WSB Journal of Business and Finance
Year 2019, Vol. 53, No. 2
eISSN 2657-4950

Machine learning model development for predicting road transport GHG emissions in Canada

Mohd Jawad Ur Rehman Khan[a1], Anjali Awasthi[a]

[a]*Concordia Institute for Information Systems Engineering (CIISE), Faculty of Engineering & Computer Science, Concordia University, Montreal, Quebec H3G 1M*

DOI 10.2478/WSBJBF-2019-0022

Abstract

Prediction of greenhouse gas (GHG) emissions is important to minimise their negative impact on climate change and global warming. In this article, we propose new models based on data mining and supervised machine learning algorithms (regression and classification) for predicting GHG emissions arising from passenger and freight road transport in Canada. Four models are investigated, namely, artificial neural network multilayer perceptron, multiple linear regression, multinomial logistic regression and decision tree models. From the results, it was found that artificial neural network multilayer perceptron model showed better predictive performance over other models. Ensemble technique (Bagging & Boosting) was applied on the developed multilayer perceptron model, which significantly improved the model's predictive performance.

Keywords: Supervised machine learning; Feature selection; Multiple Linear Regression; Multilayer Perceptron, Ensemble learning; Bagging.

1 Introduction

Greenhouse gases, commonly referred as GHGs, are the natural and anthropogenic gaseous constituents of the atmosphere. They absorb and emit radiations emitted by earth's surface, atmosphere and clouds at specific wavelengths between spectrums of thermal infrared radiation (Metz et al., 2007). The intensity of GHGs has increased quickly because of the increased anthropogenic activities along with the increase in population, increasing earth's temperature. GHGs absorb the energy radiated by the sun causing the lower layer of the atmosphere to trap the heat and raise its temperature; this phenomenon is called natural greenhouse gas effect. This natural phenomenon got amplified since the advent of industrialisation and urbanisation.

The major greenhouse gases are carbon dioxide (CO_2), methane (CH_4), nitrous oxide (N_2O), hydrofluorocarbons (HFCs), sulphur hexafluoride (SF_6) and perfluorocarbons (PFCs). Of these major gases, the most dominant is CO_2, which accounts 77% of global CO_2 equivalent causing global warming (Metz et al., 2007).

Road transport emission models are extensively focused on estimating vehicle emissions inventory by considering only freight relevant and meteorological data for, for example, vehicle types, fuel type and driving speed. Many studies focus on only calculating emission factors using several emission monitoring and inventorying tools/models to calculate the emission with respect to region, vehicle type and so on. In addition,

- Prediction of current and future GHG emissions using inventorying software is complicated as predefined input variables and extensive field knowledge are required.
- Limited machine learning models are available for modelling GHG emissions from road transport sector using independent and not predefined input variables.

[1] Corresponding author:.e-mail address: Anjali.awasthi@concordia.ca

2 Research Approach

In this article, we present an alternative method for modelling and predicting GHG emissions specifically from road transportation (passenger and freight). The models are developed using machine learning approach because, the models learn the relationship between inputs and outputs by adapting and capturing historical data and the underlying functional relationship and with the help of learning on historical data, future predictions are performed on unseen data set.

Machine learning models are less complex, need a small number of inputs and have minimal in-depth field knowledge, and most notably, inputs are not predetermined.

We undertook the study of data mining and machine learning models to predict the GHG emission caused by road transportation using socioeconomic, demographic and emission input data for Canada. We developed models using logical, perceptron and statistics techniques and algorithms, that is, decision tree (C4.5), multilayer perceptron and multiple linear regression and multinomial logistic regression, respectively. Ensemble learning techniques will implement model performance improvement techniques (ensemble learning) on the best performing machine learning model to further improve its performance.

3 Literature review

The main source of GHG emission data is GHG inventories (National Inventory Submissions 2017). These inventories contain a large number of input parameters, which are used to calculate total emissions. Each model uniquely uses this parameter to determine the final total emission. Most emission sectors such as oil and gas, electricity, transportation, heavy industry and buildings are the product of a statistical parameter of the respective source, that is, activity data (A) and an emission factor (EF) (Winiwarter et al., 2001).

GHG Emissions=Activity data ×Emission factor

Activity data refer to the estimated quantitative amount of human activity resulting in emissions during a given time period, for example, the total amount of fossil fuel burned is the activity data for fossil fuel combustion sources (Government of Canada, Environment and Climate Change Canada 2017).

The emission factor is the average emission rate of a given GHG for a given source, relative to units of activity. It relates the quantity of a pollutant released to the atmosphere with an associated activity, for example, kilograms of particulate emitted per mega gram of coal burned.

Research gap

A review of studies in the area points out that a limited number of studies have been performed on the topic of road transport GHG emissions using data mining and machine learning models and using independent and widely available indicators such as socio-economic parameters, emission data and fuel efficiency compared with pre-determined input variables. The following were the gaps observed in the literature:

- Emission models are extensively focused on estimating vehicle emissions inventory by considering activity and emissions factors based on freight relevant, distance and meteorological data.
- Limited number of studies has been performed on the topic of road transport GHG emissions projection by using data mining and machine learning models.

Ensemble learning techniques have not been used for improving model's performances for road transport GHG emissions modelling.

4 Solution approach

We used data mining approaches to develop alternate models for emissions prediction. Data mining is about explaining the past and predicting the future using data analysis and modelling. It is a multi-disciplinary domain that combines statistics, machine learning and database technology (Sayad 2011). The most significant application of data mining is machine learning. In this section, we will discuss data mining approaches used in this article.

Feature selection method helps in achieving the following aims (Shardlow, 2016):

- To reduce the size of the problem – reducing compute time and space required to run machine learning algorithms.
- To improve the predictive accuracy of classifiers: first by removing noisy or irrelevant features and second by reducing the likelihood of overfitting to noisy data
- To identify which features may be relevant to a specific problem.

In our research, we implemented RReliefF filter method for feature selection using attribute evaluator and search method of the Wakaito Environment of Knowledge Analysis (WEKA) to determine the set of relevant input indicators amongst the field of socio-economic, demographic and emission data.

A. Multiple Linear Regression

The purpose of linear regression analysis is to evaluate the relative impact of a predictor variable on a particular outcome. Regression with the single attribute is called as simple linear regression and regression with multiple attributes is called as multiple linear regression. Multiple linear regression helps in the easy fitting of models, which depends linearly on their attributes. Regressions are extensively used statistical tool in various practical applications, majority of them being forecasting and predictive modelling (Yan et al., 2009)

Considering a given data set $\{ y_i, x_{i1}, x_{i2}, \ldots, x_{ik}\}$ where i = 1 to n, the linear Regression model is given by (Lang, H. 2013)

$$y_i = \beta_0 + x_{i1}\beta_1 + x_{ik}\beta_k + e_i$$

where $i = 1,2,3, \ldots n$,
y_i is the dependent variable,
x_{ik} is the independent variable for the dependent variable y_i,
β_k is the unknown parameters (to be estimated from data),
e_i is the error term.

B. Multinomial Logistic Regression

Logistic regression, also called logit model, is a statistical modelling technique. It evaluates the relationship between multiple independent variables and categorical dependent variable and estimates the probability of occurrence of an event by fitting data to a logistic curve. Depending on the type and value of dependent variable, logistic regression can be classified as binary and multinomial logistic regression models (Hosmer & Lemeshow 2000). When the dependent variable is not dichotomous and is consisted of more than two categories, a multinomial logistic regression can be used (Hosmer et al., 2013).

The impact of independent variables is usually explained in terms of odds, as multinomial logistic regression estimates the probability of an event occurring over the probability of an event not occurring. The multinomial logistic function is used when the dependent variable has *k* possible outcomes. MNL uses a linear predictor function *f(k, i)* to predict the probability that observation *i* has outcome *k.*

The function can be described as follows:
$f(k,i) = \beta_{0,k} + \beta_{1,k}x_{1,i} + \beta_{2,k}x_{2,i} + \ldots + \beta_{M,k}x_{M,i}$
$f(k,i) = \beta_{0,k} + \beta_k . X_i$
where
X_i is the set of independent variable.
β_k is set of regression coefficients associated with outcome *k.*

Unfortunately, the probability given by this function is not a good model because extreme values of *x* will give values of $\beta_{0,k} + \beta_k . X_i$, and these values do not fall between 0 and 1. The logistic regression solution to this problem is to transform the odds using the natural logarithm (Peng et al., 2002).

When there are *K* possible categories of the response variable, the model consists of $k - 1$ simultaneously logit equation. With multinomial logistic regression, we model the natural log odds as a linear function of the explanatory variable:

Logit (Y) = ln $\frac{P_r(y_i=k-1)}{P_r(y_i=k)} = \beta_{0,k} + \beta_k . X_i$

to calculate and interpret the effect of an independent variable; it is good to take exponential of both sides of the equation to get predicted probabilities (Wattimena 2014).

$$P_r(Y_i = k-1) = \frac{e^{\beta_{k-1}.X_i}}{1 + \sum_{k=1}^{k-1} e^{\beta_k.X_i}}$$

The probability of the reference category, '*K'* can be calculated as (Wang 2005)

$$\left(P_r(Y_i = k)\right) = 1 - \left(\frac{e^{\beta_{k-1}.X_i}}{1 + \sum_{k=1}^{k-1} e^{\beta_k.X_i}}\right)$$

C. C4.5 Decision Tree

There were few limitations for the ID3 algorithm, and in 1993, Ross Quinlan proposed C4.5 to overcome those limitations. C4.5 algorithm acts similar to ID3 but improves a few of ID3 behaviours (Hssina et al., 2014):

- Possibility to use continuous data,
- Using unknown (missing) values,
- Ability to use attributes with different weights,
- Pruning the tree after being created.

The best attribute is chosen as the test at the root node of the tree. If the attribute is discrete in nature, a descendant of the root node is created for each possible value of this attribute. If the attribute is continuous in nature a descendant of the root node is created for each possible discretised interval of this attribute. Next, the training samples are sorted to the suitable descendant node. Furthermore, the process is repeated using the training samples related with each descendant node to choose the best attribute specific at that point in the tree, for testing. **J48** is an implementation of the C4.5 algorithm in the WEKA data-mining tool.

Alike ID3 the statistical test used in C4.5 also uses an entropy-based measure for allocating an attribute to each node in the tree. Similar to ID3, the data is sorted at every node of the tree to determine the best splitting attribute. The difference is C4.5 uses gain ratio impurity method to evaluate the splitting attribute (Quinlan 1993). At every node, C4.5 selects data attribute that best splits data into subsets rich in one class or the other. The selection criterion is the normalised information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalised information gain is chosen to make the decision (Quinlan 1993) (Hssina et al., 2014). The information gain ratio is given by:

$$GainRatio(p,T) = \frac{Gain\ (p,T)}{SplitInfo(p,T)}$$

where

$$Gain(p,T) = Entoropie(p) - \sum_{j=1}^{n}(p_j \times Entoropie\ (p_j))$$

$$SplitInfo(p,test) = -\sum_{j=1}^{n} P'\left(\frac{j}{p}\right) \times \log(P'\left(\frac{j}{p}\right))$$

$P'\left(\frac{j}{p}\right)$ is the proportion of elements present at the position p, taking the value of *j*th test.

