ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

Machine Learning for Classification of Traffic Congestion – New Traffic Management Perspective

Sriram Sridhar[1]

Publication Date: 2026/01/29

Abstract: With growing economic status, many people are buying vehicles which reflects their growing stature. Sadly, this is not backed by good traffic management systems which are increasingly leading to clogging of roads and unhealthy environment prevails leading to deterioration of health as well. Maps such as those provided by major players offset some of the problems faced by including traffic density in the path planning. But it still does not consider the weather conditions and emission levels in the area while suggesting a route. Also, the experience of past users of the route is not given any consideration of the route quality. This means that the estimation of traffic congestion goes beyond the accepted parameters of traffic density alone but is more intrinsically complicated than that. This also implies that there is no clear rule that can be written to classify the traffic congestion levels of the area.

This means the only solution to estimate the traffic congestion levels of the region is to develop a robust machine learning model for the task of classifying the traffic congestion levels. By using this machine learning approach, we can use several feature parameters to map and model the traffic congestion levels to input and for which a rule-based approach fails as learning from past data is the only solution. This can result in better path planning and infrastructure making commute healthy, faster resulting in improved quality of life. In this paper, we therefore apply a few classification models and determine its performance using the accuracy metric, training time and model stability. We therefore recommend the best fitting model from among the two models we have applied to this task of multi-class classification. We believe that this will transform the way transport is planned and pave way for a cleaner and healthier environment in the long run.

Keywords: *Path Planning, Sustainable Infrastructure, Traffic Congestion, Clean Environment.*

How to Cite: Sriram Sridhar (2026) Machine Learning for Classification of Traffic Congestion – New Traffic Management Perspective. *International Journal of Innovative Science and Research Technology*, 11(1), 2164-2172. https://doi.org/10.38124/ijisrt/26jan1185

I. INTRODUCTION

Economies are on the rise. Industrialization and scientific progress has meant that what was seemingly impossible and a distant reality is now easily within reach. Automotive is one such area where we have made significant progress. We have made the transition from ox-carts to vehicles at a frenetic pace. This is largely true of developed and to a major extent the developing countries of the world. This coupled with the fact that employment opportunities are increasing has meant that more people have these vehicles in possession.

There is however a flip side to it. Infrastructure development and planning hasn't yet matched the frenetic pace of vehicle purchase. This has led to poor planning of traffic leading to lower life expectancy, pollution and most importantly irritatingly long travel times. Maps being offered by major players offset some of the problems as they offer the shortest path possible from source to destination. But it still leaves a lot to be desired as traffic density is the only parameter for calculating the best possible path. It seems that a more comprehensive and a holistic approach to path planning is needed. This means that traffic congestion needs to be seen beyond the accepted parameter of traffic density alone.

What this also means is that as we add more and more inputs to determining the congestion levels there is no clear rule set that can be used to arrive at this congestion classification of a route. Machine Learning offers a very good alternative to this rule based approach as it relies on building a model that maps the input feature set to the output which is the congestion level of a route.

In this paper, we therefore use machine learning to classify the congestion level of routes using many inputs. We will apply a couple of different supervised learning techniques for this multi-class classification task and estimate its performance using the accuracy metric, training time and model stability.

The paper is organized as follows. The following section briefly discusses the background work done in this area. This is then followed by a brief discussion on the

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

algorithms used in this work. The next section describes the background preparation in this work. This is then followed by experimental procedure used in this work. This is then followed by the results obtained in this work and the recommendation on the best fitting model for this multiclass classification. The last section is devoted to the future work in this area.

II. LITERATURE SURVEY

This section lists the previous work done in the topic of traffic congestion classification. Only most important references are listed here for the sake of brevity. [1] lists the time-dependent nature of traffic congestion. They use data from advanced sensors as inputs to the classification model and consider factors such as environment, social events etc.. as influencers of traffic congestion. Finally, Machine Learning (ML) models namely Decision Tree (DT), Naive Bayes (NB), K-Nearest Neighbor (KNN), Random Forest (RF), Support Vector Classifier (SVC), Logistic Regression (LR), and Multilayer Perceptron (MLP) and show the best model for the task. [2] uses machine learning algorithms namely Random Forest, K-Nearest Neighbour, XG-Boost as well as Artificial Neural Networks (ANN) for traffic jam prediction and offer a comparative evaluation of each. [3] presents another work on traffic congestion prediction using machine learning. However, they treat it as a regression task as the use of the metric RMSE supports this.

To the best of our knowledge the comparative evaluation of logistic regression (One vs Rest), KNN classifier, Random Forests with the Extreme Learning Machines (ELM) which is known for its fast training time without compromising on the accuracy on the traffic congestion classification is not yet explored. Also, factors such as weather, public opinion on the route quality as well as other factors are seldom explored. This paper addresses this research direction to offer a comparative evaluation which can be exploited to create intelligent traffic management systems.

III. TECHNICAL BRUSHUP

➢ *Logistic Regression (One-Vs-Rest)*

Supervised Learning is largely composed of two main types of tasks – Regression, Classification. Regression is mainly used to solve problems where we are predicting continuous valued outputs. Classification refers to problems where the inputs are used for estimating countably finite labels. Since the problem here we are focusing on is that of multi-class classification we will focus on Logistic Regression (LR) which is one such technique useful for the classification task. But the conventional LR is not adept at handling multi-class classification. Hence, we use the One vs Rest LR technique as one of the algorithms for this task. The following are the steps:

- Decompose the multi-class classification problem into several binary classification problems.
- A separate logistic regression model is trained for each of these binary classes. Each model is able to distinguish between one class vs the rest.
- A voting mechanism happens when presented with a new point. Each model is called to assign the probability of the class it was trained for. The class with the highest probability is the assigned class for that instance.

➢ *Extreme Learning Machines (ELM)*

ELM was born out of the ideology that the then ANNs were not exactly mimicking the human brain. It took far too long to train the model without guaranteed success. Typically, gradient based approaches for training weights and biases is employed which is disadvantageous for the following reasons:

- The convergence depends on the learning rate. The higher learning rate faster is the convergence, but it could cause dangling problems. On the other hand, lowering the learning rate will take long time to converge.
- The convergence could hit a tumbling block as the gradient is zero even at local minimum, which means the most optimal solution may never be arrived at.

