# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**Jnana Sangama, Belagavi – 590018**

**PROJECT REPORT**

**on**

# "SYSTEMATIC ASSESMENT OF CYBER-PHYSICAL SECURITY OF EMS FOR CONNECTED AND AUTOMATED ELECTRIC VEHICLES"

Submitted in partial fulfilment for the award of the degree

**Bachelor of Engineering**

in

**Computer Science and Engineering**


**Submitted by**

**Gagan R**

**1ST20CS039**

**Under the Guidance of**

**Prof. Suchitra Devi A**

**Asst. Professor**

**Department of CSE**


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## SAMBHRAM INSTITUTE OF TECHNOLOGY

**M. S. Palya, Bengaluru – 560097**

**2023-2024**

# SAMBHRAM INSTITUTE OF TECHNOLOGY

**M. S. Palya, Bengaluru – 560097**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# CERTIFICATE

Certified that the Project work entitled "**SYSTEMATIC ASSESMENT OF CYBER-PHYSICAL SECURITY OF ENERGY MANAGEMENT SYSTEM FOR CONNECTED AND AUTOMATED ELECTRIC VEHICLES**" carried out by Mr. **Gagan R , 1ST20CS039**, bonafide student of **SAMBHRAM INSTITUTE OF TECHNOLOGY** in partial fulfilment for the award of **BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING** of **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**, Belagavi during the year **2023-2024**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

| | | |
|---|---|---|
| **Prof. Suchitra Devi A**<br>Assistant Professor<br>Dept. of CSE<br>SaIT, Bengaluru | **Dr. T. John Peter**<br>HOD<br>Dept. of CSE<br>SaIT, Bengaluru | **Dr. H. G. Chandrakanth**<br>Principal<br>SaIT, Bengaluru |

**EXTERNAL VIVA:**

**Name of the Examiners** **Signature with date**

**1.**

**2.**

# SAMBHRAM INSTITUTE OF TECHNOLOGY

**M. S. Palya, Bengaluru – 560097**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# DECLARATION

I'm the student of 8th Semester, Dept. of CSE, SaIT doing the Final Year Project, declare that

[1] The Hardware/Software is not purchased/brought from any outside originations.

[2] The Hardware/Software is not from any other previous final year engineering projects of VTU.

[3] Our Project work is as per VTU norms, and we have followed the rules and regulations.

Violating any of the above conditions, we will accept the action taken by the Department/College/ VTU in this regard.

The Title of the Project is
**SYSTEMATIC ASSESMENT OF CYBER-PHYSICAL SECURITY OF ENERGY MANAGEMENT SYSTEM FOR CONNECTED AND AUTOMATED ELECTRIC VEHICLES**

The project work is guided by:
**Prof. Suchitra Devi A**
Asst. Professor
Dept. of CSE, SaIT.

| No. | Student Name | USN | Signature |
|---|---|---|---|
| 1 | **Gagan R** | **1ST20CS039** | _____ |

# ABSTRACT

In this project, a systematic assessment of cyber physical security on the energy management system for connected and automated electric vehicles is proposed, which, to our knowledge, has not been attempted before. The generalized methodology of impact analysis of cyber-attacks is developed, including novel evaluation metrics from the perspectives of steady-state and transient performance of the energy management system and innovative index-based resilience and security criteria. Specifically, we propose a security criterion in terms of dynamic performance, comfortability, and energy, which are the most critical metrics to evaluate the performance of an electronic control unit (ECU). If an attack does not impact these metrics, it perhaps can be negligible. Based on the statistical results and the proposed evaluation metrics, the impact of cyber-attacks on ECU is analyzed comprehensively. The conclusions can serve as guidelines for attack detection, diagnosis, and countermeasures.

# ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. I would like to take this opportunity to thank them all.

I would like to express our heartfelt thanks to **Dr. H. G. Chandrakanth, Principal, Sambhram Institute of Technology**, whose valuable guidance has been the one that helped us to complete the project.

I would like to express our profound gratitude to **Dr. T. John Peter, HOD, Department of CSE, Sambhram Institute of Technology**, for his suggestions and his instructions have served as the major contribution towards the completion of the project.

I would like to extend our impassioned thanks and admiration to our **guide, Prof. Suchitra Devi A, Assistant Professor, Department of CSE, Sambhram Institute of Technology**, for their able guidance, regular source of encouragement and assistance throughout this project.

I would like to thank all the teaching and non-teaching staff members of the Computer Science Department, who have helped me directly or indirectly for the successful completion of the project.

Finally, I would like to thank my Parents and Friends who have helped me with their valuable suggestions and guidance for the completion of the project.

**Gagan R**
**1ST20CS039**

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

With the significant increase in the traffic, road, and environmental information enabled by vehicle-to infrastructure/cloud/vehicle communications, the connected and automated vehicle (CAV) technology can significantly enhance the driving safety, comfort, and energy efficiency. However, since a large number of embedded ECUs are integrated into networks, it also brings cyber-security concerns. As demonstrated by recent examples, the vehicles are vulnerable to cyber-attacks, allowing an attacker to circumvent the vehicle control systems, which would lead to severe consequences such as disabling brakes, turning off headlights, and taking over steering. For example, cyber-attacks on anti-lock braking systems in demonstrate that a malicious attacker can modify the feedback measurements through wheel speed sensors and cause life-threatening situations. Spoofing attacks on the global positioning system (GPS) may result in course deviation in an autonomous vehicle. Some cyberattacks through direct (by connecting with onboard diagnostics (OBD-II) port) and remote (through wireless channels like Bluetooth) access have also been reported in the literature. Furthermore, cyber-attacks in connected and automated vehicles (CAVs) through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) are discussed and have received increased attention in real-life scenarios in the last two years.

The integration of Connected and Automated Electric Vehicles (CAEVs) into modern transportation systems offers substantial benefits, including enhanced efficiency, reduced emissions, and improved safety. However, this technological advancement also introduces significant cyber-physical security challenges, particularly within the Energy Management System (EMS) that governs energy consumption and operational optimization of these vehicles. As CAEVs become increasingly connected, they are more susceptible to a range of cyber and physical threats that could compromise their functionality and safety. This project aims to systematically assess the cyber-physical security of EMS for CAEVs, identifying potential vulnerabilities, evaluating associated risks, and proposing robust mitigation strategies to safeguard the integrity and reliability of these critical systems in the face of evolving security threats.

In particular, due to the connection with battery charging infrastructure, more centralized control architecture, and higher electrification, the cyber-physical security in connected and

automated electric vehicles (CAEVs) is receiving much more attention compared to an internal combustion engine (ICE) vehicle. For example, the connectivity between CAEVs, charging stations and smart grid may expose the CAEVs to the cyberattacks. Compared to conventional cyber approaches for ICE vehicles, for instance, that focus on a vehicle's entry points cyber-physical security monitoring can serve as a second line of protection because an abnormal system measurement is a clear indicator for potential cyber-attacks. However, cyberphysical security on CAEVs is still in its infancy. Due to the lack of security monitoring, they are prone to a wide range of cyber-attacks ranging from conventional eaves-dropping and denial of service (DOS) attacks to man-in-the-middle (MiTM) attacks that degrade the vehicle's performance [14]. The consequences can be catastrophic as they have the ability to cause physical damage to vehicles, people, and the infrastructure (the grid). There have been some preliminary works on cyber-security of battery management systems.

In this project , we propose a systematic vulnerability assessment of CAEVs, and the main contributions are as follows:

• A framework of impact analysis of cyber-physical security on core control systems in CAEVs, such as electronic stability control, antilock brake system that focuses on driving safety, advanced driver assistance system, and energy management system is presented.

• For the vulnerability assessment of the EMS in a CAEV, we design a model predictive control (MPC) based system that optimizes both the instantaneous driving velocity and torque allocation to reduce energy consumption, which is considered as one of the applications of EMS. Based on the system, we develop innovative index-based evaluation metrics in terms of dynamic performance, comfortability, energy, security, and resilience. If an attack does not impact these metrics, it perhaps can be negligible.

• The impact of cyber-attacks are analyzed under specific and statistical results, and the vulnerability of the vehicle to each attach type is discussed based on the evaluation metrics and security criteria. The conclusions can serve as guidelines for attack detection and countermeasures.

## OBJECTIVE OF THE PROPOSED WORK

- To reduce energy consumption.
- To provide guidelines for attack detection and countermeasures.
- To develop a framework of impact analysis of cyber-physical security on core control systems in CAEVs, such as electronic stability control, antilock brake system that focuses on driving safety, advanced driver assistance system, and energy management system.
- To design a model predictive control (MPC) based system for vulnerability assesment.

## PROBLEM STATEMENT

Cyber-physical security challenges in CAEVs remains significant:

(1) Most of the existing works are developed for connected and automated ICE vehicles rather than CAEVs.

(2) Only safety-critical systems are addressed while long-term specification like efficiency performance (e.g., energy management system (EMS)) receives little attention. It is essential particularly for CAEVs because of the limited battery capacity and the 'range anxiety.'

