Smart Sensing System for Real-time Automatic Traffic Analysis of Highway Rest Areas

Final Report

METRANS Project 17-11

April 2018

Principal Investigator:

Mohammad Mozumdar, Ph.D.

Department of Electrical Engineering

California State University, Long Beach

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the accuracy of the data and information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation, and the METRANS Transportation Center in the interest of information exchange. The U.S. Government, and California State University, Long Beach assume no liability for the contents or use thereof. The contents do not necessarily reflect the official views or policies of the State of California, CSULB, or the Department of Transportation. This report does not constitute a standard, specification, or regulation.

ABSTRACT

State transportation agency spends millions of dollars annually to maintain and improve the service provided to the drivers in the highway rest areas. In order to collect traffic data in real-time, Researchers can use the vehicle data in the rest areas. Therefore, it is helpful immensely to update the existing safety policies in the rest areas. Transportation agencies don't have any automated systems to perform "automatic" and "real-time" vehicle identification and classification in the highway rest areas. Motivated by a dire need to enhance and modernize the transportation system, we propose an advanced modular system that will integrate a smart sensor to extract a rest area traffic pattern in real-time. Currently, Caltrans collects traffic data from Automated Vehicle Classification (AVC) stations and also manual census collected in the specific locations. However, this technology is too expensive, time-consuming, and disruptive; therefore it has not been used widely in many different locations.

In recent years, There have been many significant improvements in MEMS sensors domain with respect to size, cost and accuracy. Moreover, extreme miniaturization of RF transceivers and low power micro-controllers have motivated researchers to develop small and low power sensors and radio equipped modules. These sensors are gradually replacing traditional wired sensor systems. These modules which are often called "sensor mote" (size of a quarter) communicate with other sensor nodes and build an intelligent network of sensors. Because of the miniaturization and low power consumption, these sensor motes are extremely efficient due to their low power budget. We propose a wireless MEMS sensor based automatic vehicle classification and identification system for highways rest areas.

Our developed Automatic Vehicle Classification and Identification (AVCI) system consists of two parts, AVCI sensor nodes containing magneto-resistive and accelerometer sensors. These sensors calculate speed and axles respectively. The next part, the system proposes a Access Point (AP) which collects data from sensor motes and calculate speed, axles counts and then it classifies the collected data based on Federal Highway Administration (FHWA) 13-categories Scheme-F[5]. The AP includes a RF transceiver to communicate with the sensor motes and also a GPRS (General Packet Radio Service) shield to transmit aggregated traffic data to the county or regional traffic data collection center.

TABLE OF CONTENTS

ILLUSTRATIONS

DISCLOSURE

This research project report was funded by a grant from the U.S Department of Transportation through the METRANS Transportation Center (METRANS) under a cooperative agreement between the University of Southern California (USC) and California State University, Long Beach (CSULB).

ACKNOWLEDGEMENTS

The authors acknowledge support given from U.S Department of Transportation through the METRANS under a cooperative agreement between the University of Southern California (USC) and California State University, Long Beach (CSULB). We would also like to acknowledge all graduate and undergraduate students who worked very hard day-and-night to get the best result possible.

I. SCOPE

This research proposal targets the design of smart sensing system for real-time automatic traffic analysis of highway rest areas. We developed low-power sensing platforms, optimized power-saving algorithms, communications protocols, and machine-learning models to yield a novel and modular multi-nodal sensing systems that will help traffic analysis of highway rest areas efficiently. This could be a part of nationwide efforts of Intelligent Transportation Systems (ITS) for smart connected roads. To the best of our knowledge, Caltrans (or similar entity at nationwide) doesn't have any installed ITS that can perform "automatic" and "real time" vehicle identification and classification for highway rest areas. Our proposed system will reveal high grained traffic data such that user will be able to know for "each" vehicle the time of entry and exit in the rest area and, it's classification (based on axles). The developed smart sensing and data interpretation system will maintain small foot-print, significantly cost-effective (compare to existing available systems), and will be capable of automatic identifying and classifying each vehicle in high way rest area in real-time.

II. INTRODUCTION

Public rest areas located along highways throughout the United States that allow quick access and free 24-hour availability to basic amenities, such as parking and restrooms. Other amenities, such as vending machines, pay phones, picnic tables, and travel information, are also often available. Public rest areas are typically designed to serve the needs of a broad range of travelers, including vacation/recreational travelers, commercial vehicle operators, commuters, motorcyclists, bus tours, and others.

Beside passenger vehicles, rest area is also a safe legal truck parking place for the long-haul truck. Long-haul truck drivers are subject to hours-of-service regulations issued by the Federal Motor Carrier Safety Administration. These regulations govern the hours that truck drivers can drive during a consecutive shift. Current regulations mandate that truck drivers carrying freight are permitted to drive up to 11 hours following 10 off-duty [6]. While the rationale for mandating truck driver rest is well motivated and understood, the practicalities of truck drivers finding safe and legal locations to rest is a separate challenge. Along highways, truck stops, and public rest areas comprise the majority of spaces publicly available for a truck driver to use. These locations are limited in number and in size, and they are also spaced along the highway at distant intervals. If a truck driver, approaching the hours-of-service driving limit encounters a cluster of truck rest areas that are full, he or she is generally faced with the choice of driving some additional highway distance to the next available rest area or park illegally. Our designed system will be able to provide real time traffic classification for highway rest areas, hence transportation agency could publish this parking availability information in the road side variable electronic message board for truck (as well as passenger vehicle) drivers.

The California SRRA (Safety Roadside Rest Area) system is an attractive and safe place where travelers restore their energy and driving alertness. Traveler convenience and comfort are the top priority of Caltrans SRRA system, but it serves more expansive goals that benefit the community and economy. The state of California has 87-unit SRRA system currently includes 4,378 parking spaces and was constructed between 1958 and 1984 [7]. More than 100 million visitors use it annually, with usage exceeding one million at each of 47 SRRA units, 2 million at each of seven SRRA units, and more than 3 million at each of two SRRA units. The two units that comprise the Aliso Creek SRRA in northern San Diego County together receive 6.4 million visitors each year. Current maintenance and operations expenditures for the SRRA system are approximately $0.13 per user visit, or $13

million annually [7]. Our developed AVCI system will help transportation agency to collect detailed traffic data from high way rest areas, and hence it will facilitate the policy maker to take better decisions to improve the uses of highway rest areas. Only in the state of California, our developed system could make impact highway rest areas uses of 100 million visitors yearly.

III. METHODOLOGY

Currently available traffic sensor systems such as: inductive loop, video, sonar, radar, magnetic, capacitive, PVDF wire, and pneumatic treadle, are costly and use electrical power from the power distribution network. Automatic Vehicle Classification (AVC) systems currently installed by Caltrans and others can cost nearly hundred thousand of dollars and require direct power supply from utility poles [8]. Regardless, in-pavement sensors are still popular, due to their accuracy, ability to provide direct information with very little ambiguity, ability to monitor road conditions (i.e. presence of ice), all while not requiring a human operator. Our developed approach will require the installation of a **"quarter" sized sensor platform** that includes a wireless transceiver module, sensors, a low power microcontroller, and other minor electronics modules. Furthermore, the sensors will communicate with Access Point (AP) wirelessly, therefore the need to cut into the pavement, as required in inductive loop approach, will no longer be necessary. The power savings in our developed method would be achieved at two levels, at the node level and the network level. At network level, energy savings can be achieved through an optimized balance of computation and communication to AP node, improved hardware design, and also by employing a power-aware communication protocol. At the node level, the sensor node can be operated at different power modes such as Low Power Mode (LPM) modes, also known as sleep mode, built into their microprocessor platform. In addition, node level power consumption can be optimized through reducing the active processing time. Such low

power operating modes can be easily controlled by interrupt-driven program flow. To comply with a stringent power budget, more savings can be achieved by adjusting the sampling rate on-the-fly.

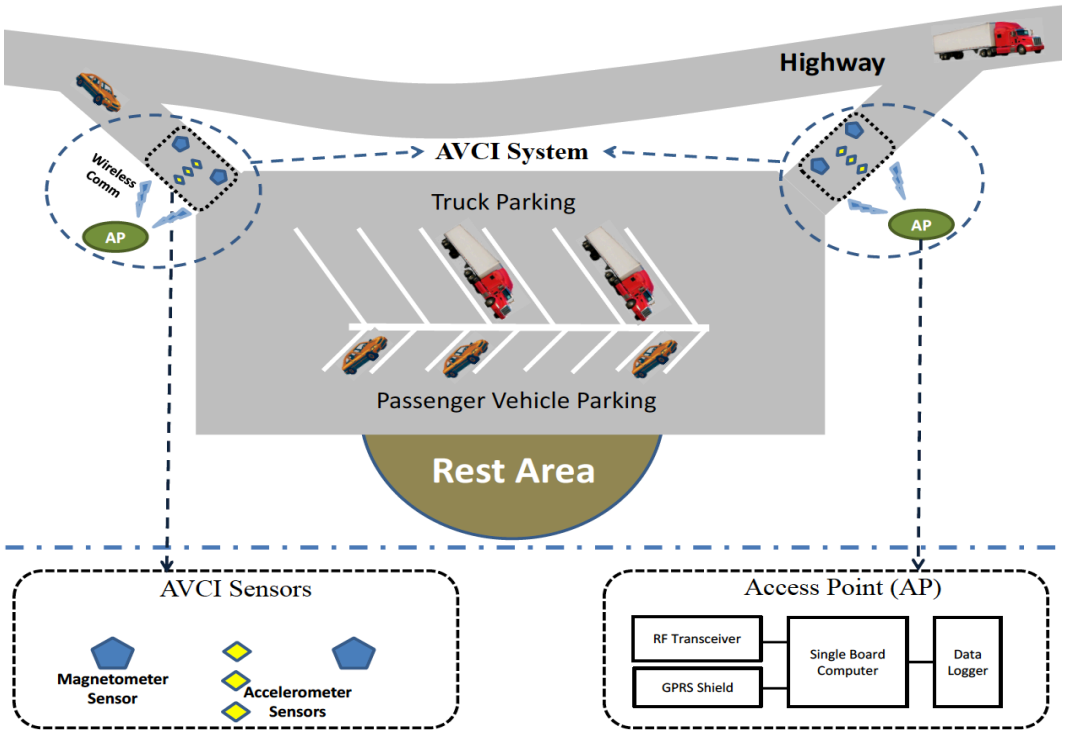

Figure 1: Automatic vehicle classification and identification system for highway rest areas

3.1 System Architecture

Our developed prototype of Automatic Vehicle Classification and Identification (AVCI) system includes two subsystems: AVCI sensors (consist of magnetometer and accelerometer sensors) and AP (algorithm modules for vehicle classification and identification). The AVCI system needs to be installed in entry and exit of the rest area as shown in Figure 1. Magnetometer sensor detects magnetic (ferrous) metals which can identify vehicle presence. In our design, we will use two magnetometer sensors in a fixed distance so that we can accurately calculate vehicle presence and speed. On the other hand, accelerometers detect pavement vibration when a vehicle travels over their detection zones and it will be used for detecting vehicle axles.

Magnetometers sensor data changes in the magnetic field caused by a vehicle and it transmits the vehicle's magnetic signatures to the AP (shown in Figure 2). The signatures are processed by the *Vehicle Speed and Clustering Module (VSCM)* running on the AP to calculate the speed and vehicle grouping. The accelerometers locate the axle peaks based on vibration data and send the peak location time to the AP. An application called *Axle Count Detection Module(ACDM)* performs peak clustering to group axle peaks from the same vehicle. Another application, called *Vehicle Identification and Classification Module (VICM)*, combines output from VSCM and ACDM (vehicle length, number of axles and axle spacing between each axle pair) to perform classification and identification. Using a predefined vehicle classification scheme, the class of the vehicle will then be determined and logged. The most common classification scheme, the FHWA 13-categories Scheme F, will be used for this study [5]. The magnetometer sensors, accelerometer sensors, and AP will be time synchronized to within couple of microseconds. Consequently, even if sensors report their measurements asynchronously, the AP will align magnetometer readings with corresponding accelerometer readings.

