A System Architecture Design: Integrating Random Forest, Natural Language Processing, and Internet of Things to Predict Technical Carnapping in the Philippines

Dr. Jonell V. Ocampo
Assistant Professor: Department of Industrial and Information Technology
Cavite State University – Carmona
Carmona City, Philippines
e-mail: jonell.ocampo@cvsu.edu.ph

Dr. Jonathan M. Caballero
Associate Professor: Computer Science Department, College of Science
Technological University of the Philippines – Manila
Manila, Philippines
e-mail: jonathan_caballero@tup.edu.ph

Abstract— Technical carnapping, or *rent-tangay*, is a new deceptive scheme of stealing a car by virtue of a rental contract. Typical car anti-theft systems can only provide the location and status of the car. This means that by the time the vehicle is stolen, it is often too late. Even with the use of technology, this makes it hard for car owners and operators to prevent this new illegal scheme. This study features an architecture design for a car anti-theft system that integrates the use of natural language processing (NLP), random forest (RF) model, and internet of things (IoT) in predicting technical carnapping or rent-tangay in the Philippines. It highlights three major components, which are the black box or the hardware module, the mobile application, and the website application. The black box is responsible for gathering data inputs, including geographical location, recorded audio, and sensor outputs. The NLP pipeline is responsible for mining and processing text-based data from the audio recording. Whereas the RF model is responsible for scoring all of the inputs and using them to predict technical carnapping. The model developed in the study scored 100% in recall, 96.30% in accuracy, and 85.71% in F1 score. This implies the success and effectiveness of the design in predicting technical carnapping. This achievement significantly contributes to the body of work that focuses on developing security systems for cars, especially by effectively and efficiently implementing NLP and machine learning to the system. The study pushes the technological boundaries that can be explored in designing and developing car security systems.

Keywords-Natural Languange Processing; Random Forest Algorithm; Internet of Things; Car Anti-theft System

I. Introduction

Globally, vehicle theft is an issue [1]. The criminal act of driving away a car without the consent of the owner [2] rises per year [1, 3, 4]. According to statistics, a vehicle is stolen every 23 seconds [2]. In addition, in 2021 alone, 35,000 cars were reportedly stolen [5]. These incidents typically happen in parking lots [6]. At the point where a car is stolen and driven away, a crucial battle against time happens to recover it [7, 8]. As time passes after the theft, the chance of recovering the car becomes less. The stolen cars are often disassembled or sold if not recovered immediately [7]. A simple installation of an anti-theft system in a car, such as a global positioning system (GPS) device, makes a significant difference in recovering a car after theft [8]. The exploitation of advanced technologies as anti-theft and security systems plays a vital role in dissuading thieves from committing their acts [9].

A range of sophisticated technological innovations are already used to fight nefarious acts toward vehicles. Researchers and developers often use microcontrollers as the main processing unit of anti-theft systems, including ESP32 [3], Arduino [8], and Raspberry Pi [10]. These devices are capable of integrating different analog and digital components, taking the inputs, processing them, and producing the desired outputs. Common car anti-theft systems have global systems for mobile communications (GSM) that enable them to communicate with remote devices [1, 3, 4]. GSM is often integrated with GPS for tracking and short message service (SMS) features for notification and response purposes [1, 3, 4]. This GSM, GPS, and SMS setup allows the security device to have a car immobilization feature. With a predefined command that is sent via SMS, it can trigger an active component and stop the engine of a stolen vehicle [1, 3, 4]. In terms of early detection of theft, the internet of things (IoT) is widely used to monitor input signals coming from different sensors and input components, including cameras, scanners, and limit switches. This allows owners to detect unusual movements and monitor their vehicles remotely [5, 8, 10]. Innovators also found a way to enable authentication and fast identification of car owners. This is done by using advanced components such as radio frequency identification (RFID) scanners, fingerprint scanners, and face recognition [6, 8, 9, 10]. The concept of identifying the face of the driver and cross-referencing it with the plate number and the actual vehicle is implemented in private parking spaces. When a hit is detected, an alarm will be activated, and the exit barrier of the parking space will not open [2, 6, 9]. This prevents criminals from carjacking a car and driving it away from parking spaces. It is an advanced technological feat that utilizes machine learning (ML) in recognizing and cross-referencing images [6, 9]. In a more unusual approach to countering car theft, blockchain technology and near-field communication (NFC) were used to quickly find and recover stolen cars [7]. Fig. 1 shows a typical design of a car anti-theft system. It contains the basic components of a car security system, which include input devices, a processing unit, a communication module, and output devices.

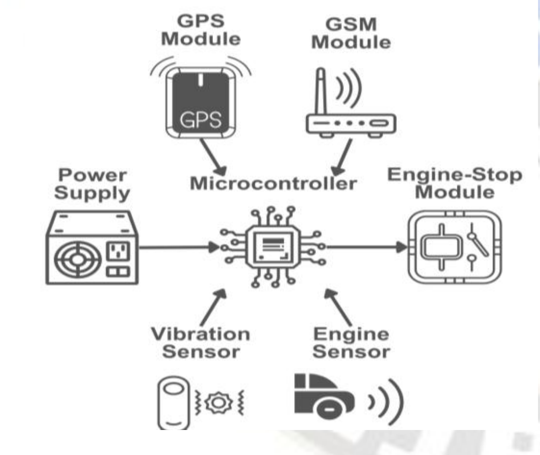

Figure 1. A typical design of a car anti-theft system.