# CHAPTER 2

# LITERATURE SURVEY

A literature survey or a literature review in a project report shows the various analyses and research made in the field of interest and the results already published, taking into account the various parameters of the project and the extent of the project. Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system & guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

A literature survey is a text of a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews use secondary sources, and do not report new or original experimental work. Most often associated with academic-oriented literature, such as a thesis, dissertation or a peer-reviewed journal article, a literature review usually precedes the methodology and results sectional though this is not always the case. Literature reviews are also common in are search proposal or prospectus (the document that is approved before a student formally begins a dissertation or thesis). Its main goals are to situate the current study within the body of literature and to provide context for the particular reader. Literature reviews are a basis for researching nearly every academic field. demic field. A literature survey includes the following:

- Existing theories about the topic which are accepted universally.
- Books written on the topic, both generic and specific.
- Research done in the field usually in the order of oldest to latest.
- Challenges being faced and on-going work, if available.

Literature survey describes about the existing work on the given project. It deals with the problem associated with the existing system and also gives user a clear knowledge on how to deal with the existing problems and how to provide solution to the existing problems Objectives of Literature Survey

- Learning the definitions of the concepts.
- Access to latest approaches, methods and theories.
- Discovering research topics based on the existing research
- Concentrate on your own field of expertise– Even if another field uses the same words, they usually mean completely.
- It improves the quality of the literature survey to exclude sidetracks– Remember to explicate what is excluded.

Before building our application, the following system is taken into consideration:

The growing range of cyber-security risks shown above has been promoting the development of vehicle cyber-security techniques for both theoretical and application aspects. The efforts can be categorized into two schemes. The first scheme focuses on the ability to prevent malicious attacks. For instance, throughout the vehicle development cycle, automakers can define core performance requirements of subsystems to automotive parts suppliers, and then the subsystems are designed by considering its security within the software. To prevent malicious attacks through direct contact with the OBD-II port, the communication protocol of the OBD-II is kept secret to the public. Several critical practices, like secure hardware, secure software updates, penetration testing, and code reviews, are also widely used by the automotive industry . Besides, approaches concerning information security during driving, such as message authentication and encryption, the firewall between external networks and vehicle devices are also taken into consideration .

Although these conventional vehicle cybersecurity and information-security approaches can be used to prevent attacks, they alone cannot guarantee the security of the whole system. Therefore, cyber-physical security from the control perspective that concentrates on improving the resilience of the automotive control system to attack should be addressed, including impact analysis , attack detection and diagnosis , and resilient control While these efforts provide some technical foundations, cyber-physical security challenges in CAEVs remains significant:

(1) Most of the existing works are developed for connected and automated ICE vehicles rather than CAEVs.

(2) Only safety-critical systems are addressed while long-term specification like efficiency performance (e.g., energy management system (EMS)) receives little attention. It is essential particularly for CAEVs because of the limited battery capacity and the 'range anxiety.'

For instance, the authors provide a physics-driven approach to assess the vulnerability of EV batteries, and the results have shown that cyber-attacks can lead to faster deterioration in power capability and battery life. Furthermore, most of the existing literature is cyber-based methods and rely heavily on communication technology. There is little work on impact analysis on cyber-attacks. Although there have been some researches focusing on impact analysis of cyber threats on cyber-physical systems, e.g., electric systems and smart grids, they mainly focus on few metrics such as active (or reactive) power, system frequency, node voltage, and power angle. For example, the authors analyzed the data integrity attacks on automatic generation control loop for smart grids; the cybersecurity policies for flexible alternating current transmission devices are discussed and presented the impact of integrity attacks on electric market operations; used reachability methods in graph theory to assess the risks and vulnerabilities of two-area power systems. For a complicated control system in a CAEV, such as safety system (electronic stability control, antilock brake, etc.), auto driving system (adaptive cruise control system, lane keep assistance, etc.), and EMS (torque split optimization, battery management system, etc.), more detailed models and metrics should be considered to evaluate the system comprehensively. For example, the upper autonomous controller or human driver requires a fast and accurate dynamic response, reasonable power output, low torque ripple, as well as minimizing energy consumption in various drive cycles. These performances should be particularly addressed for control systems in CAEVs while these approaches for electric systems and smart grids are unfeasible. In summary, it is essential to emphasize the cyber-security challenge of the ECUs in CAEVs, and novel methodologies of vulnerability assessment should be developed.

# CHAPTER 3

# METHODOLOGIES

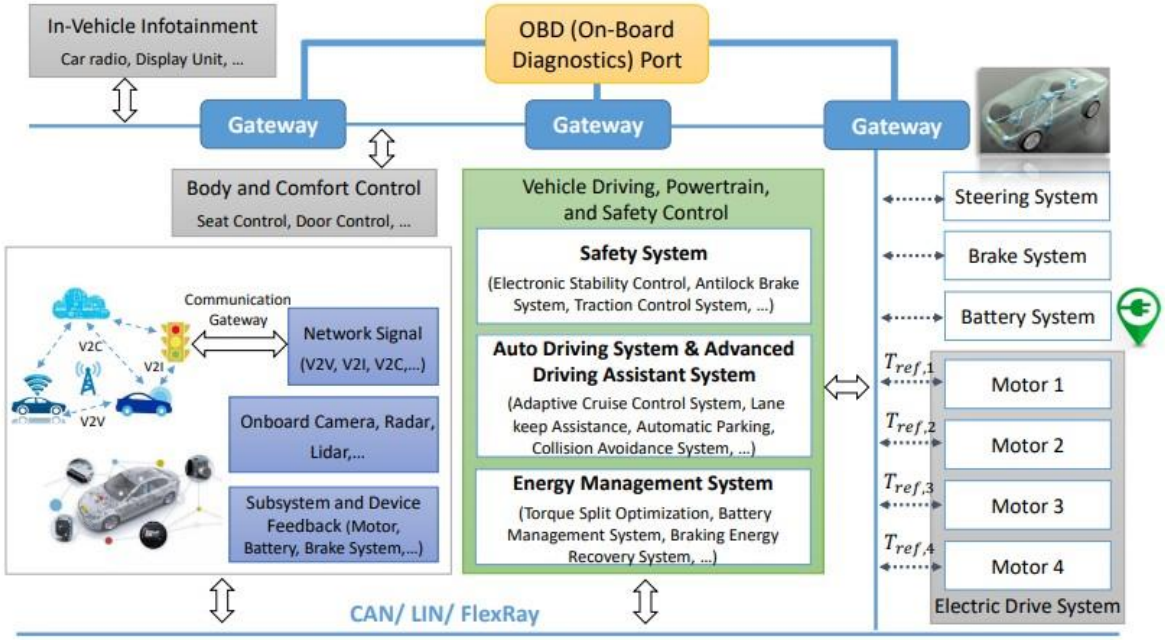


Fig.3.1 System Architecture

Generalized methodology for cyber attack impact analysis: This involves creating new evaluation metrics to assess the impact of cyberattacks on the EMS.

These metrics consider:

• Steady-state performance: How the attack affects the system's stable operation.

• Transient performance: How the attack impacts the system's behavior during sudden changes.

## 1. Elliptic Curve Cryptography and Blockchain:

- It assures power security and maintains an adequate balance between demand and supply.
- It is not suitable for automated electric vehicles.

## 2. Charging Scheduling Algorithm:

- optimize the EV charging behaviors, reduce charging costs, and balance load.

- It suits for load balancing.
- It is not discuss about the cyber physical security.

## 3. Blockchain:

- The proposed methods transformative to other cyber-physical system applications.

## Training of ML models:

In this module we will apply the various machine learning algorithms to train the preprocessed data to predict the attacks.

In this module we are comparing the performance of following algorithms.

- Random forest
- KNN
- Decision Tree
- Logistic Regression
- AdaBoost

## AdaBoost:

AB is a tree-based ensemble classifier that incorporates many weak classifiers to reduce misclassification errors. It selects the training set and iteratively assigns the weights depending on the previous training precision for retraining the algorithm. In order to train any weak classifier, an arbitrary subset of the full training set is used and AB assigns weights to each instance and classifier. The following equation defines the combination of several weak classifiers:

$$H(x) = Sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$

where H(x) defines the output of the final model through combining the weak classifiers and ht(x) represents the output of classifier t for input x and αt specifies the weight assigned to the classifier.

## Random Forest:

RF is a decision tree-based ensemble classification method and follows the split and conquer technique in the input dataset to create multiple decision-making trees (known as the forest). It works in two phases. At first, it creates a forest by combining the 'N' number of decision trees and in the second phase, it makes predictions for each tree generated in the first phase. The working process of the RF algorithm is illustrated below:

1) Select random samples from the training dataset.

2) Construct decision trees for each training sample.

3) Select the value of 'N' to define the number of decision trees.

4) Repeat Steps 1 and 2.

5) For each test sample, find the predictions of each decision tree, and assign the test sample a class value based on majority voting.

## Decision Tree:

DT follows a top-down approach to build a predictive model for class values using training data-inducing decision-making rules. This research utilized the information gain method to select the best attribute. Assuming Pi , the probability such that xi€D, exists to a class Ci , and is predicted by |Ci, D|/|D|. To classify instances in the dataset D, the required information is needed, and the following equation calculates it:

$$Info(D) = -\sum_{i=1}^{m} P_i \log_2(P_i)$$

## KNN:

KNN classifies the test data by utilizing the training data directly by calculating the K value, indicating the number of KNN. For each instance, it computes the distance between all the training instances and sorts the distance. Furthermore, a majority voting technique is employed to assign the final class label to the test data. This research applies Euclidean distance to

$$D_e = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$$

calculate the distances among instances. The following equation represents the Euclidean distance calculation:

## LOGISTIC REGRESSION:

Based on a given dataset of independent variables, logistic regression calculates the likelihood that an event will occur, such as voting or not voting. Given that the result is a probability, the dependent variable's range is 0 to 1. In logistic regression, the odds—that is, the likelihood of success divided by the probability of failure-are transformed using the logit formula. The following formulae are used to express this logistic function, which is sometimes referred to as the log odds or the natural logarithm of odds.