Fortunately, the inventors of the ELM [4] demonstrate that analytical (matrix based) approach was possible to train hidden node to output node parameters for the Single Layer Feedforward Neural Networks (SLFN) while randomly initializing the input node to hidden node parameters. They show that it is possible to find that the hidden node to output node parameters by a simple inverse matrix operation called the Moore Penrose inverse.

Mathematically, let $\boldsymbol{N}$ represent the number of distinct samples $(\mathbf{x_i}, \mathbf{t_i})$, $\boldsymbol{N}$~ represents the number of hidden nodes, activation function $\mathbf{g(x)}$ we have the formula as:

$$\sum_{i=1}^{N\sim} \beta_{(i)} g_i x_{(j)} = \mathbf{o_j} \text{ where } j = 1,2 \ldots . N \qquad (1)$$

Where w_i is the i^{th} hidden node and the input nodes, β_i is the weight vector connecting the i^{th} hidden node and the output nodes and b_i is the threshold of the i_{th} hidden node.

That standard SLFNs with N~ hidden nodes can approximate these N samples with zero error means that the following equation holds:

$$\mathbf{H}\beta = \mathbf{T} \qquad (2)$$

Where H is called the hidden layer output matrix.

The special solution to (2) which is also one of the least square solutions can be obtained by calculating the inverse which is called the Moore Penrose generalized inverse of the matrix H.

➢ *KNN Classifier*

In the KNN classifier, we calculate the distance of the sample point with each of the labelled data points. We then choose the value of K which is the number of data points to be considered for maximum voting for deciding on the class label of the new sample point. Typically, the value of K

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

should neither be too small nor too large. Furthermore, it is advisable to choose an odd value for K.

➢ *Random Forest Classifier*

Random Forest is an extension of the decision trees where each tree is a flow of decision making starting from the root node all the way to the leaf node. Here, we use several trees and then perform an average optimization across trees to arrive at the classification.

➢ *Traffic Management Dataset*

The machine learning dataset used in this dataset is taken from Kaggle which is called the smart mobility dataset [5] which relates to traffic management making cities more liveable. It considers factors beyond traffic density alone for the purpose of classifying the congestion levels in the area. Specifically, the following factors are considered as the input features, and the dataset has around 5000 datapoints.

- Traffic Data: Vehicle count, speed, road occupancy, and traffic light status.
- Weather & Accidents: Weather conditions and accident reports impacting congestion.
- Social Network Sentiment: Public opinions on mobility and congestion from social media.
- Smart Mobility Factors: Ride-sharing demand, parking availability, and public transport delays.
- Environmental Impact: CO_2 emissions and pollution levels.
- Target Variable
- Traffic Congestion Level: Categorized as Low, Medium, or High, based on traffic density, speed, and road occupancy

IV. BACKGROUND PREPARATION

➢ *The Following Data Wrangling Steps Apply to the Dataset.*

- Remove the rows with any missing column values in the dataset.
- Convert categorical input features into numerical numbers using the one hot encoding process.
- Convert the output labels using label encoding (one hot encoding avoided to prevent it from being a multi-label classification).

The above exercise will help prepare us for the step of applying machine learning models.

V. EXPERIMENTAL APPROACH

The following steps are adopted for the experimental approach in this study. The same is shown pictorially by a block diagram in Figure 1.

➢ Conduct the background preparation mentioned above.

➢ Divide the dataset into training and test sets into ratios of 60-40.

➢ Apply the ML models for the classification task.

➢ We tested the stability of the models by training and testing for 50 cycles while evaluating the performance as mentioned in step 5. This is particularly relevant for the ELM model as we show empirically that the random initialization of input to hidden nodes does not affect model stability much.

➢ Use the accuracy metric and training time to evaluate the model performance.

➢ Recommend the best fitting model.

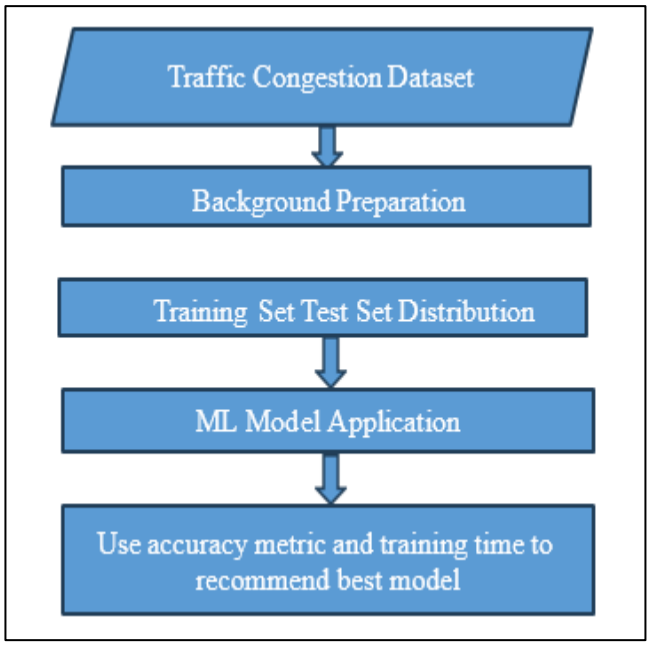

Fig 1 Block Diagram Showing the Experimental Study

VI. RESULTS

The following are the results on the application of the two algorithms. The algorithms are evaluated mostly by the training time and the accuracy metric. We begin our empirical study with the algorithm ELM on the dataset.

