ISSN 2063-5346

ARTIFICIAL INTELLIGENCE APPROACH TOWARDS CRITICAL PROBLEM SOLUTION – AN INNOVATIVE MACHINE LEARNING REAL-TIME INTELLIGENT TRAFFIC SYSTEM DESIGN

Sai K Kamila[1], Michael Katz[1], Jon Guenther[1], Biswajit Brahma[2], Subhendu Kumar Pani[3], Nilayam Kumar Kamila[4]

Article History: Received: 05.06.2023 Revised: 18.07.2023 Accepted: 12.08.2023

ABSTRACT:

In today's world there are several difficulties faced by the travelers to commute from one location to another location. Few of the difficulties are time of the day, days of the week, unpredicted accidents, unplanned roadworks, scheduled road repairs etc. Apart from these, there are certain environment parameters, for instance, weather variables such as rain density and wind speed which affect the traffic flow patterns significantly. In this area, much research has been accomplished, which we'll discuss in our literature survey section. Furthermore, in this article, we'll present an artificial intelligence model with the integrated predicted traffic data and its impact with respect to the weather parameters. The model's design and implementation architectural significance is presented through our mathematical analysis and simulated prediction accuracy over the other predicted models.

Keywords: Machine Learning, System design, Intelligent Model, Artificial Intelligence.

[1]Charter School of Wilmington, Wilmington Delaware, USA
[2]McKesson Corporation, San Francisco, California, USA
[3]Krupajal Engineering College, Biju Patnaik University of Technology, Odisha, India
[4]Capital One, Wilmington, DE 19801 USA

DOI:10.48047/ecb/2023.12.9.295

Introduction

Real-time traffic analysis is identified as one of the most critical real-time issues due to its high dynamicity and impact from several known and unknown factors. Although these variables are mostly dependent on the season, they have an impact year-round. Further, approximately 6 million vehicle collisions are registered annually, of which adverse weather conditions lead to nearly 1,235,000 accidents, accounting for 21% of all accidents, as the Federal Highway Administration recorded [2]. Therefore, the effect of the atmosphere on traffic flow and intensity has become an increasing concern over the past few years [3]. We do expect to notice a strong correlation between weather and traffic flow with high temperatures leading to greater traffic congestion and lower temperatures leading to less congestion. Traffic congestion has led to an increase in numerous problems in the previous years. For instance, traffic slows economic growth while increasing fuel consumption [18]. McCarthy emphasizes how traffic "congestion can cost cities billions of dollars each year" [19]. In fact, statistics demonstrate how road congestion can cost companies more than $27 billion a year plus extra petroleum cost. Moreover, traffic crowding provokes reduction of working hours and productivity overall [18]. Traffic obstruction is also a principal cause of rising air pollution: recent studies illustrate correlation among excessive health issues for individuals living near major traffic roadways [20].

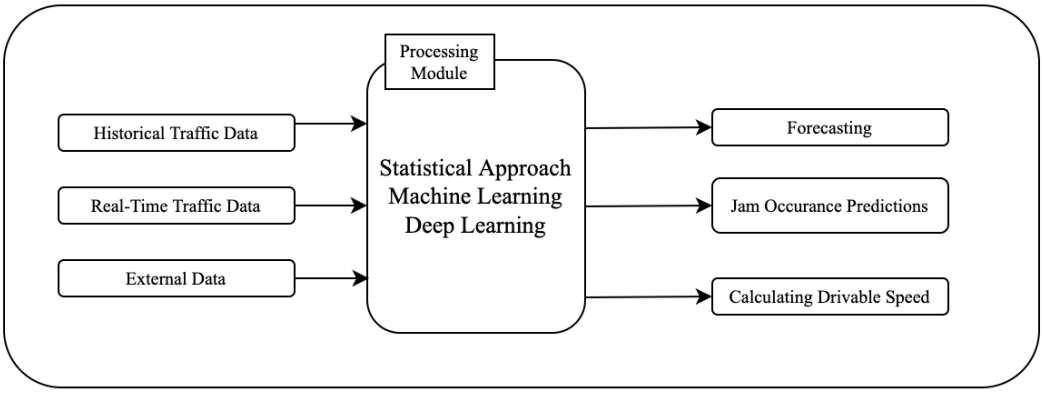


Figure 1: Existing System Design Overview for Traffic Data Processing

As per the recent publication with the industrial latest approach [24], figure 1 shows a summarized detail of the processing modules to solve the traffic congestion problems. The historical traffic data, real-time traffic data and external data (weather and social) are injected to the processing module to provide the traffic predictions and calculations for an ideal driving speed.

Due to the dramatic detrimental effects traffic blockage has on the modern world, it is an augmented concern for traffic management and authorities. With this article, we hope to enhance productivity in workplaces and reduce fuel consumption. Enhancing productivity will have a direct influence on building a strong competitive economy [21]. In addition, since vehicles will be aware of the traffic situation, statistics will mostly show a decrease in the amount of air pollution due to traffic congestion; for example, traffic decreased by more than 40% in the pandemic, hence records highlight an acceptable diminution in ozone pollution [22].

This article is organized as shown in figure 2. Overall article details are presented in the abstract and introduced in this section. In the next section, we share the background and motivation followed by the literature review. The proposed design's methodology, process, and system design are presented in successive sections,

respectively. Finally, we present the mathematical analysis and results sections extensively. And we end this article with our cohesive conclusion, acknowledgements, and references.

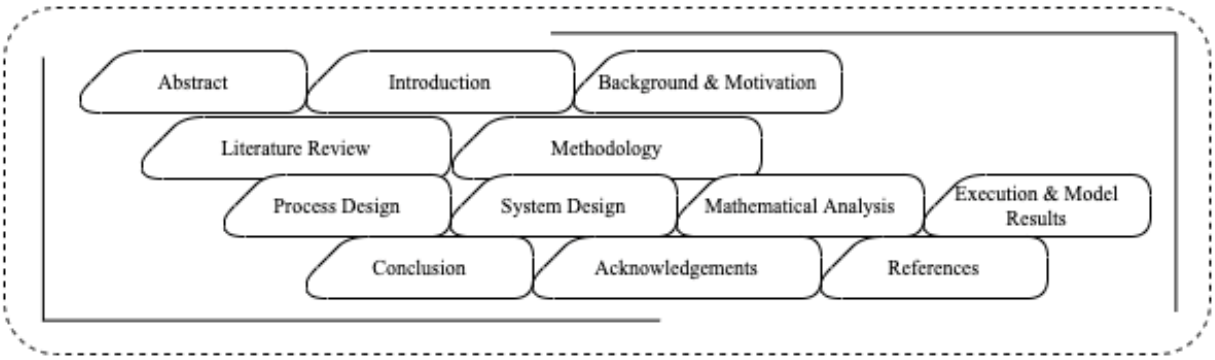


Figure 2: The details of the Article Organization

Further, in this article we aim to accomplish these core central objectives:

1. Demonstrate a strong correlation between real-time weather and traffic flow data and study the recent developing and existing traffic patterns.
2. Design a real-time system which collects, integrates, filters, and analyzes the data.
3. Build and implement a proof of concept to provide a safe travel decision with the collected data and proposed design.

Nevertheless, in this article our limitations consist of the following:

1. This proof of concept is applicable only to a single segment of the geographic region and not applied in multiple different segments: this could be a potential scope for future research.
2. Social data inclusion for traffic prediction is determined as the future scope, hence not part of this article's presentation scope.
3. The data collected in this article is from a single source API. Hence the data accuracy is not validated with other available sources. Multiple sources of the same data collection, integration, and filtration is not in this article's scope. However, data accumulated from multiple sources will undoubtedly help in data validations, and may lead to a better result and increased accuracy.

2 Background & Literature Review

Previously, AI models and applications have been enhanced to show real-time traffic based on accidents at a particular location in real-time; however, environment parameters integration and prediction has not yet been highly polished. For instance, Google Maps and Apple Maps utilize real-time traffic data based on vehicles' performances on the road to show a current traffic condition. But these platforms don't predict how the traffic will be the next day. As Akhtar et al. mentions the forecasting models have dramatic scope to improve prediction accuracy, but a multitude of current research primarily focuses on only one parameter [6].