The design in Fig. 1 highlights the use of a microcontroller as the processing unit of the anti-theft system. It also has a GPS module for tracking and a GSM module for communication purposes. The vibration sensor and engine-start detector are used in the design to monitor the status of the car. If an unusual movement happens and the car starts, a notification will be sent to the owner via SMS. When the car is driven away from an initial point, it will be verified and tracked using the GPS module, and an alarm will be triggered. The owner can send a message to the car's anti-theft system, and the engine of the car will stop. The entire system is powered by a power supply setup. Although this security system may work fine against common cases of car theft, it will not be enough to stop other forms of vehicle thievery, such as technical car-napping, *rent-tangay,* and the *rent-sangla or benta scheme* [11].

The *rent-tangay* scheme happens when an individual steals a vehicle by virtue of a rental contract or intentionally does not return the vehicle after the stated rent duration in the contract [11]. Whereas the *rent-sangla or benta* scheme is the act of selling or pawning a vehicle that is stolen by virtue of a rental contract. All of these schemes are executed without the knowledge of the legal owner or lessor of the vehicles [11].

Since a typical car anti-theft system is not sufficient to counter other forms of vehicle thievery, this study highlights a system architecture design that aims to predict technical car-napping, *rent-tangay*, or *rent-sangla/benta* schemes. This anti-theft system is capable of converting audio signals into text, analyzing them, and detecting predefined words related to car theft. Further, inputs coming from various sensors will be processed and scored together with the analyzed audio recording. If the total score of the inputs reaches the threshold, it will send an alarm to the owner, and a trigger to stop the engine of the vehicle can be activated. The system is a combination of hardware and software technologies. The hardware is comprised of Orange Pi, Arduino, Wi-Fi dongle, GPS tracking system, audio receiver, power cut sensor, detachment sensor, kill-engine module, power supply, and battery. On the other hand, the software is composed of mobile and website applications. It is the integration of natural language processing (NLP), random forest (RF), and Internet of Things (IoT) into a single system.

II. Related Work

Technological developments that aim to fight vehicle theft can be categorized into two. The first are devices or mechanisms installed in the vehicle; these are used to detect intrusions or forced entry, unusual movements, and displacement of the car [1, 3, 4]. The second are systems installed on stations such as parking lots or checkpoints, which are primarily used in the search and recovery of the stolen cars [2, 6, 9]. This section of the paper will only discuss works that focus on the first category because of its direct relevance to the study.

The work in [1] features a cost-efficient concealed device installed in the car. It has an authentication feature that requires a password to be entered before the car can start. If the password entered is correct, the device will allow the ignition of the engine. If the password entered is incorrect, the device will send an alert message to the owner of the car via the SMS feature. The owner can then send a message to the device to jam the vehicle and prohibit the start of the engine. At the same time, this event will trigger the device to send the location of the car every 30 seconds using the GPS and GSM modules until it is deactivated.

The main circuit board of the device where the controller unit is installed is customized and built with cost-efficient materials that are available in the market. As for the recommendation, the authors stated that other controllers can be explored to make the device more dynamic [1].

The concept and the use of modules in [1] is supported in [4], where a car anti-system was built using a controller unit, GPS module, GSM module, limit switches, and relays. The work in [4] does not have authentication features, and the initiation of the workflow happens when the owner becomes aware of the stolen car. The owner can send a message to the device installed in the car via the SMS feature to get its location using the GPS module. The device will then reply with a message to the owner containing its location using the longitude and latitude. These coordinates can be inputted in Google Maps so the vehicle can be located. Unlike the work in [1], the tracking feature of the work in [4] is initiated manually. Another difference is that the work in [4] can control the doors of the car aside from its engine and send alert messages to the concerned authority. This is done by programming in their contact information to the GSM module before deploying the anti-theft system.

Compared to the works in [1, 4], which utilize GSM for its communication feature, the work in [5] revolutionizes the method by using the IoT. In addition, the work in [5] features another device aside from the one installed in the car. The two devices are a transmitter and a receiver, which are both using Zigbee technology to communicate with each other. A power supply, Arduino Uno, and an LCD module are also present in both the transmitter and receiver. The combination of these two devices acts as a detector of stolen cars. When a car is stolen and driven away, the receiver acts as a tracker by detecting the signal coming from the transmitter. If the stolen car is detected, the system can be initiated to stop the movement of the car. The details of the car as well as the incident can be seen on the website or IoT dashboard of the anti-theft system.

The work in [8] acts as a car tracker with a three-level authentication feature. Similar to the work in [5], it utilizes IoT with the help of a Wi-Fi module. It is also similar to the work in [1] that requires a password before the engine can start; the difference is the addition of another two authentication features, which are the RFID and fingerprint scanners. The three-level authentication feature starts with the RFID scanner. If the RFID tag scanning attempt is successful, the driver needs to scan his or her fingerprint for the second level. The last authentication feature requires the input of a password on the keypad matrix. If the system detects three failed attempts in any of the security levels, it will automatically send a notification to the owner via a mobile application indicating that someone is trying to access the car. Along with the notification, the car anti-theft system will also send its location with the help of the GPS module. The mobile application, interfaced with Google Maps, will show the location of the vehicle on the map.

The work in [3] integrates the features of the works in [1, 4, 5, 8] in one car anti-theft system. It utilizes IoT for communication purposes just like the work in [8]. In addition, it uses the combination of GSM and GPS modules to send the location of the vehicle to the owner via SMS [1, 4, 8]. If a car theft happens, the owner can control the engine of the car by sending an SMS device to the device [1, 4, 5]. The device can also sound an alarm to indicate an emergency and cause panic to the thieves. The system uses an ESP32 microcontroller to process all the input, interface the components, and control the output. It is powered by a regulated 9-volt battery.