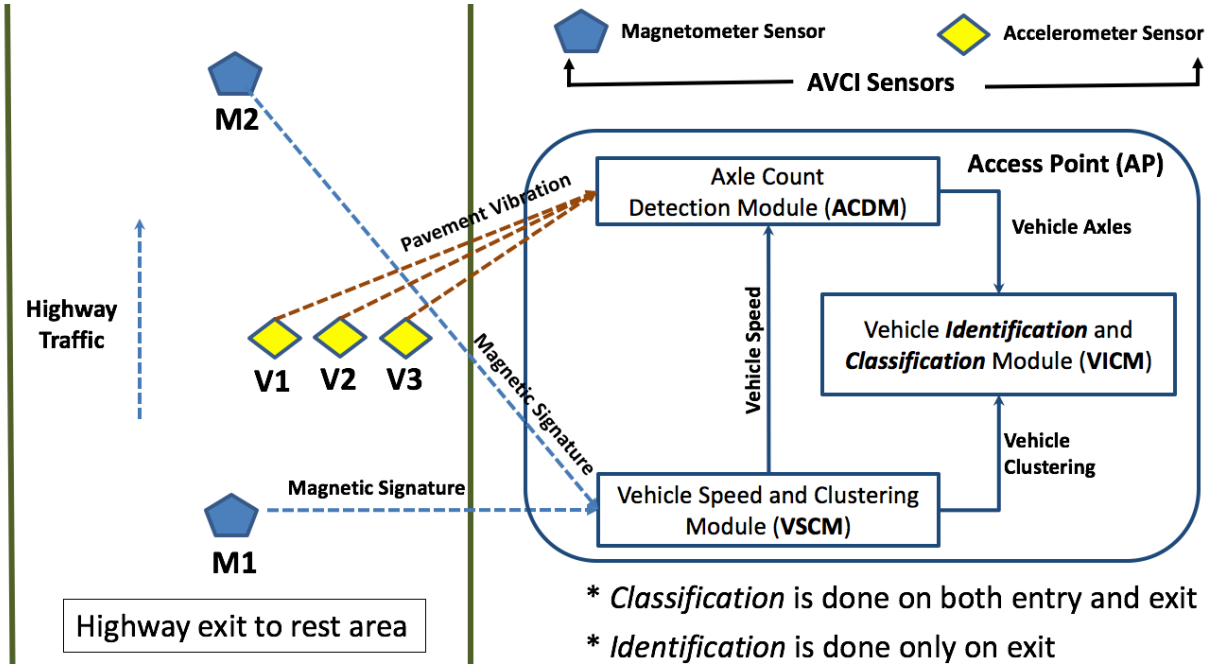

Figure 2: Hardware and firmware components of AVCI

IV. HARDWARE DEVELOPMENT

In order to produce the desired performance and provide for a long endurance system, new hardware was designed to take account for the low power requirements of the project scope. While the new hardware uses the same processor and radio interface from the previous version, the CC430F5137, but switch out the old Honeywell magnetometer for a LIS3MDL magnetometer from STMicroelectronics. This new magnetometer has several benefits over the older sensor: uses SPI as opposed to I^2C for lower power communication, uses less power during sampling, has a wider set of options for full scale range detection, and can generate interrupts when the magnetic field magnitude exceeds a programmable threshold, thereby allowing for the system to only sample from the device when a vehicle is nearby. Readings are updated by the magnetometer's on-board controller in bulk and a data ready (#DRDY) signal is sent to the sensor node controller to indicate new data is available to be read. Magnetometer data is read over a 3-wire SPI bus.

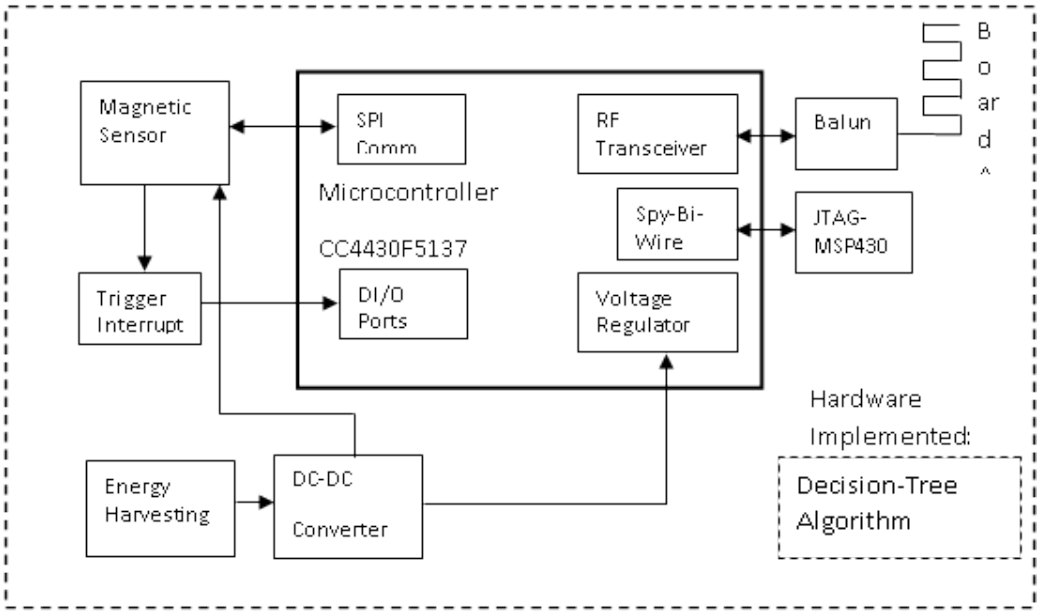

Figure 2: A simplified Wireless Sensor Node block diagram of the hardware architecture

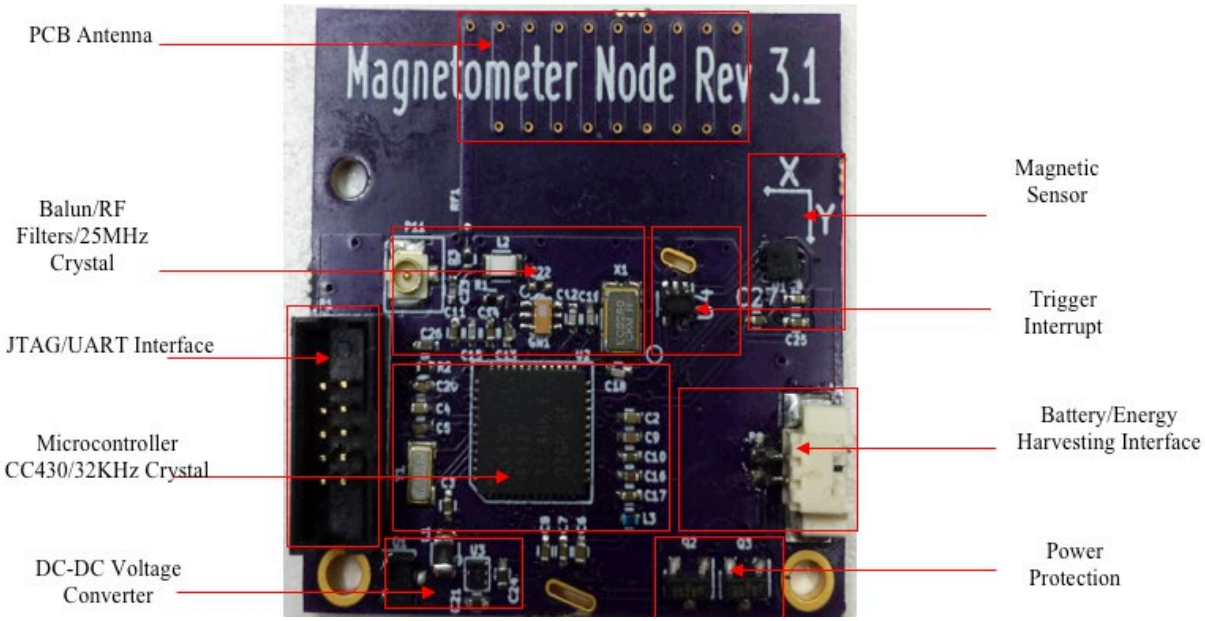

Figure 3: Wireless Sensor node desinged for vehicle detection

The new design also switched out the chip antenna for a meandering PCB antenna as a feasibility approach to implement future cost reduction options, should it become necessary for a production version. This design still utilizes the 915MHz ISM frequency band.

The sensor node uses a 10-pin, 1.27mm pitch header for programming and interfacing to a power analysis module, referred to as the Node Analyzer (NA), shown in Figure 5. The NA is a custom designed power logging tool which allows for sensor power profiling. The NA contains a microcontroller that will be able to communication through a UART to the sensor node and put the sensor node into the commanded state so the power utilization might be read. The NA contains power monitoring circuitry as well as a digitally controlled, variable voltage regulator. The power sourced from the NA is fed into the sensor node during operation while the power usage of the device is closely monitored via the NA.

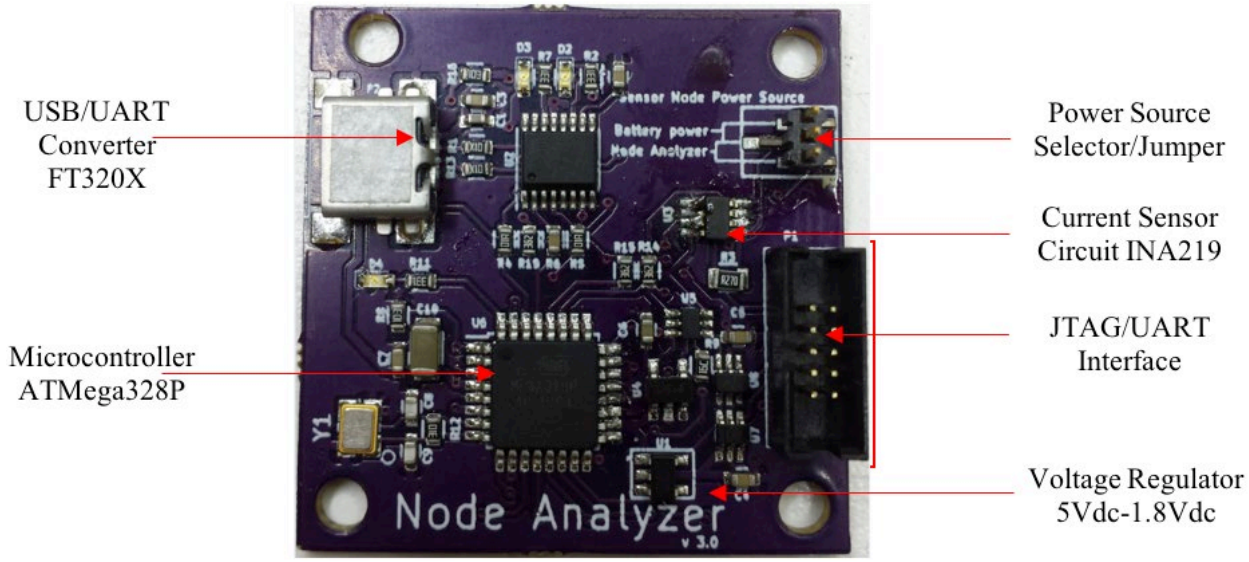

Figure 4: Node Analyzer (NA)

To program the Sensor Node, MSP-FET, or MSP-FET430UIF is required as well as a programming adapter board designed to convert the programmer's 14-pin, 2.54mm pitch header to the 10-pin,

1.27mm pitch header on the sensor node. Sensor nodes are programmed and debugged over the Spy-Bi-Wire interface. The sensor node boards utilize the 10-pin header to reduce the overall sensor node size. The programming setup requires the use of an adapter board to convert the 14-pin, 2.54mm programmer header to the 10-pin, 1.27mm header used on the sensor node (shown in Figure 6).