Table 1: Content of previous research accomplished in this area.

Year	Author	Comments
2010	Mario Cools et al.	Lower temperatures reduce traffic intensity; higher temperatures increase traffic intensity.
2010	R. G. Hoogendoorn et al.	Fog (low visibility) led to a decrease in speed and acceleration.
2021	Mahmuda Akhtar et al.	Short-term traffic flow prediction is based on different traffic parameters.
2019	Zhengyang Lu et al.	Adverse winter conditions drastically impact traffic at signalized intersections.
2003	Andrew D. Stern et al.	Weather conditions lead to traffic delays even if there's low traffic flow.
2018	Yichuan Peng et al.	Intelligent Transportation Systems can mitigate negative effects of severe weather, related to reduced visibility.
2022	Hui Bi et al.	The results showcase a strong correlation between weather parameters and traffic relativity.
2010	Nour-Eddin El Faouzi et al.	European initiatives have been launched to weather response for real-time traffic surveillance.
2022	Ahmed Abohassan et al.	Investigates the level of friction in low temperatures for traffic safety.
2022	Shenghan Zhou et al.	Accounts traffic parameters for short-term traffic prediction employing deep learning models.
2022	Suvitha D et al.	Deep learning time series dependent study to determine the future traffic logistics.
2022	R. Sathiyaraj et al.	Smart Traffic Prediction is an indicator of traffic congestion which aids to avoid the crowding, relying on traffic parameters.
2022	Deekshetha H. R et al.	Prediction of real-time traffic data status utilizing previous years' traffic data records.
2020	Yuanli Gu et al.	An enhanced Bayesian combination model with deep learning for traffic flow forecasting utilizing historical traffic flow data.
2020	Gaurav Meena et al.	Various algorithms are employed to generate predictions based on traffic data and models.

Most authors generate models based on only one type of parameter (i.e., traffic or weather), or even more specifically (i.e., accident reports, traffic flow, temperature, visibility, etc.). Given Table 1, research shows strong correlation between weather and traffic [5, 4, 7, 9], most encompassing numerous models with low temperature relation to increased traffic accidents. Furthermore, past research includes deep learning models, machine learning models, and Artificial Intelligence (AI) implementations, [16] however, they only forecast the current data of that day.

Moreover, current navigating maps utilize a method of crowdsourcing. Crowdsourcing is where the object sends a signal to the server, in this case its location, then the server accumulates this information and sends updated status back to the object. Nevertheless, this provides real-time data for users who are already on the road. On the other hand, in some cases, though historical traffic data is applied and utilized in the process of creating a predictive traffic congestion model, traffic flow is the only parameter considered. As mentioned by Lv et al., current traffic flow projection mechanisms employ shallow traffic prediction models and continue to be inadequate for real-time applications and/or future objectives [17].

3 Methodology

Two Application Programming Interfaces (API), which join programs in order to perform an intended function created around sharing data or enact a previously established process, were utilized to gather weather and traffic data. Specifically, we used the TomTom Move Traffic Portal to collect traffic flow data from a certain segment in Maryland and Weather API to accumulate weather data from the same location and time. Figure 3 below summarizes the fundamental idea based on how the module is formed.

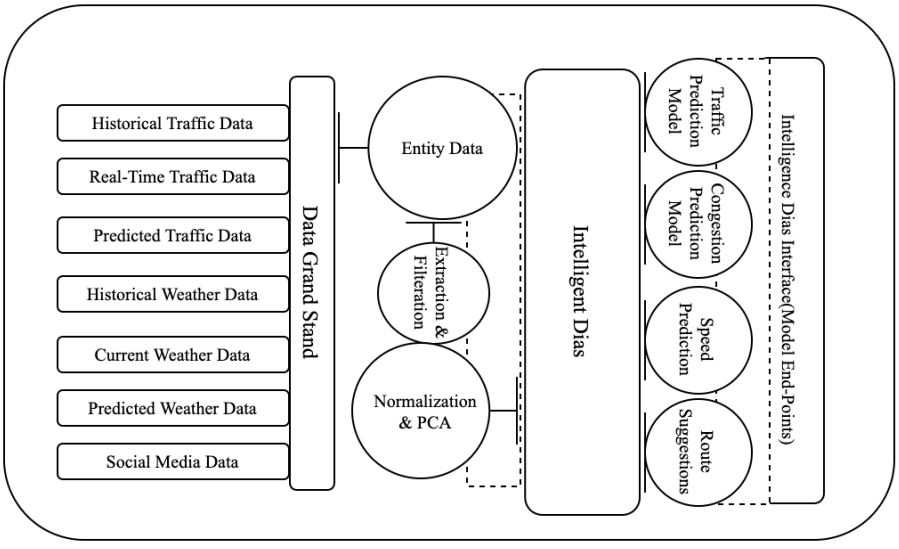


Figure 3: Proposed System Design Overview for Traffic Data Processing

The proposed design is based on an improvement mentioned in the industry published design [24]. As depicted in figure 3, the new proposed model comprises of two main sub-modules e.g., data repository grandstand (DRGS) and the Intelligent Model Dias (IMD). The DRGS module pulled the data from various sources e.g. weather api and traffic api and social data api. In this article we modeled the design only with the scope of traffic and weather api. The same data is extracted, filtered and normalized in the Intelligent Dias module.

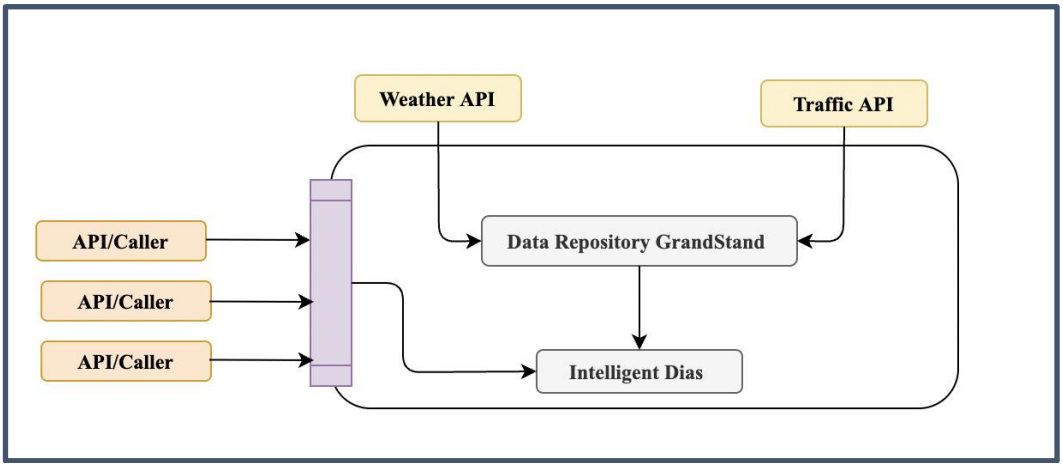


Figure 4: Data Flow for the Proposed Design with DRS and ID Applications

As shown in figure 4, Data is obtained through the two API functions, and then integrated through the Data Repository. Consequently, the data points are fed into the Intelligent Dias which provides us with an analytical prediction visual. In other words, Weather API and Traffic API were both called and stationed at the Data Repository Grandstand, a processor application. Then those data points were tested to generate an intelligent model which can predict the traffic flow level given the weather. Basically, callers will then retrieve information on how the real-time weather will impact traffic flow from the intelligent model.

3.1 Utilized Applications' Details

Each of the applications introduced in Figure 1 are detailed extensively below:

1. **Weather API**
   As mentioned before, the weather API was collected from Weather API [1]. We assembled numerous parameters through this and recorded a variety of changes within the components, as time progressed. Below in Figure 2, we have outlined the particular variables we captured. Moreover, the region and time were also documented throughout the process to maintain similarity between the collection point of weather and traffic data. The real-time weather data was updated every 15 minutes and it is evident in our observations that the weather data was continuously changing without errors.