Compared to the other works previously discussed, the work in [10] highlights advanced features such as face detection and face recognition. The car anti-theft system in the study uses Raspberry Pi as its main processing unit. A Raspberry Pi camera is used to take images of the driver. A GSM module is used for communication purposes, and a GPS module is used to identify the location of the vehicle, just like in [1, 4, 8]. The face detection is done with the help of a library that contains the Haar Cascade method. On the other hand, the face recognition is done using the local binary pattern histogram technique. If the system detects an unknown driver, it will send the image of the driver and the location of the car to the Telegram application that is interfaced to the system through the Telegram application programming interface (API). This aims to alert the owner to be able to take precautionary measures.

III. RELATED LITERATURE

This section discusses the studies related to NLP and the studies considered in using random forest (RF) as the artificial intelligence (AI) model in the study.

The language used by humans is ambiguous, contextual, full of slang, and guided by complex grammatical rules. Without utilizing a proper method, it would be extremely difficult for a machine to comprehend and process human language [12]. The car anti-theft system featured in this study used NLP as one of its major features because it enables the car anti-theft system to understand and evaluate human language effectively. NLP is a method capable of processing data coming from speech or text. Specifically, it can detect, extract, and categorize data coming from unstructured sources. Ultimately, it makes the data understandable to machines, allowing it to be subjected to further processing. NLP may involve several processes depending on its application. It includes a method called tokenization, which is a process that splits text-based data into different units, such as words, phrases, or sentences [13]. Another NLP method is called part-of-speech tagging. This is where the type of the words in the data is identified; the type may include adjectives, verbs, and nouns. Similarly, NER identifies the type of the words, but instead of part of speech, it detects dates, places, and names in the data [14]. To make the process faster, the texts in the data are reduced to their base forms in a process called stemming [15]. Since human language structure is guided by grammar rules, these are detected in a method called parsing [14]. Adding to the overall efficiency of NLP, a method called stop-words removal is conducted to eliminate words that add little to no meaning to the texts. After cleaning and preprocessing the data, it undergoes vectorization that converts it to numerical code, which can be easily processed by a machine [15]. One application of NLP is highlighted in the work in [16], where the method is used to predict financial risk. In the study, they used text-based data from social media platforms, detailed financial reports, and financial-related news articles. The data were extracted, preprocessed, and used to train and test a factorization model. The study concludes a series of positive results in terms of predicting financial risks. In another study, the authors in [17] used NLP to predict recurrence of breast cancer. The researchers mined text-based data from medical progress notes and patient pathology reports and used it to test and evaluate deep learning models and ML classifiers. The result of the study highlights the significant potential of NLP and ML in predicting the recurrence of breast cancer. The work in [18] is another medical application of NLP. In the study, the technique was used to classify the feeding status of infants. According to the study, almost 1000 digital records of mothers and their infants were used to train and test ML classifiers. The method used a ratio of 70:30 in training to testing data. The NLP method in the study used processes including removal of punctuation, numbers, and stop words. Based on the results, the NLP framework achieved an outstanding accuracy in identifying the feeding status of infants.

The design of the car anti-theft system in the study requires a prediction technique to process the output of the NLP pipeline. This technique should be able to put scores on different parameters that would holistically tell in advance if there will be an incident of *rent-tangay* or not. Amongst different classifier and regression algorithms, RF was used in the study to perform the task. The RF is a type of supervised ML algorithm that can handle relationships within parameters that may seem complex for other learning algorithms [19, 20]. Instead of using only one decision tree, RF uses multiple decision trees constructed by leveraging ensemble learning. The algorithm obtains the prediction mean by combining several decision trees that are formed using random features and samples [21]. The greater the number of decision trees and the deeper they get, the more they reduce overfitting in the technique. This creates a more potent and significant classification and regression tool [22]. According to the work in [23], the RF method includes fundamental steps, such as bootstrap sampling, random feature selection, decision tree construction, and aggregation. These steps are normally conducted chronologically. To make sure that the training data for each tree in the RF are varied, subsets of the original data are randomly selected as replacements in the process called bootstrap sampling. In terms of reducing overfitting and expanding the capacity of the model in handling datasets, a bagging technique is used to consider random subsets of the data features that will be applied for the nodes of each individual decision tree. This technique is called random feature selection. After creating random subsets of the original dataset and features, the decision trees can be created. This is done by splitting the data into subgroups based on a criterion. For the final step, according to the work in [23], the outputs of the decision trees in the model are integrated into one; this is then used to give the final classification or prediction. The work in [24] showcases another approach to implementing the RF algorithm. Based on the study, the process starts by preprocessing the data. Then, it will be divided into training and testing data. Next, features will be extracted, and it will proceed with training the model using the RF algorithm. After the training of the model, it will be tested using the data for testing. The prediction can now be made after this step.

The edge of the RF algorithm over other machine learning algorithms is concluded in various studies. Based on [25], RF was used among other ML algorithms to predict stroke disease. Demographical and behavioral data were used to train and test learning algorithms, including RF, logistic regression, and decision trees. According to the results, RF outperformed the other algorithms by achieving an accuracy and F1 score of 94%. In [19], RF was used to analyze and predict house pricing based on real estate features. Results showed that the RF got the highest accuracy and F1 score against decision tree, support vector machine (SVM), logistic regression (LR), and K-nearest neighbor (KNN) algorithm. Similarly, the work in [22] highlights a comparative analysis of various learning algorithms to identify what is best in predicting soil erosion status. The algorithms used were RF, Naïve Bayes (NB), KNN, LR, SVM, linear discriminant analysis (LDA), and stochastic gradient descent (SGD). Based on the results, RF got the highest score of 97% for accuracy, recall, precision, and F1 score.