## Conclusion:

This methodology outlines a systematic approach to cybersecurity anomaly detection, encompassing data preprocessing, feature engineering, model training.

# CHAPTER 4

# SYSTEM ANALYSIS

## 4.1 Existing System

In particular, due to the connection with battery charging infrastructure, more centralized control architecture, and higher electrification, the cyber-physical security in connected and automated electric vehicles (CAEVs) is receiving much more attention compared to an internal combustion engine (ICE) vehicle. For example, the connectivity between CAEVs, charging stations and smart grid may expose the CAEVs to the cyberattacks. Compared to conventional cyber approaches for ICE vehicles, for instance, that focus on a vehicle's entry points, cyber-physical security monitoring can serve as a second line of protection because an abnormal system measurement is a clear indicator for potential cyber-attacks. However, cyber physical security on CAEVs is still in its infancy.

### Limitation:

Due to the lack of security monitoring, they are prone to a wide range of cyber-attacks ranging from conventional eaves-dropping and denial of service (DOS) attacks to man-in-the-middle (MiTM) attacks that degrade the vehicle's performance.

## 4.2 Proposed System

In this project, we propose a systematic vulnerability assessment of CAEVs, and the main contributions are as follows:

• A framework of impact analysis of cyber-physical security on core control systems in CAEVs, such as electronic stability control, antilock brake system that focuses on driving safety, advanced driver assistance system, and energy management system is presented.

• For the vulnerability assessment of the EMS in a CAEV, we design a model predictive control (MPC) based system that optimizes both the instantaneous driving velocity and torque allocation to reduce energy consumption, which is considered as one of the applications of EMS. Based on the system, we develop innovative index-based evaluation metrics in terms of

dynamic performance, comfort ability, energy, security, and resilience. If an attack does not impact these metrics, it perhaps can be negligible.

• The impact of cyber-attacks are analyzed under specific and statistical results, and the vulnerability of the vehicle to each attach type is discussed based on the evaluation metrics and security criteria. The conclusions can serve as guidelines for attack detection and countermeasures

**Advantages:**

- High Accuracy
- Avoid over fitting of data

## 4.3 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- OPERATINAL FEASIBILITY

### 4.3.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### 4.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### 4.3.3 Operational Feasibility

**Operational Requirements:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# CHAPTER 5

# SOFTWARE REQUIREMENTS

System Requirement Specification (SRS) is a central report, which frames the establishment of the product advancement process. It records the necessities of a framework as well as has a depiction of its significant highlight. An SRS is essentially an association's seeing (in composing) of a client or potential customer's frame work necessities and conditions at a specific point in time (generally) before any genuine configuration or improvement work. It's a two-way protection approach that guarantees that both the customer and the association comprehend alternate's necessities from that viewpoint at a given point in time.

The SRS talks about the item however not the venture that created it, consequently the SRS serves as a premise for later improvement of the completed item. The SRS may need to be changed, however it does give an establishment to proceed with creation assessment. In straightforward words, programming necessity determination is the beginning stage of the product improvement action.

The SRS means deciphering the thoughts in the brains of the customers – the information, into a formal archive – the yield of the prerequisite stage. Subsequently the yield of the stage is a situated of formally determined necessities, which ideally are finished and steady, while the data has none of these properties.

## 5.1 Functional Requirements

This section describes the functional requirements of the system for those requirements which are expressed in the natural language style.

1. Create a desktop application using tinter framework which contains road vehicle simulation.

2. System will monitor the activities in and detect the attacks.

3. Show the simulation.

## 5.2 Non-Functional Requirements:

These are requirements that are not functional in nature, that is, these are constraints within which the system must work.

- The program must be self-contained so that it can easily be moved from one Computer to another. It is assumed that network connection will be available on the computer on which the program resides.

## Capacity, scalability and availability.

- The system shall achieve 100 per cent availability at all times.
- The system shall be scalable to support additional clients and volunteers.

## Maintainability.

The system should be optimized for supportability, or ease of maintenance as far as possible. This may be achieved through the use documentation of coding standards, naming conventions, class libraries and abstraction.

## Randomness, verifiability and load balancing.

The system should be optimized for supportability, or ease of maintenance as far as possible. This may be achieved through the use documentation of coding standards, naming conventions, class libraries and abstraction. It should have randomness to check the nodes and should be load balanced.

## Assumptions and Dependencies:

1. Existing logging and monitoring infrastructure.

2. Access to system and network activity data.

3. Integration with existing security tools and systems.

## 5.3 Hardware Requirements

- Processor Type : Intel Core$^{TM-}$ i5

- Speed : 2.4 GHZ
- RAM :8 GB RAM
- Hard disk : 80 GB HDD

## 5.4 Software Requirements

- Operating System : Windows 64-bit
- Technology : Python
- IDE : PythonIDLE
- Tools : Anaconda
- Python Version : Python 3.6

### 5.4.1 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software[30] and has a community-based development model, as do nearly all of Python's other implementations. Python and C Python are managed by the non-profit Python Software Foundation.

**History**

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum's long influence on Python is reflected in the title given to him by the Python community: Benevolent Dictator For Life (BDFL) – a post from which he gave himself permanent vacation on July 12, 2018.

Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x[37] and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. In January 2017, Google announced work on a Python 2.7 to Go transcompiler to improve performance under concurrent workloads.

**Features and philosophy**

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has filter(), map(), and reduce() functions; list comprehensions, dictionaries, and sets; and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

Beautiful is better than ugly

Explicit is better than implicit

Simple is better than complex

Complex is better than complicated

**Methods**

Methods on objects are functions attached to the object's class; the syntax instance.method(argument) is, for normal methods and functions, syntactic sugar for Class.method(instance, argument). Python methods have an explicit self parameter to access instance data, in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).

**Libraries**

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

**Graphical user interfaces**

- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Image processing

**Development environments**

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Other shells, including IDLE and IPython, add further abilities such as auto-completion, session state retention and syntax highlighting.

As well as standard desktop integrated development environments, there are Web browser-based IDEs; SageMath (intended for developing science and math-related Python programs); PythonAnywhere, a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing scientific computing.

**Implementations**

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several select C99 features. It compiles Python programs into an intermediate bytecode which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including Windows and most modern Unix-like systems. Platform portability was one of its earliest priorities.

# CHAPTER 6

# SYSTEM DESIGN

The system “design” is defined as the process of applying various requirements and permits it physical realization. Various design features are followed to develop the system the design specification describes the features of the system, the opponent or elements of the system and their appearance to the end-users

This project provides a general guidance for vulnerability assessment of core control systems for CAEVs, with which the impact of cyber-attacks on different critical systems and signals, as well as the interaction between these subsystems can be analyzed comprehensively. We are simulating the results of the data in CAEV networks.

We have designed this application as a simulation where vehicles will move on a road by receiving commands from ECU and then we will monitor vehicle velocity to detect normal and attack scenarios and then record energy consumption in both scenarios and report.

## 6.1 SYSTEM ARCHITECTURE

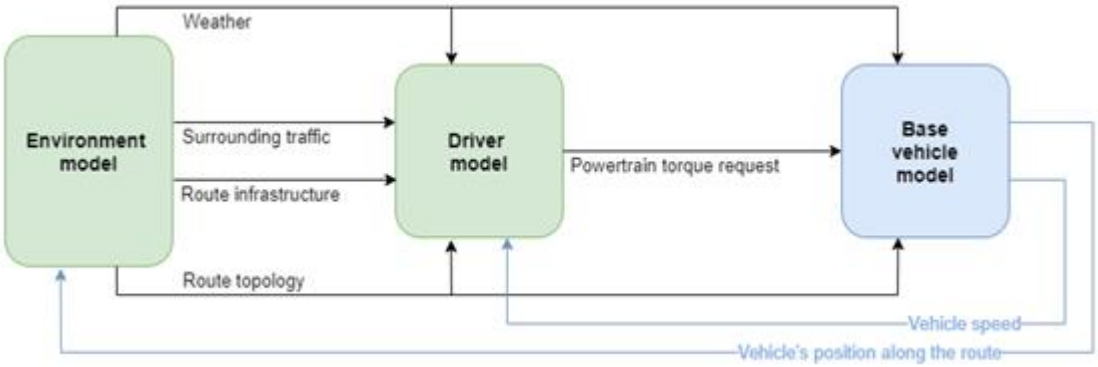


**Figure: 6.1 System architecture**

The simulation framework has been used in a first phase to aid project partners in the sizing of components for their demonstrators by supporting the engineering decisions. The framework is later used to develop the advanced energy and thermal management strategies in a virtual environment prior to being implemented and tested in the demonstrator vehicles.

## 6.2 Data flow diagram

**Level:0** Describes the overall process of this project. we are passing EMS Evs data System will read the data and simulate the data and provide the result.