Figure 5 illustrates all the components captured after calling the Weather API. For our scope, we extracted the below listed data points:

a. Location
   a. Name
   b. Region
   c. Country
   d. Latitude & Longitude
   e. Time Zone
   f. Local Time
b. Current
   a. Last Updated
   b. Temperature (Celsius & Fahrenheit)
   c. Day/Night
   d. Basic Description of Weather

e. Wind (mph & kph)
f. Wind Degree & Direction
g. Pressure (mb & in)
h. Precipitation (mm & in)
i. Humidity
j. Percentage of Cloud
k. Feels-Like Temperature (Celsius & Fahrenheit)
l. Visibility (km & m)
m. Gust (mph & kph)


```
{
    "location": {
        "name": "Port Deposit",
        "region": "Maryland",
        "country": "USA",
        "lat": 39.62,
        "lon": -76.08,
        "tz_id": "America/New_York",
        "localtime_epoch": 1670113117,
        "localtime": "2022-12-03 19:18"
    },
    "current": {
        "last_updated_epoch": 1670112900,
        "last_updated": "2022-12-03 19:15",
        "temp_c": 13.7,
        "temp_f": 56.7,
        "is_day": 0,
        "condition": {
            "text": "Partly cloudy",
            "icon": "//cdn.weatherapi.com/weather/64x64/night/116.png",
            "code": 1003
        },
        "wind_mph": 14.8,
        "wind_kph": 23.8,
        "wind_degree": 295,
        "wind_dir": "WNW",
        "pressure_mb": 1020.0,
        "pressure_in": 30.13,
        "precip_mm": 0.0,
        "precip_in": 0.0,
        "humidity": 86,
        "cloud": 57,
        "feelslike_c": 12.3,
        "feelslike_f": 54.1,
        "vis_km": 10.0,
        "vis_miles": 6.0,
        "uv": 1.0,
        "gust_mph": 18.1,
        "gust_kph": 29.2
    }
}
```


Figure 5: Weather API data components captured from the Weather API

1. Traffic API


```
{
    "flowSegmentData": {
        "frc": "FRC2",
        "currentSpeed": 33,
        "freeFlowSpeed": 33,
        "currentTravelTime": 70,
        "freeFlowTravelTime": 70,
        "confidence": 1,
        "roadClosure": false,
        "coordinates": {
            "coordinate": [
                {
                    "latitude": 39.736517695645283,
                    "longitude": -75.541795702128113
                },
                {
                    "latitude": 39.736799771000634,
                    "longitude": -75.541590890745752
                },
                {
                    "latitude": 39.737266177109490,
                    "longitude": -75.541259503480362
                }
            ]
        },
        "@version": "traffic-service-flow 1.0.070"
    }
}
```


Figure 6: Traffic API data components captured from the Traffic API

Likewise, Traffic API was captured in a similar manner from TomTom's Move Traffic Portal. Unlike WeatherAPI's multiple components, we only captured two central traffic flow parameters: the current/free flow speed and the current/free flow travel time. These two components mainly summarize the traffic flow at a given instance. Again, the latitude and longitude coordinates are measured here as well to ensure that both weather and traffic data are consistent with each other.

Figure 6 shows the actual parameters captured including the real-time traffic flow time and speed.

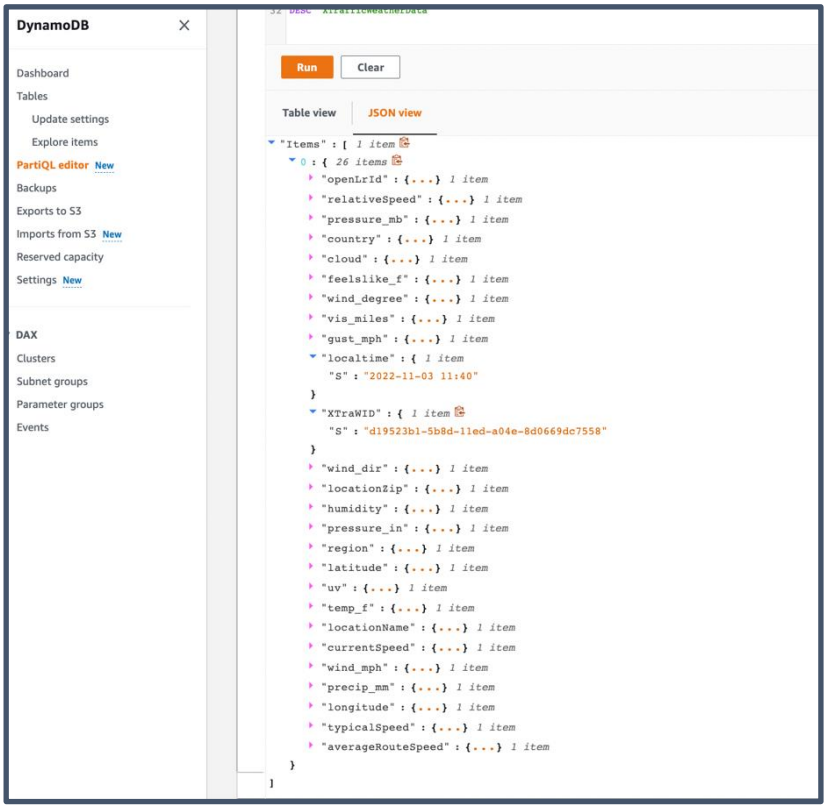


Figure 7: Data points stored in the DRGS Data Store.

1. **Data Repository Grand Stand**
   The Data Repository Grand Stand (DRGS) is a processor application for the weather and traffic data. Here, both the weather and traffic data are stored in an AWS DynamoDB table. AWS, also known as Amazon Web Services, is a platform offered by Amazon; in our research, we employed a specific part of their services, the DynamoDB table. Essentially, as these APIs were called, they were directly stored in a DynamoDB table for later reference.

2. **Intelligent Model Dias**
   The Intelligent Model Dias (IMD) is ultimately the final goal which connects the users and this traffic predictive model. This portion was built using Project Jupyter. The weather and traffic data which were combinedly stored in the DynamoDB table were compiled to generate this weather impacting traffic model. Furthermore, 15% of the data was employed as testing data whereas the rest were utilized to train the model. For

instance, from the total recorded weather and traffic data, 15% was kept aside for testing the model's accuracy whereas the other 75% was used to build and train the model. Figure 8 showcases how the two data sets were trained and tested and generated how accurate/efficient the results were on a real-time basis.

The AIML Regression(T, W, d) algorithm provides the details of the traffic dataset $T = \{(K_i, V_i)\}$ and weather dataset $W = \{(K_i, V_i)\}$ extracted integration. The applied dataset is $D_t = \{(T_t - S_t) \cup W_t\}$ at a $t - time\ inteval$. The split of the training dataset and test data set on feature space is also determined with the random dataset of 85% of the original data $TR(X)_t = Random(0.85)[D_t], TS(X)_t = (D_t - TR(X)_t)$.

Algorithm: AIMLRegression(T, W, d)

Initialization:

$T = \{(K_i, V_i) | i = 1..n, K \in Traffic\ Parameters\ and\ V \in Parameter\ Values\}$
$W = \{(K_i, V_i) | i = 1..n, K \in Weather\ Parameters\ and\ V \in Parameter\ Values\}$
$D_t = \{(T_t - S_t) \cup W_t |\ at\ t - time\ interval\}$;
where D_t is All Traffic Data Set except $S_t(Speed\ Data)$

begin

While$(t > 0$ && $size(D_t) > D_{th} + d\ Days)$ do
$TR(X)_t = Random(0.85)[D_t], TS(X)_t = (D_t - TR(X)_t)$
$TR(Y)_t = Random(0.85)[S_t], TS(Y)_t = (S_t - TS(Y)_t)$
$TR(X)_t.PCA = PCA(TR(X)_t)$
$TR(Y)_t.PCA = PCA(TR(Y)_t)$
$LR(TR(X)_t, TS(X)_t), where\ LR\ is\ Logistic\ Regression$
$LR(TR(Y)_t, TS(Y)_t)$
end
$T = T - Data(d\ Days)$
$W = W - Data(d\ Days)$
AIMLRegression(T, W, d)

end

Similarly, the training and test dataset on the output is also determined with $TR(Y)_t = Random(0.85)[S_t], TS(Y)_t = (S_t - TS(Y)_t)$ respectively. The history is maintained in the DRGS submodules, but the latest data from traffic and weather is refreshed with the logic of $T = T - Data(d\ Days)$ and $W = W - Data(d\ Days)$ as specified in the algorithms in the scope of IMD module.