➢ *ELM*

- *Training Test Set (60 – 40)*

✓ *Activation Function - ReLU*

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

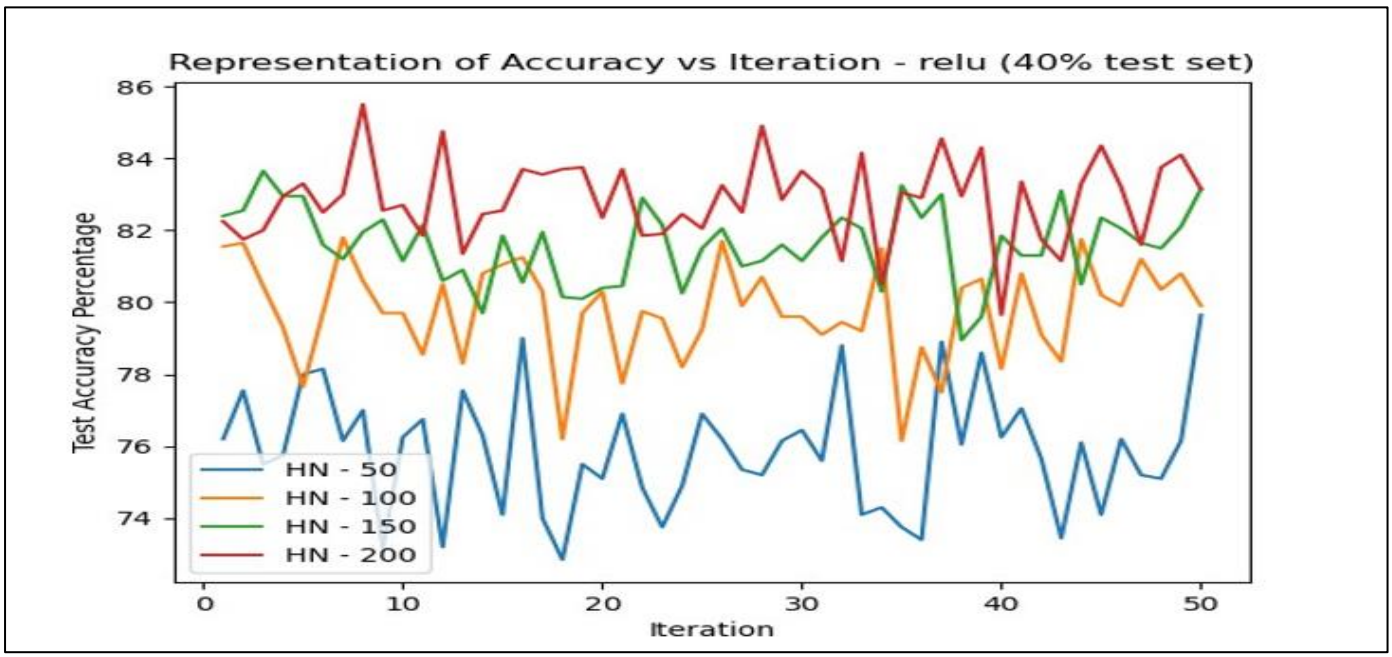

Fig 2 Testing Accuracy Estimation for the Dataset Using the ReLU Activation Function.

Table 1 Statistics for Figure 2

Hidden Neurons	Standard Deviation (σ)	Mean Training Time
50	1.6684	0.042s
100	1.3453	0.068s
150	1.0580	0.103s
200	1.1503	0.156s

✓ *Observation:*

It can be seen from figure 2 that with the hidden neurons at 200 the testing set classification accuracy is at its highest with lowest standard deviation. However, from Table 1 due to the increase in the number of hidden neurons the number of connections increase due to which the number of weights and biases increase resulting in marginally higher training time. However, it is still very reasonable considering the side of the dataset.

✓ *Activation Function – Leaky ReLU*

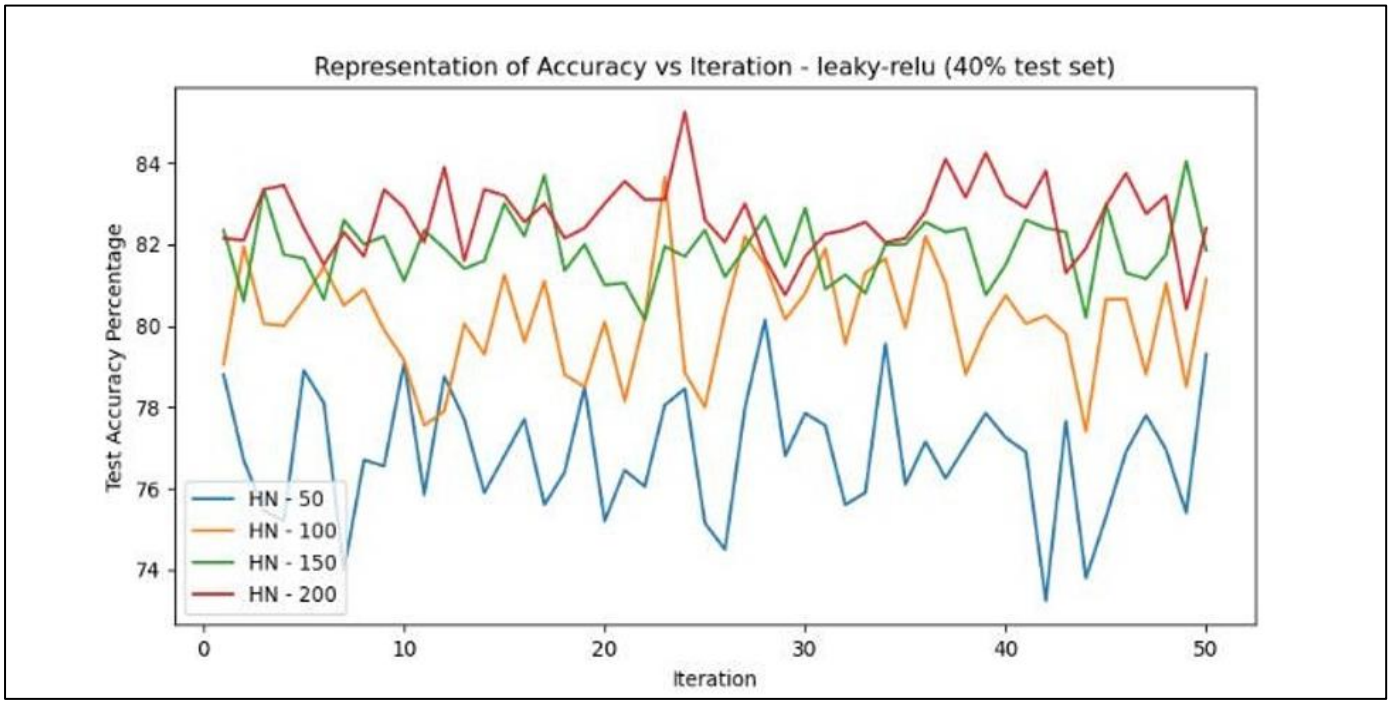

Fig 3 Testing Accuracy Estimation for the Dataset Using the Leaky ReLU Activation Function.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