A. SENSOR NODE HARDWARE DESIGN

As described in the hardware architecture above, the block diagrams are shown here in a schematic format to provide detailed information of each components and circuitry used in the wireless sensor hardware. As shown in Figure 7, this schematic capture and PCB layout is done using the open source KiCad EDA software.

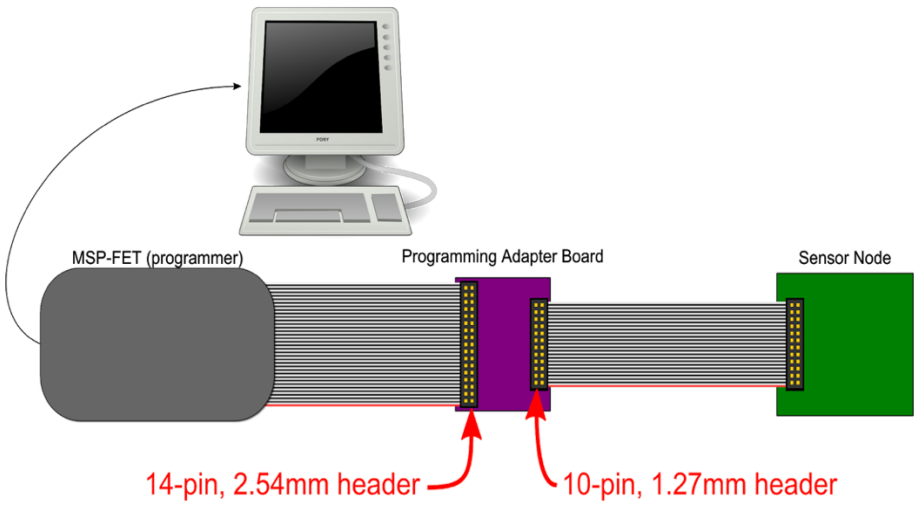

Figure 5: Sensor node programming setup

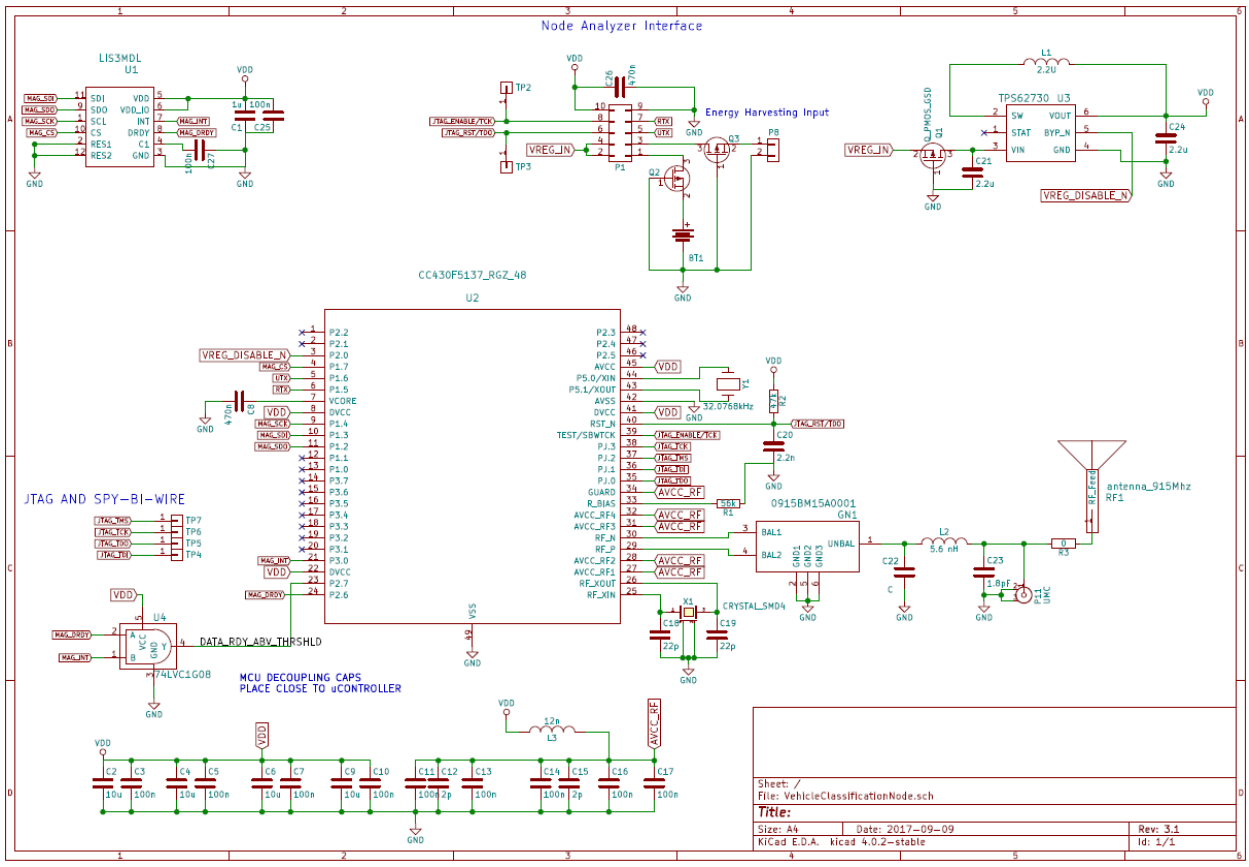

Figure 6: PCB layout of the Wireless Sensor Node

B. HARDWARE COMPONENTS

The Wireless Sensor Node that we designed and tested for the vehicle classification algorithms is shown in Figure 4 and 7. The following major components on the boards and its features are elaborated further in this section.

Microcontroller – The CC430F5137 is SoC from Texas Instruments designed for low-power wireless communication applications with its built-in RF transceiver. It consists of two Universal Serial Communication Interfaces (USCI), where USCI-B0 is used for the SPI to interface to the magnetometer. In the schematic the SPI lines where mark as MAG_SDI, MAG_SDO, MAG_SCK, and MAG_CS, which stands for the magnetic sensor to the microcontroller with the corresponding description: Slave Data Input (SDI), Slave Data Output (SDO), Slave Clock (SCK), and the Chip

Select (CS), respectively. These lines serve the data and configuration commands between the CC430 microcontroller and the magnetometer.

Magnetometer – the LSI3MDL is connected through the SPI lines. The sensor is capable of providing interrupt capability for when the magnetic field intensity changes beyond some determined threshold. The magnetometer is able to select a full-scale range of ±4/±8/±12/±16 gauss.

Trigger Interrupt – 74LVC1G08 is an AND gate logic circuit which allows the processor to be configured to interrupt when both the magnetic field is above the interrupt threshold and there is data ready.

Energy Harvesting Interface – is an input port for an energy harvesting power source beside the Lithium-based, 3.3V 1Ah CR2477 battery coin-cell. In this project, two MOSFETs are used for reverse polarity protection for both input power sources.

DC-DC Converter – TPS62730 – is step-down DC-DC converter optimized for ultra-low power wireless applications. It has an input voltage range from 1.9VDC to 3.9VDC where in our case we convert the 3.6VDC to 2.1VDC. Also, the TPS62730 provides up to 100mA output current and allows the use of tiny and low cost chip inductors and capacitors to further reduce the size of the sensor node.

Balance-Unbalance (Balun) – 0915BM15A0001 is a 915MHz impedance matching circuitry for RF transceiver of the CC430F5137 chip. It is connected in between the microprocessor chip's RF front and the inductor-capacitor (LC) bandpass filter of the antenna.

PCB-Antenna – the radio antenna is a PCB design, etched to the tracing on the PCB. This approach provided for a medium size antenna that fits perfectly on the 30mm x 30mm PCB and is low cost. This antenna design was based on the Evaluation Module (EM) board recommendation for 868/915/955 MHz PCB spiral-type found in Application Note 058 from Texas Instruments. This approach allows for a reduce component count and serves as a feasibility study as an option for cost reduction.

Spy-Bi-Wire JTAG Protocol - is a serial communication protocol for programming and debugging firmware using Spy-Bi-Wire interface on the programmer. The pins from the microcontroller are then connected to the Molex 10-pin header for connection to the MSP-FET programmer.

C. HARDWARE FOR POWER PROFILING

Another board that we developed is the Node Analyzer (shown in Figure 5 and 8), as discussed in the above section was used in order to determine the power consumption of the Wireless Sensor Node board in the actual implementation. The NA is composed of a "current sensor" that was used for monitoring the power consumption of the attached sensor node. The main idea of implementing low power profiling to the SN board is to be able to make sure the sensors can keep on running for years without replacing the battery. To realize this scenario, we developed this board - Node Analyzer board (NA), which is used to determine the power consumption of the SN board. The schematic of the Node Analyzer board is shown in Figure 8.

D. NODE ANALYZER BOARD

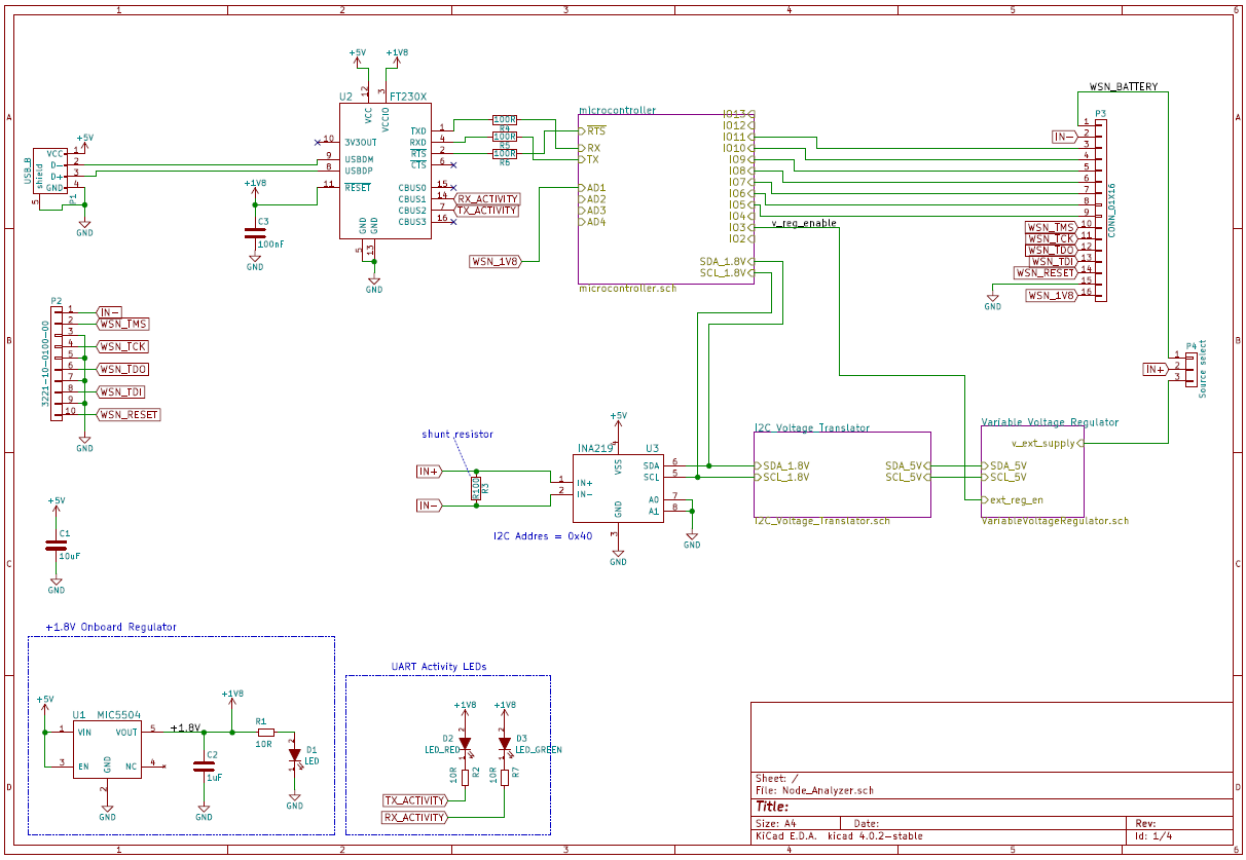

Figure 7: Schematic diagram of the Node Analyzer

The NA board is mainly composed of the following chips:

ATMega328P: Microcontroller that reads the current consumed in the SN board through other component on the NA board which is the **Current Sensor INA219** chip. It is also composed of a voltage regulator that provides the power to the SN board while under test. Also, a **UART-to-USB chip FT320X** is used to interface to a PC that runs the **Node Analyzer Companion Software (NACS)** that provides the control of the power profiling sequences and reads the log data to view the power consumption. The NA board is able to control and analyze the power profiling sequences of the SN board and to be able view the power consumption parameters while under test.