3.2 Process Design

```
speedData.loc[(speedData['averageRouteSpeed'] >= 72.5), 'safeTravel'] = 0
speedData.loc[(speedData['averageRouteSpeed'] < 72.5), 'safeTravel'] = 1
speedData = speedData['safeTravel']
speedData
X_train, X_test, y_train, y_test =
               train_test_split(weatherData,speedData,test_size=0.15,random_state=0)

scaler = StandardScaler()

# Fit on training set only.
scaler.fit(X_train)

# Apply transform to both the training set and the test set.
X_train_pca = scaler.transform(X_train)
X_test_pca = scaler.transform(X_test)
logisticRegr = LogisticRegression()
logisticRegr.fit(X_train,y_train)

y_train_hat =logisticRegr.predict(X_train)
train_accuracy = accuracy_score(y_train, y_train_hat)*100
print("Accuracy for our Training dataset with PCA is : {:.3f}%".format(train_accuracy))
y_test_hat=logisticRegr.predict(X_test)
test_accuracy=accuracy_score(y_test,y_test_hat)*100
test_accuracy
print("Accuracy for our Testing dataset with PCA is : {:.3f}%".format(test_accuracy) )

logisticRegr = LogisticRegression()
logisticRegr.fit(X_train_pca,y_train)

y_train_hat =logisticRegr.predict(X_train_pca)
train_accuracy = accuracy_score(y_train, y_train_hat)*100
print("Accuracy for our Training dataset with PCA is : {:.3f}%".format(train_accuracy))

y_test_hat=logisticRegr.predict(X_test_pca)
test_accuracy=accuracy_score(y_test,y_test_hat)*100
test_accuracy
print("Accuracy for our Testing dataset with PCA is : {:.3f}%".format(test_accuracy) )
```

Figure 8.1: Extracted Data from Weather & Traffic API Integration & Regression

It is tedious to manually call each API and store the data one at a time; therefore, in order for the API's to be called automatically continuously in a certain time interval, we wrote a central python code which continuously ran and presented us with numerous stored data points.

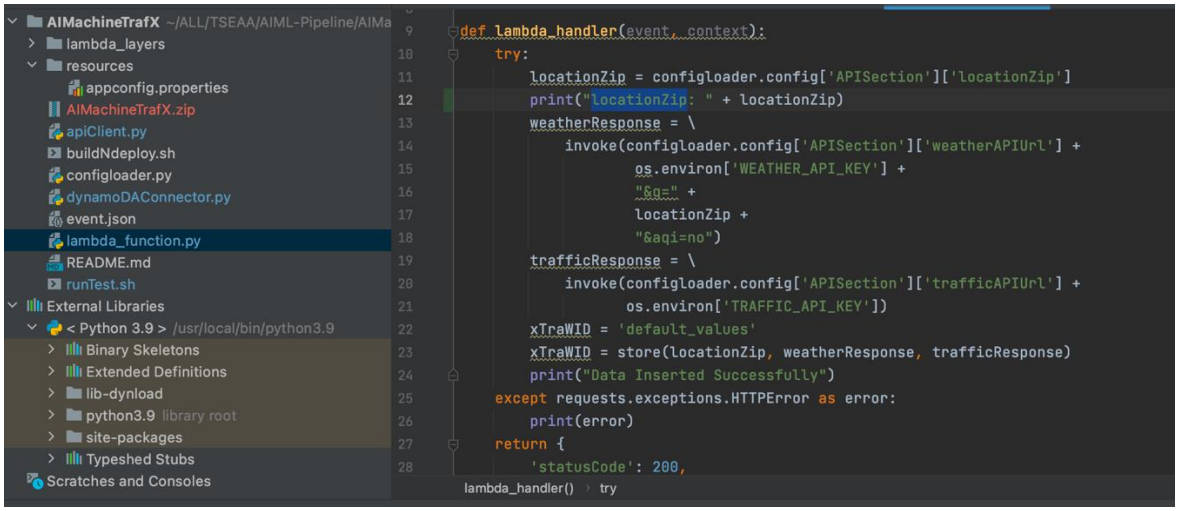


Figure 8.2: central code for extracting the data points and storing them.

As shown in the code above, this lambda function will call the API through the API configuration keys, located in the configloader.py file, and store the locationZip, weatherResponse, and trafficResponse in the DynamoDB table (xTraWID). When data is inserted successfully, the function will print 'Data Inserted Successfully' otherwise it'll show an error.

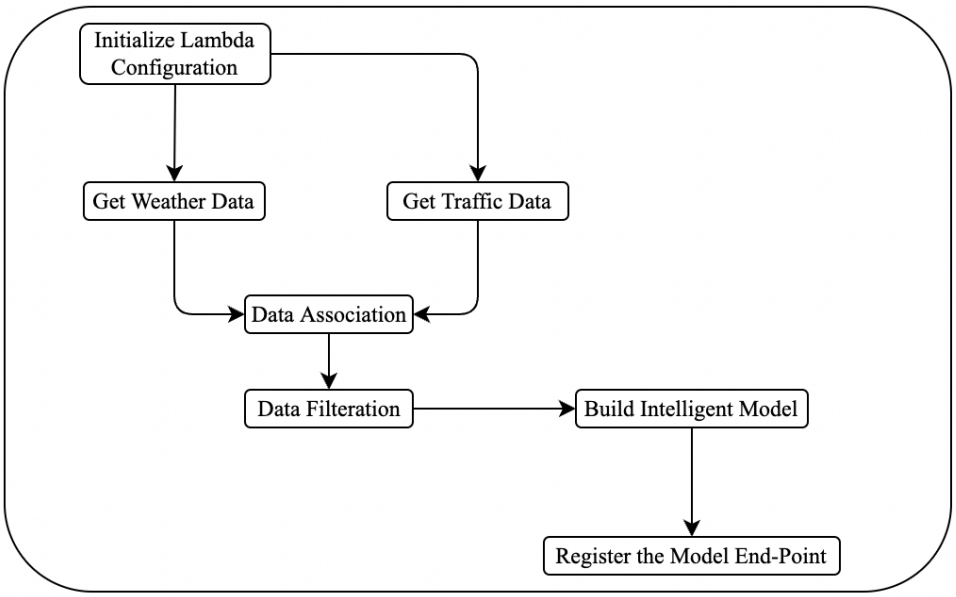


Figure 9: Lambda Integration for DRGS & IMD Sub-Module Automation

Figure 9 summarizes the concept of how this module is built: initialization, combination, filtration, and presentation. This automates the DRGS and IMD sub-module, so as to capture the Weather and Traffic API's historical data, current data and predicted data and process in the regular intervals to make the Intelligent model an effective and efficient model.

In addition to the basic flow diagram shown in Figure 2, our lambda processor will be configured with the API keys, and when the lambda function is executed, it'll gather weather and traffic data. Consequently, both the data sets are associated in the Data Association Model; further, the data filtration process will remove all unwanted fields and create a field that is needed to be processed. This field and the data points are then utilized by the intelligence model which creates a model base and is finally registered as an end-point for clients to utilize.