D. Multilayer Perceptron

The most significant invention in the field of soft computing is neural networks (NNs), inspired by biological neurons in the human brain. The concepts of NNs were first mathematically modelled by McCulloch and Pitts (McCulloch et al., 1943). Single-layer perceptron (SLP) and multi-layer perceptron (MLP) are two types of FNN. The difference between the two types is the number of perceptron. SLP has a single perceptron, and MLP has more than one perceptron. MLPs are proficient in solving nonlinear problems (Werbos 1974). The applications of MLPs are categorised as pattern classification, data prediction and function approximation.

The MLP model is a flexible and general-purpose type of ANN composed of one input layer, one or more hidden layers and one output layer (Dawson et al., 1998). MLPs are fully connected feed-forward nets with one or more layers of nodes between the input and the output nodes. Upon receiving a given number of inputs, each neuron calculates a linear combination of the inputs using synaptic weights w_i to generate the weighted input *z*; then, it provides an output *y* via an activation function.

Figure 1 shows an MLP with three layers, where the number of input nodes is *n*, the number of hidden nodes is *h* and the number of output nodes is *m*.

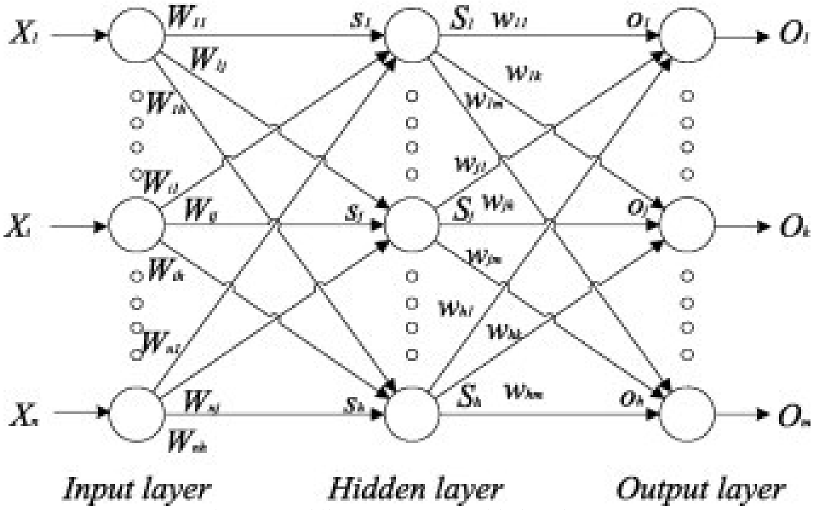


Figure 1 Multilayer perceptron with three layers

The output of the MLP is calculated as follows (Mirjalili et al., 2014):

Step 1: The following equation first calculates the weighted sums of inputs:

$$s_j = \sum_{i=1}^{n} (W_{ij} X_i) - \theta_j, \qquad j = 1,2, \dots\dots h$$

where n is the number of the input nodes, W_{ij} shows the connection weight from the ith node in the input layer to the jth node in the hidden layer, θ_j is the bias (threshold) of the jth hidden node, and X_i indicates the ith input.

Step 2: The output of each hidden node is calculated as follows:

$$S_j = sigmoid\ (s_j) = \frac{1}{(1 + e^{(-s_j)})}\ ,\ \ j = 1,2, .. h$$

Step 3: After calculating the outputs of hidden nodes, the final outputs are defined as below:

$$o_k = \sum_{j=1}^{h} (W_{jk} S_j) - \theta'_k, \qquad k = 1,2, \dots\dots m$$

$$O_k = sigmoid\ (o_k) = \frac{1}{(1 + e^{(-o_k)})}\ ,\ \ k = 1,2, .. m$$

where W_{jk} is the connection weight from the jth hidden node to the kth output node and θ'_k is the bias (threshold) of the kth output node.

The most important parts of MLPs are the connection weights and biases. Training an MLP involves finding optimum values for weights and biases to achieve desirable outputs from certain given inputs (Mirjalili et al., 2014). Back propagation algorithm is a popular technique to train the MLP model. The comparison of the network's output to the target value is initiated, and the difference (or error δ) is calculated. The error parameter is used during the weight-correction procedure (Lek and Park 2008).

Pseudo code for Back propagation learning algorithm in the MLP (Lek and Park 2008):

```
1. Randomise the weights $w$ to small random values,
2. Select an instance $t$, a pair of input and output patterns from the training set,
3. Apply the network input vector to the network,
4. Calculate the network output vector $z$,
5. Calculate errors for each of the outputs $k$, the difference ($\delta$) between the desired output and the network output,
6. Calculate the necessary updates for weights $\Delta w$ in a way that minimises this error,
7. Add up the calculated weights' updates $\Delta w$ to the accumulated total updates $\Delta w$,
8. Repeat steps 2–7 for several instances comprising an epoch,
9. Adjust the weights w of the network by the updates $\Delta w$,
10. Repeat steps 2–9 until all instances in the training set are processed. This constitutes one iteration,
11. Repeat the iteration of steps 2–10 until the error for the entire system (error $\delta$ defined above or the error on cross-validation set) is acceptably low, or the predefined number of iterations is reached.
```

The backpropagation algorithm executes gradient descent on the error surface by adjusting each weight as shown in Figure 2. The adjustment in weight is made in proportion to the gradient of the surface at its location. The learning rate η and the momentum term α play a vital role in the learning process of backpropagation network.

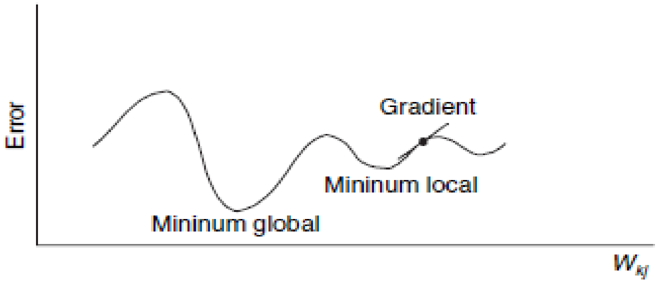


Figure 2 Error surface as function of a weight showing gradient and local and global minima. Source: (Lek and Park 2008)

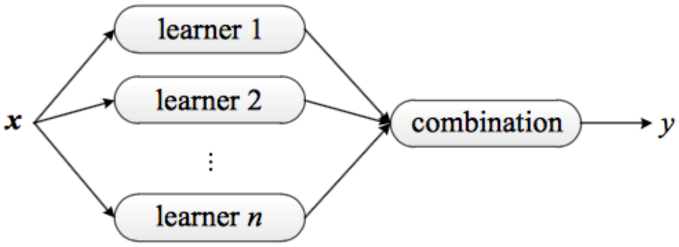


Figure 3 General ensemble architecture. Source: (Zhou 2012).

Efficient selection of the values of these parameters is important to avoid the network getting into oscillation and getting stuck in local minimum (Lek and Park 2008).

E. Ensemble Learning

Ensemble learning techniques train multiple classifiers instead of just one classifier to solve the same learning problem (Zhou 2012). The resulting classifier is generally more accurate than any of the individual classifiers making up the ensemble (Opitz et al., 1999) as shown in Figure 3. An ensemble contains a number of classifiers called base learners. Base learners are usually generated from training data by a base learning algorithm that can be decision tree, neural network or other kinds of learning algorithms. Ensemble methods construct a set of learners and combine them. Base learners are also called as weak learners because the generalisation power of an ensemble is usually stronger than base learner and, hence, provide improved prediction accuracy. The individual decisions of base learners in an ensemble are combined in some way (usually either by averaging or weighted/unweighted voting) to classify new examples.

Extensive work by the researchers in this domain led to the birth of a popular method for creating accurate ensembles, that is, bagging (Breiman 1996). In this research, we used ensemble methods such as bagging to improve the prediction accuracy of best-performing machine learning model for GHG emission by road transport in Canada.

Bagging

It is most commonly known as bootstrap aggregation. The two important elements of bagging algorithm are bootstrap and aggregation (Breiman 1996).

Bagging deploys bootstrap sampling to obtain the data subsets for training the base learners. Consider a training data set containing m number of training examples; sampling with replacement will generate a sample of m training examples. Some original examples may appear more than once, whilst some original examples are not present in the sample. Repeating the process T times, T samples of m training examples are obtained. Then, from each sample, a base-learner/classifier can be trained by applying the base-learning/classifier algorithm (Zhou 2012) (Breiman 1996). Each bootstrap replicates contain, on an average, 63.2% of the original training set, with multiple repetitions of example from the training set. In addition, bagging reduces variance (Breiman 1996; Dietterich 2000). Bagging uses voting for classification and averaging for regression to aggregate the outputs of the base learners (Zhou 2012). Bagging algorithm is shown in Figure 4.

