networks: Time-series forecasting," *Appl. Sci.*, vol. 11, no. 21, p. 10335, Nov. 2021.
 - [21] R.-G. Cirstea, T. Kieu, C. Guo, B. Yang, and S. J. Pan, "EnhanceNet: Plugin neural networks for enhancing correlated time series forecasting," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 1739–1750.
 - [22] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, Jan. 2019.
 - [23] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Phys. A, Stat. Mech. Appl.*, vol. 519, pp. 127–139, Apr. 2019.
 - [24] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, Solitons Fractals*, vol. 135, Jun. 2020, Art. no. 109864.
 - [25] S. Elsworth and S. Güttel, "Time series forecasting using LSTM networks: A symbolic approach," 2020, *arXiv:2003.05672*.
 - [26] N. E. Johnson, O. Ianiuk, D. Cazap, L. Liu, D. Starobin, G. Dobler, and M. Ghandehari, "Patterns of waste generation: A gradient boosting model for short-term waste prediction in New York city," *Waste Manage.*, vol. 62, pp. 3–11, Apr. 2017.
 - [27] D. Hoornweg, P. Bhada-Tata, and C. Kennedy, "Environment: Waste production must peak this century," *Nature*, vol. 502, no. 7473, p. 615, 2013.
 - [28] D. J. Tonjes and K. L. Greene, "A review of national municipal solid waste generation assessments in the USA," *Waste Manag. Res., J. Sustain. Circular Econ.*, vol. 30, no. 8, pp. 758–771, Aug. 2012.
 - [29] C. Perera and M. Fernando, "Comparison of multiple linear regression and artificial neural network models for the prediction of solid waste generation in Sri Lanka," *Trans. Mach. Learn. Data Min.*, vol. 13, no. 1, pp. 3–25, 2020.
 - [30] M. Abbasi and A. El Hanandeh, "Forecasting municipal solid waste generation using artificial intelligence modelling approaches," *Waste Manag.*, vol. 56, pp. 13–22, Oct. 2016.
 - [31] R. Sha'Ato, S. Y. Aboho, F. O. Oketunde, I. S. Eneji, G. Unazi, and S. Agwa, "Survey of solid waste generation and composition in a rapidly growing urban area in central Nigeria," *Waste Manag.*, vol. 27, no. 3, pp. 352–358, Jan. 2007.
 - [32] Y. Wei, Y. Xue, J. Yin, and W. Ni, "Prediction of municipal solid waste generation in China by multiple linear regression method," *Int. J. Comput. Appl.*, vol. 35, no. 3, pp. 136–140, 2013.
 - [33] L. Zhang, Z. Yuan, J. Bi, and L. Huang, "Estimating future generation of obsolete household appliances in China," *Waste Manag. Res., J. Sustain. Circular Economy*, vol. 30, no. 11, pp. 1160–1168, Nov. 2012.
 - [34] L. Xu, P. Gao, S. Cui, and C. Liu, "A hybrid procedure for MSW generation forecasting at multiple time scales in Xiamen city, China," *Waste Manag.*, vol. 33, no. 6, pp. 1324–1331, Jun. 2013.
 - [35] M. Abbasi, M. Abduli, B. Omidvar, and A. Baghvand, "Forecasting municipal solid waste generation by hybrid support vector machine and partial least square model," *Int. J. Environ. Res.*, vol. 7, pp. 27–38, 2013.
 - [36] S. H. Yusoff, U. N. K. A. Din, H. Mansor, N. S. Midi, and S. A. Zaini, "Neural network prediction for efficient waste management in Malaysia," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, pp. 738–747, 2018.
 - [37] S. A. Ali and A. Ahmad, "Forecasting MSW generation using artificial neural network time series model: A study from metropolitan city," *Social Netw. Appl. Sci.*, vol. 1, no. 11, pp. 1–16, Nov. 2019.
 - [38] U. Soni, A. Roy, A. Verma, and V. Jain, "Forecasting municipal solid waste generation using artificial intelligence models—A case study in India," *Social Netw. Appl. Sci.*, vol. 1, no. 2, p. 162, Feb. 2019.
 - [39] D. M. S. H. Dissanayaka and S. Vasanthapriyan, "Forecast municipal solid waste generation in Sri Lanka," in *Proc. Int. Conf. Advancements Comput. (ICAC)*, Dec. 2019, pp. 210–215.
 - [40] P. Juba, "Forecasting solid waste generation in Juba town, south Sudan using artificial neural networks (ANNs) and autoregressive moving averages (ARMA)," *J. Environ. Waste Manag.*, vol. 4, no. 2, pp. 211–223, 2017.
 - [41] V. Patel and S. Meka, "Forecasting of municipal solid waste generation for medium scale towns located in the state of Gujarat, India," *Int. J. Innov. Res. Sci.*, vol. 2, no. 9, pp. 4707–4716, 2013.
 - [42] V. Cerqueira, L. Torgo, and C. Soares, "Machine learning vs statistical methods for time series forecasting: Size matters," 2019, *arXiv:1909.13316*.
 - [43] A. Mwenda, D. Kuznetsov, and S. Mirau, "Time series forecasting of solid waste generation in Arusha city-Tanzania," *Math. Theory Model.*, vol. 4, no. 8, pp. 29–39, 2014.
 - [44] F. Marandi and S. Fatemi Ghomi, "Time series forecasting and analysis of municipal solid waste generation in Tehran city," in *Proc. 12th Int. Conf. Ind. Eng. (ICIE)*, 2016, pp. 14–18.
 - [45] A. Chauhan and A. Singh, "An arima model for the forecasting of health-care waste generation in the Garhwal region of Uttarakhand, India," *Int. J. Services Oper. Informat.*, vol. 8, no. 4, pp. 352–366, 2017.
 - [46] I. Rimaitytė, T. Ruzgas, G. Denafas, V. Račys, and D. Martuzevičius, "Application and evaluation of forecasting methods for municipal solid waste generation in an eastern-European city," *Waste Manag. Res., J. Sustain. Circular Economy*, vol. 30, no. 1, pp. 89–98, Jan. 2012.
 - [47] E. Owusu-Sekyere, E. Harris, and E. Bonyah, "Forecasting and planning for solid waste generation in the kumasi metropolitan area of ghana: An arima time series approach," *Int. J. Sci.*, vol. 11, 2013.
 - [48] Z. Ceylan, S. Bulkan, and S. Elevli, "Prediction of medical waste generation using SVR, GM (1,1) and ARIMA models: A case study for megacity Istanbul," *J. Environ. Health Sci. Eng.*, vol. 18, no. 2, pp. 687–697, Dec. 2020.
 - [49] S. Azadi and A. Karimi-Jashni, "Verifying the performance of artificial neural network and multiple linear regression in predicting the mean seasonal municipal solid waste generation rate: A case study of fars province, Iran," *Waste Manag.*, vol. 48, pp. 14–23, Feb. 2016.
 - [50] M. K. Tiwari, S. Bajpai, and U. Dewangan, "Prediction of industrial solid waste with ANFIS model and its comparison with ANN model—A case study of Durg-Bhilai twin city India," *Int. J. Eng. Innov. Technol. (IJEIT)*, vol. 6, no. 2, pp. 192–201, 2012.
 - [51] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