UART communication: The UART connection to the node analyzer will enable for a set of commands. Some specific to the analyzer and some intended to be sent to the sensor nodes. The

profiles sent to the sensor nodes are static and arguments are not needed. Therefore, a single byte will be encoded to determine what operation to take. Commands for the analyzer are for power settings, or resets. The node analyzer will continuously send power information and a timestamp of the operation to the PC. There is a companion application that provides a GUI of this information. This data is the main advantage which will enable better refinement and iterations of device settings or algorithms.

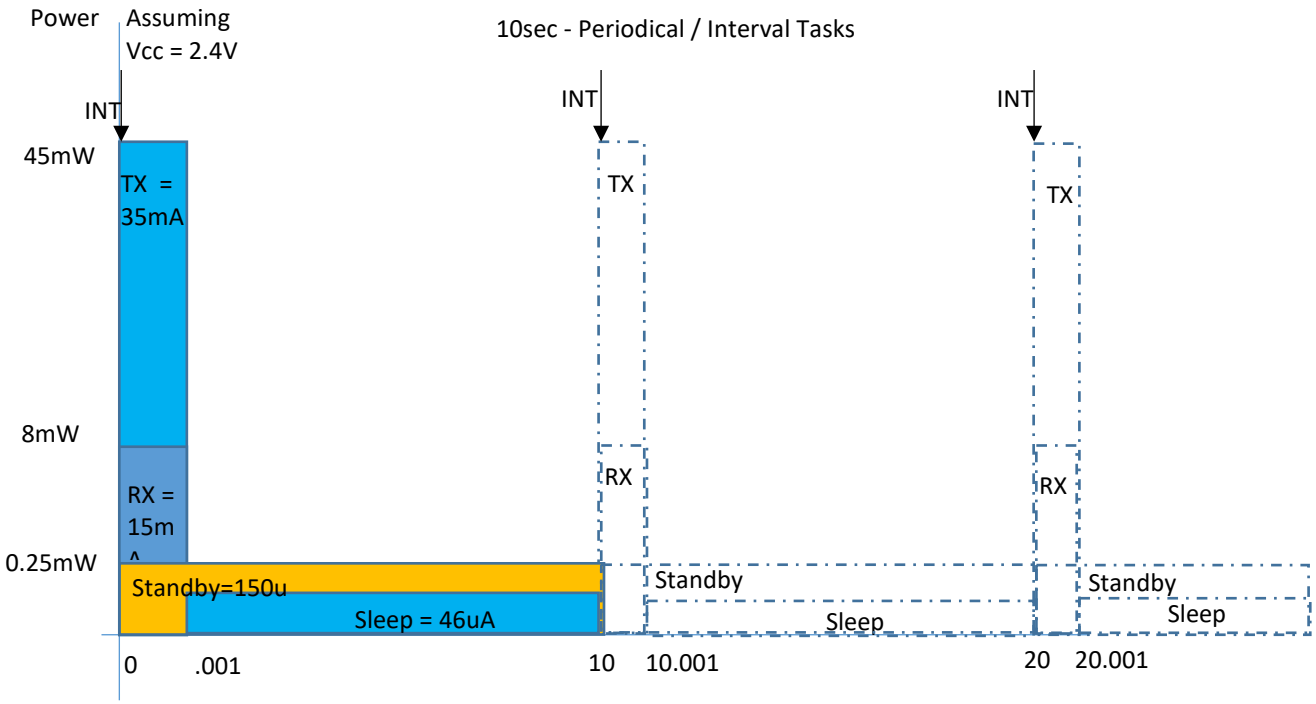

Figure 8: Sample power profiling capture, denoting power consumption levels at various operating modes.

E. POWER PROFILING TASK

We have included here an example of a power profiling graph that could be applied to the Wireless Sensor Board which can be controlled by the Node Analyzer board. This power profile graph shown in Figure 9 indicates that when an interrupt from a vehicle passing over the Sensor Node, the SN starts to process the data, extract the features, perform the Decision-Tree algorithm, and transmit the result for about 0.001ms with a current consumption of 35mA to the receiver. Once the processing

the transmission is done the Wireless Sensor Board goes back to Standby Mode or Sleep Mode, consuming approximately 156uA and 46uA, respectively, until the next interrupt occurs. While no interrupt occurs, the SN board will remain in a low-power mode, thus, consuming less power that would allow the battery to last for years without any replacement.

V. VEHICLE CLASSIFICATION ALGORITHMS

A. PREPROCESSING

The preprocessing block generally performs two operations: calibration and dimensions modification. The calibration operation is not a part of the data flow and is usually done during the initial phases of a sensor's installation and placement underneath the ground. The first task of calibration is to nullify the effects of the magnetic fields generated by the surroundings into which the node is embedded.

This part of calibration is essential, as different locations will have different magnetic fields. Initially, the magnetometer is designed such that its base value (the value of the sensor under no load condition, or under ideal conditions, i.e. when no car is passing over it) always points to zero Gauss; when it is deployed underneath the surface of the Earth, however, the base value is altered due to the local presence and intensity of the Earth's magnetic field. This change in the base value is irreversible and, moreover, the value of the change will vary by location.

Therefore, setting the base value of the sensor to zero or to some constant will produce inconsistent measurements at different locations. Thus, calibration of the base value of a sensor is a significant factor in developing an efficient system because it allows for nullifying the impact of Earth's Magnetic Field over the sensor. Once a base value corresponding to a sensor is obtained, the value is then normalized to zero in order to synchronize all the MLVC systems.

Figure 9 shows the normalization of various base values associated with various sensors placed at different locations of the earth.

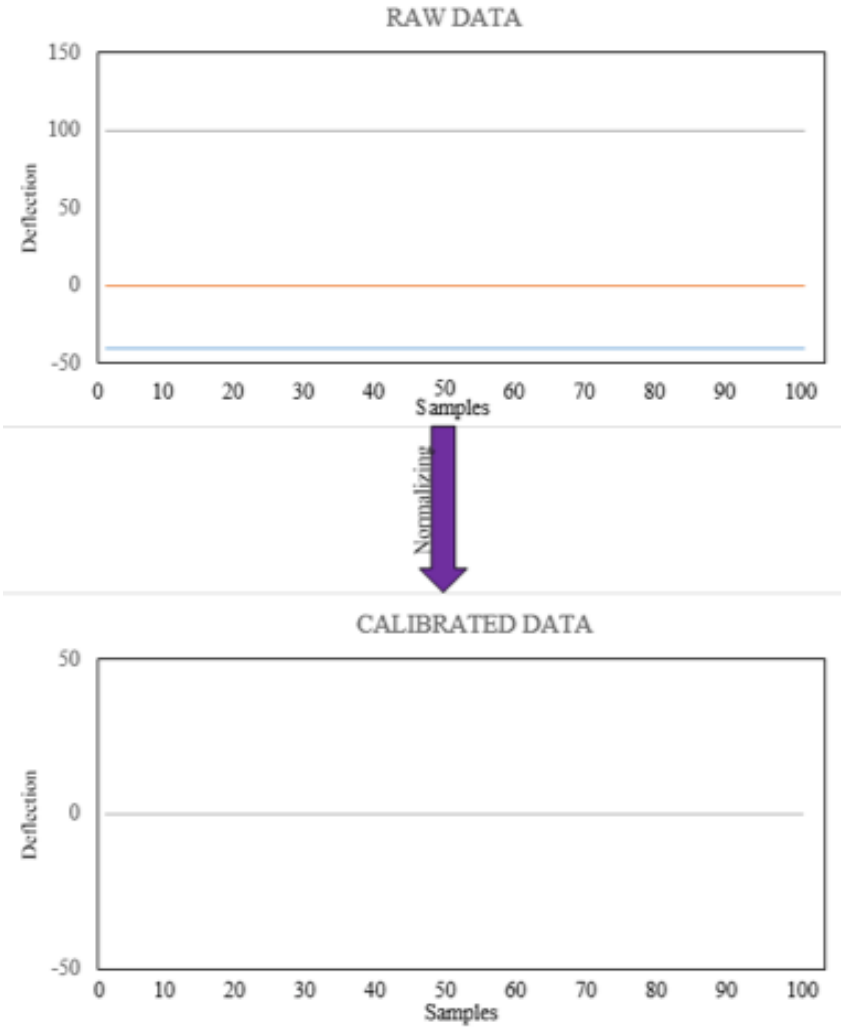

Figure 9: Conversion of raw base values into calibrated base values

After calibrating and normalizing different base values for different sensors placed at various locations, preprocessing further determines a threshold value for each sensor. The threshold level is set so as to avoid false detection of vehicles, which is often caused by pedestrians or by metal objects lying over the ground in the range of a sensor. The logic behind this process is that the metallic

composition of a vehicle is higher than that of a person or other objects; thus, the threshold is set to trigger only for magnetic fields typical of a modern vehicle. The node is programmed in such a way that once a value higher (or lower in case of negative values) than this threshold value is received, it automatically opens a sample window that records all the values (magnetic signature) until the point where the value again becomes nearly equal to this threshold value and remains there for at least 10 samples, as shown in Figure 10. For a given vehicle, an average of around 100 samples are obtained per vehicle axis per run.