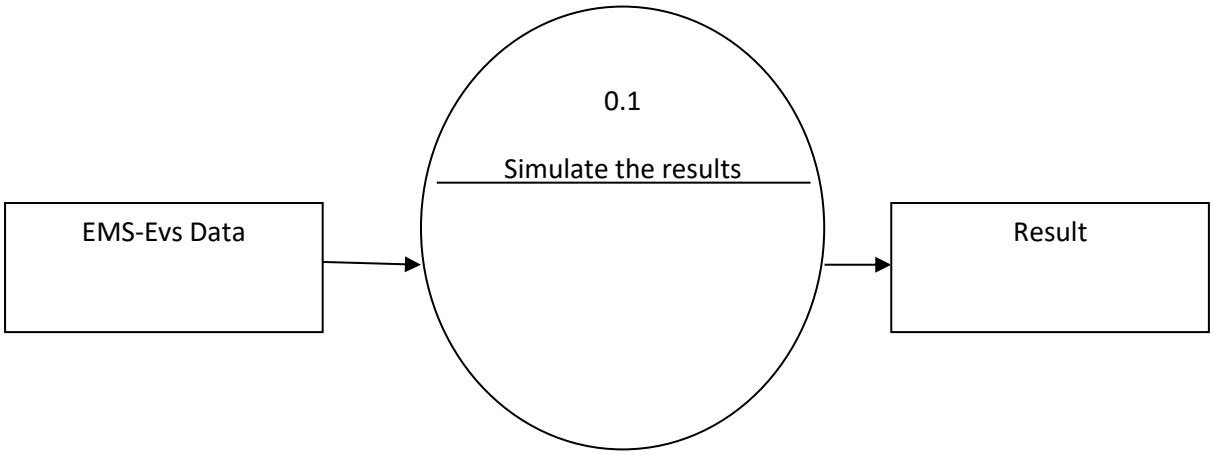


**Figure 6.2 Data flow diagram**

## 6.3 Detailed Design

### 6.3.1 DFD Level 1

**Level 1:** Describes the first stage process of this project. we are passing EMS Dataset as dataset to the system will preprocess and extract the important features.

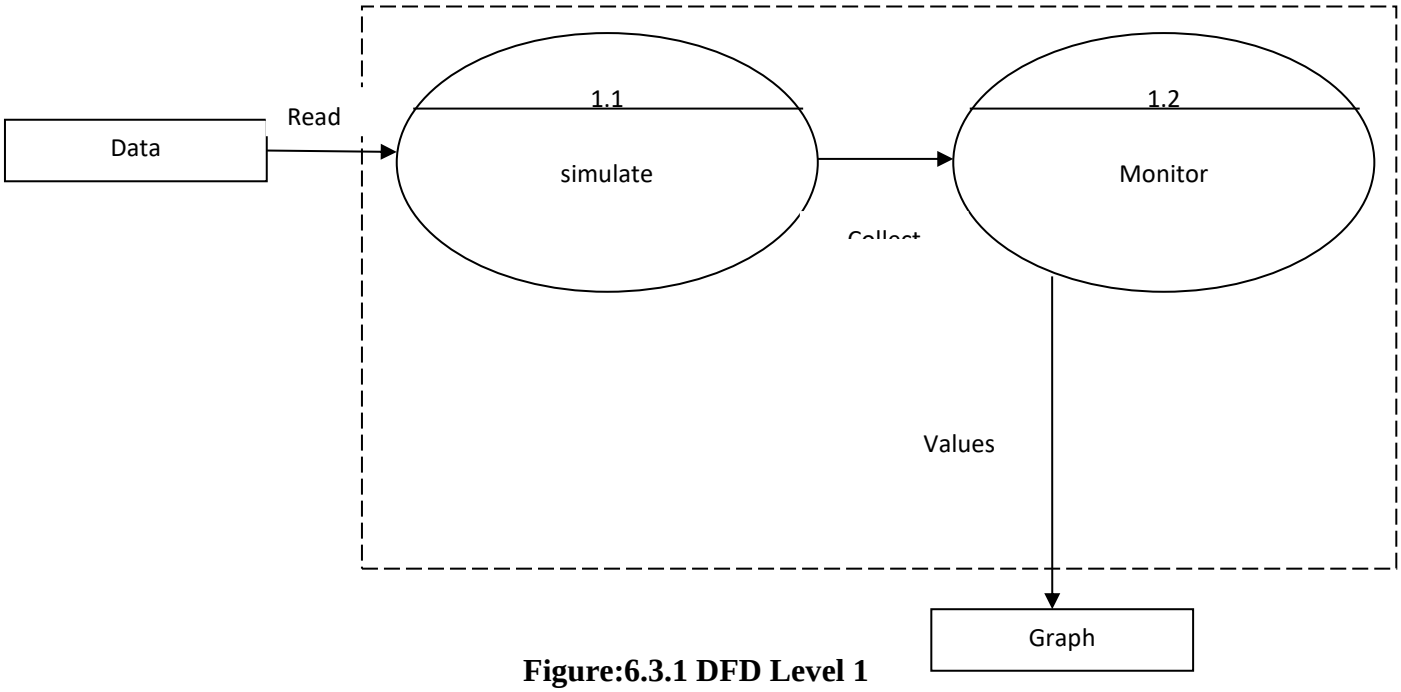


**Figure:6.3.1 DFD Level 1**

**Level-2:**

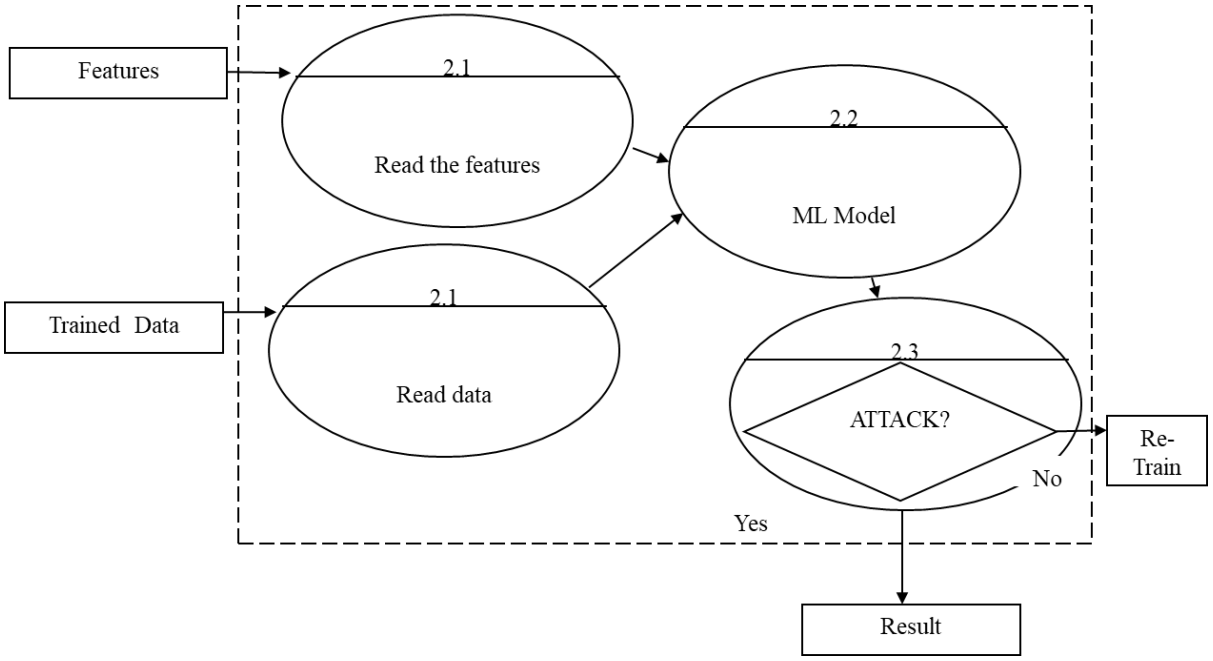


**Figure : 6.3.1. DFD Level 2**

**Level 2:** Describes the final stage process of this project. we are passing extracted features from level 1and trained data as a input the system will detect attacks using ML model from the given input.

## 6.3.2 Use case diagram

The Below figure represent use case diagram of proposed system, where user inputs data of EMS , the algorithm work to generate the identified output as attack or normal. The actor and use case is represented. An eclipse shape represents the use case namely input image, pre-process, training, recognition and output.