Applications of RF include prediction of athlete performance [26], weather [27], heart disease [23], brain stroke [24], and surveying air pollutants [20]. On the other hand, application of the combination of NLP and RF includes fake news detection [13], phishing attack detection [14], breast cancer recurrence prediction [17], and classification of infant feeding status [18].

IV. ARCHITECTURE DESIGN OF THE CAR ANTI-THEFT SYSTEM

The system architecture design in Fig. 2 represents a comprehensive vehicle tracking and control system that integrates multiple components to ensure seamless data flow and effective monitoring.

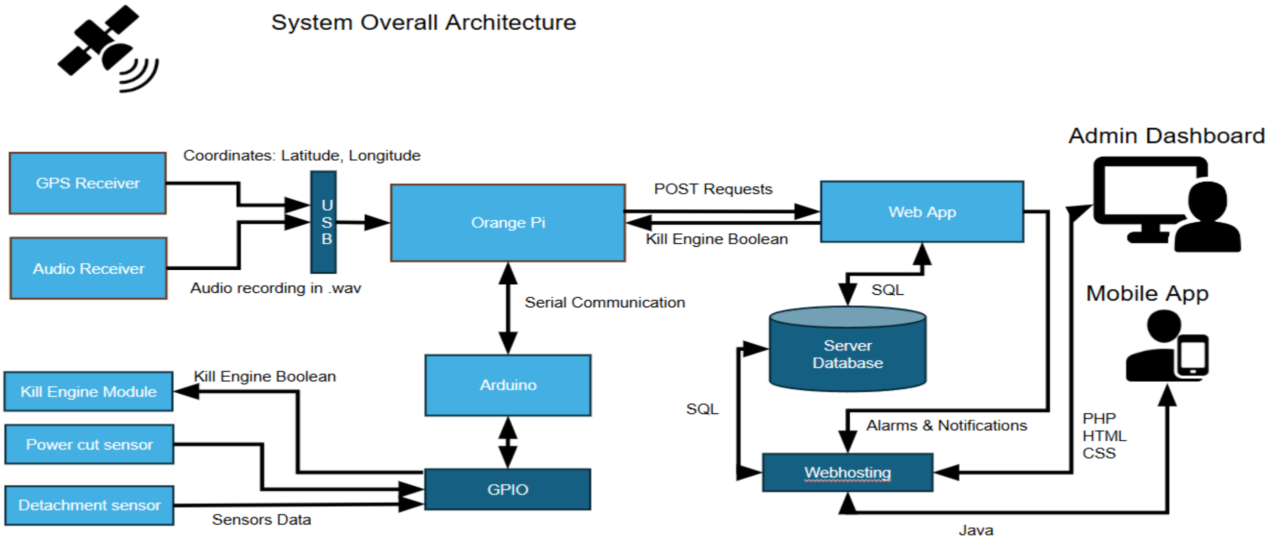

Figure 2. The architecture design of the car anti-theft system in the study.

Among the key components are a web application, a server database, an Orange Pi, an Arduino, a GPS receiver, an audio receiver, and a mobile app. The GPS receiver obtains the vehicle's location, and the audio receiver records sound. Both are sent to the Orange Pi for processing. The Orange Pi acts as the central processing device, receiving data from the GPS and audio receivers and exchanging serial data with the Arduino. The Arduino then controls the kill engine and communicates with a number of sensors using inputs from the Orange Pi.

The final component of the system is the web hosting service, which stores the server database and controls Java-based alert and notification communication. This service sends updates to the admin dashboard and mobile app, which provide control and monitoring user interfaces. While the admin dashboard allows administrators to control the system, the mobile application, which was developed using hypertext preprocessor (PHP), hypertext markup language (HTML), and cascading style sheets (CSS), offers a mobile-friendly interface for users to monitor and control the vehicle tracking system. The entire architecture is designed to ensure that data is efficiently gathered, processed, stored, and sent among the several components, providing users with a responsive and reliable tracking and control system.

A. Mobile Application Architecture

The mobile application architecture in Fig. 3 is intended to guarantee smooth communication between a mobile application and a website, enabling effective vehicle tracking and administration. The WebView engine, at the heart of this architecture, is essential to rendering web content in the mobile application. The WebView engine makes sure that users may interact with the data and features of the web application straight from their mobile devices by transforming online material into a displayable format. By preserving consistency across the online and mobile environments, this integration makes it possible to create a responsive and user-friendly experience.

The mobile application's data control is another key component of the architecture. The WebView engine is in charge of overseeing the data transfer and presentation between the mobile environment and the web application, making sure that all information is correctly transferred and presented. In order to retrieve or update vehicle tracking data, the mobile application must handle POST requests that are delivered to the web server. JavaScript is essential to this process because it allows for real-time updates and dynamic interactions within the mobile application, guaranteeing that users always have access to the most recent data.

The user's surroundings are also considered in the architecture of the mobile application, namely with regard to internet connectivity and user interactions. Clicking buttons and other user inputs are handled by the mobile phone environment. These inputs are essential for interacting with the program and sending commands. It also makes sure the application keeps a steady network connection, which allows it to communicate with the web server continuously. The car tracking system depends on real-time data interchange; therefore, this is essential to its operation. In addition, the file manager in a mobile environment is in charge of any data or files that are kept locally on the device, which enhances the application's overall effectiveness and performance.

Lastly, the architecture emphasizes how JavaScript and Java work together to make it easier for the WebView engine to communicate with the mobile environment. This combination guarantees the application's ability to manage user inputs and process data, among other complex activities, all the while preserving a seamless user experience. The mobile application is an essential part of the vehicle monitoring system since it can provide a strong and dependable performance by utilizing the advantages of both JavaScript and Java. This architecture makes sure that every part of the system functions as a whole, giving consumers a smooth and useful tool for managing and tracking cars.