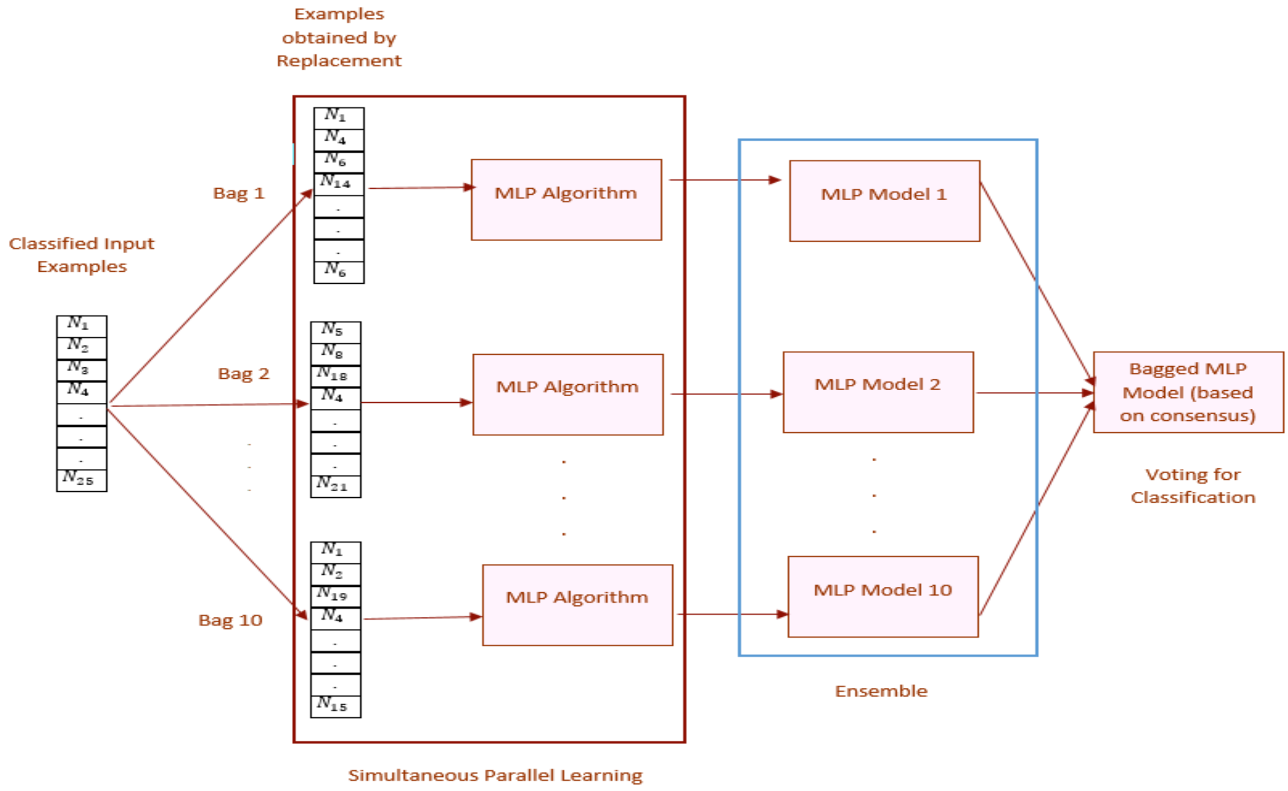
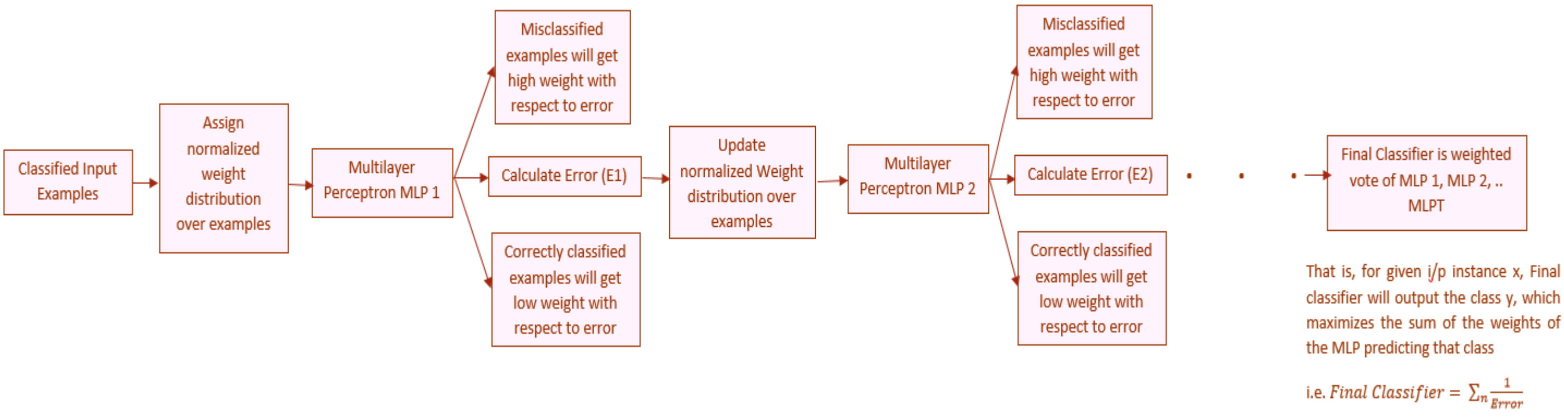

Examples obtained by Replacement
Bag 1
Classified Input Examples
Bag 2
Bag 10
N_1 N_2 N_3 N_4 . . . N_{25}
N_1 N_4 N_6 N_{14} . . . N_6
N_5 N_8 N_{18} N_4 . . . N_{21}
N_1 N_2 N_{19} N_4 . . . N_{15}
MLP Algorithm
MLP Algorithm
MLP Algorithm
MLP Model 1
MLP Model 2
MLP Model 10
Bagged MLP Model (based on consensus)
Voting for Classification
Ensemble
Simultaneous Parallel Learning

Figure 4 Bagging algorithm

Boosting

Boosting (Freund et al., 1996) ensemble method produces a series of classifiers. On the basis of the performance of the previous classifier(s) in series, the training set used for each member classifier of the series is chosen. Boosting algorithm is shown in Figure 5.

According to the logic of boosting algorithm, it gives less emphasis on correctly classified examples by the classifier in series and gives more emphasis on previously misclassified examples by a classifier in series by choosing them more frequently compared to correctly predicted examples. In general, the boosting algorithm tries to generate new classifiers that are better able to predict examples for which the current ensemble's performance is poor (Opitz et al., 1999).


Classified Input Examples
Assign normalized weight distribution over examples
Multilayer Perceptron MLP 1
Misclassified examples will get high weight with respect to error
Calculate Error (E1)
Correctly classified examples will get low weight with respect to error
Update normalized Weight distribution over examples
Multilayer Perceptron MLP 2
Misclassified examples will get high weight with respect to error
Calculate Error (E2)
Correctly classified examples will get low weight with respect to error
. . .
Final Classifier is weighted vote of MLP 1, MLP 2, .. MLPT

That is, for given i/p instance x, Final classifier will output the class y, which maximizes the sum of the weights of the MLP predicting that class

i.e. $Final\ Classifier = \sum_n \frac{1}{Error}$

Figure 5 Boosting algorithm

The most popular boosting procedure is AdaBoost-M1 (Adaptive Boosting). This procedure allows continuing adding weak learners until some desired low training error is achieved.

4 Application for Canada

We divided the application to Canada into three different phases, as shown in Figure 6; in this article, we focus on phase 2, that is, machine learning model development. We adapted the data mining/machine learning techniques to perform predictive modelling of GHG emissions caused by road transportation (passenger and freight). We will implement ensemble learning algorithm to improve the predictive performance of best performing model.

Data collection: The socio-economic, demographic and emission data were collected from GHG inventory sink of Canada, Statistics Canada, CAFC targets and fleet average website and trading economics. GHG inventory sink reports emission figures by vehicle type, Statistics Canada and trading economics reports values for socio-economic indicators, and we used transport policy.net for fleet average reports for fleet fuel efficiency values for passenger cars and light duty trucks.

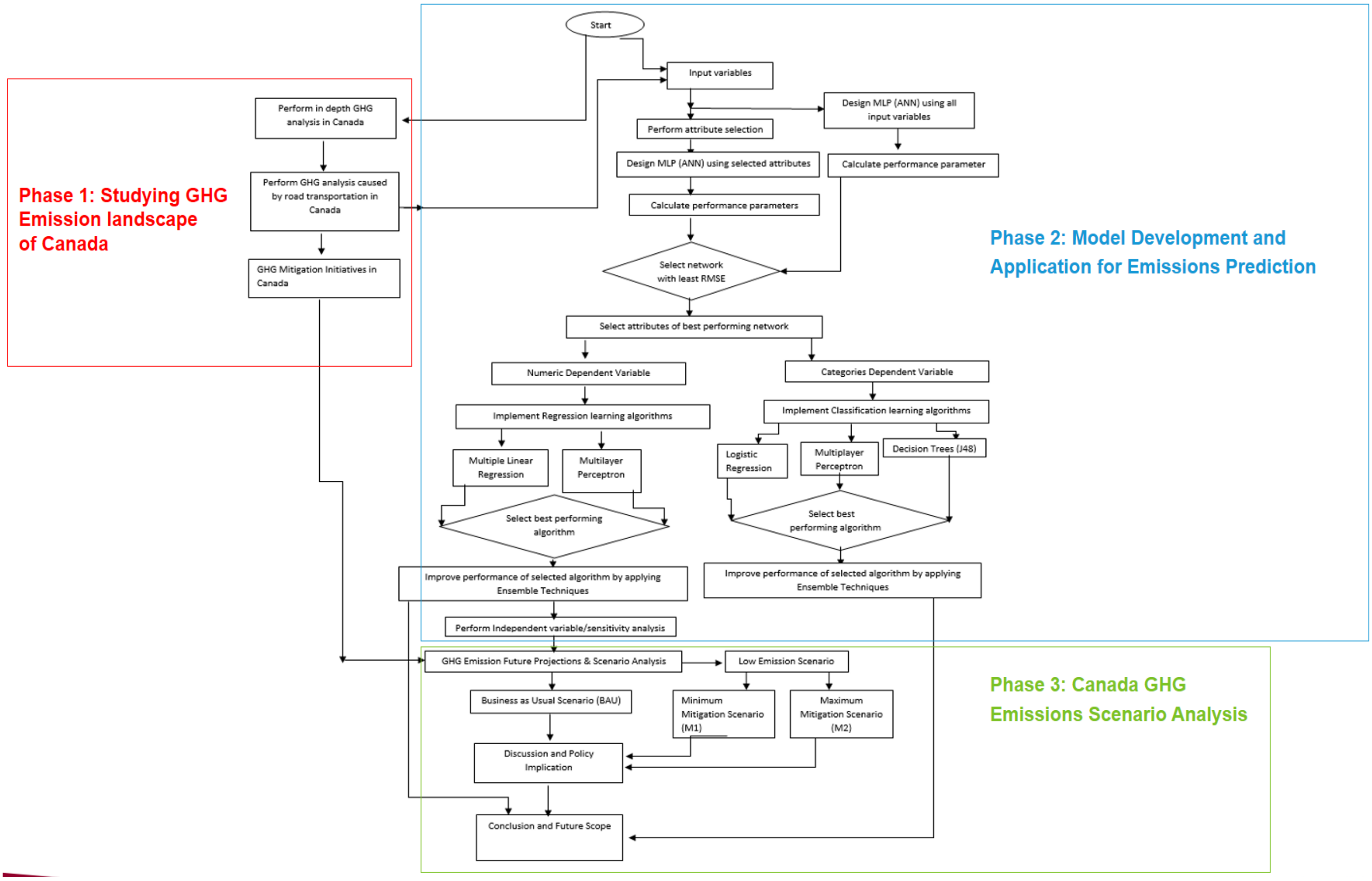

Phase 1: Studying GHG Emission landscape of Canada
Perform in depth GHG analysis in Canada
Perform GHG analysis caused by road transportation in Canada
GHG Mitigation Initiatives in Canada
Start
Input variables
Design MLP (ANN) using all input variables
Perform attribute selection
Design MLP (ANN) using selected attributes
Calculate performance parameter
Calculate performance parameters
Select network with least RMSE
Phase 2: Model Development and Application for Emissions Prediction
Select attributes of best performing network
Numeric Dependent Variable
Categories Dependent Variable
Implement Regression learning algorithms
Implement Classification learning algorithms
Multiple Linear Regression
Multilayer Perceptron
Logistic Regression
Multiplayer Perceptron
Decision Trees (J48)
Select best performing algorithm
Select best performing algorithm
Improve performance of selected algorithm by applying Ensemble Techniques
Improve performance of selected algorithm by applying Ensemble Techniques
Perform Independent variable/sensitivity analysis
GHG Emission Future Projections & Scenario Analysis
Low Emission Scenario
Business as Usual Scenario (BAU)
Minimum Mitigation Scenario (M1)
Maximum Mitigation Scenario (M2)
Discussion and Policy Implication
Conclusion and Future Scope
Phase 3: Canada GHG Emissions Scenario Analysis

Figure 6 Research steps

Phase 2: Supervised learning model development (Regression and Classification) and applications for emissions prediction

3.1 Attribute selection (ranking)

We implemented RReliefF Algorithm in WEKA (capable of performing RReliefF). We used an input vector X [Year, Carsales, Gasoline Price CAD Later, GDP transportation, Interest Rate, CPI, Car Emission, Light Trucks Emission, Medium Trucks Emission, Heavy Trucks Emission, Buses Transit Emission, Population(million), Passenger Car Fuel Efficiency, Light Duty Truck Fuel Efficiency, Total GHG (only Road)] 25*15. In WEKA explorer, we chose attribute evaluator and search method and observed the rank of input attributes. Figure 7 shows the rank of attributes as determined by WEKA for GHG emission prediction.