3.3 System Design

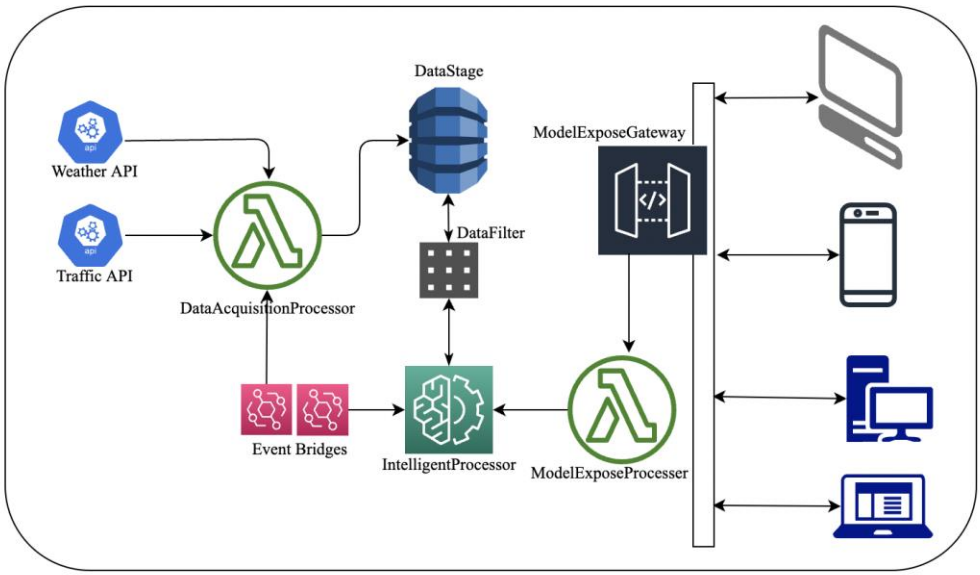


Figure 10: Proposed Model's System Design & Automation

The entire system is designed by combining these various components. Certainly, the weather API and traffic API play a crucial role in granting us our required data sets, nevertheless, without the precise lambda processor, specific trigger, data storage, and data filtration mechanisms, the model would not be conveniently exposed to the client users.

Figure 10 shows how the proposed model works in the cloud environment with the automated data capture and integration with the Intelligent Model and also how the data will be processed to be presented to external users for the usage.

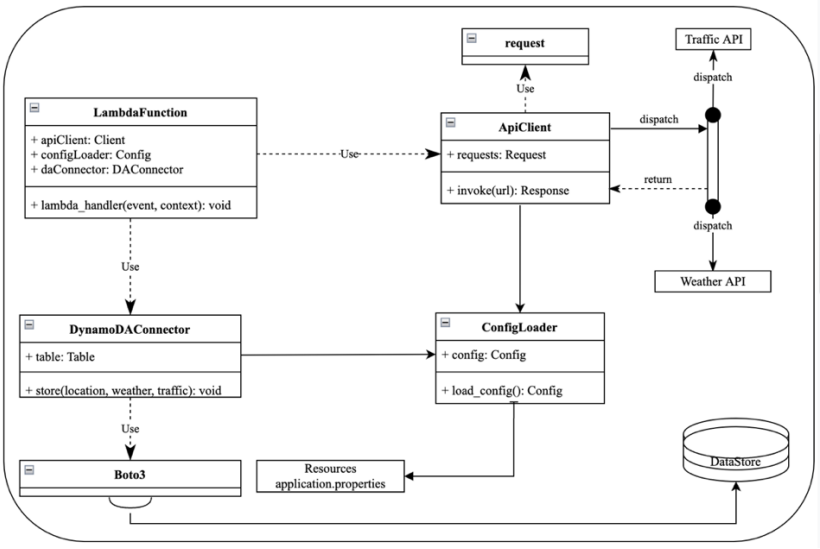


Figure 11: Low Level Design Diagram for DRGS Sub-Module

When the Data Acquisition Processor (DAP) will be executed by Event Bridges periodically, the Lambda Processor will accrue respective data from the external APIs. Event Bridges is a platform framed and constructed to trigger the DAP: for our proof-of-concept model, we manufactured our DAP to be triggered in 30-mins segments. The data assembled by DAP is stored in the Data Stage Platform (DSP). DSP maneuvers Data Filtration Management (DFM) to filter the necessary fields which will be employed by the Intelligent Processor Application (IPA). The IPA will process filtered data and construct a model base which will be exposed to the model's endpoint. The model's endpoint is accessed by a Model Exposed Processor (MEP) which is published to the external world through a Model Exposed Gateway (MEG).

In DRGS Sub-Module, we used ApiClient and DAConnector used by the LambdaFunction. ApiClient is employed to interact with the Traffic API and Weather API with the in-build request module of the implemented code (in our case it's Python 3.x). Similarly DAConnector is a connector to interact with the Data Store (in this case it's AWS Dynamo Database) using the Boto3 module of the implemented language code.

In the IMD module, IntelligentDias uses the StandardScaler to make the PCA and LogisticRegression to make the regression evaluation. The AccuracyScore module ensures to evaluate the accuracy between the training data set and test dataset.

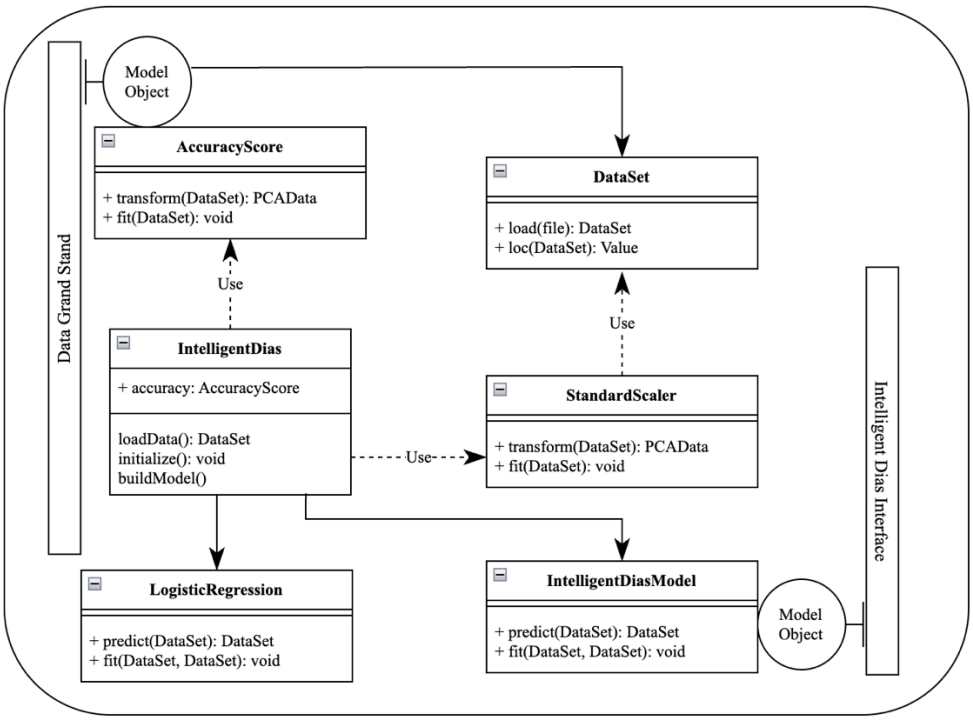


Figure 12: Low Level Design Diagram for IMD Sub-Module

Intelligent Dias Model is used to expose the Model object through our Intelligent Dias Interface for the external users to utilize the traffic predictions and safe travel speed data.

Mathematical Analysis:

For the transformation from input space's properties x_i with the initial weight vector w_i with linear interceptor $b$as shown in below equation.

$$z = w.x + b = \left[\sum_{i=1}^{n} w_i x_i\right] + b;$$

For the probability notation,

$$\sigma(z) = {}^{1}/_{(1+e^{-z})} = {}^{1}/_{[1+exp(-z)]}$$

$$p(y=1) = \sigma(w.x+b)$$

$$= {}^{1}/_{[1+exp(-(w.x+b))]}$$

$$= \frac{1}{1+exp[-([\sum_{i=1}^{n} w_i x_i]+b)]}$$

$$p(y=0) = 1 - \sigma(w.x+b)$$

$$= 1 - \left[{}^{1}/_{[1+exp(-(w.x+b))]}\right]$$

$$= \left[\frac{exp(-(w.x+b)}{1+exp(-(w.x+b))}\right]$$

$$= \left[\frac{exp[-([\sum_{i=1}^{n} w_i x_i] + b)]}{1 + exp[-([\sum_{i=1}^{n} w_i x_i] + b)]}\right]$$