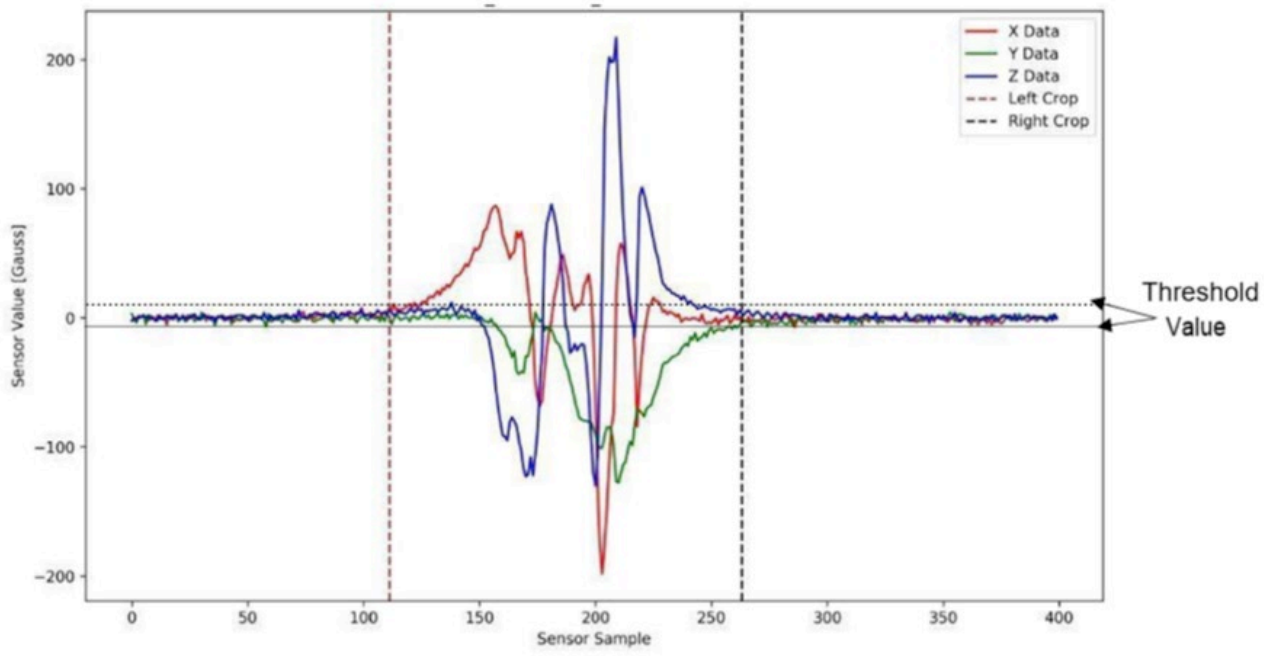

Figure 10: Detection window

After calibration is performed and vehicle measurements are taken, the system performs the next phase of axis or dimensions modification. Axis modification is yet another important part of preprocessing, done to improve the accuracy of the algorithms. The inputs to the preprocessing block are the magnitudes of the values obtained along X, Y, and Z axes, while the outputs are along XY, XYZ, and Z axes, where:

$$XY = \sqrt{(X^2 + Y^2)} \tag{3.1}$$

$$XYZ = \sqrt{(X^2 + Y^2 + Z^2)} \tag{3.2}$$

The above conversion was necessary in order to maximize the sensor's efficiency. The sensor is oriented in such a way that X and Y are at angle of 0 degrees relative to the ground and the Z axis is perpendicular to it, as shown in Figure 7. Normally the sensor is placed such that a vehicle should always approach it along the X-axis, as depicted in Figure 11, which means approaching perpendicular to both the Y- and Z-axis. But at times it could be possible that the sensor can be approached from the Y-axis or from both X and Y. The reasons behind this are that there could be a change in the orientation of the sensor due to human error or it could be possible that the sensor is placed at a road intersection where vehicles are moving at right angles to each other. This could degrade the overall efficiency of the algorithm, as we can then have two magnetic signatures of a single car with values along the X- and Y- axis interchanged because we have considered the X-axis as our default one. To avoid this potentiality, we combined X- and Y-axis values to obtain values that are orientation-independent.

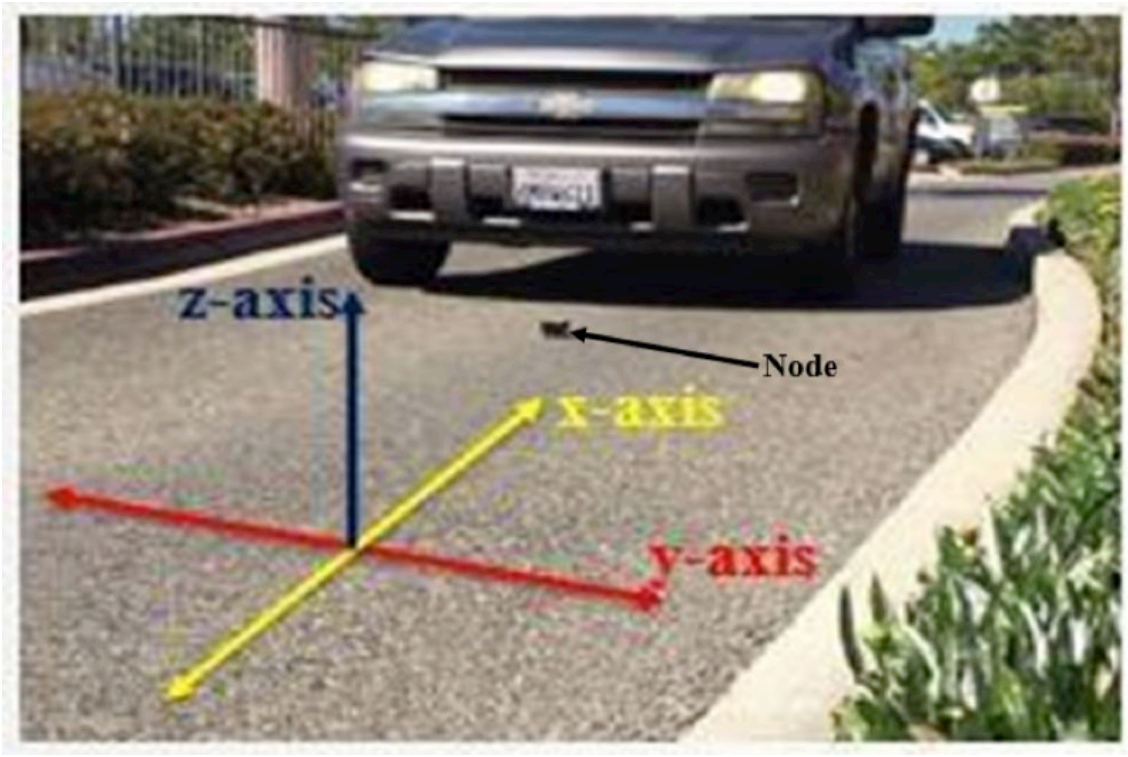

Figure 11: Placement and orientation of the sensor

Next, it is necessary to consider the other equation. Since the Z-axis is perpendicular to the ground, its values are almost unaffected by the orientation of the sensor. Because of this, we send the Z-axis values unaltered at the output of the preprocessing block. Still, a rare possibility for error exists, which can alter the Z-axis values. This could be seen in cases where the sensor is not properly aligned with the ground or in cases where the roads are more elevated on one side than on the other. In such situations, the sensor could be tilted and the Z-axis could make an acute or an obtuse angle with the ground instead of a right angle, thus resulting in a different magnetic signature than what is produced while the sensor is placed properly. Thus, we also provide the XYZ output so as to negate any possibility of error from misalignment. In all, then, the XY, XYZ, and Z values are fed to the feature extraction process to obtain features from the modified magnetic signature, which is discussed next.

B. FEATURE EXTRACTION

The features are extracted from each one of the 30 runs of a vehicle and for each run a total of 32 features are extracted from the values along the XY, XYZ, and Z axes. Specifically, 11 of these features correspond to each of the XYZ-and the Z-axis, and 10 correspond to the XY- axis, from a single run of a vehicle. Table 1 illustrates each one of them.

TABLE 1. Different Types of Features

Features	XY axis	XYZ axis	Z axis
Maximum	xymax	xyzmax	zmax
Minimum	xymin	xyzmin	zmin
Mean	xymean	xyzmean	Zmean
Standard Deviation	xystdev	xyzstdev	zstdev
Variance	xyvar	xyzvar	zvar
Ratio	_	xyzratio	zratio
RMS Value	xyrms	xyzrms	Zrms
P2P Value	xyp2p	xyzp2p	zp2p
P2P-RMS Value	xyp2prms	xyzp2prms	zp2prms
Kurtosis	xykurtosis	xyzkurtosis	zkurtosis
Skewness	xyskewness	xyzskewness	Zskewness

A detailed description of all the features is as follows:

1) Maximum

The Maximum feature of XY, XYZ, and Z axis determines the maximum value along each dimension. The Max feature for each axis is represented as follows:

$$xymax = \max(xy_{1,}xy_{2,}----,xy_{100,}) \tag{3.3}$$

$$xyzmax = \max(xyz_{1,}xyz_{2,}----,xyz_{100,}) \tag{3.4}$$

$$zmax = \max(z_{1,}z_{2,}----,z_{100,}) \tag{3.5}$$

where 100 corresponds to the total number of samples.

2) Minimum

The Minimum feature of XY, XYZ, and Z axis determines the minimum value along each dimension. The Min feature for each axis is represented as follows:

$$xymean = \frac{\sum_{i=1}^{100}(xy_i)}{100} \tag{3.9}$$

$$xyzmean = \frac{\sum_{i=1}^{100}(xyz_i)}{100} \tag{3.10}$$

$$zmean = \frac{\sum_{i=1}^{100}(z_i)}{100} \tag{3.11}$$

where, i stands for the i^{th} sample and 100 signifies the total number of samples.

$$xystdev = \sqrt{\frac{\sum_{i=1}^{100}(xy-xymean)^2}{100}} \tag{3.12}$$

$$xyzstdev = \sqrt{\frac{\sum_{i=1}^{100}(xyz_i-xyzmean)^2}{100}} \tag{3.13}$$

$$zstdev = \sqrt{\frac{\sum_{i=1}^{100}(z-zmean)^2}{100}} \quad (3.14)$$

where, i stands for the i^{th} sample and 100 signifies the total number of samples.

3) Variance

Var stands for variance, which is equal to the square of standard deviation. It describes the level of variation from the mean value. It is represented as follows:

$$xyvar = \frac{\sum_{i=1}^{100}(xy_i-xymean)^2}{100} \quad (3.15)$$

$$xyzvar = \frac{\sum_{i=1}^{100}(xyz_i-xyzmean)^2}{100} \quad (3.16)$$

$$zvar = \frac{\sum_{i=1}^{100}(z_i-zmean)^2}{100} \quad (3.17)$$

4) Ratio

Ratio is one of the features most important to classification. There could be a case in which we have multiple different readings along the Z and XYZ axis for the same vehicle on

different days, depending on the level of air inflated inside the tire of that vehicle, with a flat tire being the worst case scenario; such variations could be minor or potentially significant. We can analyze that our Max and Min features would fail in such a case. The Ratio feature works as a remedy to this problem. Ratio is expressed as the ratio of the maximum value to the minimum value. Thus, whatever the case, the ratio will remain constant throughout for a particular vehicle as the minimum and maximum values will increase or decrease by the same factor. Hence, this feature is a boon in vehicle classification along the Z and XYZ axis (the above problem will not affect XY axis). The feature is expressed as follows:

$$xyzratio = (\frac{xyzmax}{xyzmin}) \quad (3.18)$$

$$zratio = \left(\frac{zmax}{zmin}\right) \quad (3.19)$$

5)RMS

RMS stands for root mean square and the feature is represented as follows:

$$xyrms = \sqrt{\frac{\sum_{i=1}^{100} xy}{100}} \quad (3.20)$$

$$xyrms = \sqrt{\frac{\sum_{i=1}^{100} xyz}{100}} \quad (3.21)$$

$$zrms = \sqrt{\frac{\sum_{i=1}^{100} z}{100}} \quad (3.22)$$

where, i stands for the i^{th} sample and 100 signifies the total number of samples.