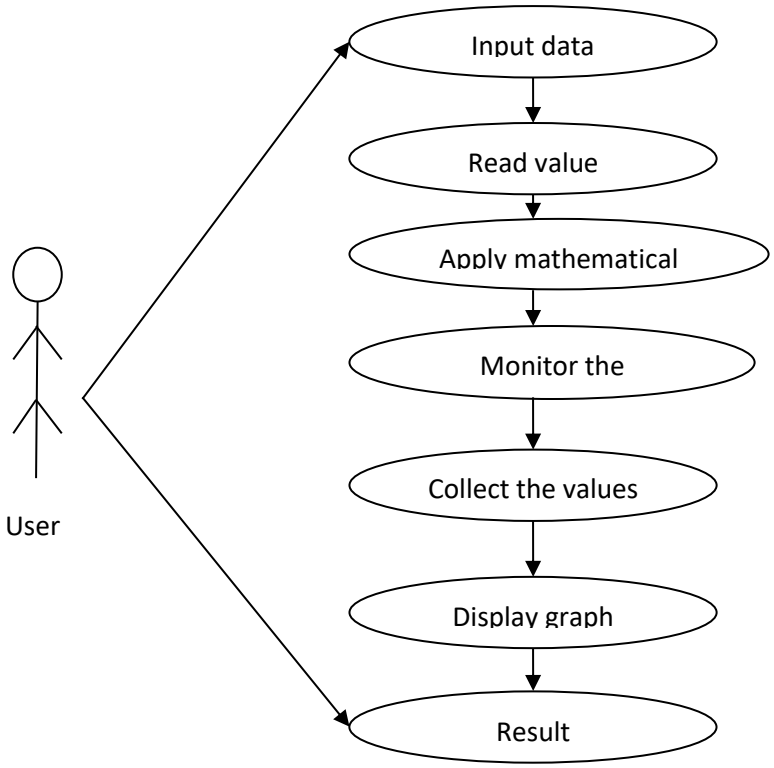


**Figure : 6.3.2 Use Case Diagram**

## 6.3.3 Activity Diagram

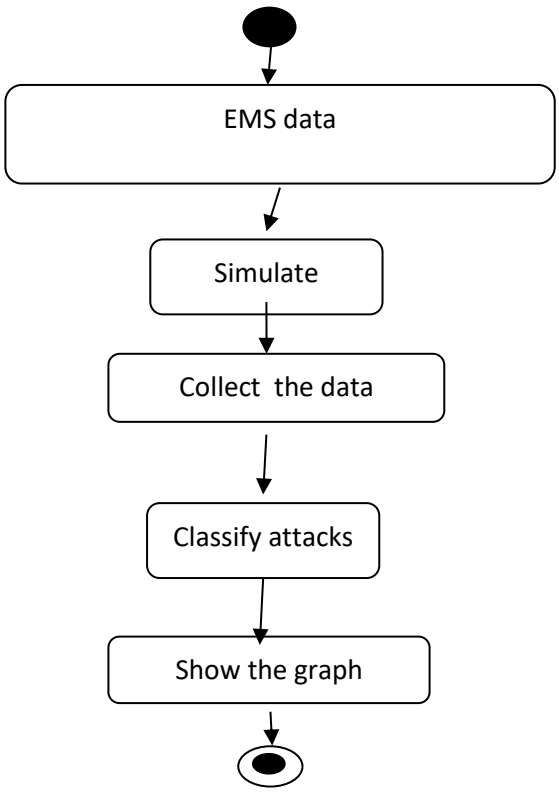


**Figure 6.3.3 Activity Diagram**

### 6.3.4 Sequence Diagram

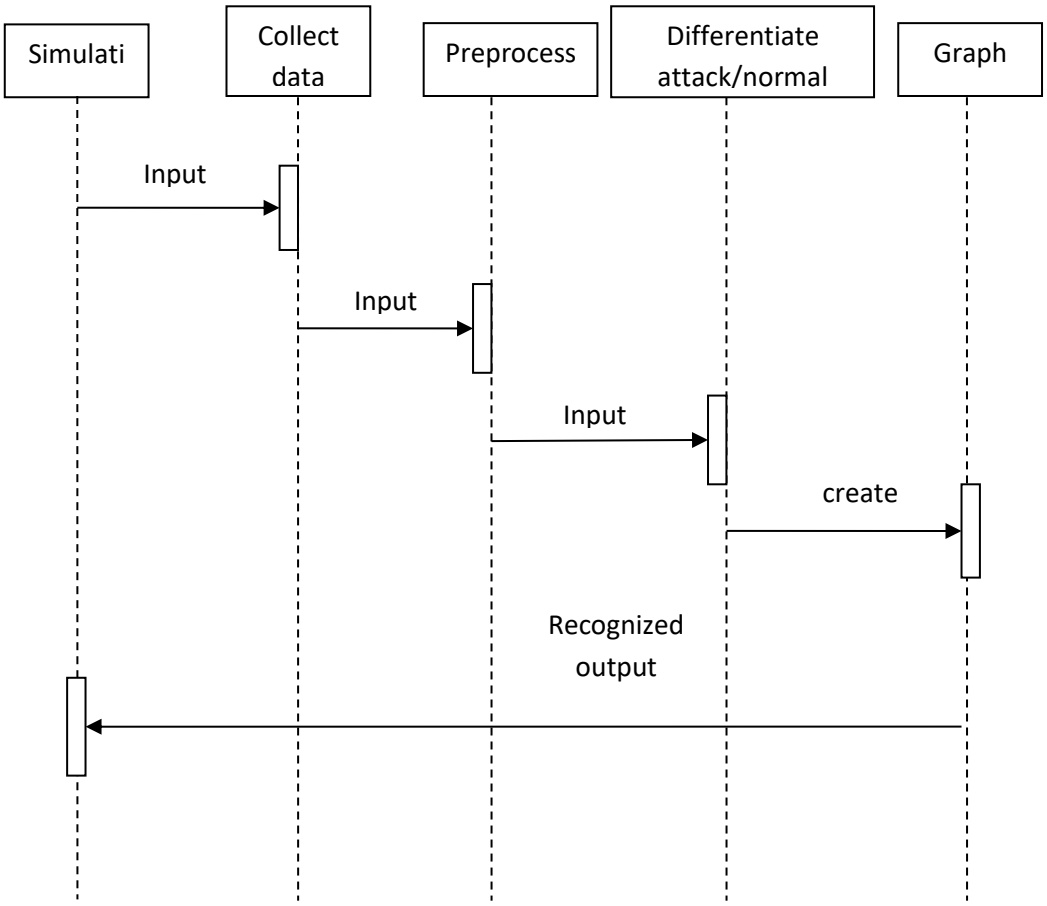


**Figure : 6.3.4 Sequence Diagram**

A sequence diagram shows a parallel vertical lines, different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in order in which they occur. The above figure represents sequence diagram, the proposed system's sequence of data flow is represented.

### 6.3.5 Object Diagram

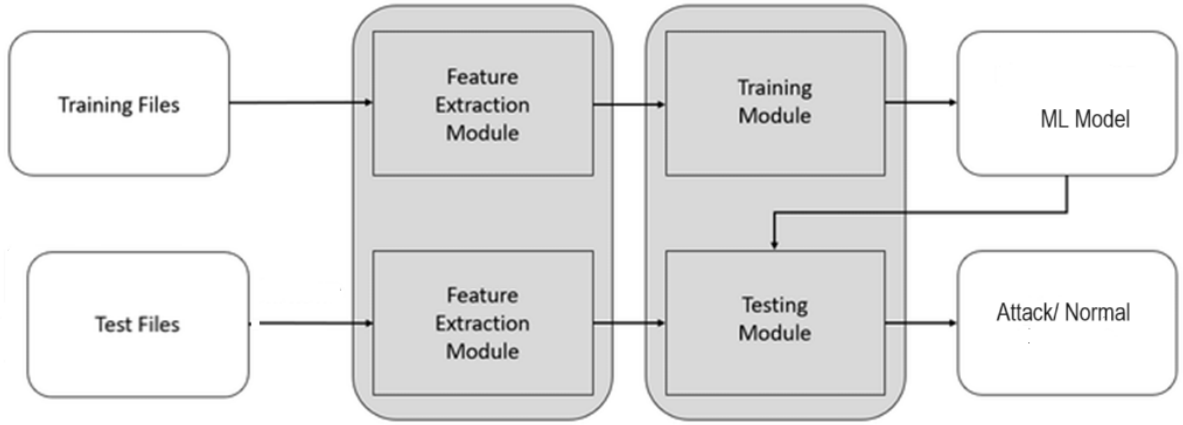


**Figure 6.3.5 Object Diagram**

# CHAPTER 7

# IMPLEMENTATION

## Dataset collection:

- We use the EMS-EVs dataset to evaluate the performance of the proposed intelligent attack detection method. The dataset is a benchmark dataset for network intrusion, which is an improved version of the EMS-EVs dataset.
- The Dataset contains 125,973 training traffic samples and 22,554 test traffic samples. For training seven weeks of network traffics were collected in the form of raw tcpdump format, and the following two weeks of network traffics were also collected for testing.
- To make the attack detection task realistic, there are many attacks that did not appear during the collection phase of training data. Attacks fall into four main categories according to their characteristic: DOS, U2R, R2L, Probe.

## Data Preprocess:

- A preprocessing step is to transform the input data into a matrix format that can be vectorized. The common operations include data sampling, data cleansing and data dimensionality reduction. The processing step is the same during the training and testing phase.

## Classification of attacks:

- In this module we will use the models which gives more accuracy to classify the types of attacks in Energy Management Systems in Electric Vehicles.

## Simulations:

In this model we are simulating the road with four vehicles which are collecting communication data and monitoring the road network . Our results shows the graph differences of attack communication and normal communication of EVs in energy uasage, braking, etc.