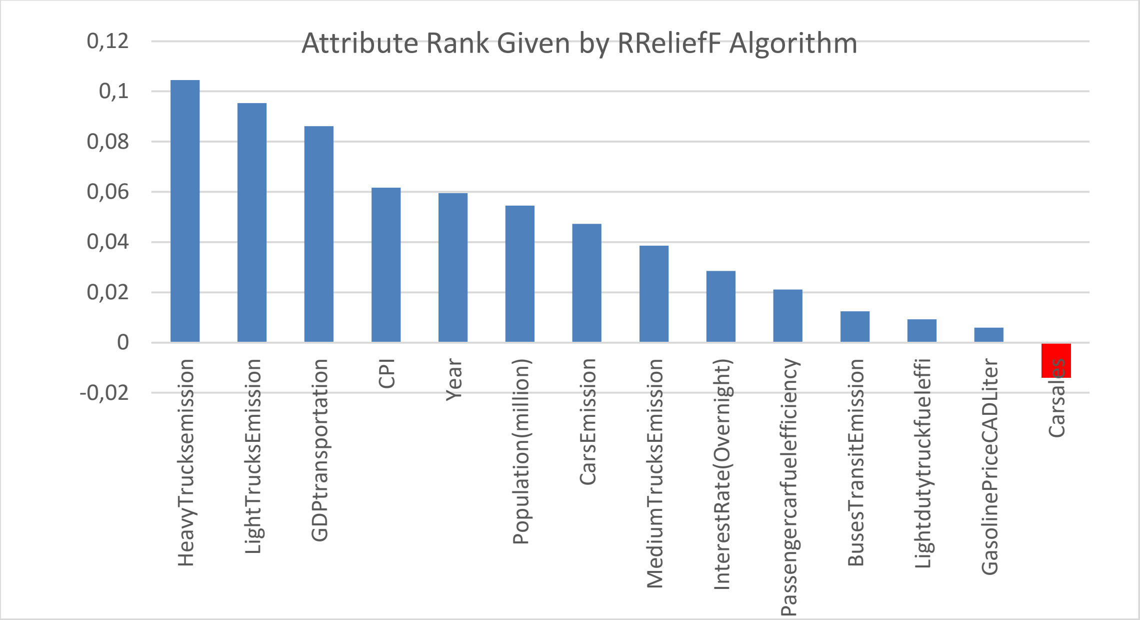


Figure 7 Attribute rank given by RReliefF algorithm

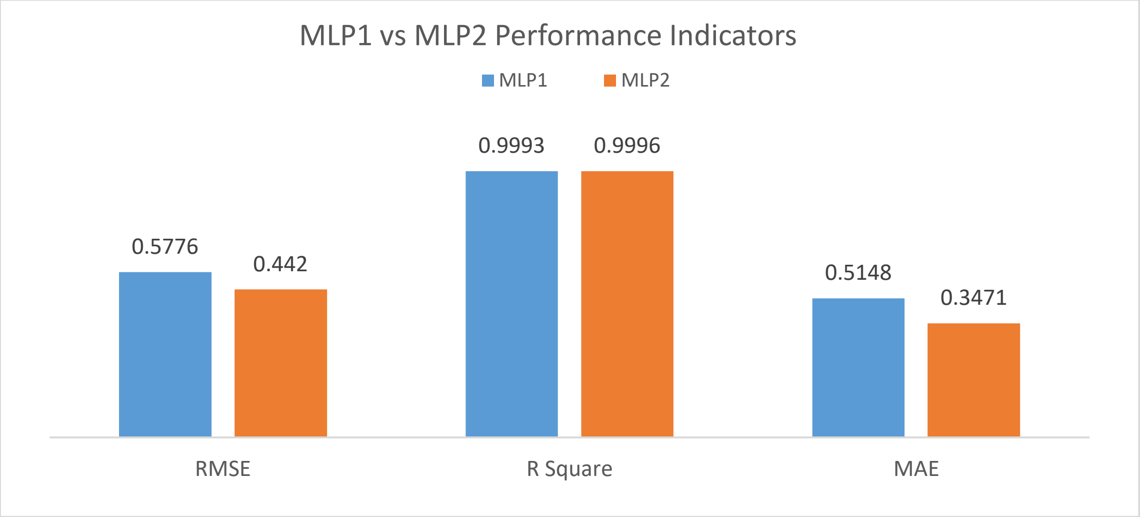


Figure 8 MLP1 vs. MLP2 performance indicators

Two MLP models are developed to authenticate relevant attributes given by RReliefF Algorithm. Model MLP1 with all input attributes and Model MLP2 with Relief algorithm selected attributes (excluding car sales since it got negative ranking). The MLP models are developed in WEKA. Total numeric values of GHG emission by road transport were selected as the dependent variable, and the remaining attributes were used as covariates.

The prediction accuracy of numeric GHG emission was evaluated with the help of performance indicators. MLP2 with attributes selected by RReliefF algorithm performs better compared with MLP1 with all available inputs as attributes.

Figure shows the results that after removing less influencing attribute (car sales), the model MLP2 error rates of RMSE and MAE decreased to 0.442 and 0.3471, respectively, and correlation coefficient value slightly increased to 0.9996, proving that generalising performance of machine learning models will improve with relevant input attributes.

3.2 Machine learning model development for emissions predictions

We developed supervised machine learning models for nominal and numeric GHG emission data.

A. Model development for nominal GHG values

To explore classification supervised machine learning models, we converted our numeric dependent variable into a nominal variable. There are two approaches to deal with the multi-class problem for classifiers one-vs-one (OVO) and one-vs-all (OVA) (Galar et al., 2011). We used OVA in WEKA tool.

Classified socio-economic, emissions and fuel efficiency data were selected with 10-fold cross-validation technique to avoid the problem of overfitting and to check the generalisation by the model when applied to independent/unknown data set. The voted averaged evaluation results after 10-fold cross-validation were given by WEKA under cross-validation summary. On the 11th run, WEKA runs the algorithm on the data set and provides the final deplorable model.

1. Multinomial Logistic Regression

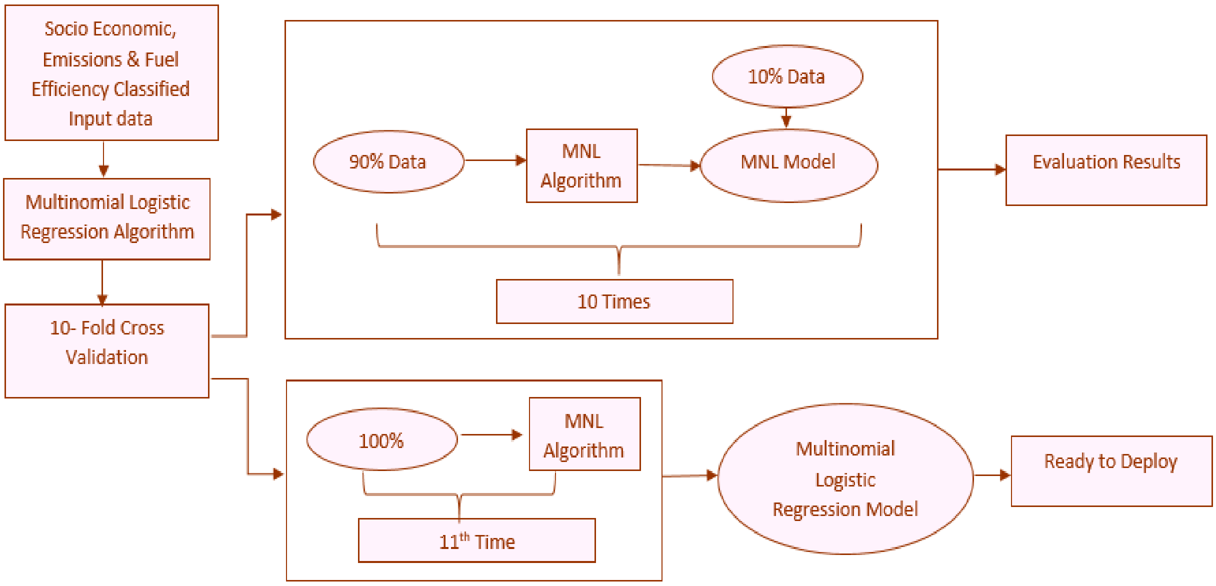

Socio Economic, Emissions & Fuel Efficiency Classified Input data
Multinomial Logistic Regression Algorithm
10- Fold Cross Validation
90% Data
MNL Algorithm
10% Data
MNL Model
10 Times
Evaluation Results
100%
MNL Algorithm
11th Time
Multinomial Logistic Regression Model
Ready to Deploy

Figure 9 Multinomial logistic regression model development

Multinomial logistic regression model development is shown in Figure 9.

2. Decision Tree

As shown in Figure 10, light duty truck efficiency has been chosen as the root node. It has the highest information gain and gain ratio compared with other attributes and, hence, was selected as the best splitting attribute. Analysing the below C4.5 decision tree given by WEKA, we can see that the algorithm calculates a threshold 10.8; in this case, it has two branches, that is, the values less than 10.8 and those greater than 10.8.