Table 2 Statistics for Figure 3

Hidden Neurons	Standard Deviation (σ)	Mean Training Time
50	1.5316	0.049s
100	1.3213	0.0928s
150	0.8440	0.1402s
200	0.8990	0.2494s

✓ *Observation:*

It can be seen from figure 3 that the trend continues even for the leaky ReLU activation as setting the number of hidden neurons to 200 gives the highest accuracy. However, the average training time has also increased a little as seen from Table 2 but is negligible.

✓ *Activation Function – Sigmoid*

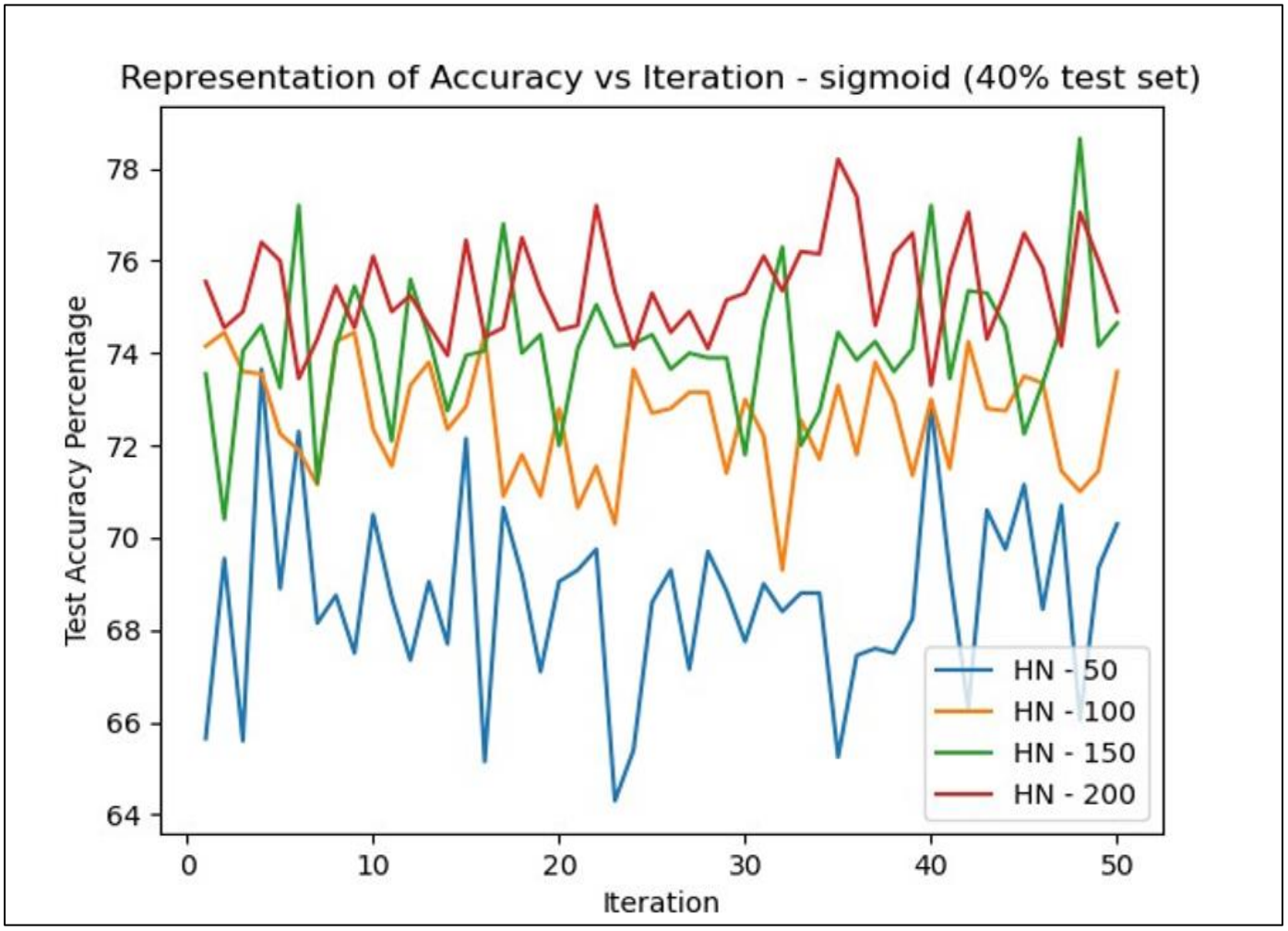

Fig 4 Testing Accuracy Estimation for the Dataset Using the Sigmoid Activation Function.

Table 3 Statistics for Figure 4

Hidden Neurons	Standard Deviation (σ)	Mean Training Time
50	2.008	0.058s
100	1.201	0.1422s
150	1.509	0.1815s
200	1.071	0.3141s

✓ *Observations:*

It can be seen that from figure 4 that the accuracy percentages has dipped a little in comparison to relu or leaky relu. However, the training time is comparable and so is the standard deviation as seen from Table 3.