# CHAPTER 8

# RESULTS

## Adaboost

In [24]:
```
from sklearn.ensemble import AdaBoostClassifier

adaboost_classifier = AdaBoostClassifier(
    n_estimators=10,
    learning_rate=1.0,
    random_state=0
)

adaboost_classifier.fit(X_train, y_train)
```

Out[24]:
```
AdaBoostClassifier(n_estimators=10, random_state=0)
```

In [25]:
```
evaluate(adaboost_classifier,X_train, X_test, y_train, y_test)
```

```
TRAINIG RESULTS:
================================
CONFUSION MATRIX:
[[  77  305]
 [   8 7610]]
ACCURACY SCORE:
0.9609
CLASSIFICATION REPORT:
                     0           1  accuracy    macro avg  weighted avg
precision     0.905882    0.961466  0.960875     0.933674      0.958811
recall        0.201571    0.998950  0.960875     0.600260      0.960875
f1-score      0.329764    0.979849  0.960875     0.654807      0.948808
support     382.000000  7618.000000  0.960875  8000.000000   8000.000000
TESTING RESULTS:
================================


 TRAINIG RESULTS:
 ================================
 CONFUSION MATRIX:
 [[  77  305]
  [   8 7610]]
 ACCURACY SCORE:
 0.9609
 CLASSIFICATION REPORT:
                      0           1  accuracy    macro avg  weighted avg
 precision     0.905882    0.961466  0.960875     0.933674      0.958811
 recall        0.201571    0.998950  0.960875     0.600260      0.960875
 f1-score      0.329764    0.979849  0.960875     0.654807      0.948808
 support     382.000000  7618.000000  0.960875  8000.000000   8000.000000
 TESTING RESULTS:
 ================================
 CONFUSION MATRIX:
 [[  17   88]
  [   3 1892]]
 ACCURACY SCORE:
 0.9545
 CLASSIFICATION REPORT:
                      0           1  accuracy    macro avg  weighted avg
 precision     0.850000    0.955556    0.9545     0.902778      0.950014
 recall        0.161905    0.998417    0.9545     0.580161      0.954500
 f1-score      0.272000    0.976516    0.9545     0.624258      0.939529
 support     105.000000  1895.000000    0.9545  2000.000000   2000.000000
```

In [26]:
```
y_pred = adaboost_classifier.predict(X_test)
a2 = accuracy_score(y_test, y_pred) * 100
p2 = precision_score(y_test,y_pred,average='macro') * 100
r2 = recall_score(y_test,y_pred,average='macro') * 100
f2 = f1_score(y_test,y_pred,average='macro') * 100
```

Fig.8.1: Result of AdaBoost

## Random Forest

In [27]:
```
from sklearn.ensemble import RandomForestClassifier

random_forest_classifier = RandomForestClassifier(
    n_estimators=100,
    random_state=0
)


random_forest_classifier.fit(X_train, y_train)
```

Out[27]:
```
RandomForestClassifier(random_state=0)
```

In [28]:
```
evaluate(random_forest_classifier,X_train, X_test, y_train, y_test)
```

```
TRAINIG RESULTS:
===============================
CONFUSION MATRIX:
[[ 382    0]
 [   0 7618]]
ACCURACY SCORE:
1.0000
CLASSIFICATION REPORT:
              0       1  accuracy  macro avg  weighted avg
precision   1.0     1.0       1.0        1.0           1.0
recall      1.0     1.0       1.0        1.0           1.0
f1-score    1.0     1.0       1.0        1.0           1.0
support   382.0  7618.0       1.0     8000.0        8000.0
TESTING RESULTS:
===============================
 ==============================
CONFUSION MATRIX:
[[   4  101]
 [   0 1895]]
ACCURACY SCORE:
0.9495
CLASSIFICATION REPORT:
                   0           1  accuracy    macro avg  weighted avg
precision   1.000000    0.949399    0.9495     0.974699      0.952055
recall      0.038095    1.000000    0.9495     0.519048      0.949500
f1-score    0.073394    0.974043    0.9495     0.523719      0.926759
support   105.000000  1895.000000    0.9495  2000.000000   2000.000000
```

In [29]:
```
y_pred = adaboost_classifier.predict(X_test)
a3 = accuracy_score(y_test, y_pred) * 100
p3 = precision_score(y_test,y_pred,average='macro') * 100
r3 = recall_score(y_test,y_pred,average='macro') * 100
f3 = f1_score(y_test,y_pred,average='macro') * 100
```

Fig.8.2: Result of Random Forest

# CHAPTER 9

# SNAPSHOTS

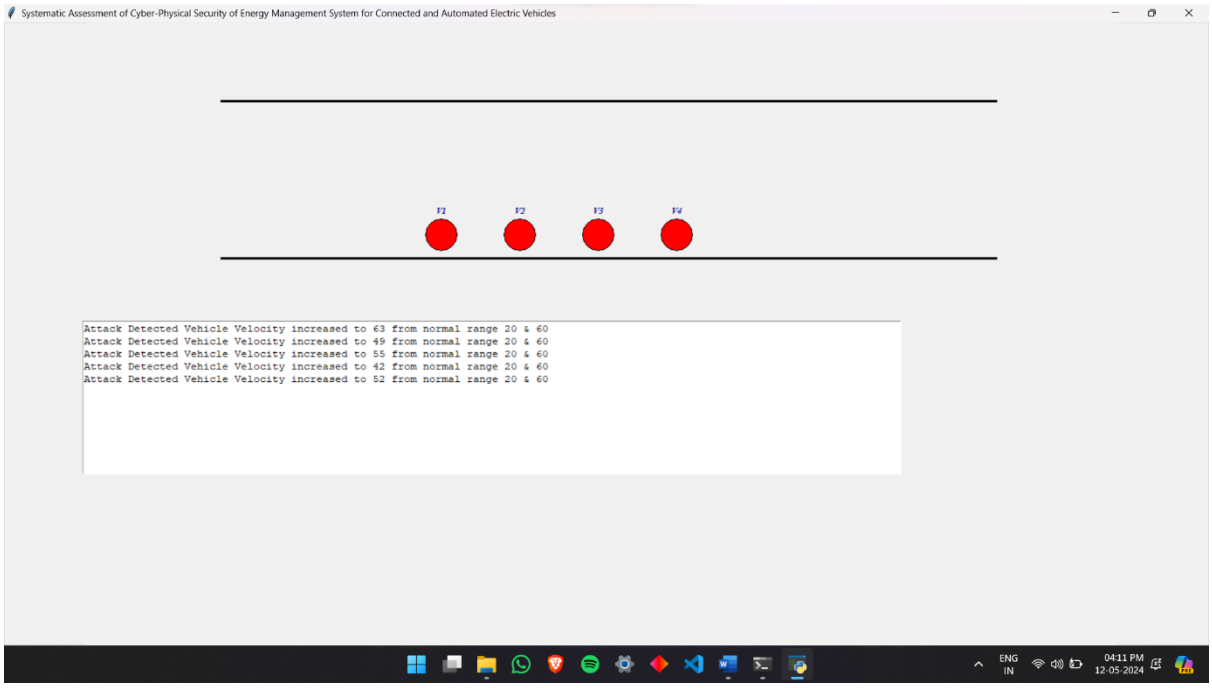


**Figure:.9.1: Execution**.

In above screen black lines indicate road and the red balls indicates vehicles, then vehicles will start moving by receiving commands from ECU

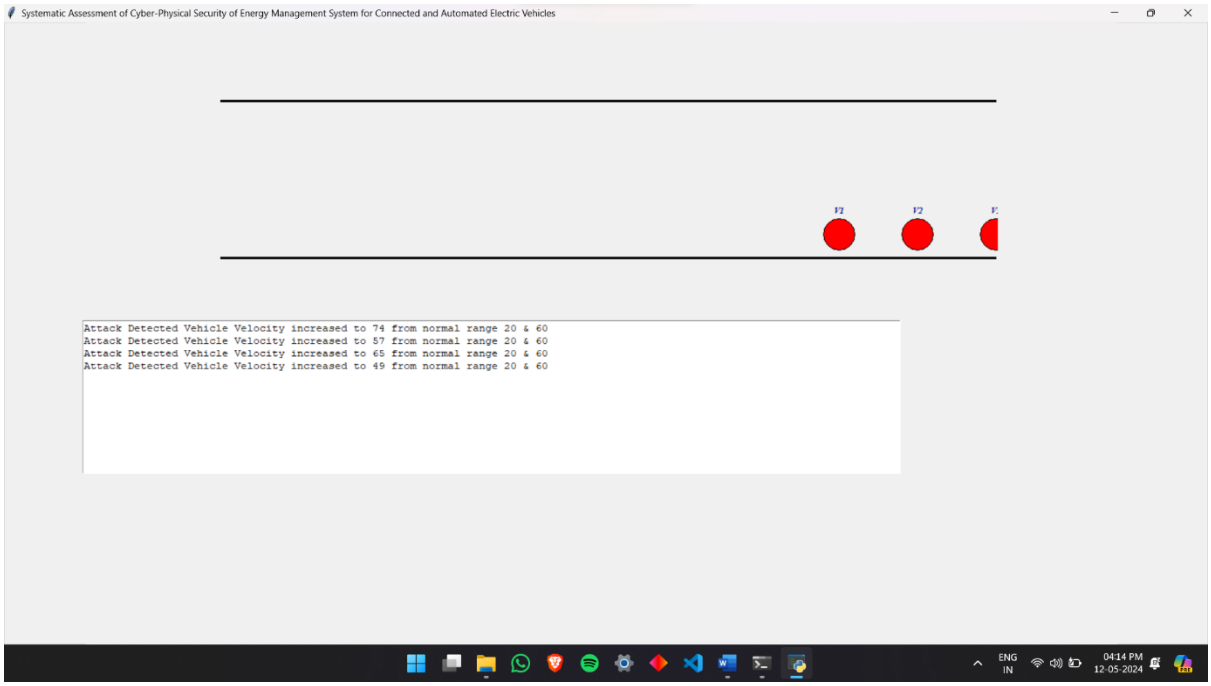


**Figure:9.2: Execution**

In above screen in text area we are displaying abnormal command received by vehicles based on velocity. If there is sudden change in velocity from normal range then attack will be detected.