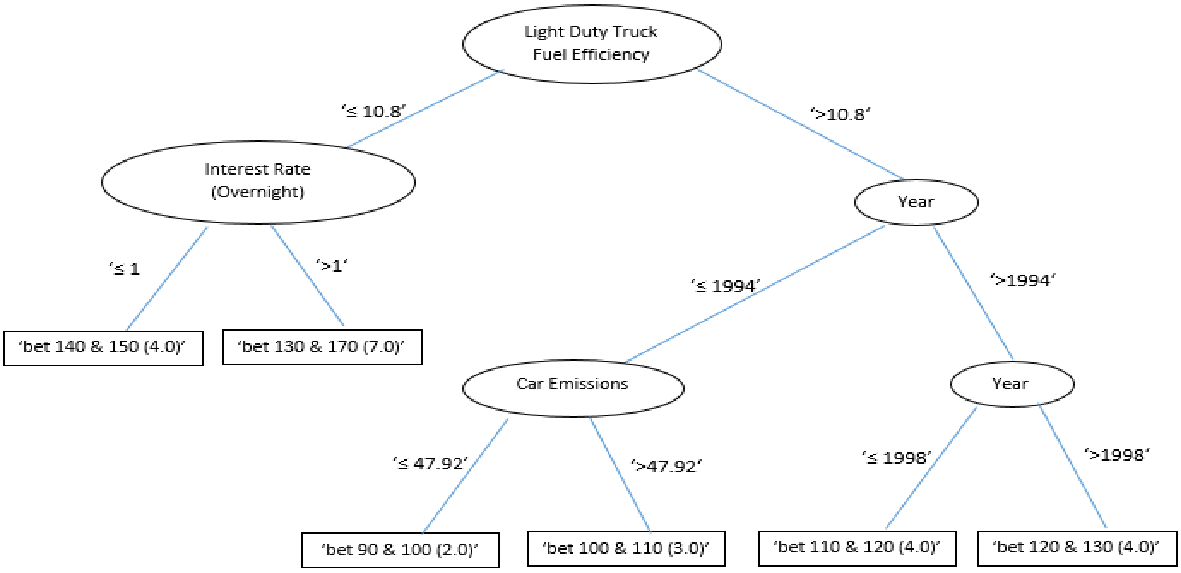


Figure 10 Decision tree model

3. Multilayer Perceptron

Learning parameters plays a vital role in fine-tuning of MLP model; in case the performance parameters given by cross validation are not satisfactory; the network can be fine-tuned by changing learning rate, momentum and number of epochs (or training time). Hence, cross-validation is an important validation technique because its results impact the network training.

Figure 11 shows the MLP model development. On the 11th run, WEKA develops the MLP network, which is shown in Figure 12. MLP NN for categorical dependent data is a three-layer network: input layer, hidden layer and output layer. The weights are given for each attribute that feeds into each sigmoid node plus the threshold (bias) weight. The output nodes have a feed of weight and threshold from the nine hidden neurons.

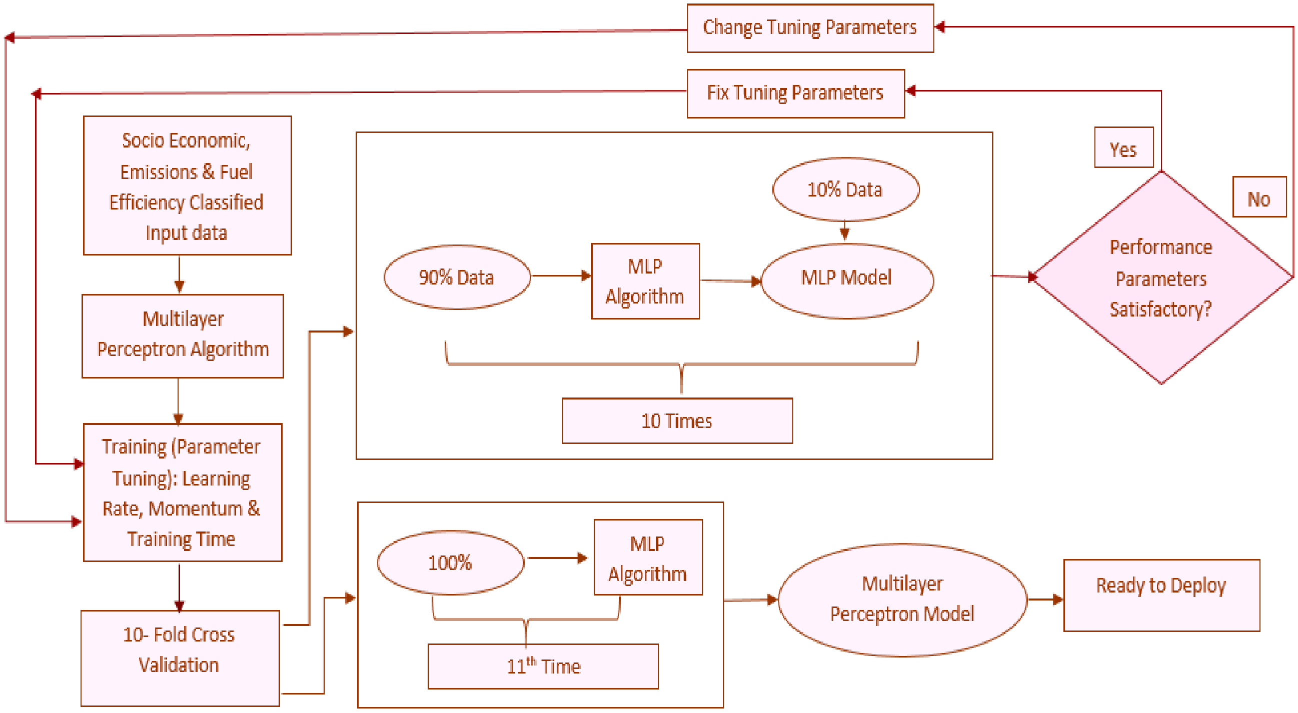

Change Tuning Parameters
Fix Tuning Parameters
Socio Economic, Emissions & Fuel Efficiency Classified Input data
Multilayer Perceptron Algorithm
Training (Parameter Tuning): Learning Rate, Momentum & Training Time
10- Fold Cross Validation
10% Data
90% Data
MLP Algorithm
MLP Model
10 Times
Yes
No
Performance Parameters Satisfactory?
100%
MLP Algorithm
11th Time
Multilayer Perceptron Model
Ready to Deploy

Figure 11 Multilayer perceptron model development

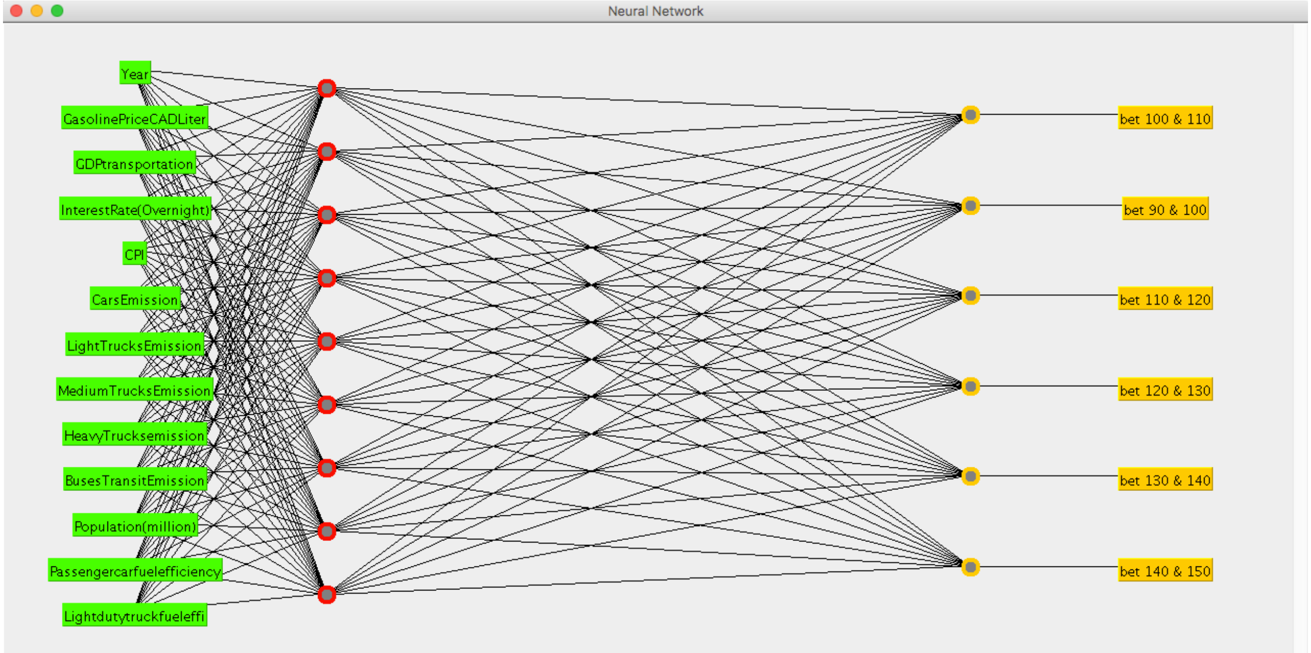


Figure 12 Multilayer perceptron network

4. Bagging

The bagging MLP model development is shown in Figure 13; we used 10 iteration for bagging algorithm and used 10 fold cross-validation, which means for each bag, 10 MLP classifiers were trained and combined using averaging. Finally, for classification, majority voting is performed for all 10 bags and the model is selected. Figure 12 shows MLP for bagging network.

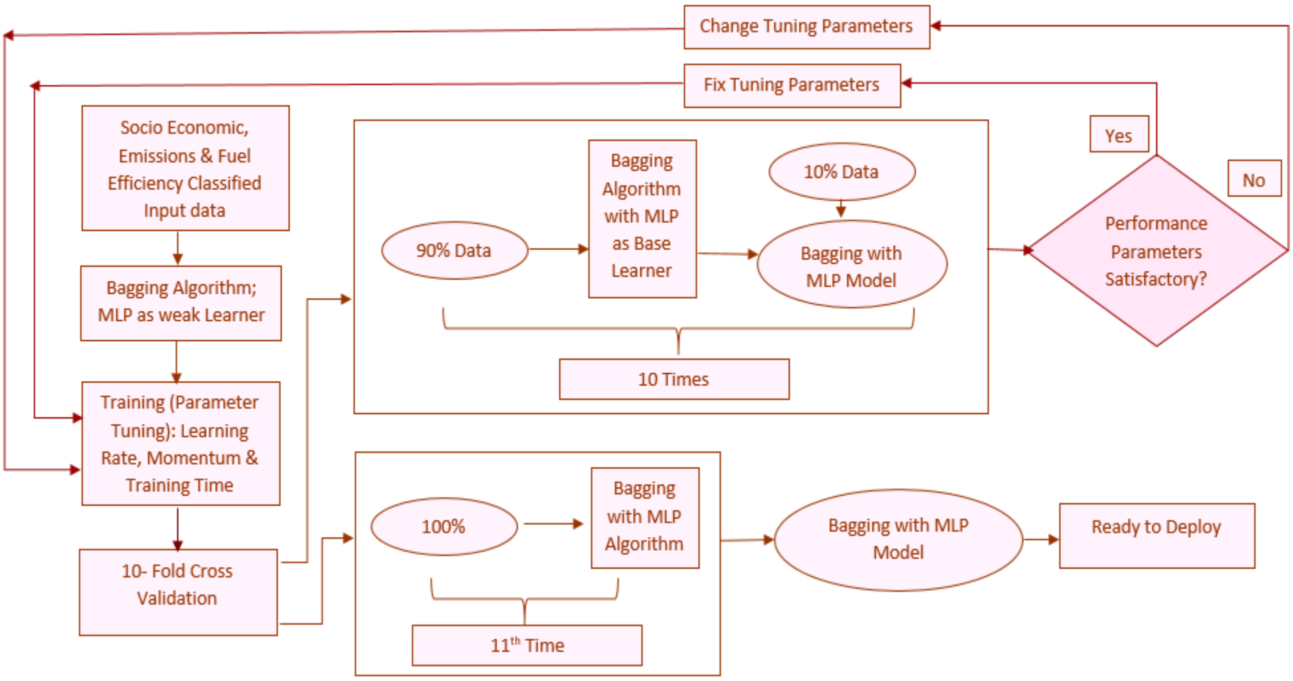

Change Tuning Parameters
Fix Tuning Parameters
Socio Economic, Emissions & Fuel Efficiency Classified Input data
Bagging Algorithm; MLP as weak Learner
Training (Parameter Tuning): Learning Rate, Momentum & Training Time
10- Fold Cross Validation
90% Data
Bagging Algorithm with MLP as Base Learner
10% Data
Bagging with MLP Model
10 Times
Yes
No
Performance Parameters Satisfactory?
100%
Bagging with MLP Algorithm
11th Time
Bagging with MLP Model
Ready to Deploy