✓ *Activation Function – Sine*

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

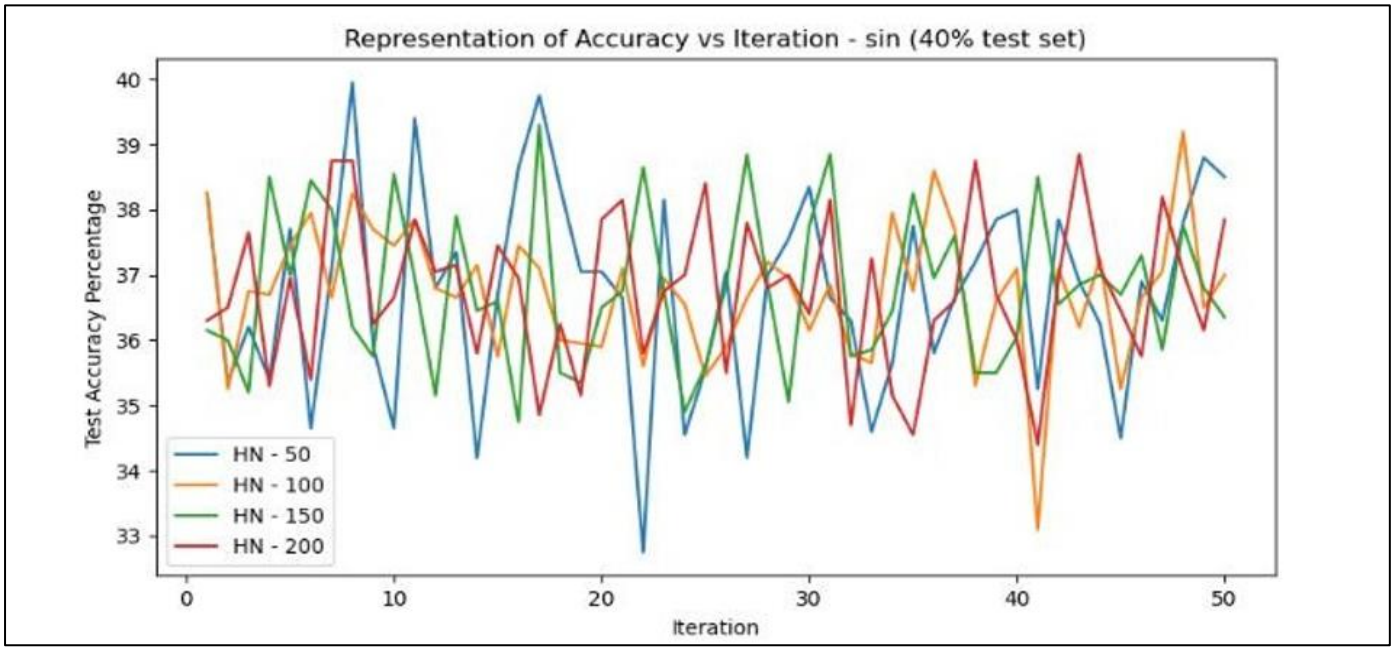

Fig 5 Testing Accuracy Estimation for the Dataset Using the Sine Activation Function.

Table 4 Statistics for Figure 5

Hidden Nerons	Standard Deviation (σ)	Mean Training Time
50	1.557	0.051s
100	1.0317	0.090s
150	1.183	0.116s
200	1.159	0.1745s

✓ *Observation:*

It can be seen that from figure 5 that the accuracy percentage has dipped significantly meaning that sin is not such a good fit to the data. But from Table 4 it can be seen that there is not much difference in the standard deviation and training time.

➢ *Logistic Regression*

- *Training Test Set (60 – 40)*

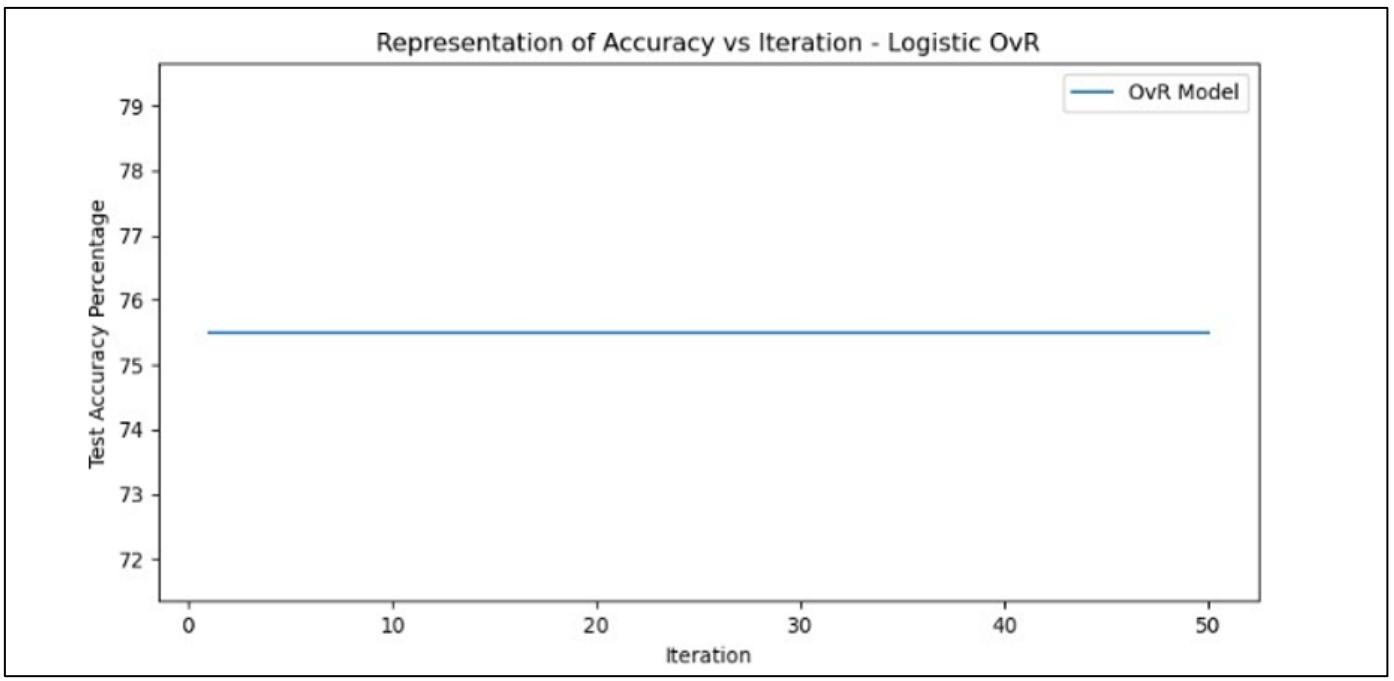

Fig 6 Testing Accuracy Estimation for the Dataset Using the Logistic OvR Function.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

Table 5 Statistics for Figure 6

Standard Deviation (σ)	Mean Training Time
0	0.61s

✓ *Observations:*

It can be seen that from figure 6 the accuracy of OvR is debatable. While it can be inferred that OvR performs better the sin function considerably in the ELM it still lags behind the ReLU and Leaky ReLU function in the ELM. From Table 5 it can be seen that training time is not significantly different.