- [52] D. Niu, F. Wu, S. Dai, S. He, and B. Wu, "Detection of long-term effect in forecasting municipal solid waste using a long short-term memory neural network," *J. Cleaner Prod.*, vol. 290, Mar. 2021, Art. no. 125187.
- [53] G. Bandyopadhyay and S. Chattopadhyay, "Single hidden layer artificial neural network models versus multiple linear regression model in forecasting the time series of total ozone," *Int. J. Environ. Sci. Technol.*, vol. 4, no. 1, pp. 141–149, Dec. 2007.
- [54] W. Xu, H. Peng, X. Zeng, F. Zhou, X. Tian, and X. Peng, "A hybrid modelling method for time series forecasting based on a linear regression model and deep learning," *Appl. Intell.*, vol. 49, no. 8, pp. 3002–3015, 2019.
- [55] X. Sun, M. Liu, and Z. Sima, "A novel cryptocurrency price trend forecasting model based on LightGBM," *Finance Res. Lett.*, vol. 32, Jan. 2020, Art. no. 101084.
- [56] L. Yermal and P. Balasubramanian, "Application of auto ARIMA model for forecasting returns on minute wise amalgamated data in NSE," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCCIC)*, Dec. 2017, pp. 1–5.
- [57] *Ballarat, Australia Dataset*. Accessed: Nov. 18, 2022. [Online]. Available: <https://data.gov.au/data/dataset/ballarat-garbage-collection>
- [58] *Austin, Texas Dataset*. (2019). [Online]. Available: <https://data.world/cityofaustin/mbnu-4wq9>
- [59] J. Barnard and X.-L. Meng, "Applications of multiple imputation in medical studies: From AIDS to NHANES," *Stat. Methods Med. Res.*, vol. 8, no. 1, pp. 17–36, Jan. 1999.
- [60] A. C. Acock, "Working with missing values," *J. Marriage Family*, vol. 67, no. 4, pp. 1012–1028, Nov. 2005.
- [61] R. Lall, "How multiple imputation makes a difference," *Political Anal.*, vol. 24, no. 4, pp. 414–433, 2016.
- [62] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, and M. Martín, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artif. Intell. Med.*, vol. 50, no. 2, pp. 105–115, 2010.
- [63] M. Saad, M. Chaudhary, F. Karray, and V. Gaudet, "Machine learning based approaches for imputation in time series data and their impact on forecasting," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 2621–2627.
- [64] X. Zhang, C. Yan, C. Gao, B. Malin, and Y. Chen, "XGBoost imputation for time series data," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Jun. 2019, pp. 1–3.
- [65] Y. Huang, C. Du, Z. Xue, X. Chen, H. Zhao, and L. Huang, "What makes multi-modal learning better than single (provably)," in *Proc. NeurIPS*, 2021, pp. 1–13.
- [66] F. J. Doblas-Reyes, R. Hagedorn, and T. N. Palmer, "The rationale behind the success of multi-model ensembles in seasonal forecasting—II. Calibration and combination," *Tellus A*, vol. 57, no. 3, pp. 234–252, May 2005.
- [67] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *Proc. AMSS 16th Int. Conf. Comput. Modelling Simulation (UKSim)*, Mar. 2014, pp. 106–112.
- [68] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [69] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.
- [70] M. Navratil and A. Kolkova, "Decomposition and forecasting time series in the business economy using prophet forecasting model," *Central Eur. Bus. Rev.*, vol. 8, no. 4, p. 26, 2019.
- [71] E. S. Gardner, Jr., "Exponential smoothing: The state of the art," *J. Forecasting*, vol. 4, no. 1, pp. 1–28, 1985.
- [72] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [73] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.
- [74] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.
- [75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [76] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," 2012, *arXiv:1211.5063*.
- [77] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," 2020, *arXiv:2001.08317*.
- [78] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
- [79] A. Sbrana, A. L. D. Rossi, and M. C. Naldi, "N-BEATS-RNN: Deep learning for time series forecasting," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2020, pp. 765–768.
- [80] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. Van Pottelbergh, M. Pasička, A. Skrodzki, N. Huguenin, M. Dumonal, J. Kościsz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, and G. Grosch, "Darts: User-friendly modern machine learning for time series," 2021, *arXiv:2110.03224*.