Figure 13 Bagging MLP model development

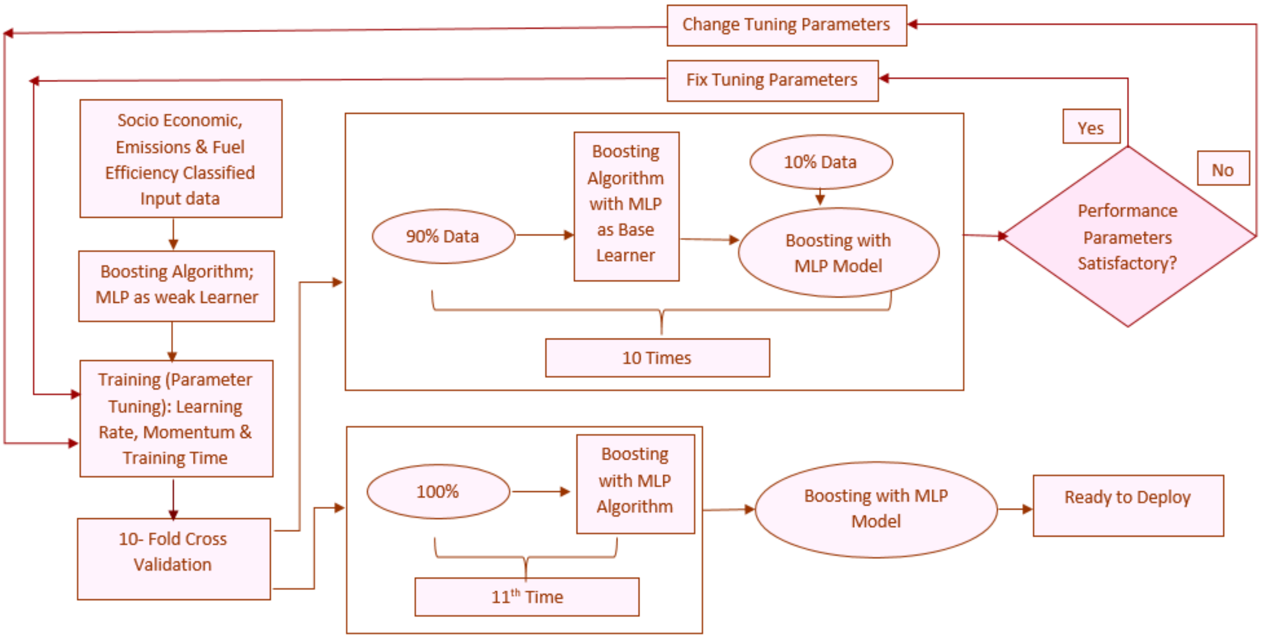

Change Tuning Parameters
Fix Tuning Parameters
Socio Economic, Emissions & Fuel Efficiency Classified Input data
Boosting Algorithm; MLP as weak Learner
Training (Parameter Tuning): Learning Rate, Momentum & Training Time
10- Fold Cross Validation
90% Data
Boosting Algorithm with MLP as Base Learner
10% Data
Boosting with MLP Model
10 Times
Yes
No
Performance Parameters Satisfactory?
100%
Boosting with MLP Algorithm
11th Time
Boosting with MLP Model
Ready to Deploy

Figure 14 Boosting MLP model development

5. Boosting

The boosting algorithm with 10 iterations was also evaluated using 10-fold cross-validation technique. The boosting algorithm invokes weak learner (base algorithm) repeatedly in a series of rounds. The boosting MLP model development is shown in Figure 14. We used 10 iterations for boosting algorithm, and we used 10-fold cross-validation, which means, for each boosting iteration, 10 MLP classifiers were trained and combined using averaging. Finally, for classification, majority voting is performed for all 10 bags and the model is selected. That is, for a given input x, final classifier will output the class y, which maximises the sum of weights of MLP predicting that class. Figure 12 shows MLP with boosting network.

B. Model development for numeric GHG values

Classified socio-economic, emissions and fuel efficiency data were selected with 10-fold cross-validation technique to avoid the problem of overfitting and to check the generalisation by the model when applied to independent/unknown data set. Total numeric values of GHG emission by road transport were selected as the dependent variable, and the remaining attributes were used as covariates. The averaged evaluation results after 10-fold cross-validation were given by WEKA under cross-validation summary.

1. Multiple Linear Regression

The MLR model is developed in WEKA. On the 11th run, WEKA runs the multiple linear regression algorithm on the data set and provide MLR model, as can be seen in Figure 15.

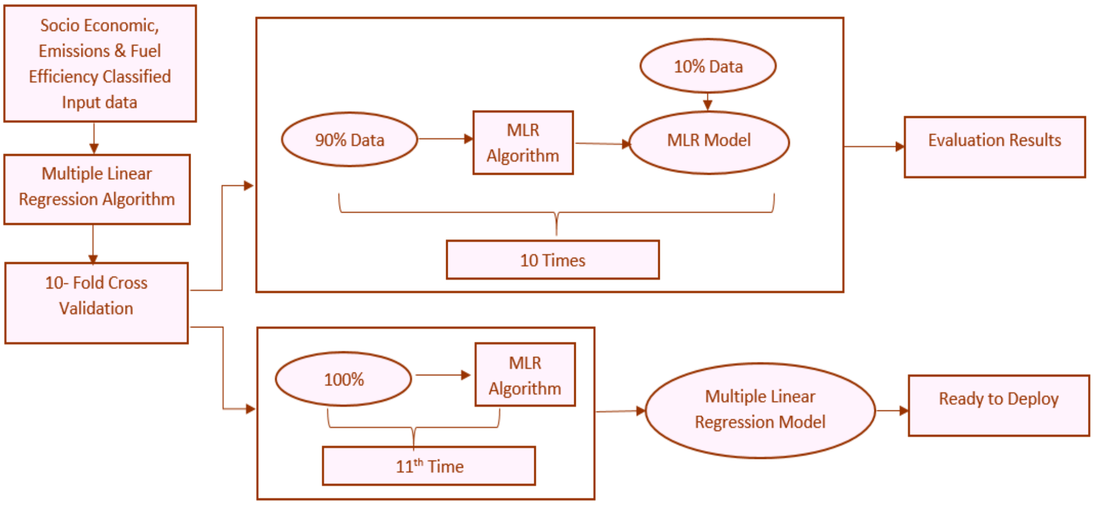

Socio Economic, Emissions & Fuel Efficiency Classified Input data
Multiple Linear Regression Algorithm
10- Fold Cross Validation
90% Data
MLR Algorithm
10% Data
MLR Model
10 Times
Evaluation Results
100%
MLR Algorithm
11th Time
Multiple Linear Regression Model
Ready to Deploy

Figure 15 Multiple linear regression model development

2. Multilayer Perceptron

The MLP model is developed in WEKA. Learning parameters plays a vital role in fine-tuning of MLP model; in case the performance parameters given by cross-validation are not satisfactory, the network can be fine-tuned by changing learning rate, momentum and number of epochs (or training time). Therefore, cross-validation is an important validation technique because its results impact the network training.

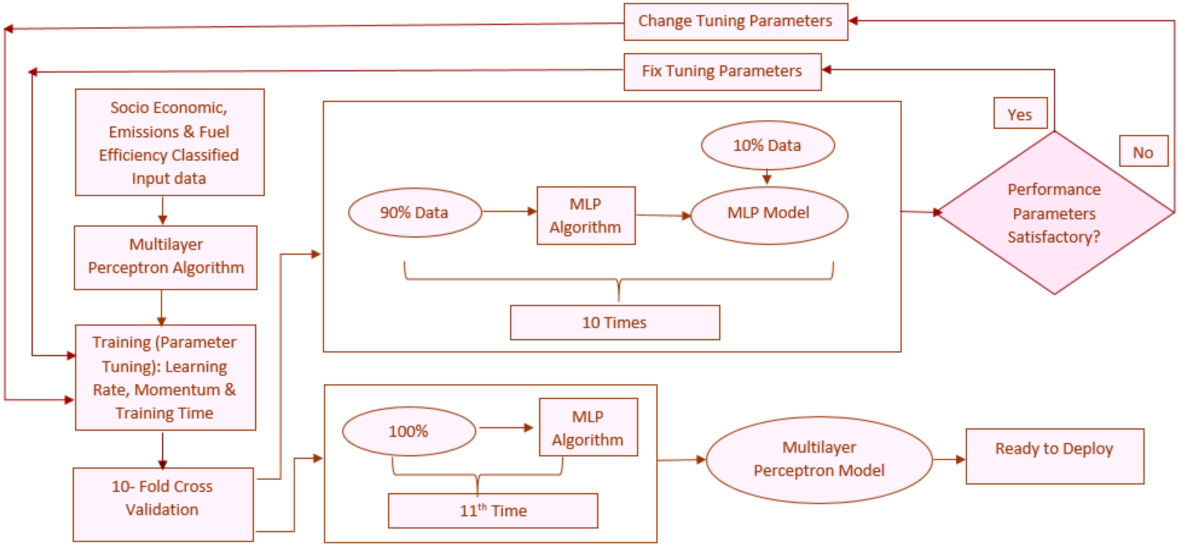

Change Tuning Parameters
Fix Tuning Parameters
Yes
No
Socio Economic, Emissions & Fuel Efficiency Classified Input data
Multilayer Perceptron Algorithm
Training (Parameter Tuning): Learning Rate, Momentum & Training Time
10- Fold Cross Validation
90% Data
MLP Algorithm
10% Data
MLP Model
10 Times
Performance Parameters Satisfactory?
100%
MLP Algorithm
11th Time
Multilayer Perceptron Model
Ready to Deploy

Figure 16 MLP model development

The MLP model development is shown in Figure 16. On the 11th run, WEKA develops the MLP network, which is shown in Figure 17. It is a two-layered feedforward network with backpropagation setting. The training is performed using gradient descent algorithm. We used 10-fold cross-validation technique to avoid the problem of overfitting and to check the generalisation by the model when applied to independent/unknown data set.