So, the decision function is

$$= \begin{cases} 1, if\ p(y = 1|x) > 0.5 \\ 0, otherwise \end{cases}$$

Let the training model dataset be

$$D = \left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)\right\}$$

This is composed of design matrix X and the output response matrix be y. Hence,

$$D = \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(n)} \\ \vdots & \ddots & \vdots \\ x_n^{(1)} & \cdots & x_n^{(n)} \end{bmatrix}, x_i = \begin{bmatrix} x_i^{(1)} \ldots & x_i^{(m)} \ldots & x_i^{(n)} \end{bmatrix} \text{ and } x_i = \begin{bmatrix} y^{(1)} \\ y^{(m)} \\ y^{(n)} \end{bmatrix}$$

The linear model is

$$y = x\Theta + \epsilon$$

Where ϵ is the noise vector, and Θ be the learn parameter vector.

$$\epsilon = (\epsilon_1, \epsilon_2, \ldots \epsilon_n)$$

Hence the predicted model is

$$\hat{y} = X\widehat{\Theta}\ where\ \hat{y} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(m)} \\ \hat{y}^{(n)} \end{bmatrix}$$

A suitable learned vector $\widehat{\Theta}$ will minimize the noise vector ϵ i.e., $y = \hat{y}$. This closeness can be captured with loss function as below,

$$C(\Theta, \mathrm{D}) = \frac{1}{n}\sum_{i=1}^{n} C_i(\Theta), where\ \ C_i(\Theta) = C_i\left(\widehat{\Theta}, y^{(i)}, \hat{y}^{(i)}\right)$$

$C_i(\Theta)$ is i^{th} sample. We need to choose $\widehat{\Theta}$ so close to Θ so that $\Theta = \widehat{\Theta}$. If we choose the loss function as quadratic loss, then we've

$$C_i(\Theta) = \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

So,

$$C(\Theta, \mathrm{D}) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

$$= \frac{1}{n}\|y - \hat{y}\|^2$$

The learned parameter vector as

$$\widehat{\Theta} = \min_{\hat{y}=x\Theta} \frac{1}{n}\|y - \hat{y}\|^2 = \min\|y - x\Theta\|^2$$

The solution for minimization can be derived with respect to Θ as

$$\frac{\partial\widehat{\Theta}}{\partial\Theta} = 0\ i.e. \frac{\partial\|y - x\Theta\|^2}{\partial\Theta} = 0$$

$$\frac{\partial(y - x\Theta)^T(y - x\Theta)}{\partial\Theta} = 0$$

$$\frac{\partial[y^T y - 2y^T x\Theta + \Theta^T \mathrm{x}^T x\Theta]}{\partial\Theta} = 0$$

$$-2\mathrm{y}^T x + 2\mathrm{x}^T x\Theta = 0$$

$$\mathrm{y}^T = \mathrm{x}^T\Theta, \mathrm{i.e.}\,\Theta = (\mathrm{x}^T)^{-1}\mathrm{y}^T$$

So, with this $\Theta = (\mathrm{x}^T)^{-1}\mathrm{y}^T$ value with training set, the $\widehat{\Theta}$ error could be minimized which will make the regression prediction with close accuracy.

4 Execution and Results

Upon executing our DAP system, Figure 4, we get the results as shown in Figure 9.

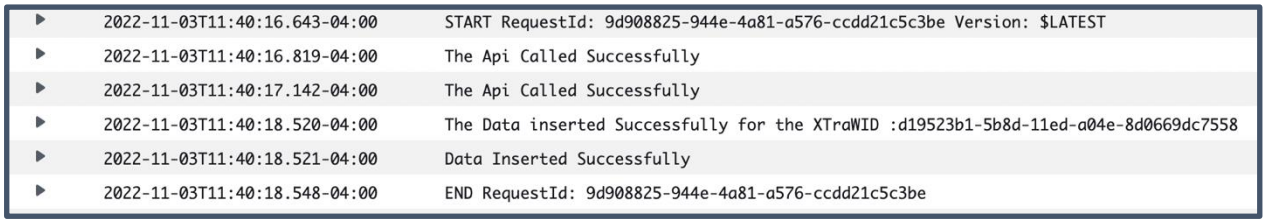


Figure 9: The result after Lambda function was run and data was inserted in the DynamoDB table.

'The Api Called Successfully' was displayed twice for the two respective APIs called (weather and traffic). The next line exhibits that the data was inserted into the DynamoDB table: the XTraWID (XTraffic Weather ID) was a key ID for the platforms to communicate about where the data should be placed. Hence, since the XTraWID = 'd19523b1-5b8d-11ed-a04e-8d0669dc7558' the data accumulated was signaled to store at the corresponding place.

Furthermore, after generating our IPA model, Figure 5, we received the accuracy percentage as shown in Figure 10.

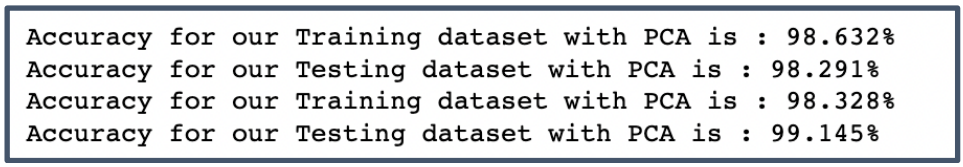


Figure 10: The accuracy rate for the IPA model with the training and testing dataset for the two sets of data collected.

PCA refers to Principal Component Analysis, a technique for comprehending large data sets with various numbers or dimensions and summarizing the overall data to a smaller set for easier visualization and analysis [23]. As mentioned earlier, the training dataset was 75% of the data collected and the testing dataset was 15% of total data, refer to Figure 5 (section 3.1).

4.1 Intelligent Model Results

When data was filtered – trained and tested – we experimented with a variety of speeds to see which speed illustrates a clear distinction between safe and unsafe driving conditions. The graphs below exhibit green data points where weather conditions are good and therefore traffic flows at a higher speed, safe zone; the blue data points demonstrate where undesired weather conditions for driving purposes are present and therefore traffic speed is lower, unsafe zone. In addition, some graphs have overlapping blue and green points, this indicates other traffic influential factors (i.e., road accidents, construction, etc.) which is not discussed in this article. While viewing the graphs, *blue is unsafe and green is safe.*