OSHAN MUDANNAYAKE received the B.Sc. degree from the University of Colombo. He is currently a Research Assistant at the University of Colombo School of Computing. His research interests include reinforcement learning and game theory.



DISNI RATHNAYAKE received the B.Sc. degree from the University of Colombo, Sri Lanka. She is currently working as a Research Assistant at the University of Colombo School of Computing, Sri Lanka. Her research interests include data analysis, machine learning, and deep learning applications.



JEROME DINAL HERATH received the Ph.D. degree from the Computer Science Department, State University of New York (SUNY) at Binghamton, USA. He is currently a Security Data Scientist at Obsidian Security, USA. His research interests include applications of machine learning and deep learning, adversarial artificial intelligence (AI), and explainable AI.



DINUNI K. FERNANDO received the Ph.D. degree from the Computer Science Department, State University of New York (SUNY) at Binghamton, USA. She is currently a Senior Lecturer at the University of Colombo School of Computing. Her research interests include virtualization, networking, applied machine learning, and security.



MGNAS FERNANDO (Member, IEEE) received the Ph.D. degree from the University of Colombo, Sri Lanka. He is currently a Professor of computer science at the University of Colombo School of Computing. His research interests include data mining, ICT education in Sri Lanka, algorithms, MIS, e-government, applied machine learning, and blended learning.