3. Bagging

Bagging performs better on the unstable base classifier, where minor changes in the training set can lead to major changes in the classifier output. An MLP is an example of the unstable classifier. The bagging algorithm with 10 iterations/bags was also evaluated using 10-fold cross-validation technique. So for each bag, 10 MLP classifiers were trained and combined. To aggregate the outputs of the base learner, bagging algorithm uses averaging for regression.

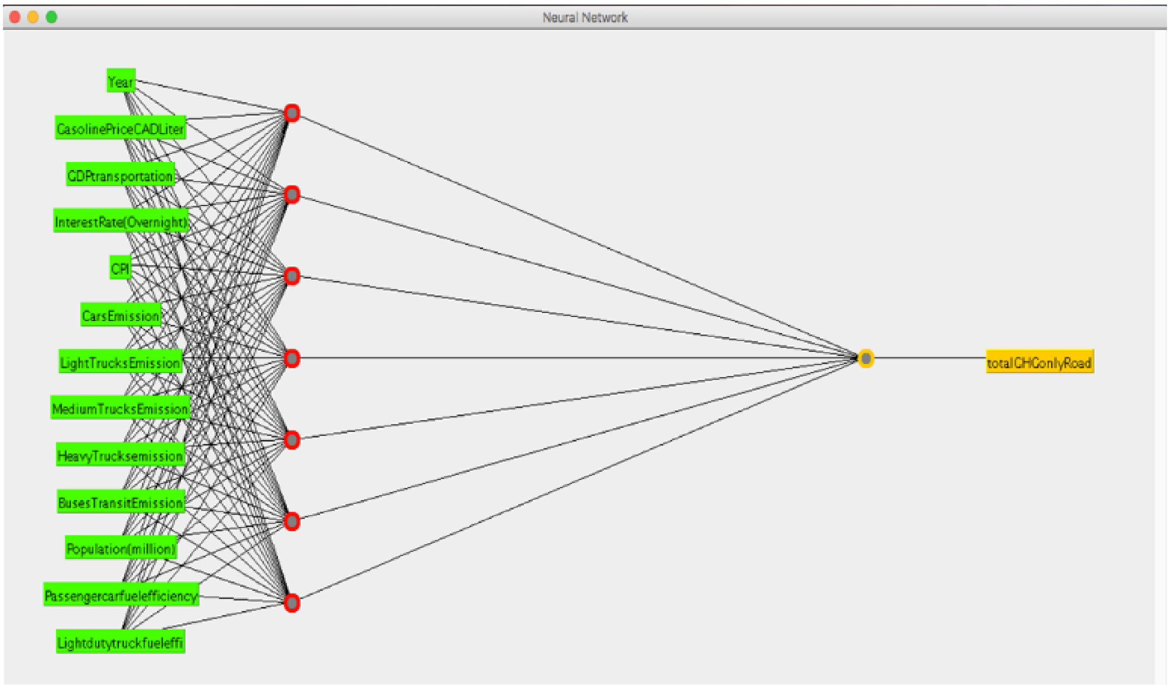


Figure 17 Multilayer perceptron network

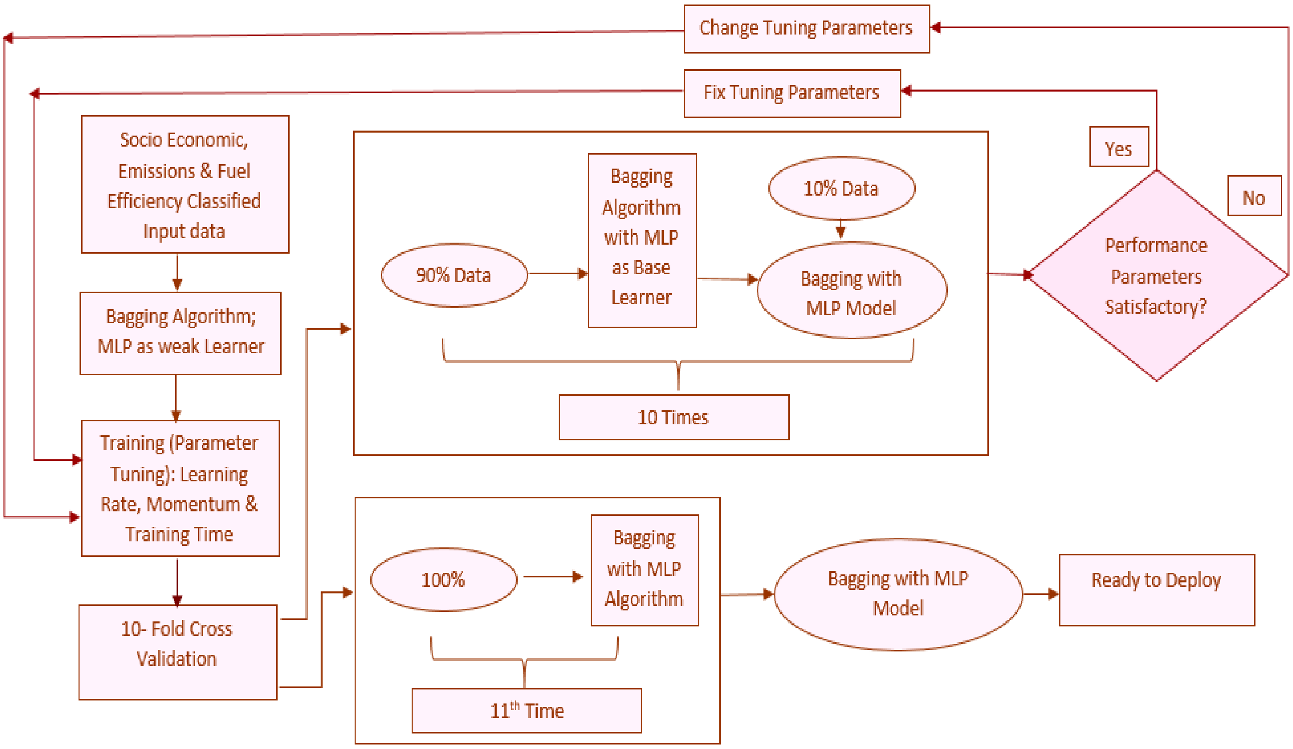

Change Tuning Parameters
Fix Tuning Parameters
Socio Economic, Emissions & Fuel Efficiency Classified Input data
Bagging Algorithm; MLP as weak Learner
Training (Parameter Tuning): Learning Rate, Momentum & Training Time
10- Fold Cross Validation
90% Data
Bagging Algorithm with MLP as Base Learner
10% Data
Bagging with MLP Model
10 Times
Yes
No
Performance Parameters Satisfactory?
100%
Bagging with MLP Algorithm
11th Time
Bagging with MLP Model
Ready to Deploy

Figure 18 Bagging MLP model development

The bagging MLP model development is shown in Figure 18; we used 10 iterations for bagging algorithm, and we used 10-fold cross-validation, which means, for each bag 10 MLP, classifiers were trained and combined using averaging. Finally, for regression, averaging is performed for all 10 bags and the model is selected. The final developed MLP with bagging network is shown in Figure 17 is a two-layered feedforward network with backpropagation setting. The training is performed using gradient descent algorithm.

C. Results

For the implementation of learning algorithms mentioned in methodology, we used WEKA. In this section, we outline the algorithm performance measures and results of algorithm application and improvement.

1. Performance Evaluation Metrics

The performance of models was assessed by the below-mentioned metrics:

Root Mean square Error

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

Mean Absolute Error

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

R-square/Coefficient of Determination

The degree of correlation among the observed and predicted values with values close to 1.0 was calculated to demonstrate good model performance (Mashaly et al., 2016).

The performance parameter is calculated as below:

$$R^2 = \frac{\left(n \sum_{i=1}^{n} y_i\hat{y}_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} \hat{y}_i\right)^2}{\left(n \sum_{i=1}^{n} y_i{}^2 - \left(\sum_{i=1}^{n} y_i\right)^2\right)\left(n \sum_{i=1}^{n} \hat{y}_i{}^2 - \left(\sum_{i=1}^{n} \hat{y}_i\right)^2\right)}$$

where

y_i is the observed value

$\hat{y}_i$ is the predicted value

n is the number of observations

Cohen's Kappa Statistics

It evaluates the portion of hits that can be credited to the classifier itself relative to all the classifications that cannot be credited to chance alone (Carletta 1996). Kappa statistics is given by

$$\frac{n\sum_{i=1}^{m} TP - \sum_{i=1}^{m} T_{ri}T_{ci}}{n^2 - \sum_{i=1}^{m} T_{ri}T_{ci}}$$

where *TP* is the number of true positives for each class, *n* is a total number of examples and *m* is a number of class labels. T_{ri} is row count and T_{ci} is column count. Cohen's Kappa ranges from −1 through 0 to 1. These values indicate total disagreement, random classification, and perfect agreement, respectively (Viera et al., 2005). For ideal data modelling, the value Kappa statistics will approach to 1.

Accuracy

It measures the capacity of the predictive model to classify correctly; it is the proportion of the total number of predictions that were correct.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

F measure

It is harmonic mean of precision and recalls, that is, it can be interpreted as a weighted average of precision and recall; F measure calculates the accuracy of a test (Sasaki 2007).

$$F\ measure = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

For ideal data modelling, the F measure value should approach to 1.

Area under the curve

For comparison, classifiers have to reduce the two-dimensional representation of classifier performance into a single scalar value. The most common method is to calculate the area under the ROC curve, abbreviated as AUC (Hanley et el., 1982). The AUC is a portion of the area of the unit square; hence, its value will always be between 0 and 1.

The area under the ROC curve (AUC) is calculated by the trapezoid rule, (de Menezes et al., 2017).

$$AUC = \sum_{i=1}^{n}(x_{i+1} - x_i)\ \left(\frac{y_{i+1} + y_i}{2}\right)$$

where *i* is the threshold of the curve from which the pair of points (x_i, y_i) are taken.

The AUC measures the success of the model in correctly classifying TP and TN. Usually, as a general rule as stated by Zhou et al. (2009), if AUC ≥0.8, the discrimination is said to be excellent.