- *Training Test Set (60 – 40)*

For the KNN classifier, we run the training and testing of the dataset by incrementally changing the value of K nearest neighbours by 2. The model gave the optimum performance at K = 11. The same has been shown below.

➢ *KNN Classifier*

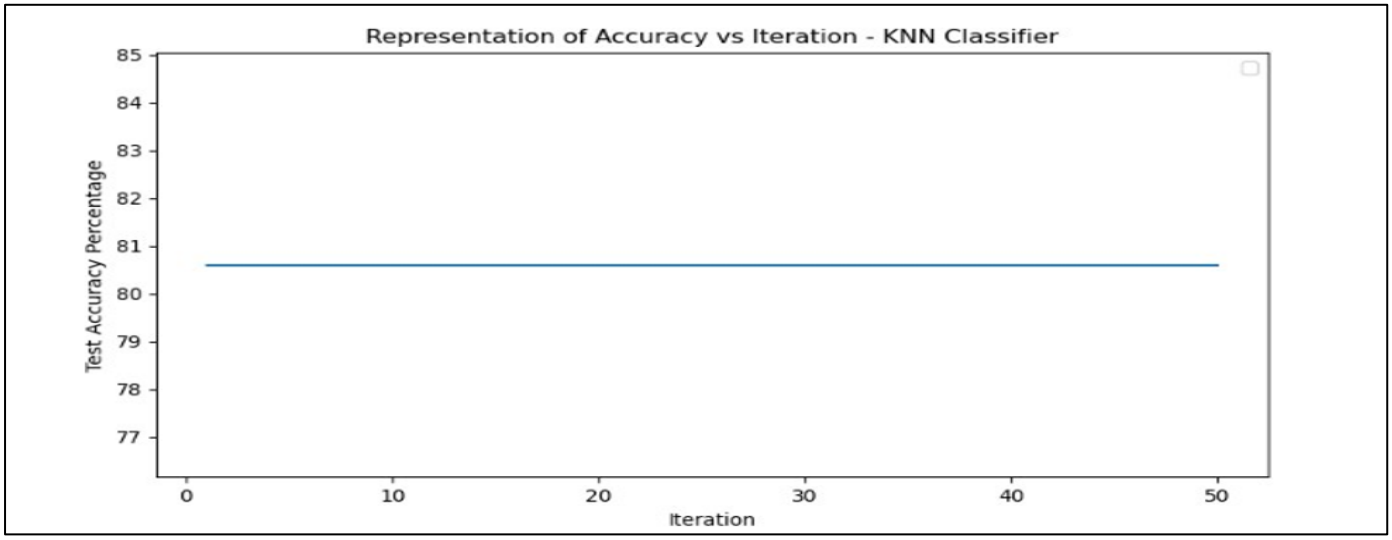

Fig 7 Testing Accuracy Estimation for the Dataset Using the KNN Classifier Function with K = 11.

Table 6 Statistics for Figure 7

Standard Deviation (σ)	Mean Training Time
0	0.03s

➢ *Random Forest Classifier*

- *Training Test Set (60 – 40)*

The following diagram represents the testing accuracy for the random forest classifier.

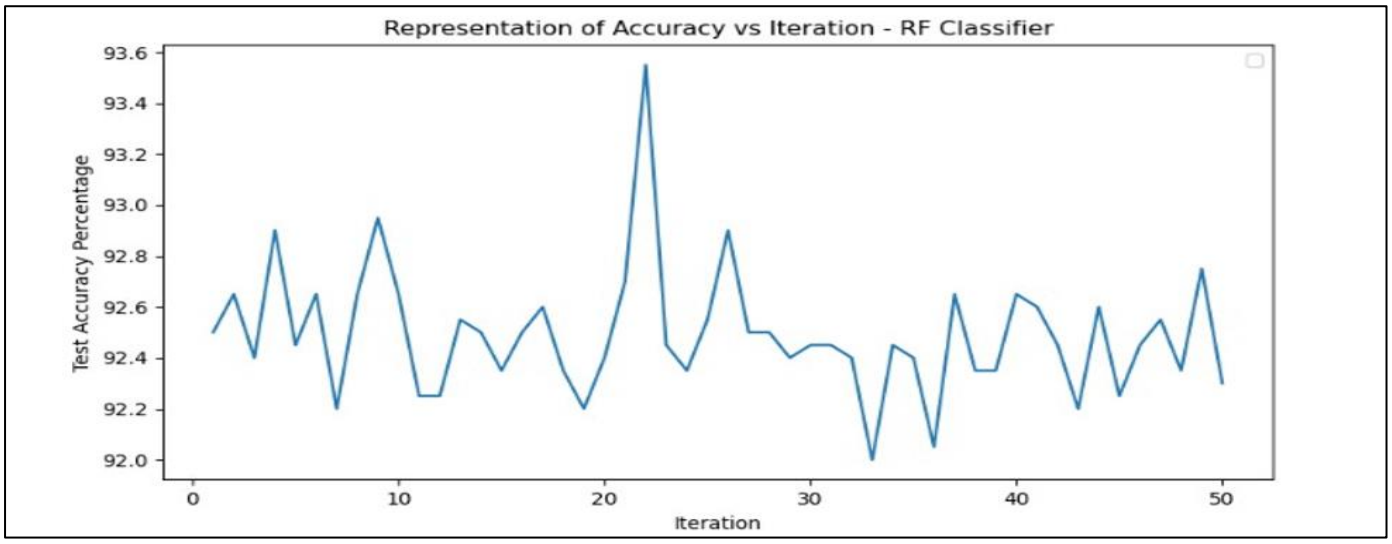

Figure 8 - Testing Accuracy Estimation for the Dataset Using the RF Classifier Function.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

Table 7 Statistics for Figure 8

Standard Deviation (σ)	Mean Training Time
0.33	44s

➢ *SVC Classifier*

- *Training Test Set (60 – 40)*
  For the support vector machines classifier, we vary the kernel functions and verify the accuracy of the model.