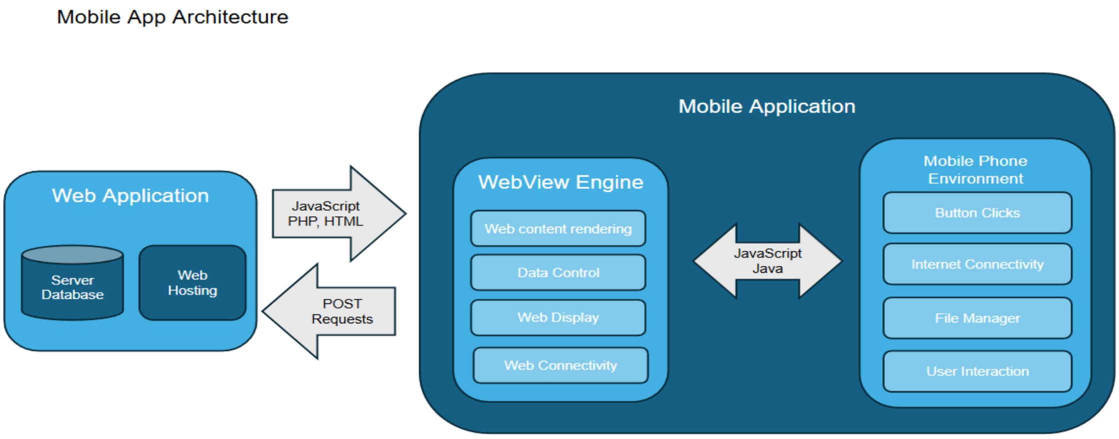

Figure 3. The mobile application architecture used in the study.

B. *Website Application Architecture*

This system architecture in Fig. 4 is sophisticated and integrates several data sources and processing levels. This architecture ensures modularity and scalability by designing each component to perform a specified function. Starting from the beginning, the Orange Pi provides raw data, such as GPS locations, audio recordings, and sensor data, to the data parser, which serves as an entry point for the system. This parser is essential because it separates the data into several streams for additional processing, such as preparing audio recordings for speech-to-text conversion or translating GPS coordinates into position information. The architecture enables real-time data processing through the use of a specialized parser, which is crucial for applications that must respond quickly to changes.

The architecture uses specific modules for data extraction and matching and data scoring (audio analysis) after the parsing stage. To pinpoint crucial moments, these elements take pertinent textual data and compare it to predetermined hot words. This is when the analytical powers of the architecture are useful. Based on the existence of particular keywords, the scoring system assesses the data and can take additional actions, such as issuing warnings or using a decision tree to make judgments. The architecture's capacity to prioritize events and react properly is emphasized by the usage of an audio analysis-based scoring system, which is essential for security-focused applications.

The architecture process includes location data and behavior analysis in addition to textual data analysis. The modules for coordinate-to-location conversion and location analysis evaluate vehicle movements to ascertain driving behavior and convert GPS data into addresses that can be read by humans. Comparably, behavior analysis employs sensor data to track the status of the car, including power outages and unapproved system disconnections. The creation of a thorough understanding of the vehicle's operations—which may subsequently guide decision-making—requires these assessments. These layers are integrated into the architecture to provide constant monitoring of the vehicle's position and physical state, which serves as a strong basis for operational management and security.

The decision-making and communication components, which include the language model, SMS alert application programming interface (API), decision tree, and security algorithm, are the last in the architecture and are in charge of making intelligent decisions based on the data analysis, like sending alerts or commands like the kill engine Boolean to control the vehicle remotely. The data formatter and POST request in JavaScript object notation (JSON) modules, the last stage of the architecture, make sure that the data is appropriately formatted for storage or other actions, such as updating the database or sending commands back to the Orange Pi. This thorough end-to-end sequence, which includes data reception, decision-making, and action, is an excellent example of a well-planned, secure, and responsive web application architecture that can handle intricate interactions in real-time.

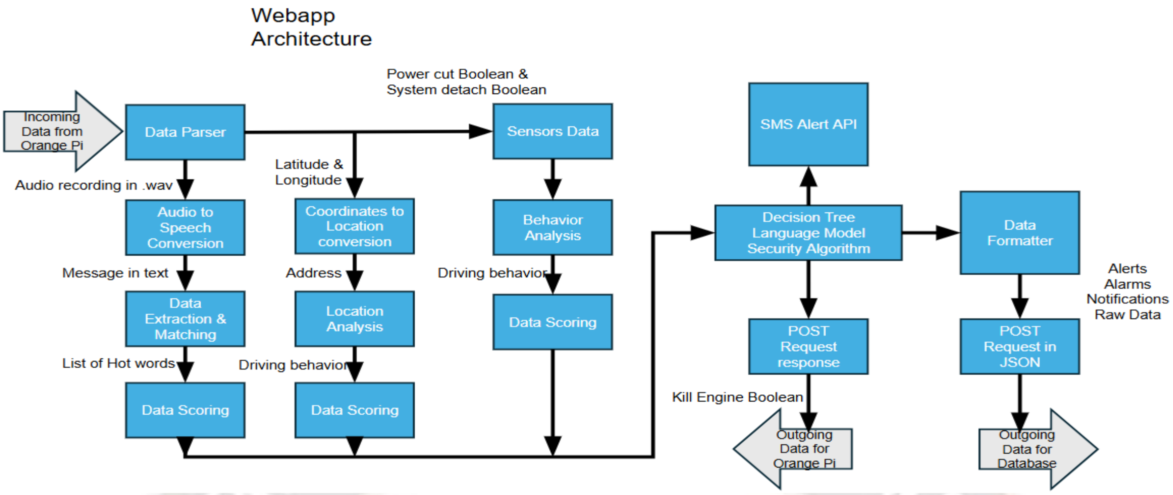

Figure 4. The website application architecture used in the study.