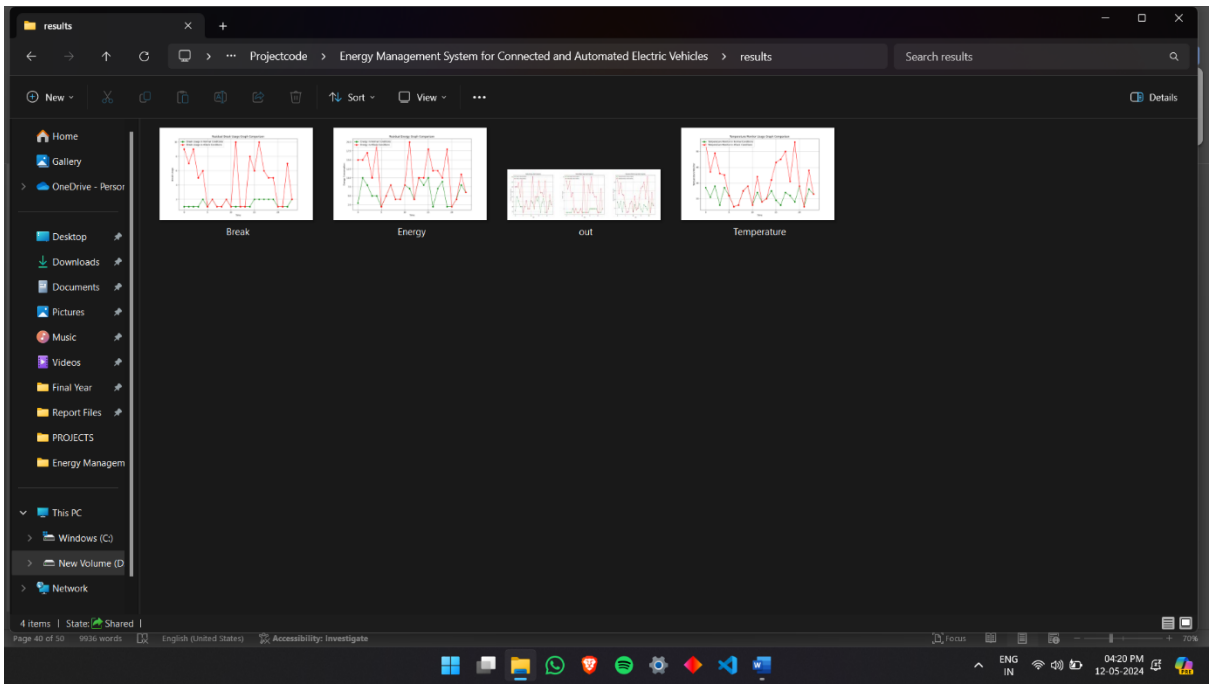


**Figure:9.3: A separate folder to Store the Graph Obtained**

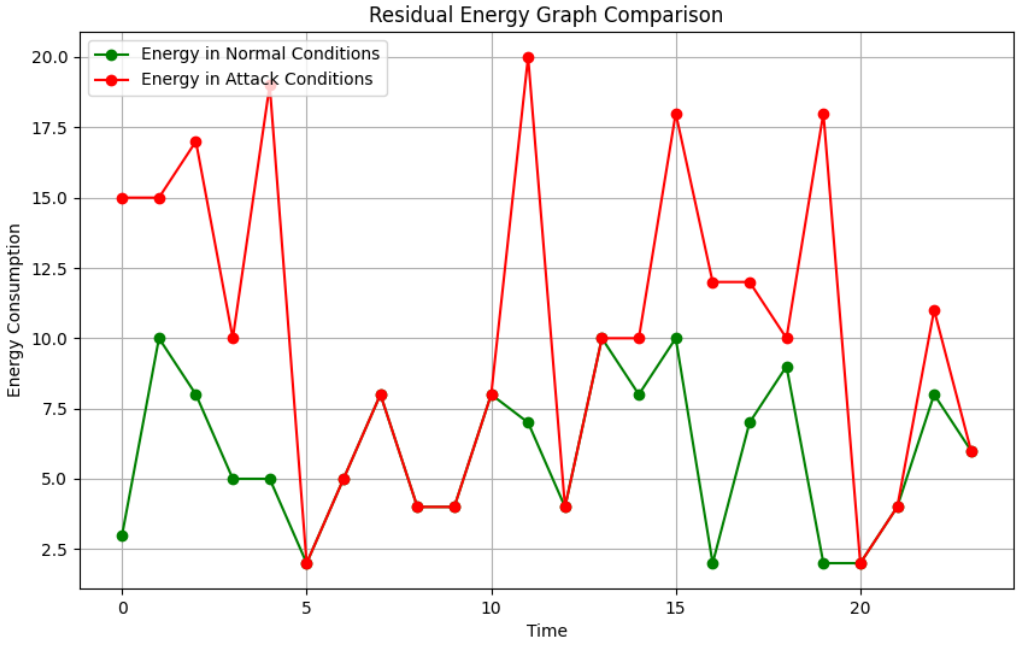


**Figure:9.4: Energy Graph**

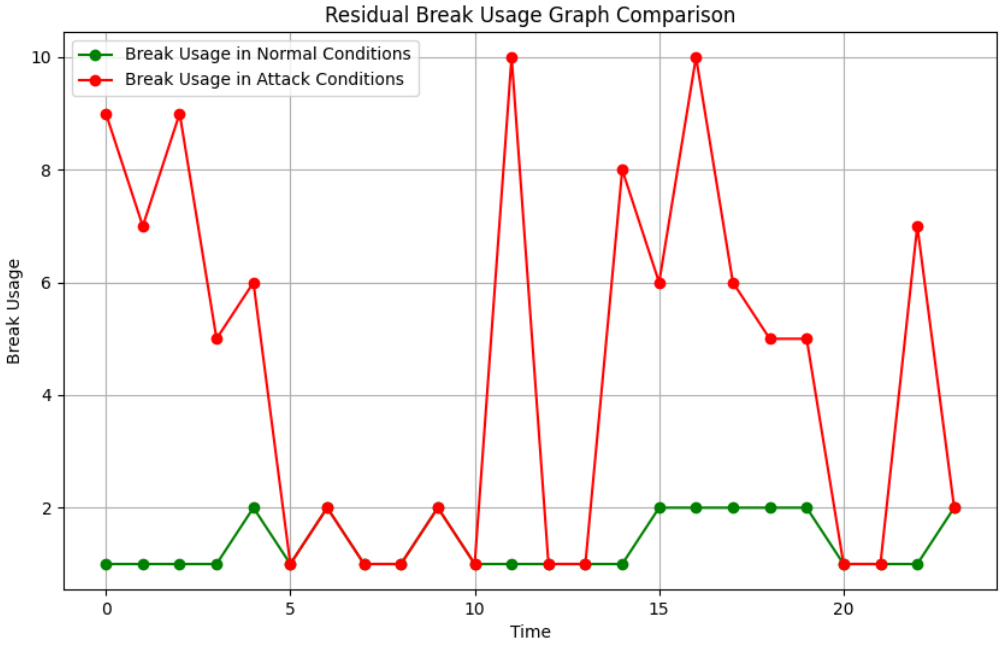


**Figure:9.5: Break Graph**

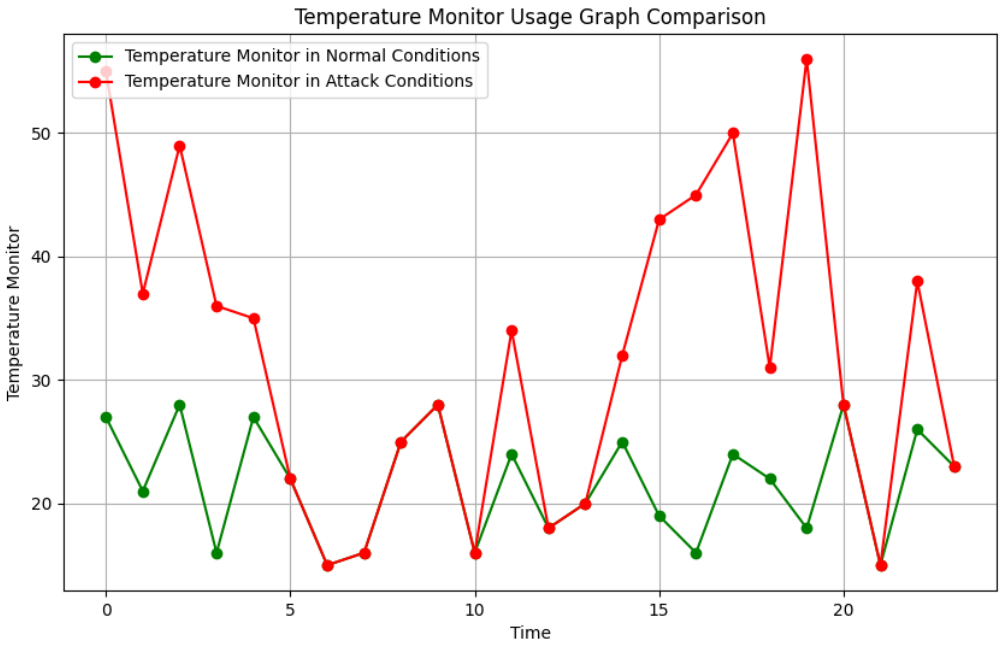


**Figure:9.6: Temperature Graph**

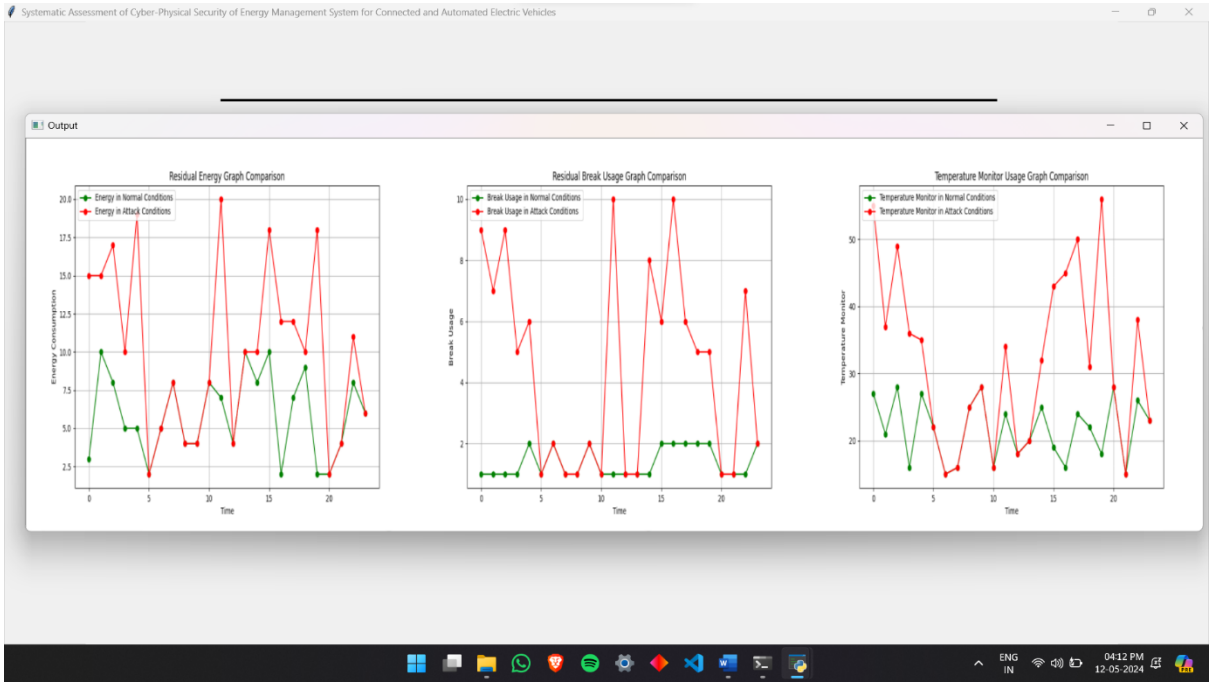


**Figure:9.7: Output**

# CHAPTER 10

# SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## 10.1 TYPES OF TESTS

### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

## System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

## White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

## Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests,

must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

## Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

## Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

## Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

## Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

## Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## 10.2 Test cases:

**Test Case 1:**

| | |
|---|---|
| **Test Case#** | UTC01 |
| **Test Name** | Network simulation |
| **Test Description** | To test network is simulating correctly or not |
| **Input** | No. of vehicles |
| **Expected Output** | System shows in the simulation as per the no. of vehicle |
| **Actual Output** | System display accordingly |
| **Test Result** | Success |

**Test Case 2:**

| | |
|---|---|
| **Test Case#** | UTC02 |
| **Test Name** | Network simulation |
| **Test Description** | To test network is simulating correctly or not |
| **Input** | No. of vehicles, energy consumptions as zero |
| **Expected Output** | Show the error |
| **Actual Output** | Shown error |
| **Test Result** | Success |

**Test Case 3:**

| | |
|---|---|
| **Test Case#** | UTC03 |
| **Test Name** | Prediction of Intrusion |
| **Test Description** | To test whether predicting network Intrusion or not? |
| **Input** | Network values |
| **Expected Output** | It Should predict Intrusion |
| **Actual Output** | Predicted intrusion as per the trained data |
| **Test Result** | Success |

**Test Case 4:**

| | |
|---|---|
| **Test Case#** | UTC04 |
| **Test Name** | Test case for importing valid python libraries |
| **Test Description** | To test whether an algorithm to implement congestion nodes works without sklearn and keras models |
| **Input** | Import all valid libraries sklearn, tkinter and keras libraries |
| **Expected Output** | An error should be thrown specifying "error importing libraries sklearn, tkinter and keras libraries" |
| **Actual Output** | An error is thrown |
| **Test Result** | Success |

# CHAPTER 11

# CONCLUSION AND FUTURE ENHANCEMENT

## CONCLUSION

This project provided a general guidance for vulnerability assessment of core control systems for CAEVs, with which the impact of cyber-attacks on different critical systems and signals, as well as the interaction between these subsystems can be analyzed comprehensively. As a case study, we have developed an MPC-based EMS for CAEVs with four in-wheel motors and presented a systematic vulnerability assessment on cyber-threats. Then, innovative index-based evaluation metrics in terms of dynamic performance, comfortability, energy, and system security and resilience are established to evaluate the critical performance. Following, we give a few remarks on practical applications and future works. In the project, we have demonstrated that an attacker can degenerate the overall performance of the vehicle through data integrity attacks, e.g., higher velocity tracking error and torque ripples, lower energy efficiency, and even instability. For the developed MPC-based EMS, the results have shown that all of the evaluation metrics, including the proposed indices of recovery time and resilience, can reflect the impact of various cyber-attacks. Then, by using these metrics, one can develop data-based or model-based detection and diagnosis approaches in practical applications. Also, the statistical results can help to identify the critical signals, so that they pay more attention to it when designing a system.