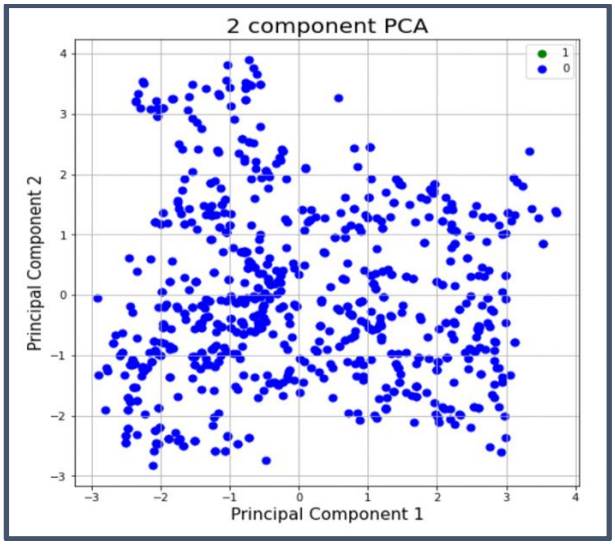


Figure 11: Speed less than 120 mph

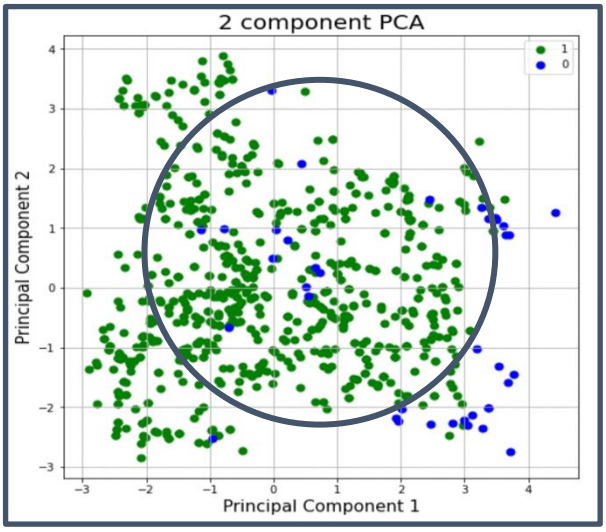


Figure 12: Speed less than 100 mph

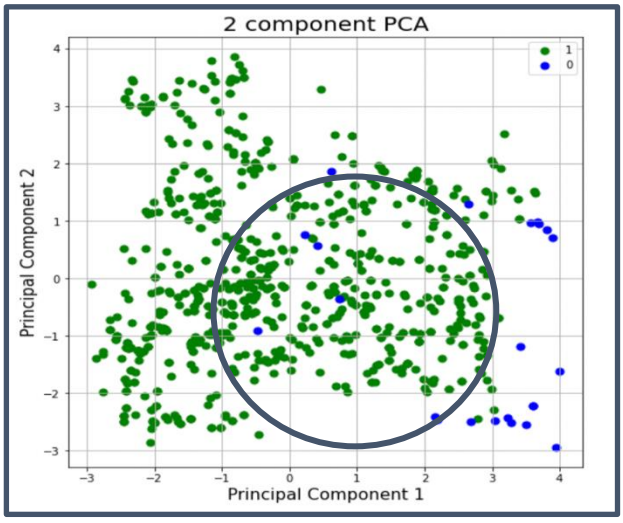


Figure 13: Speed less than 90 mph

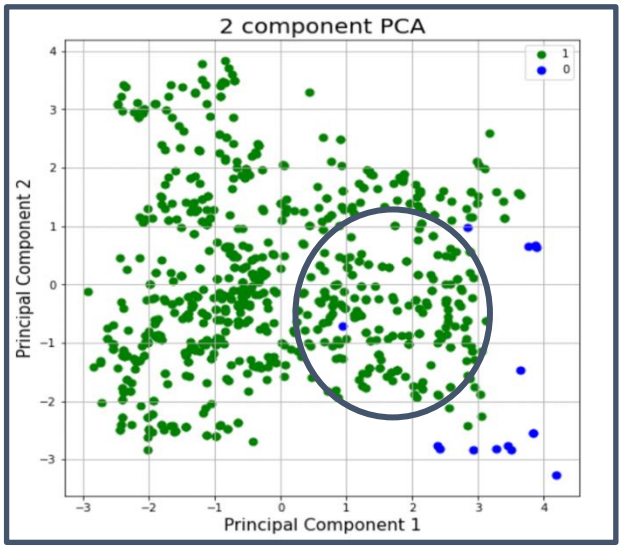


Figure 14: Speed less than 80 mph

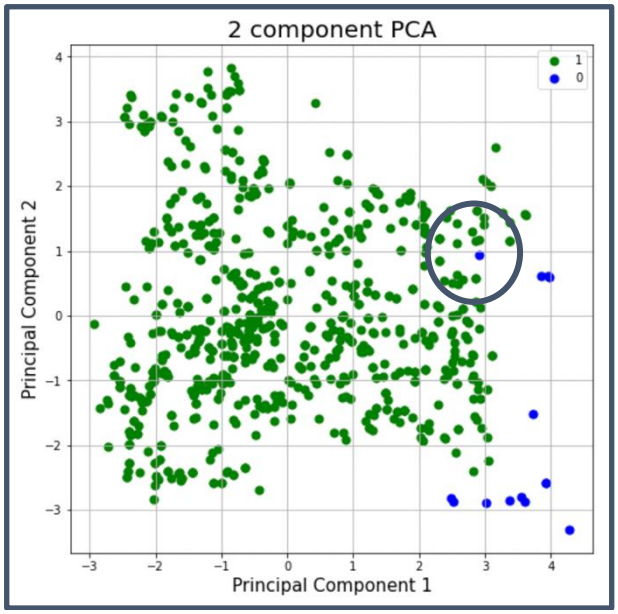


Figure 15: Speed less than 75 mph

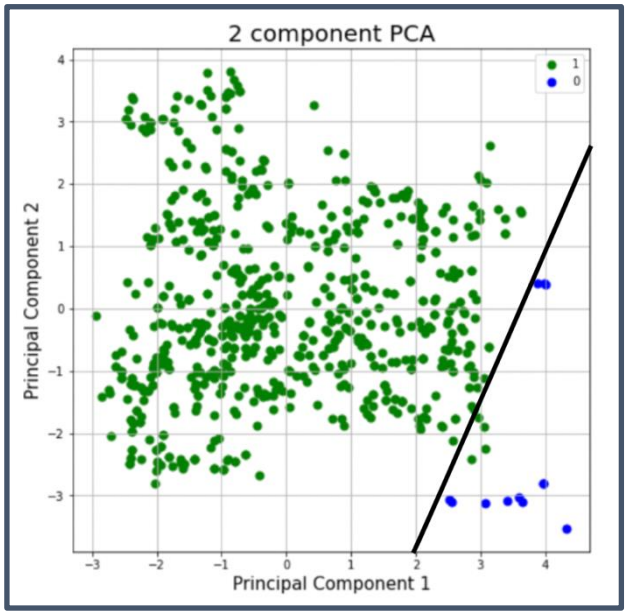


Figure 16: Speed less than 72.5 mph

4.11 IPA Results Discussion

Figure 11 displays only blue data points because based on the data we gathered none of the traffic speeds captured was above 120 mph, and all points fell under the category of below 120 mph thus 120 mph presented an inconclusive conclusion. Hence, we decreased the speed to under 100 mph (Figure 12), yet there's no clear distinction between the safe and unsafe zone. Therefore, we continued decreasing the speed limit (Figure 13, 14, 15), the circled area spotlights the overlap (which are outliers – speed lessened due to other causes); based on the correlation, with lower speeds the circle's area decreased developing a clear distinction between the safe and unsafe zone as emphasized in Figure 16. With this, we established that 72.5 mph sketched an explicit difference between the two zones: nevertheless, based on the given scenarios, we can expect the safe speed data to be in the range of 70 mph to 80 mph, but we would need to study this with different road segments rather than just one, which is not in the scope of this article.

As per the study by the author in [24], and [25] we observe that the existing model is working with 87.5% of accuracy through Random Forest methods, 90% through KNN and DL models. Convolutional Neural Network CNN model works with 89.5% accuracy.

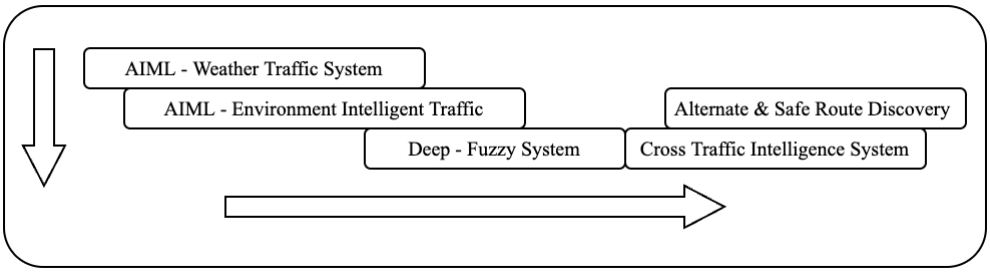
ML Approaches	% Accuracy
Random Forest (RF)	87.5%
K-Nearest Neighbours(KNN)	90%
Deep Learning (DL)	90%
Convolutional Neural Network (CNN)	89.5
ML Integrated HCP PCA with Logical Regression (MLI-HCP)	98.29

However, our proposed model works with the 98.29% of accuracy when taken the models with historical data, current data and predicted data (available in the APIs)

and when the features transformed with principal component analysis data.

Conclusion

We've studied and reviewed a variety of sources on how traffic and weather greatly correlate with each other. And based on our background and literature review, Table 1, we accomplished that though authors recognize this correlation, numerous rely on a specific parameter to approach the view of traffic flow. This research provides us a model in which we learn how to design a real-time system of traffic patterns; in fact, this design indicates a sustainable and reliable mechanism to collect, integrate, filter, and analyze data to effectively observe a safe travel speed. The model's design predicts the traffic speed in which it is safe or unsafe to travel, and this is our in progress and future scope of this research.