C. Hardware Block Diagram

Fig. 5 shows the block diagram of the hardware components of the study. The solid line represents the supply line of each block, and the broken line refers to the signal flow of the system. The direct current (DC) power supply is required for the single-board computer (SBC) that is present in the vehicle. For the tracking system, a GPS receiver module was used to harness coordinates that will be fed to the SBC. For the speech recognition module, a condenser microphone will be used to capture audio signals inside the vehicle, and they will be transmitted to the SBC. Then, both data sets will be processed and will be displayed in the web and mobile application. A Wi-Fi module is necessary in order for the black box to communicate with the software application. The SBC is also capable of controlling the engine cut-off feature and can send SMS notifications depending on the control signal coming from the software system.

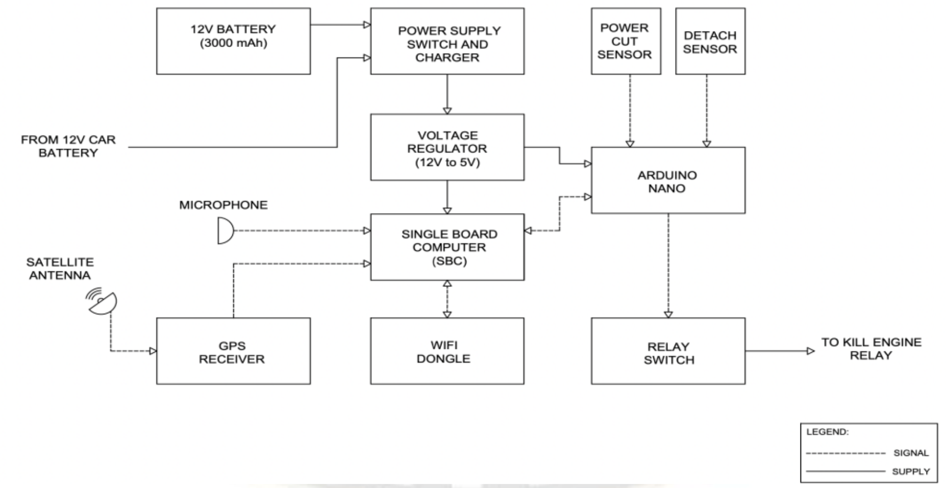

Figure 5. The hardware block diagram used in the study.

V. TESTING RESULTS AND DISCUSSION

This section discusses the results of the tests conducted during the study. The results came from areas including the predictive model testing, confusion matrix, performance matrix, and average word error rate for English and Tagalog conversions.

A. Predictive Model Testing Results

Performance assessment of engine kill predictions in different conditions is crucial for ensuring the reliability and safety of automotive systems. By systematically evaluating how well the model predicts engine shutdown based on various alerts—such as driving outside the destination, vehicle warnings, and power cuts—developers can identify strengths and weaknesses in the predictive algorithm. This analysis not only helps in refining the model to reduce false positives and negatives but also ensures that the system responds accurately under diverse driving scenarios.

In model testing, such performance assessments are vital as they provide insights into the robustness and reliability of predictive algorithms. Understanding the conditions under which the model succeeds or fails allows developers to enhance its accuracy and reduce false positives and negatives.

Table I shows the 30 test cases with corresponding 3 inputs and 2 outputs. The logic input condition represents the location warning, black box warning, and communication warning, while the output is the expected and actual system response of a kill engine status. Data 0 and 1 represent the deactivation or activation of logic input and kill engine response. Table II contains the description of each test case that explains the scenario to identify good cases or a kill engine. "Good case" means that the car is in safe condition and no security alerts are detected, while the "kill engine case" occurs when a very high alert level is detected by the system.

On the other hand, overestimation took place whenever a good case condition was predicted as a kill engine. Based on the result, from 30 cases, 26 of them are correctly predicted as good cases, 3 are predicted as kill engine, and 1 case is an overestimation.

This iterative testing and analysis process is essential for building confidence in the model's deployment in real-world scenarios, ensuring that it can make safe and effective operational decisions in diverse driving environments. Ultimately, thorough model testing not only supports the technical advancement of automotive systems but also fosters trust and safety for end-users.

TABLE I. PERFORMANCE ASSESSMENT OF KILL-ENGINE PREDICTION BASED ON LOGIC INPUTS

No.	LOGIC INPUT CONDITION			KILL ENGINE STATUS		
	Location	*Black box*	*Communi-cation*	*Expected*	*System*	*Remark*
1	0	0	0	0	0	Correct
2	0	1	1	0	0	Correct
3	0	0	0	0	0	Correct
4	1	1	1	1	1	Correct
5	1	0	0	0	0	Correct
6	0	0	0	0	0	Correct
7	0	0	0	0	0	Correct
8	1	1	0	0	0	Correct
9	0	0	0	0	0	Correct
10	0	0	0	0	0	Correct
11	0	0	0	0	0	Correct
12	0	0	1	0	0	Correct
13	0	0	0	0	0	Correct
14	0	0	0	0	0	Correct
15	0	0	1	0	0	Correct
16	0	0	0	0	0	Correct
17	0	0	0	0	0	Correct
18	0	0	0	0	0	Correct
19	0	1	0	0	0	Correct
20	0	0	0	0	0	Correct
21	0	0	0	0	0	Correct
22	1	0	0	0	0	Correct
23	0	0	1	0	0	Correct
24	1	1	1	1	1	Correct
25	0	0	0	0	0	Correct
26	0	0	0	0	0	Correct
27	0	0	0	0	0	Correct
28	1	1	0	0	1	Over
29	0	0	0	0	0	Correct
30	1	1	1	1	1	Correct

a. Legend: 0 = Deactivated, 1 = Activated; Logic Input Threshold=0.7

TABLE II. PERFORMANCE ASSESSMENT SUMMARY OF KILL-ENGINE PREDICTION

No.	Expected Response	System Response	Remarks	Description
1	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
2	0	0	Correct Prediction	Some hot words were mentioned, black box was removed from its original position
3	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode

4	1	1	Correct Prediction	Vehicle is outside geofence, black box was removed and hot words reached its threshold value
5	0	0	Correct Prediction	Vehicle is outside the geofence
6	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
7	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
8	0	0	Correct Prediction	Vehicle is outside the geofence
9	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
10	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
11	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
12	0	0	Correct Prediction	Hot words were detected but doesn’t reach the threshold level
13	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
14	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
15	0	0	Correct Prediction	Hot words reached its threshold level but location and black box warning is inactive
16	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
17	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
18	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
19	0	0	Correct Prediction	Black box was removed from its original position but no location and communication warning
20	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
21	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
22	0	0	Correct Prediction	Vehicle is outside the geofence
23	0	0	Correct Prediction	Hot words reached its threshold level but location and black box warning is inactive
24	1	1	Correct Prediction	Vehicle is outside geofence, black box was removed and hot words reached its threshold value
25	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
26	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
27	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
28	0	1	Over Estimation	Vehicle is outside geofence, black box was removed but hot words doesn’t reach the threshold level
29	0	0	Correct Prediction	No sensors were activated, kill engine module is in standby mode
30	1	1	Correct Prediction	Vehicle is outside geofence, black box is in power-cut status and hot words reached its threshold value

B. Confusion Matrix Results

The confusion matrix in Table III evaluates the performance of a classification model developed to predict vehicle conditions, specifically distinguishing between "kill engine" (positive cases) and "good cases" (negative cases). The dataset utilized for this evaluation comprised 30 scenarios, with 4 predicted as "kill engine" and 26 as "good cases." The classification outcomes were categorized into four essential metrics: true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). These metrics provide a robust foundation for quantifying the model's predictive accuracy and reliability, particularly in safety-critical applications.

The results demonstrate that the model successfully identified all 26 actual "good cases" conditions as True

Negatives, signifying a high level of effectiveness in detecting negative cases and contributing to its strong recall (sensitivity).

For true positives, the model accurately classified 3 out of 3 "kill engine," which highlights its specificity in allowing correct activations. However, the analysis revealed one false positive, wherein the model erroneously classified "good cases" as "kill engine," indicating a potential overestimation of critical conditions.

Importantly, no false negatives were observed, underscoring the model's robustness in ensuring that all valid "kill engine" conditions were accurately predicted without omission. The system's predictive model incorporated a language model, vehicle movement data, and sensor output to determine when issuing a "kill_engine" response was necessary. A dataset of 30 predefined scenarios categorized conditions as: positive (P): valid "kill engine" conditions (3 scenarios). negative (N): valid "good cases" (27 scenarios).

These results underscore the potential of combining machine learning models and language processing to enhance vehicle safety systems, particularly in accurately predicting critical conditions. The system employed a language model alongside vehicle movement data and hardware sensors to determine the necessity of a valid kill_engine response to stop the car. For the evaluation of this model's accuracy, a dataset comprising 30 predetermined scenarios was utilized. Each scenario was carefully labeled to indicate whether a kill_engine output was warranted.

In the dataset, the conditions were categorized as follows: positive (P), indicating valid kill_engine conditions, accounted for 3 scenarios, while negative (N), representing valid good conditions, included 27 scenarios. This classification allowed for a clear assessment of the model's performance in identifying when a kill_engine response is appropriate.

The results, summarized in the accompanying table, illustrate the system's effectiveness in predicting valid conditions for activating the kill_engine. By analyzing these outcomes, we can gain insights into the model's reliability and its potential for real-world application in vehicle safety systems.

These metrics enable a comprehensive understanding of the classification model's strengths and weaknesses, guiding the optimization process for better real-world application. For example, a high specificity ensures the model minimizes false alarms, while a high sensitivity guarantees that positive cases are rarely overlooked. The inclusion of metrics like the F1 score ensures that the model's performance is robust, even when faced with class imbalances. The table's detailed breakdown of formulas and values also aids researchers and practitioners in understanding how each metric is derived, offering transparency and reproducibility in the evaluation process. Such insights are crucial for refining models and ensuring they align with the specific requirements of the intended use case.

TABLE III. CONFUSION MATRIX RESULTS

	Total population 30 cases	**Predicted condition**	
		Predicted Positive (PP)	***Predicted Negative (PN)***
Actual Condition	Positive (P) 3	True Positive (TP) 3	False Negative (FN) 0
	Negative (N) 27	False Positive (FP) 1	True Negative (TN) 26

C. Performance Matrix Summary

Each metric offers a unique perspective on the model's performance. For instance, sensitivity (also known as recall) quantifies the model's ability to correctly identify positive cases, while specificity measures its accuracy in rejecting negative cases. Similarly, precision assesses how often positive predictions are correct, which is especially valuable when false positives have significant consequences.