It should be noted that besides the detailed impact analysis of cyber-threats on EMS, this paper provides a general framework of vulnerability assessment of a control system in the ECU (from the control perspective). For other systems, e.g., safety system and advanced driver assistance systems in Fig. 1, one needs to conduct a detailed impact analysis by using the potential signal inputs and objectives stated in Section II (under a variety of cyber-physical attacks according to specific demands). Particularly, for those learning-based systems, e.g., a pedestrian detection system in deep learning approaches in rough weather, although vulnerability assessment can be addressed by designing various cyber-physical attacks and evaluation metrics as described in the paper, because of the unique algorithm structure compared to traditional control methodologies, further research is needed.

# FUTURE ENCHANCEMENT:

vehicles In future enhancements of the cyber-physical security of energy management systems for connected and automated electric vehicles, there's a trajectory towards more sophisticated measures. Advanced encryption and authentication mechanisms are pivotal, ensuring secure communication among vehicles, charging stations, and the energy management system. Implementing quantum-resistant cryptography or blockchain-based solutions could fortify defenses against evolving cyber threats. Moreover, integrating machine learning for behavioral anomaly detection offers proactive surveillance, swiftly identifying and neutralizing potential breaches. Secure Over-The-Air (OTA) updates are imperative, guaranteeing the integrity and confidentiality of software and firmware updates. Building resilient cyber-physical infrastructure with redundancy and failover mechanisms strengthens the system's capacity to withstand attacks. Privacy-preserving techniques like differential privacy maintain data utility while safeguarding user privacy. Collaborative security frameworks foster information sharing and coordinated responses to emerging threats. Additionally, ongoing cybersecurity training ensures stakeholders are equipped to recognize and address security incidents effectively. These enhancements, coupled with regulatory compliance and ethical considerations, pave the way for a more secure and responsible future for connected electric.

# REFERENCES

[1] J. K. Naufal, J. B. Camargo, L. F. Vismari, J. R. de Almeida, C. Molina, R. I. R. Gonzalez, R. Inam, and E. Fersman, "A ´ 2 CPS: A vehicle-centric safety conceptual framework for autonomous transport systems," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 6, pp. 1925–1939, 2017.

[2] M. Pajic, J. Weimer, N. Bezzo, O. Sokolsky, G. J. Pappas, and I. Lee, "Design and implementation of attack-resilient cyberphysical systems: With a focus on attack-resilient state estimators," IEEE Control Systems Magazine, vol. 37, no. 2, pp. 66–81, 2017.

[3] P. Guo, H. Kim, L. Guan, M. Zhu, and P. Liu, "Vcids: Collaborative intrusion detection of sensor and actuator attacks on connected vehicles," in 2017 International Conference on Security and Privacy in Communication Systems. Springer, 2017, pp. 377–396.

[4] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham et al., "Experimental security analysis of a modern automobile," in 2010 IEEE Symposium on Security and Privacy. IEEE, 2010, pp. 447–462.

[5] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno et al., "Comprehensive experimental analyses of automotive attack surfaces." in 2011 USENIX Security Symposium. San Francisco, 2011, pp. 447–462.

[6] C. Valasek and C. Miller, "Adventures in automotive networks and control units," Technical White Paper, IOActive, 2014.

[7] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in 2013 International Workshop on Cryptographic Hardware and Embedded Systems. Springer, 2013, pp. 55–72.

[8] N. O. Tippenhauer, C. Popper, K. B. Rasmussen, and S. Capkun, "On ¨ the requirements for successful GPS spoofing attacks," in 2011 ACM Symposium on Computer and communications security. ACM, 2011, pp. 75–86.

[9] T. Zhang, H. Antunes, and S. Aggarwal, "Defending connected vehicles against malware: Challenges and a solution framework," IEEE Internet of Things journal, vol. 1, no. 1, pp. 10–21, 2014.

[10] D. Wise, "Vehicle cybersecurity dot and industry have efforts under way, but dot needs to define its role in responding to a real-world attack," Gao Reports. US Government Accountability Office, 2016.