2. Model performance and comparison of algorithm Improvement analysis for nominal data

Table 1

Performance Evaluation Metric	Multinomial Logistic Regression	Decision Tree	Multilayer Perceptron
Root mean squared error	0.3445	0.3143	0.2676

Kappa statistic	0.5394	0.6403	0.5375
Wt. Avg. ROC Area	0.903	0.826	0.935

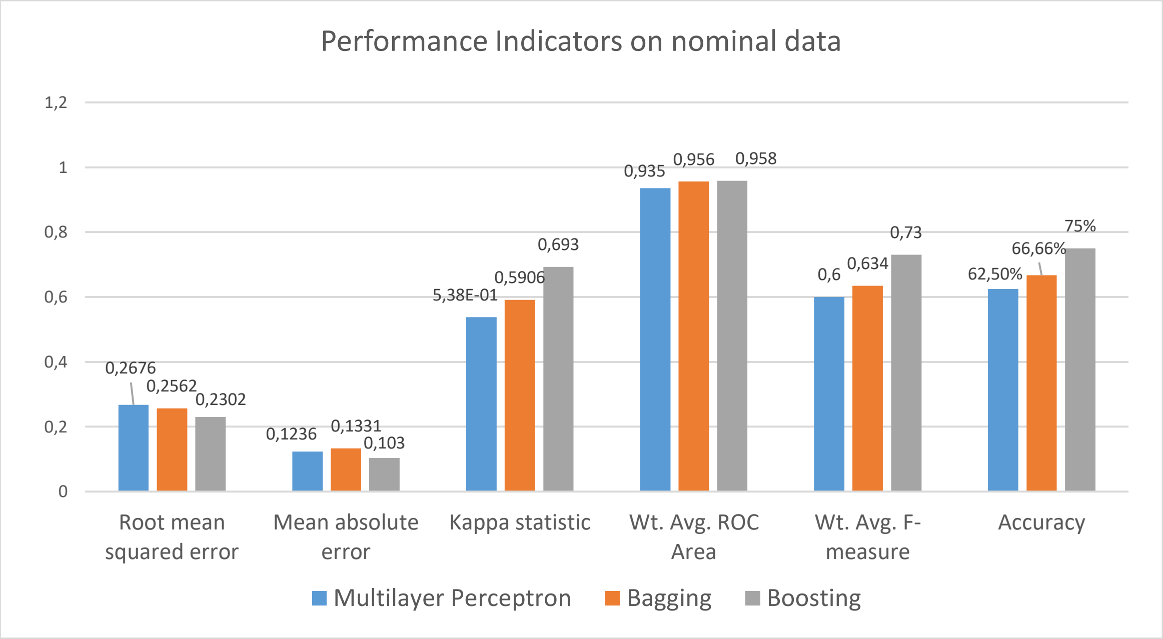


Figure 19 Performance indicators of algorithms on nominal data

As can be seen from Table 1, performance indicators for MLP model outperform decision tree and multinomial logistic regression models. Hence, we implemented ensemble techniques, that is, bagging and boosting algorithm on MLP classifier to enhance the predictive modelling capacity of this NN.

Figure 19 shows performance indicators of algorithms on nominal data; the model developed by MLP with boosting algorithm outperforms the models developed by MLP and MLP with bagging for nominal data**.**

3. Model performance and comparison of algorithm improvement analysis for Numeric data

Table 2 presents the performance evaluation of MLR and MLP models; MLP algorithm outperforms multiple linear regression in prediction performance. Hence, we implemented ensemble technique, that is,, bagging algorithm on MLP regression model to enhance the predictive modelling capacity of this NN.

Table 2

Performance Evaluation Metric	Multiple Linear Regression	Multilayer Perceptron
Root mean squared error	1.301	0.442
Correlation coefficient	0.9973	0.9996
Mean absolute error	0.7223	0.3471

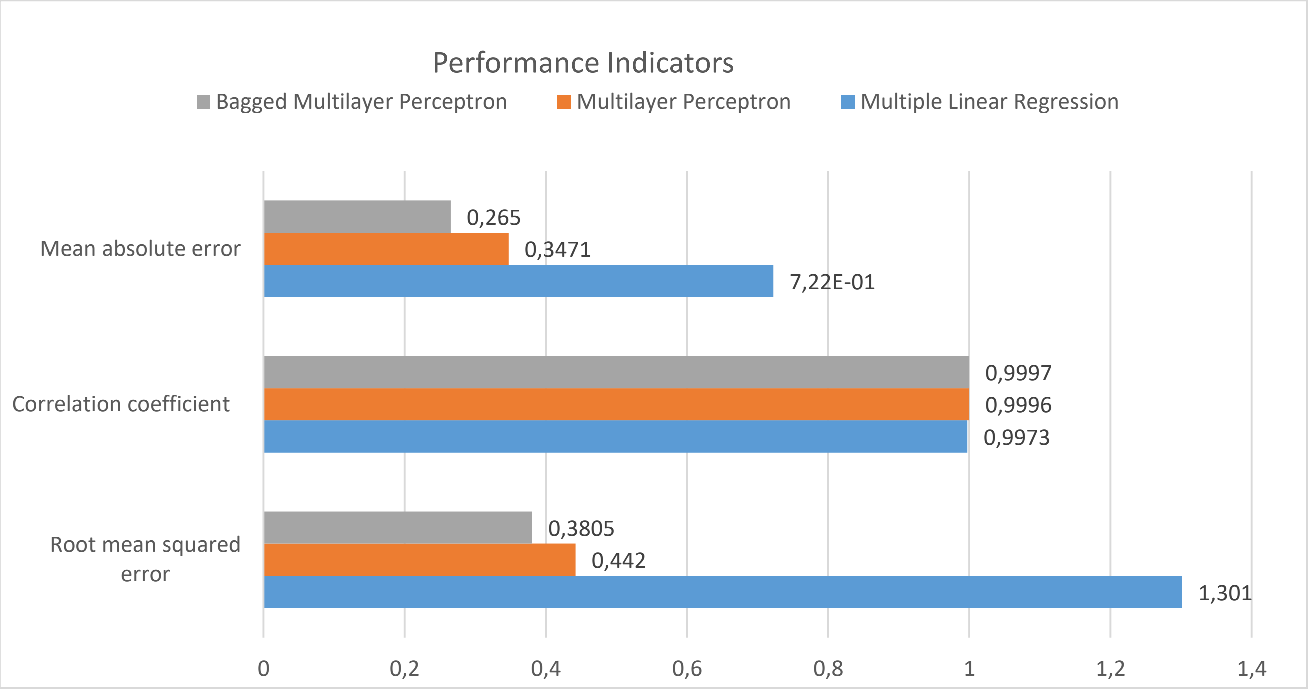


Figure 20 Performance indicators of algorithms on numeric data

Figure 20 shows that the model developed by MLP with bagging algorithm outperforms the models given by multiple linear regression and MLP. That is, for bagged MLP, the value of errors is minimum and correlation coefficient is high**.**

4 Conclusions and Future Work

Prediction of GHG emissions is important to minimise their negative impact on climate change and global warming. In this study, we presented new models based on data mining/supervised learning techniques (regression and classification) for predicting GHG emissions arising from passenger and freight road transport in Canada.

Removing less influencing attribute improves generalising performance of machine learning models. In the present study, RReliefF algorithm was implemented using WEKA for identifying relevant or most influencing attributes considered amongst socio-economic, emission and fuel efficiency for the predicting GHG emission caused by road transport. Two MLP networks were modelled, MLP1 and MLP2, and the model MLP2, which uses relevant input variables selected by RReliefF algorithm (excluding car sales), had better values of their performance evaluation parameters, that is, RMSE for MLP1 and MLP2 were 0.5776 and 0.442, respectively, and the value for R^2 for MLP1 and MLP2 were 0.9993 and 0.9996, respectively.

We developed four categories of models, namely, artificial NN MLP, multiple linear regression, multinomial logistic regression and decision tree, and evaluated their performances by the error estimated by the cross-validation technique using performance indicators. Ensemble technique (bagging and boosting) was applied on the developed MLP model, which significantly improved the model's predictive performance. For numeric GHG emissions attribute values, the artificial NN MLP model with bagging ensemble technique outperformed other models.

On the basis of the proposed work, few future research works are possible. Detailed study on most relevant and influential parameters to further improve the prediction accuracy of MLP model with bagging can be done. Furthermore, future emissions prediction can be projected and analysed.

References

Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. Computational linguistics, 22(2), 249-254.

Dawson, C. W., & Wilby, R. (1998). An artificial neural network approach to rainfall-runoff modelling. Hydrological Sciences Journal, 43(1), 47-66.

Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Springer Berlin Heidelberg.

Environment and Climate Change Canada - Environmental Indicators - Greenhouse Gas Emissions. Retrieved May 29, 2017, from http://www.ec.gc.ca/indicateurs-indicators/default.asp?lang=En&n=FBF8455E-1

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In ICML (Vol. 96, pp. 148-156).

Government of Canada, Environment and Climate Change Canada. (2017, April 13).

Hosmer, D. W., & Lemeshow, S. (2000). Special topics. Applied Logistic Regression, Second Edition, 260-351.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.

Hssina, B., Merbouha, A., Ezzikouri, H., & Erritali, M. (2014).A comparative study of decision tree ID3 and C4. 5. International Journal of Advanced Computer Science and Applications, 4(2), 13-19.

Lang, H. (2013). Topics on Applied Mathematical Statistics. KTH Teknikvetenskap, version 0.97.

S Lek Y S Park. (2008). Encyclopedia of Ecology | Multilayer Perceptron. Retrieved July 10, 2017

Mashaly, A. F., & Alazba, A. A. (2016). MLP and MLR models for instantaneous thermal efficiency prediction of solar still under hyper-arid environment. Computers and Electronics in Agriculture, 122, 146-155.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

de Menezes, F. S., Liska, G. R., Cirillo, M. A., & Vivanco, M. J. (2017). Data classification with binary response through the Boosting algorithm and logistic regression. Expert Systems with Applications, 69, 62-73.

Metz, B., Davidson, O. R., Bosch, P. R., & Dave, R. Contribution of Working Group III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change, 2007

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Let a biogeography-based optimizer train your multi-layer perceptron. Information Sciences, 269, 188-209.

Opitz, D. W., & Maclin, R. (1999). Popular ensemble methods: An empirical study. J. Artif. Intell. Res.(JAIR), 11, 169-198.

Quinlan, J. R. (1993). C4. 5: Programming for machine learning. Morgan Kauffmann, 38.

Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. The journal of educational research, 96(1), 3-14.

Sasaki, Y. (2007). The truth of the F-measure. Teach Tutor mater, 1(5).

Sayad, S. (2011). Real time data mining. Canada: Self-Help Publishers.

Shardlow, M. (2016). An analysis of feature selection techniques. The University of Manchester.

Viera, A. J., & Garrett, J. M. (2005). Understanding interobserver agreement: the kappa statistic. Fam Med, 37(5), 360-363.

Wattimena, R. K. (2014). Predicting the stability of hard rock pillars using multinomial logistic regression. International journal of rock mechanics and mining sciences, 71, 33-40.

Werbos, P. J. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. Doctoral Dissertation, Applied Mathematics, Harvard University, MA.

Winiwarter, W., & Rypdal, K. (2001). Assessing the uncertainty associated with national greenhouse gas emission inventories:: a case study for Austria. Atmospheric environment, 35(32), 5425-5440

Zhou, Z. H. (2012). Ensemble methods: foundations and algorithms. CRC press.

Yan, X., & Su, X. (2009). Linear regression analysis: theory and computing. World Scientific