6)P2P

P2P stands for peak to peak. It represents the range of a graph by calculating the difference between the highest peak and the lowest peak. The P2P feature along each axis is represented as follows:

$$xyp2p = xymax - xymin \quad (3.23)$$

$$xyzp2p = xyzmax - xyzmin \quad (3.24)$$

$$zp2p = zmax - zmin \quad (3.25)$$

7)P2PRMS

Peak to Peak RMS is defined as the ratio of the maximum amplitude among all the peak values to the RMS value. It is represented as follows:

$$xyp2prms = \frac{(\max(xymax,\ abs(xymin)))}{xyrms} \tag{3.26}$$

$$xyzp2prms = \frac{(\max(xyzmax,\ abs(xyzmin)))}{xyzrms} \tag{3.27}$$

$$zp2prms = \frac{(\max(zmax,\ abs(zmin)))}{zrms} \tag{3.28}$$

8) Kurtosis

Kurtosis is the measure of sharpness of the curve at the mean. It is important in that it provides us with the level of density of the values around the mean for each axis in a magnetic signature. Kurtosis is an important and useful feature for the classification of vehicles. Kurtosis

is represented as follows:

$$xykurtosis = \frac{\sum_{i=1}^{100}(xy_i - xymean)^4/100}{(xystdev)^4} \tag{3.29}$$

$$xyzkurtosis = \frac{\sum_{i=1}^{100}(xyz_i - xyzmean)^4/100}{(xyzstdev)^4} \tag{3.30}$$

$$zkurtosis = \frac{\sum_{i=1}^{100}(z_i - zmean)^4/100}{(zstdev)^4} \tag{3.31}$$

where, i stands for i^{th} sample and 100 corresponds to the total number of samples.

9) Skewness

The skewness is the measure of the asymmetry of a curve around the center. It helps us to analyze the measure of left leaning or right leaning of the curve from the center. This feature, together with kurtosis, serves as an excellent solution for classifying various types of vehicles.

The skewness is represented as follows:

$$xyskewnwss = \frac{\sum_{i=1}^{100}(xy_i - xymean)^3/100}{(xystdev)^3} \quad (3.32)$$

$$xyzskewnwss = \frac{\sum_{i=1}^{100}(xyz_i - xyzmean)^3/100}{(xyzstdev)^3} \quad (3.33)$$

$$zskewnwss = \frac{\sum_{i=1}^{100}(z_i - zmean)^3/100}{(zstdev)^3} \quad (3.34)$$

where, i stands for the i^{th} sample and 100 signifies the total number of samples.

Approximately 32 features are extracted from a single run of a vehicle, and 30 runs are performed per vehicle. Through this process, approximately 960 features were obtained for each of our 6 tested cars; on a whole, then, 5,760 features were obtained. All of the features associated with a given run are concatenated together to form a list of the run. Several such lists belonging to all the vehicles under test are then concatenated to form a dataset. The dataset also includes a one-to-one mapping between each run and the class to which it belongs. Therefore, the information about the class of vehicle is also preserved in the dataset. This large dataset is then shuffled by applying a randomized function to it. Even after the randomization is performed, one-to-one mapping remains preserved; this is to say that no harm is done to the dataset while randomizing it. Next, the dataset is divided into two different sub-sets, one of which contains 66% of the data for training purposes, the other contains the remaining data for testing purposes.

C. MACHINE LEARNING ALGORITHMS

The concept of machine learning is analogous to that of learning in the human mind. Many of us have heard the saying, “practice makes perfect,” but perhaps only a few have thought about the logic behind it. By performing the same task again and again, one trains one’s mind, each time, to perform the same task more efficiently in the future. The same is the case with a machine learning

algorithm. In order to obtain good results from a system or an algorithm, we have to first train it to accomplish its task; that is what machine learning is all about.

In this section, we will discuss two machine learning algorithms: CART and multi-layer perceptron. CART can be used alone for classification, while Multi-Layer Perceptron (MLP) is used alongside CART; this is because the multi-layer perceptron is incapable of performing feature reduction, which CART can easily do. We will now describe the capabilities of each algorithm, beginning with CART.

1) CART

CART stands for Classification And Regression Tree. Generally, two different types of decision trees are formed in CART: the first is for classification and is known as CART classifier; the second is for regression and is known as CART regression. In this thesis, we only employ the use of CART classifier, which, as shown in Figure 13, is a decision tree starting from the root node on the top and growing towards the final outcome–i.e. leaves. A decision tree is generally a flowchart (an upside-down tree) consisting of several conditional checks (if-else statements), represented as nodes at each level of the tree. Each node gives rise to two child nodes, as shown in Figure 12, and thus divides the whole dataset into several subsets until a pure subset is obtained. This pure subset leads towards the final product (leaves), which contain class-related information.

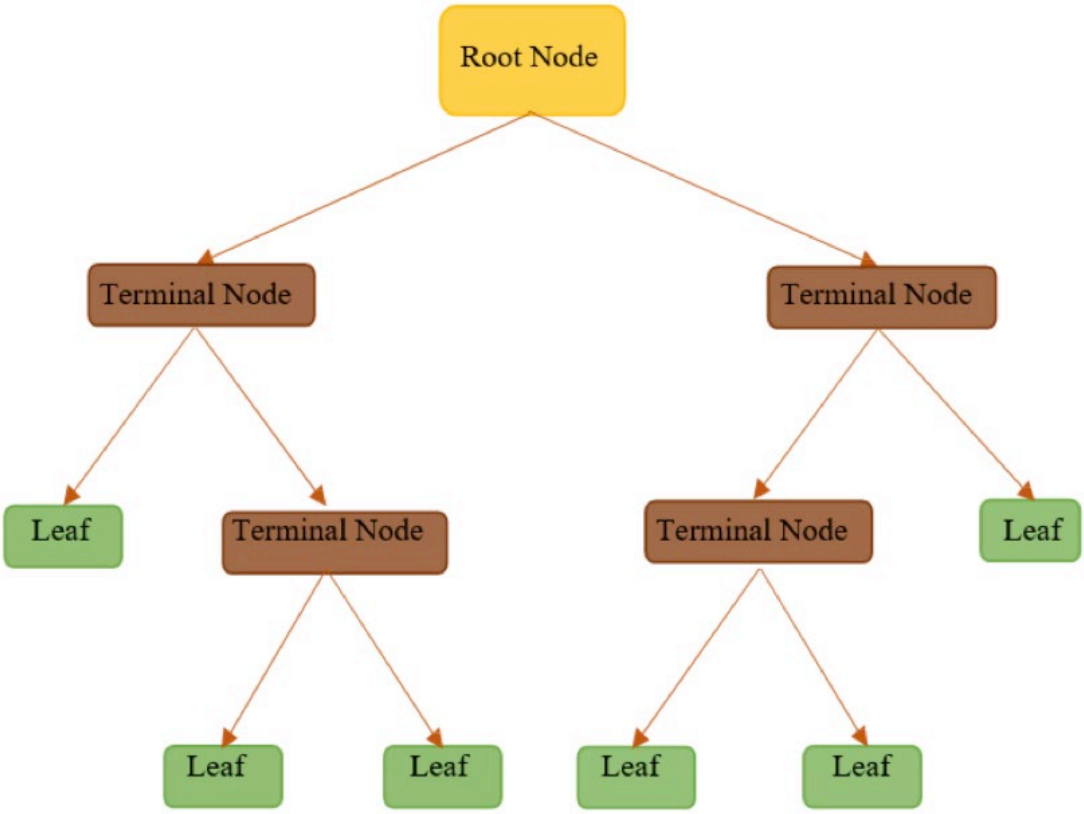

Figure 12: CART classifier

As we already know, we have a data set for training that has a one-to-one mapping between the runs and their associated vehicle's class. Next, we further divide the training set into two different sub-sets: one (x) containing all the runs and the other (y) containing the class related information while preserving the mapping. Once the subset formation is done, we use these two sub-sets as inputs to generate a decision tree by using a tree constructor function, which is expressed as follows:

$$Dectree.\ fit(x, y)$$

This function acts upon both sets simultaneously and extracts all the features from all the runs to construct a tree, while the class subset is used to form leaves. One of the advantages of

the CART classifier, and the reason we are using it in this thesis, is that it does not include all the features to create a decision tree. Rather, it performs feature reduction and chooses only those features that vary by a significant amount among all the vehicles under test, while trashing the rest. This feature of the CART classifier makes it highly efficient and accurate. The formation and operations of the decision tree can be better understood by referring to Figure 13.

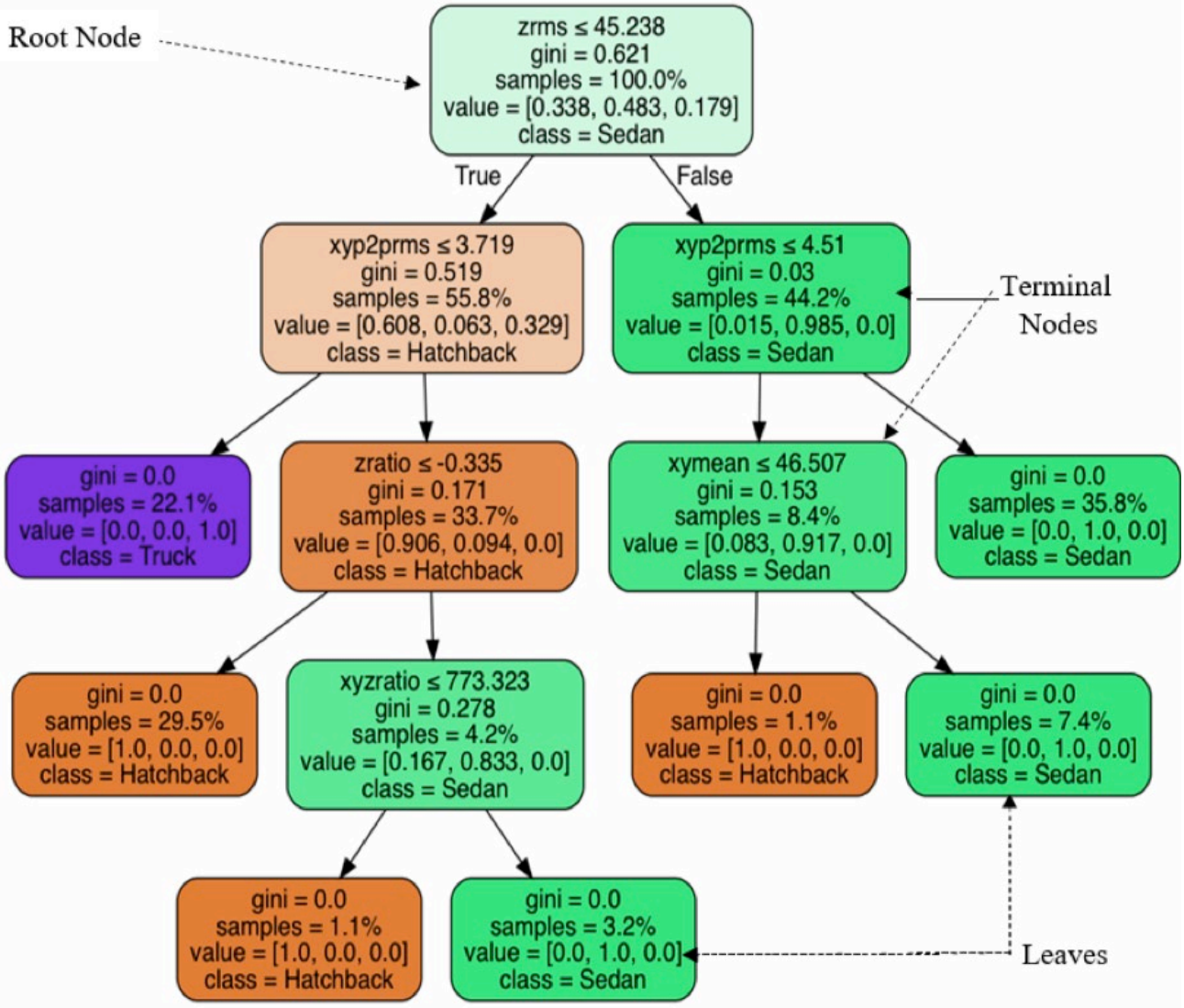

Figure 13: Formation of decision tree

As the figure shows, the tree starts with a root node. This node comprises a conditional check on a feature (p2prms), which varies the most out of the lot and, therefore, has the caliber to successfully differentiate one or more classes of vehicles from the rest. The root node gives rise to two child (terminal) nodes that consist of the same, or two different, features (one for each node) that vary less than did the one in the parent node, but more than in the remaining others. This process of tree-flourishing and node-construction goes on and on, until we reach the leaves or the final products of classification (vehicle's class) which are Sedan, Hatchback, and Truck. The level of impurity in the tree (represented by a factor called gini) decreases as we propagate from the root node out towards the leaves, with gini being the highest at the root node and approximately zero at the leaves. Hence, it can be analyzed that there is always a single root node in one decision tree that divides the whole dataset into two subsets. Following the root node, there are multiple levels of terminal nodes, with each level consisting of nodes made from features that vary more than the ones that are at a lower level and, hence, contributing to the formation of a tree.