The analysis and design model of this project's Artificial Intelligence Machine Learning and Deep Learning aspects provides a greater extension of methodologies and its significance to discover the solutioning of unique critical complications. We are inclined that with these Artificial Intelligence conceptualization implemented into different missions, essential obstacles through our future scope models e.g. AIML-Environment Intelligent, Deep – Fuzzy system, cross traffic intelligence (Road to Rail, Road/Rail to Air/Water) will provide effective prediction, analysis results, and innovative solutions.

Contribution/Acknowledgement:
Sai K Kamila: Initial Concept, Draft, Analysis & Design, Implementations, Math Analysis, Model Build & Results Comparison
Michael Katz, Jon Guenther: Draft Review, Structural Alignments, Commentary Review, Intensive Native English Proof Reading
Biswajit K Brahma & Subhendu K Pani: Contents Sanitization, Analysis & Design Review, Proof Reading, Intensive Tech Preparation Support
Nilayam K Kamila: Article Review, Guidance, Math Analysis Support, Technical Commentary Review, Tech & Result Validations

References

1. *JSON and XML weather API and Geolocation Developer API*. Weather API - WeatherAPI.com. (n.d.). Retrieved December 4, 2022, from https://www.weatherapi.com/
2. *Welcome to road weather management*. Road Weather Management - FHWA Office of Operations. (n.d.). Retrieved December 4, 2022, from https://ops.fhwa.dot.gov/weather/index.asp
3. *Modelling the effects of environmental factors on traffic flow parameters*. (n.d.). Retrieved December 5, 2022, from https://www.ripublication.com/irph/ijert19/ijertv12n12_115.pdf
4. R. G. Hoogendoorn, G. Tamminga, S. P. Hoogendoorn and W. Daamen, "Longitudinal driving behavior under adverse weather conditions: adaptation effects, model performance and freeway capacity in case of fog," 13th International IEEE Conference on Intelligent Transportation Systems,

2010, pp. 450-455, doi: 10.1109/ITSC.2010.5625046.

5. Cools, M., Moons, E., & Wets, G. (2010). Assessing the Impact of Weather on Traffic Intensity. *Weather, Climate, and Society*, *2*(1), 60–68. http://www.jstor.org/stable/24907338
6. Akhtar, M., & Moridpour, S. (2021, January 30). *A review of traffic congestion prediction using Artificial Intelligence*. Journal of Advanced Transportation. Retrieved December 4, 2022, from https://www.hindawi.com/journals/jat/2021/8878011/
7. Lu, Z., Kwon, T. J., & Fu, L. (2019, February 6). *Effects of winter weather on traffic operations and optimization of signalized intersections. Journal of Traffic and Transportation Engineering (English Edition). Retrieved December 4, 2022, from https://www.sciencedirect.com/science/article/pii/S2095756417301149*
8. *Peng, Y., Jiang, Y., Lu, J., & Zou, Y. (2018). Examining the effect of adverse weather on road transportation using weather and traffic sensors. PLOS ONE, 13(10), e0205409. https://doi.org/10.1371/journal.pone.0205409*
9. *Bi, H., Ye, Z., & Zhu, H. (n.d.). Explore scientific, technical, and medical research on ScienceDirect. ScienceDirect.com | Science, health and medical journals, full text articles and books. Retrieved December 4, 2022, from https://www.sciencedirect.com/*
10. *El Fouzi, N.-E., Billot, R., Nurmi, P., & Nowotny, B. (2010). Effects of adverse weather on traffic and safety - sirwec.org. SIRWEC. Retrieved December 5, 2022, from https://sirwec.org/wp-content/uploads/2022/04/Quebec-D-45.pdf*
11. *Abohassan, A., El-Basyouny, K., & Kwon, T. J. (2022). Effects of Inclement Weather Events on Road Surface Conditions and Traffic Safety: An Event-Based Empirical Analysis Framework. Transportation Research Record, 2676(10), 51–62. https://doi.org/10.1177/03611981221088588*
12. *Zhou, S.; Wei, C.; Song, C.; Fu, Y.; Luo, R.; Chang, W.; Yang, L. A Hybrid Deep Learning Model for Short-Term Traffic Flow Pre-Diction Considering Spatiotemporal Features. Sustainability 2022, 14, 10039. https://doi.org/10.3390/ su141610039*
13. *D, S., & M, V. (2022). Cross-domain opinion classification via aspect analysis and attention ... Wiley Online Library. Retrieved December 5, 2022, from https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6957*
14. *Sathiyaraj, R., Bharathi, A., Khan, S., Kiren, T., Khan, I. U., & Fayaz, M. (2022, April 29). A genetic predictive model approach for smart traffic prediction and congestion avoidance for Urban Transportation. Wireless Communications and Mobile Computing. Retrieved December 5, 2022, from https://www.hindawi.com/journals/wcmc/2022/5938411/*
15. *H R, Deekshetha & Madhav, A. & Tyagi, Amit. (2022). Traffic Prediction Using Machine Learning. 10.1007/978-981-16-9605-3_68.*
16. *Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao and H. Zhang, "An Improved Bayesian Combination Model for Short-Term Traffic Prediction With Deep Learning," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 1332-1342, March 2020, doi: 10.1109/TITS.2019.2939290.*
17. *Y. Lv, Y. Duan, W. Kang, Z. Li and F. -Y. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865-873, April 2015, doi: 10.1109/TITS.2014.2345663.*

18. *Sweet, Matthias. (2011). Does Traffic Congestion Slow the Economy?. Journal of Planning Literature. 26. 391-404. 10.1177/0885412211409754.*
19. *McCarthy, N. (2020, March 10). Traffic congestion costs U.S. cities billions of dollars every year [infographic]. Forbes. Retrieved December 5, 2022, from https://www.forbes.com/sites/niallmccarthy/2020/03/10/traffic-congestion-costs-uscities-billions-of-dollars-every-year-infographic/?sh=597d06b94ff8*
20. *Zhang, K., & Batterman, S. (2013, April 15). Air pollution and health risks due to vehicle traffic. The Science of the total environment. Retrieved December 5, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4243514/*
21. *Louis H. Amato, Richard J. Cebula & John E. Connaughton (2022) State productivity and economic growth, Regional Studies, Regional Science, 9:1, 180-203, DOI: 10.1080/21681376.2022.2059393*
22. *Sommer, L., Hersher, R., Jingnan, H., & Benincasa, R. (2020, May 19). Traffic is way down because of Lockdown, but air pollution? not so much. NPR. Retrieved December 5, 2022, from https://www.npr.org/sections/health-shots/2020/05/19/854760999/traffic-is-way-down-due-to-lockdowns-but-air-pollution-not-so-much*
23. *Sartorius. (2020, August 18). What is principal component analysis (PCA) and how it is used? Sartorius. Retrieved December 5, 2022, from https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186*
24. *https://www.altexsoft.com/blog/traffic-prediction/*
25. *B. Hussain, M. K. Afzal, S. Ahmad and A. M. Mostafa, "Intelligent Traffic Flow Prediction Using Optimized GRU Model," in* IEEE Access, *vol. 9, pp. 100736-100746, 2021, doi: 10.1109/ACCESS.2021.3097141.*
26. *Nilayam Kumar Kamila, Jaroslav Frnda, Subhendu Kumar Pani, Rashmi Das, Sardar M.N. Islam, P.K. Bharti, Kamalakanta Muduli,Machine learning model design for high performance cloud computing & load balancing resiliency: An innovative approach, Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 10, Part B,2022,Pages 9991-10009,ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2022.10.001.*
27. *Kamila, N.K. et al. (2023). A Review of Recent Technology Advancements on Smart Cities and its High-Performance Applications. In: Khanna, A., Gupta, D., Kansal, V., Fortino, G., Hassanien, A.E. (eds) Proceedings of Third Doctoral Symposium on Computational Intelligence . Lecture Notes in Networks and Systems, vol 479. Springer, Singapore. https://doi.org/10.1007/978-981-19-3148-2_32*
28. *Kamila, N.K., Pani, S.K., Das, R.P. et al. A near-optimal & load balanced resilient system design for high-performance computing platform. Cluster Comput 26, 1535–1550 (2023). https://doi.org/10.1007/s10586-022-03913-8*