✓ *Kernel Function – rbf*

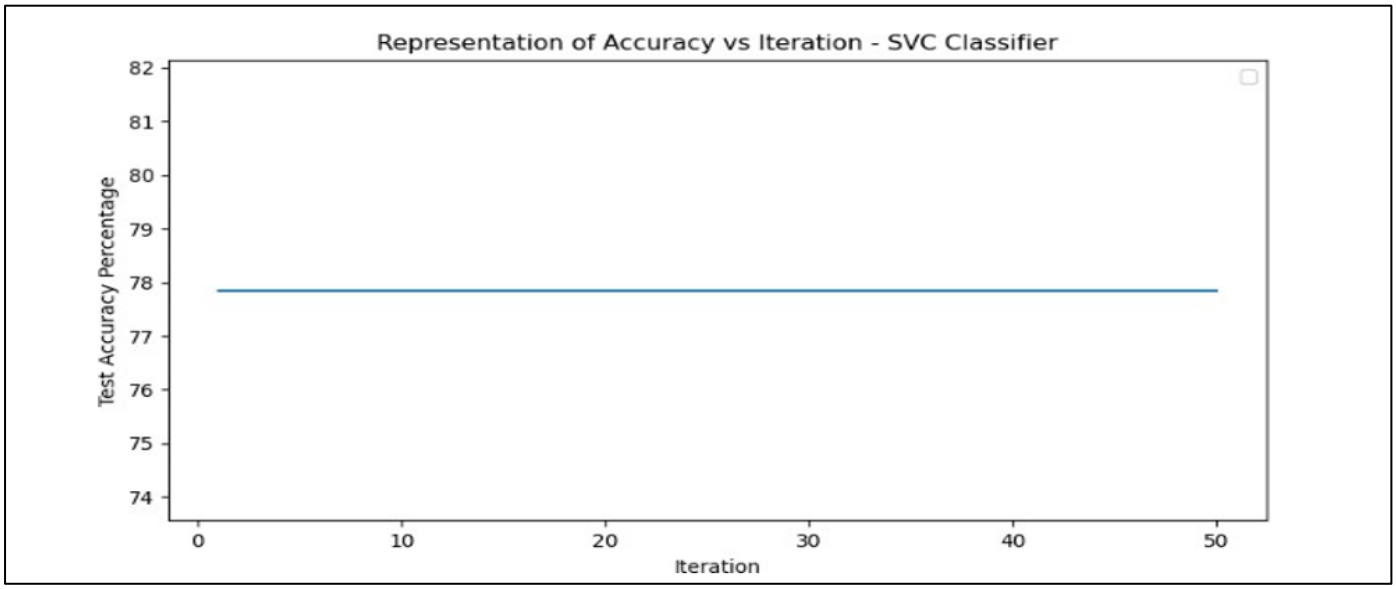

Fig 9 Testing Accuracy Estimation for the Dataset Using the rbf Function.

Table 8 Statistics for Figure 9

Standard Deviation (σ)	Mean Training Time
0	27.05s

✓ *Kernel Function – polynomial*

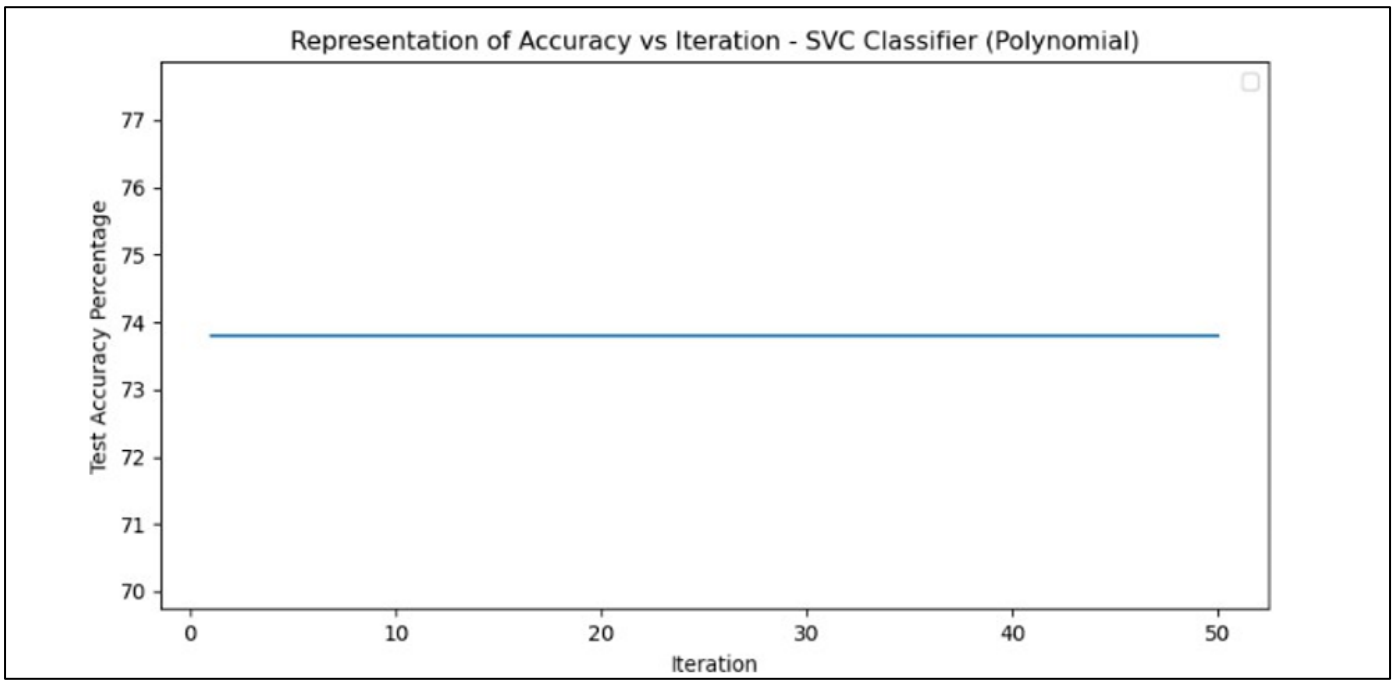

Fig 10 Testing Accuracy Estimation for the Dataset Using the Polynomial Function.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26jan1185