Once an efficient tree is created, we test its efficiency by feeding the testing dataset into the algorithm. Like the training data set, the testing data set is also divided into two subsets, but only the sub-set associated with runs is applied at the input of the algorithm, while the class- related one is used for verification. While testing, the run sub-set is applied sequentially at the input of the already-trained tree–i.e. only one run at a time is applied to the input, which is then acted upon by the algorithm in order to determine its class. The next run is only applied when the class for the previous one is determined. Therefore, sequentially over time we obtain a subset of predicted classes at the output, which is then compared with the subset of the already known classes to obtain the score and, hence, the efficiency of the algorithm.

Decision trees are effective in classification but are often affected by overfitting and pruning. These factors can degrade performance. Overfitting occurs when a tree is fed with a larger amount of data than what is actually required for construction; when this phenomenon occurs, noise and unnecessary features are also used to construct the tree, which degrades its performance. The one and only remedy to overcome overfitting is pruning, which means reducing the dataset as per the tree's requirements. Pruning is generally done manually and could result in loss of ability to detect certain relevant features. Therefore, a decision tree can only support a specific amount of data, and anything larger could reduce its efficiency. Thus, the above theory tells us that CART Classifier's usage should be limited to small- and moderately- sized datasets only. Therefore, we use a better and a different type of algorithm to deal with larger datasets. This algorithm is discussed next.

2) MLP

Where classification of vehicles is concerned, one of the most efficient and widely-used algorithms is MLP. The concept behind the formation of a MLP is derived from the function of the human brain. The brain is made up of billions of neurons, each working as an individual processing unit and contributing in decision-making processes for the entire brain–i.e. to take certain actions based on the inputs received from the senses. As shown in Figure 15, a multilayer perceptron operates similarly. It consists of several interconnected perceptrons as its basic building blocks, each performing only a small portion of the overall processing involved in a major task and, hence, contributing moderately towards the final output. In addition to its structure and basis in neural networks, there is yet another important feature of multi-layer perceptron, known as Error Backward Propagation (EBP). EBP has empowered multi-layer perceptron to outperform several similar algorithms and is the reason for its use in this thesis.

The concept of EBP will be discussed later in this section.

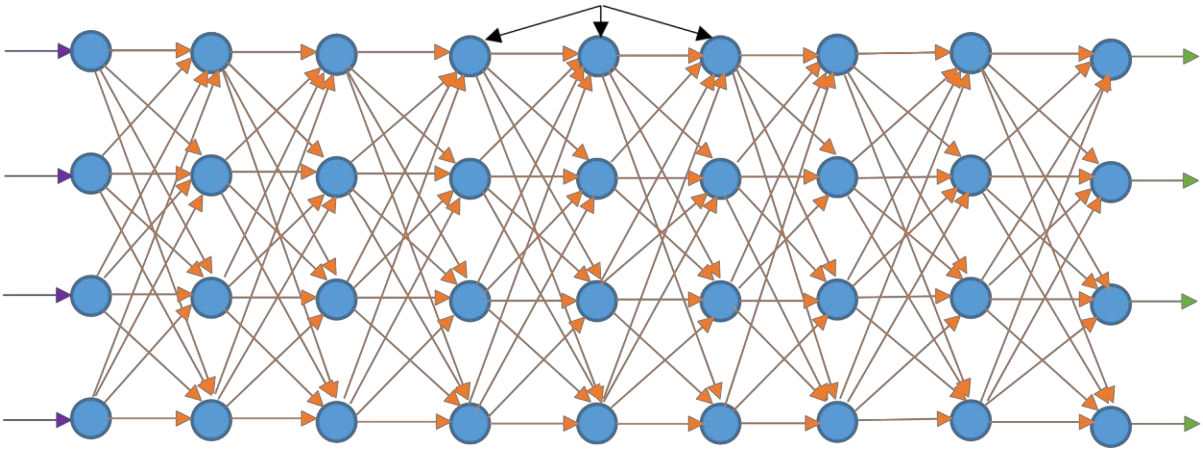

Figure 14: Overview of the structure of an MLP.

As we have already discussed, a multi-layer perceptron is formed by linking several individual perceptrons. We will now discuss individual perceptrons before returning to further explore multi-layer perceptrons. A perceptron, sometimes also called an artificial neuron, shown in Figure 15, is a small processing unit that is efficient enough to independently classify objects according to the inputs received.

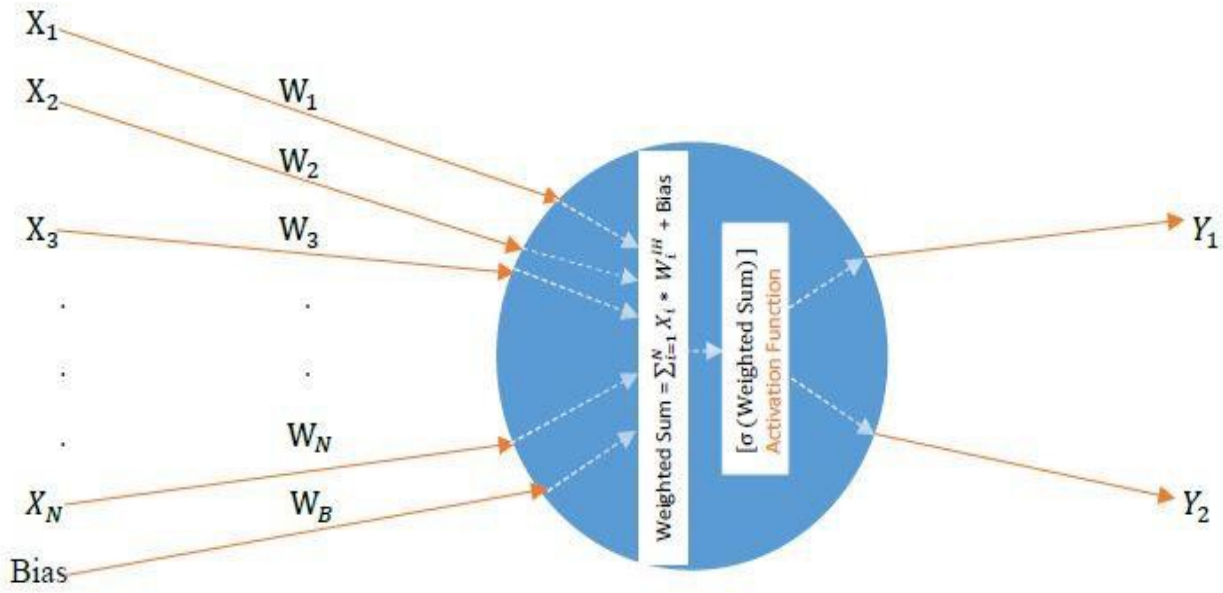

Figure 15: Diagram of a perceptron.

In order to perform classification, a perceptron initially takes in multiple inputs, then processes them with a couple of inherent functions and finally delivers an output (i.e., performs

classification) based on the values obtained from these functions. Each input is fed (or connected) to the perceptron through a single weighted connection (or a link). The name "weighted connection" is derived from the fact that each connection has a weight (or link cost) associated with it.

After obtaining inputs, the perceptron employs its first inherent function into application and calculates a weighted sum of all the inputs received. While calculating the weighted sum, a bias input is also included, whose value is always equal to 1. The bias is used because the weighted input connections fed into the perceptron do not cover all the possible values of inputs; that is, the weights are of no use if all the inputs are zero. Thus, in order to accommodate this possibility, a bias is used. The whole concept revolves around the equation of a straight line, which is:

$$Y = m \times X + b \quad (4.1)$$

Or in our case:

$$\mathrm{Y} = \sum_{i=1}^{N} (X_i \times m_i) + \mathrm{b} \quad (4.2)$$

The weighted sum is expressed as follows:

$$W_X = \sum_{i=1}^{N} (X_i \times W_i) + \mathrm{Bias} \times W_B \quad (4.3)$$

where, W_X is the weighted sum of all the inputs (X), X_i signifies the i^{th} input, W_i represents the cost or weight associated with the connection between the i^{th} input and the perceptron, Bias represents the bias input and W_B represents the cost associated with the connection of Bias.

If we compare the above two equations, it can be seen that W_i signifies m_i and (Bias ×

B_1) signifies b. Hence, it can be seen from the equation that, if all the values of X are zero, there is no significance of any m in the equation, so a bias is needed to stabilize the equation and, moreover, to stabilize the algorithm as a whole.

Once a weighted sum is obtained, the perceptron performs its second inherent function. This function, called the activation function or decision-making function, is applied over the recently-derived weighted sum to obtain a value. The activation function generally used is a sigmoid function. A sigmoid function is used because of its ability to scale down (normalize) any given value to a value within a range of 0 to 1, as depicted in Figure 12. Scaling down of values always enhances the efficiency of an algorithm; hence, a sigmoid function seems to be an optimal choice for the activation function. The sigmoid function is expressed as follows:

$$S(W_X) = \frac{1}{1+e^{-W_X}} \quad (4.4)$$

VI. CONCLUSIONS

This project's findings provide an excellent reason to consider the in-node MLVC system for identifying and classifying traffic in highway rest area. Our developed Automatic Vehicle Classification and Identification (AVCI) system could be an excellent solution for collecting data from highway rest areas to improve its uses. AVCI sensor nodes which contain magneto-resistive and accelerometer sensors for calculating speed and axles respectively. The Access Point (AP) which collects filtered sensor data from sensor motes to calculate speed, axles count also classify them based on Federal Highway Administration (FHWA)\. The AP contains RF transceiver to communicate with sensor motes and a GPRS (General Packet Radio Service) shield to send aggregated traffic data to the county or regional traffic data collection center.

.

VII. REFERENCES

[1] European Conference of Ministers of Transport (ECMT), *Managing Urban Traffic Congestion*. Paris, OECD Publishing, 2007.

[2] T. Matsuo, Y. Kaneko, and M. Matano, "Introduction of intelligent vehicle detection sensors," in *Proc. 1999 IEEE/IEEJ/JSAI International Conference on International Transportation Systems,* Tokyo, Japan, 1999, pp. 709-713.

[3] L. Lirong, Z. Yunmei, L. Shanshan, C. Ye, L. Tianqi, and X. Xinyuan, "Design of intelligent infrared vehicle detect system based on ZigBee," in *Proc. 2015 Chinese Automation Congress,* Wuhan, China, 2015, pp. 1845-1849.

[4] D. Neumann, T. Langner, and F. Ulbrich, "Online vehicle detection using Haar-like, LBP and HOG feature based image classifiers with stereo vision preselection" in *Proc. 2017 IEEE Intelligent Vehicles Symposium,* Los Angeles, CA, 2017, pp. 773-778.

[5] M. Uttarakumari, A. S. Koushik, and A. S. Raghavendra, "Vehicle detection using acoustic signatures," in *Proc. 2017 International Conference on Computing, Communication and Automation*, Greater Noida, India, 2017, pp. 1173-1177.

[6] A. N. Knaian, "A wireless sensor network for smart roadbeds and intelligent transportation systems," M.S. thesis, Dept. Elect. Eng. and Comp. Sci., MIT, Cambridge, MA, 2000.

[7] M. Bathula, M. Ramezanali, I. Pradhan, N. Patel, J. Gotschal, and N. Sridhar, "Measuring traffic in short-term construction work zones," in *Proc. 2009 International Conference on Information Processing in Sensor' Networks,* San Francisco, CA, 2009, pp. 361-362.

[8] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of Targets," *IEEE Signal Processing Magazine*, vol. 19, pp. 17-29, Aug. 2002.

[9] Z. Papp, J. Sijs, and M. Lagioia, "Sensor network for real-time vehicle tracking on road networks" presented at 2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*,* Melbourne, VIC, 2009.

[10] G. Padmavathi, D. Shanmugapriya, and M. Kalaivani, "A study on vehicle detection and tracking using wireless sensor networks," *Wireless Sensor Network*, vol. 2, pp. 173185, Feb. 2010.

[11] S. Y. Cheung, S. C. Ergen, and P. Varaiya, "Traffic surveillance with wireless magnetic sensors" presented at 12th World Conference on Intelligent Transport Systems, Berkeley, CA, 2005.

[12] Seong-eun Yoo, "A wireless sensor network-based portable vehicle detector evaluation system," *Sensors*, vol. 13, no. 1, pp. 1160-1182, Jan. 2013.

[13] J. Xing and Q. Zeng, “A model based vehicle detection and classification using magnetic sensor data,” Master’s thesis, Dept. Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, 2007.

[14] M. Liepins and A. Severdaks, “Vehicle detection using non-invasive magnetic wireless sensor network,” in *Proc. 21st Telecommunications Forum Telfor*, Belgrade, Serbia, 2013, pp. 601-604.

[15] M. J. Caruso and L. S. Withanawasam, “Vehicle detection and compass applications using AMR magnetic sensors,” Honeywell SSEC, Plymouth, MN, 2018.

[16] M. F. Duarte and Y. H. Hu, “Vehicle classification in distributed sensor networks,” *J. Parallel Distrib. Comput.*, vol. 64, pp. 826-838, Mar. 2004.

[17] E. Sifuentes, O. Casas, and R. Pallas-Areny, “Wireless magnetic sensor node for vehicle detection with optical wake-up,” *IEEE Sensors Journal*, vol. 11, no. 8, pp. 1669-1676, Aug. 2011.

[18] R. Gupta and S. R. Das, “Tracking moving targets in a smart sensor network,” in *Proc. Vehicular Technology Conference*, Orlando, FL, 2003, pp. 3035-3039.

[19] K. Vijayaraghavan and R. Rajamani. “Novel batteryless wireless sensor for traffic-flow measurement,” *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 3249-3260, May 2010.

[20] G. Zhou, L. Huang, W. Li, and Z. Zhu, “Harvesting ambient environmental energy for wireless sensor networks: A survey,” *Journal of Sensors*, vol. 2014, pp. 1-20, Jun. 2014.

[21] **M. Mozumdar**, et al. Machine learning based single sensor system for automatic vehicle classification, IEEE transactions on intelligent transportation systems (in preparation)

[22] Ying, K., Ameri, A., Trivedi, A., Ravindra D.,Patel D., **Mozumdar, M.**, "Decision Tree-based Machine Learning Algorithm for In-node Vehicle Classification", IEEE Green Energy and Systems Conference (IGESC 2015)

[23] Sharma, V., Parhad, A., **Mozumdar, M.**, "Energy Scavenging Using Piezoelectric Sensors to Power in Pavement Intelligent Vehicle Detection Systems”, METRANS International Urban Freight Conference, 2015

[24] **Mozumdar, M.**, Cruz, J., Mamidi, A., Yeh, H., “A Holistic Approach to Reduce Traffic Congestions in Highways and Surface Roads Using Real Time Traffic Data”, 2013 METRANS International Urban Freight Conference (I-NUF), Long Beach , 2013

[25] Wyman, J. H., Braley, G. A. and Stevens, R. I. (1985). Field Evaluations of FWHA Vehicle Classification Categories, Executive Summary, Materials and Research Technical Paper 84-5, Federal Highway Administration.

[26] Federal Motor Carrier Safety Administration, Interstate Truck Driver's Guide to Hours of Service, March 2015, https://www.fmcsa.dot.gov

[27] California's vision safety roadside rest area system, Moving from the past to the future, September 2016, http://www.dot.ca.gov/

[28] Gordon, R. L., Reiss, R. A., Haenel, H., Case, E. R., French, R. L., Mohaddes, A., & Wolcott, R. (1996). Traffic control systems handbook-revised edition, 1996 (No. FHWA-SA-95-032).

[29] Yang, B., & Lei, Y. (2015). Vehicle Detection and Classification for Low-Speed Congested Traffic With Anisotropic Magnetoresistive Sensor. Sensors Journal, IEEE, 15(2), 1132-1138.

[30] Lan, J., & Shi, Y. (2009, January). Vehicle detection and recognition based on a MEMS magnetic sensor. In Nano/Micro Engineered and Molecular Systems, 2009. NEMS 2009. 4th IEEE International Conference on (pp. 404-408). IEEE.

[31] Lan, J., Xiang, Y., Wang, L., & Shi, Y. (2011). Vehicle detection and classification by measuring and processing magnetic signal. Measurement,44(1), 174-180.

[32] Kaewkamnerd, S., Chinrungrueng, J., Pongthornseri, R., & Dumnin, S. (2010, June). Vehicle classification based on magnetic sensor signal. In Information and Automation (ICIA), 2010 IEEE International Conference on (pp. 935-939). IEEE.

[33] Jolevski, I., Markoski, A., & Pasic, R. (2011, June). Smart vehicle sensing and classification node with energy aware vehicle classification algorithm. In Proceedings of the ITI 2011, 33rd International Conference on Information Technology Interfaces.

[34] Y. Mimbela Luz Elena and L. A. Klein, A Summary of Vehicle Detection and Surveillance Technologies Used in Intelligent Transportation Systems. Washington, DC, USA: Fed. Highway Admin., Intell. Transp. Syst. Joint Program Off., 2000.

[35] S. Gupte and A. Papanikolopoulos, "Algorithms for vehicle classification," Minnesota Dept. Transp. (MnDOT), Saint Paul, MN, USA, Final Rep. MN/RC-2000-27, 2000.

[36] R. Martin, O. Masoud, S. Gupte, and A. Papanikolopoulos, "Algorithms for vehicle classification: Phase II," Minnesota Dept. Transp. (MnDOT), Saint Paul, MN, USA, Final Rep., MN/RC-2002-21, 2002.

[37] A. Y. Nooralahiyan, M. Dougherty, D. McKeown, and H. R. Kirkby, "A field trial of acoustic signature analysis for vehicle classification," Transp. Res. Part C, vol. 5, no. 3/4, pp. 165–177, Aug.–Oct. 1997.

[38] G. Zhang, Y. Wang, and H. Wei, "Artificial neural network method for length-based vehicle classification using single-loop outputs," Transp. Res. Rec., vol. 1945, pp. 100–108, 2006.

[39] Traffic Monitoring Guide, Fed. Highway Admin., Washington, DC, USA, 2001.

[40] I. Urazghildiiev, R. Ragnarsson, P. Ridderstrom, A. Rydberg, E. Ojefors, K. Wallin, P. Enochsson, M. Ericson, and G. Lofqvist, "Vehicle classifi- cation based on the radar measurement of height profiles," IEEE Trans. Intell. Transp. Syst., vol. 8, no. 2, pp. 245–253, Jun. 2007.

[41] B. Coifman and S. Kim, "Speed estimation and length based vehicle classification from freeway single loop detectors," Transp. Res. Part C, vol. 17, no. 4, pp. 349–364, 2009.

[42] He, Z., Zhu, H., & Yu, F. (2014, April). A vehicle detection algorithm based on wireless magnetic sensor networks. In Information Science and Technology (ICIST), 2014 4th IEEE International Conference on (pp. 727-730). IEEE.

[43] Cheung, S., Coleri, S., Dundar, B., Ganesh, S., Tan, C. W., & Varaiya, P. (2005). Traffic measurement and vehicle classification with single magnetic sensor. Transportation research record: journal of the transportation research board, (1917), 173-181.

[44] Yang, B., & Lei, Y. (2015). Vehicle Detection and Classification for Low-Speed Congested Traffic With Anisotropic Magnetoresistive Sensor. Sensors Journal, IEEE, 15(2), 1132-1138.

[45] Lan, J., Xiang, Y., Wang, L., & Shi, Y. (2011). Vehicle detection and classification by measuring and processing magnetic signal. Measurement,44(1), 174-180.

[46] Kaewkamnerd, S., Chinrungrueng, J., Pongthornseri, R., & Dumnin, S. (2010, June). Vehicle classification based on magnetic sensor signal. In Information and Automation (ICIA), 2010 IEEE International Conference on (pp. 935-939). IEEE.

[47] Kaewkamnerd, S., Pongthornseri, R., Chinrungrueng, J., & Silawan, T. (2009, September). Automatic vehicle classification using wireless magnetic sensor. In Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2009. IDAACS 2009

[48] Jolevski, I., Markoski, A., & Pasic, R. (2011, June). Smart vehicle sensing and classification node with energy aware vehicle classification algorithm. In Proceedings of the ITI 2011, 33rd International Conference on Information Technology Interfaces.

[49] He, Y., Du, Y., & Sun, L. (2012). Vehicle classification method based on single-point magnetic sensor. Procedia-Social and Behavioral Sciences, 43, 618-627.

[50] A. Haoui, R. Kavaler, and P. Varaiya, "Wireless magnetic sensors for traffic surveillance," Transp. Res. Part C, vol. 16, no. 3, pp. 294–306, Jun. 2008.

[51] Sensys networks Inc. www.sensysnetworks.com/

[52] S. Cheung and P. Varaiya, "Traffic surveillance by wireless sensor networks," California PATH Program Inst. Transp. Stud., Univ. California, Berkeley, CA, USA, Final Rep. UCB-ITS-PRR-2007-4, 2007.

[53] R. Bajwa, R. Rajagopal, P. Varaiya, and R. Kavaler, “In-pavement wireless sensor network for vehicle classification,” in Proc. IPSN, 2011, pp. 85–96.

[54] Van Dam, T., & Langendoen, K. (2003, November). An adaptive energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the 1st international conference on Embedded networked sensor systems (pp. 171-180). ACM.

[55] Stathopoulos, T., Kapur, R., Estrin, D., Heidemann, J., & Zhang, L. (2004, November). Application-based collision avoidance in wireless sensor networks. In Local Computer Networks, 2004. 29th Annual IEEE International Conference on (pp. 506-514). IEEE.

[56] Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (Vol. 3, pp. 1567-1576). IEEE.

[57] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., & Pister, K. (2000, November). System architecture directions for networked sensors. In ACM SIGOPS operating systems review

[58] Quinlan, J.R. (1993). "C4.5: programs for machine learning". San Francisco, CA: Morgan Kaufmann Publishers.