Table 9 Statistics for Figure 10

Standard Deviation (σ)	Mean Training Time
0.0	30.232s

✓ *Kernel Function – Sigmoid*

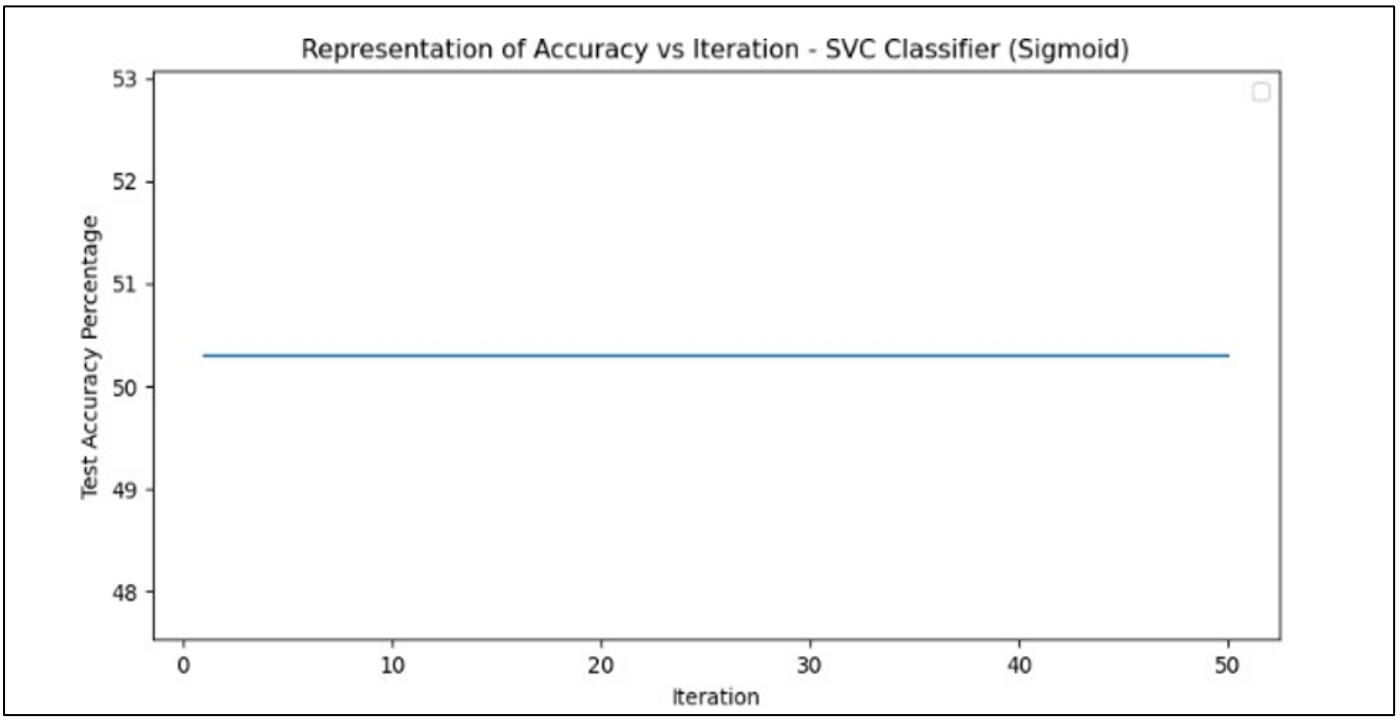

Fig 11 Testing Accuracy Estimation for the Dataset Using the Sigmoid Function.

Table 10 Statistics for Figure 11

Standard Deviation (σ)	Mean Training Time
0.0	37.382s

VII. CONCLUSIONS / FUTURE WORK

Based on empirical studies random forests performed the most optimally however at increased training time. Even the standard deviation of test accuracy is very much within acceptable limits. Future work in this area will include adding more features to this task of classifying the traffic congestion levels. It is our hope that future work in this area will result in better path planning and infrastructure support leading to cleaner and healthier cities.

ACKNOWLEDGMENT

The author Mr. Sriram Sridhar would like to thank his family and near and dear ones for their moral support without this paper would not have been possible.

REFERENCES

[1]. S. A. Ali Shah, X. Fernando and R. Kashef, "Improved Vehicular Congestion Classification using Machine Learning for VANETs," *2024 IEEE International Systems Conference (SysCon)*, Montreal, QC, Canada, 2024, pp. 1-8, doi: 10.1109/SysCon61195.2024.10553553.

[2]. Laaziza Hammoumi, Saad Farah, Mohamed Benayad, Mehdi Maanan, Hassan Rhinane,"Leveraging machine learning to predict traffic jams: Case study of Casablanca, Morocco,"Journal of Urban Management, 2025,ISSN 2226-5856, https://doi.org/10.1016/j.jum.2025.02.004.

[3]. Rafed Muhammad Yasir, Moumita Asad, Dr. Naushin Nower, Dr. Mohammad Shoyaib," Traffic Congestion Prediction Using Machine Learning Techniques", arXiv:2206.10983v4 [cs.LG] 16 Apr 2025

[4]. Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, "Extreme learning machine: Theory and applications", Neurocomputing,Volume 70, Issues 1–3, 2006, Pages 489- 501,ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2005.12.126

[5]. https://www.kaggle.com/datasets/ziya07/smart-mobility-traffic-dataset