This table provides a detailed summary of various performance metrics used to evaluate the effectiveness of a classification model. It encompasses essential measures such as sensitivity, specificity, precision, negative predictive value (NPV), and more, alongside their corresponding values and mathematical derivations. These metrics collectively assess the model's predictive capability, accuracy, and balance between identifying positive and negative cases.

Table IV shows the performance of the model having an accuracy rate of 96.67% and 100% sensitivity or recall. Data also shows the specificity of 96.30%, precision rate of 75%, and F1 score of 85.71%. It clearly supports the robustness of the model. However, there is still room for improvement, specifically in its precision rate.

TABLE IV. PERFORMANCE MATRIX SUMMARY

MEASURE	**VALUE**	**PERCENTAGE (%)**	**DERIVATIONS**
1. Sensitivity or Recall	1.0000	100	TPR = TP / (TP + FN)
2. Specificity	0.962962963	96.2962963	SPC = TN / (FP + TN)
3. Precision	0.75000	75	PPV = TP / (TP + FP)
4. Accuracy	0.9666666667	96.66666667	ACC = (TP + TN) / (P + N)
5. F1 Score	0. 8571428571	85.71428571	F1 = 2TP / (2TP + FP + FN)

D. Average Word Error Rate for English Conversation

A reasonable degree of transcription accuracy is shown by the average word error rate (WER) of 29.69% found in the evaluation of the transcription system for English talks in Table V. Substitution mistakes (4.18%), insertion errors (5.13%), and deletion errors (4.05%) are further subdivided into this WER, with insertion errors being the most common. The system's propensity to over-generate text, perhaps misinterpreting voice signals or using superfluous words to fill in gaps, is shown by the high insertion rate. This indicates room for development, especially in terms of managing the transcribed text's length and content to better match the speech input. The system's comparatively low substitution rate suggests that lexical modeling and word recognition procedures are already rather successful in identifying appropriate vocabulary in spite of these difficulties.

When compared to other studies on automated speech recognition (ASR) for English, the current WER reflects promising performance, but there is clear room for further optimization. The complexities of English conversations—ranging from varied accents and overlapping dialogue to homophones and colloquial expressions—continue to challenge the transcription system's robustness. Addressing these factors will require enhancements in acoustic modeling, incorporation of more domain-specific language models, and training on larger and more diverse datasets. Additionally, the adoption of context-aware processing techniques and improved handling of speaker variability could further enhance accuracy, making the system more suitable for real-time or high-stakes applications in English transcription.

TABLE V. AVERAGE WORD ERROR RATE (WER) FOR ENGLISH CONVERSATION

Number of files tested	Transcription with substitution, insertion and deletion			Word Error Rate (%)
	Substitution	*Insertion*	*Deletion*	
137	4.18	5.13	4.05	29.69

E. Average Word Error Rate for Tagalog Conversation

In Table VI, when compared to English transcription, surprisingly the transcription system's accuracy for Tagalog conversation was higher, with an average WER of 3.61%. Further demonstrating the model's performance of understanding Tagalog speech are the system's average substitution rate of 2.59, insertion rate of 3.56, and deletion rate of 1.67 from 61 files that were tested. This supports that the NLP model used in the system is effective and reliable.

TABLE VI. AVERAGE WORD ERROR RATE (WER) FOR TAGALOG CONVERSATION

Number of files tested	Transcription with substitution, insertion and deletion			Word Error Rate (%)
	Substitution	*Insertion*	*Deletion*	
61	2.59	3.56	1.67	3.61

VI. CONCLUSION AND FUTURE WORK

The study features an architecture design for a car anti-theft system, which utilizes NLP and RF methods in predicting technical car-napping or rent-tangay in the Philippines. The main components of the system are the hardware module, mobile application, and website. These three are well integrated into the entire system, sending and receiving data from each other. The hardware module, or the black box, successfully receives sensor outputs from the car, which include GPS location, recorded audio inside the vehicle, and the black box's power and placement status. The NLP pipeline effectively extracted and processed text-based data coming from audio recordings. The RF model, mainly used for scoring purposes, excellently scored the variables coming from the black box. The overall score was then used to predict carjacking. The model developed in the study scored 100% in recall, 96.30% in accuracy, and 85.71% in F1 score. This achievement significantly contributes to the body of work that focuses on developing security systems for cars, especially by effectively and efficiently implementing NLP and machine learning to the system. The study pushes the technological boundaries that can be explored in designing and developing car security systems. In the future, the study can be improved by adding other indicators related to car theft to be inputted and scored by the RF model. This can further increase the accuracy of the system in predicting technical carnapping.

REFERENCES

[1] K. Pande, A. Tijare, A. Peshattiwar, and S. Nitnaware, "Remote theft alert with car engine shutdown," Educational Administration: Theory and Practice, Jan. 2024, doi: 10.53555/kuey.v30i5.2181.

[2] E. Thakran, S. Bhattacharya, and S. Verma, "Proposed method for car theft prevention techniques," International Journal of Technical Research & Science, vol. Special, no. Issue3, pp. 108–113, Aug. 2020, doi: 10.30780/specialissue-icaccg2020/022.

[3] N. Mr. M. Yadav, N. Mr. S. Jadhav, N. Mr. A. Pawar, and N. Dr. B. Shinde, "Smart anti-theft system for electric vehicle," International Journal of Advanced Research in Science Communication and Technology, pp. 347–354, May 2024, doi: 10.48175/ijarsct-18655.

[4] N. S. Priyanka and G. R. Nirogi, "Design and development of sms based car engine control system to prevent car theft using GSM and GPS," International Journal of Scientific Engineering and Research (IJSER), vol. 9, no. 8, pp. 27–29, Aug. 2021, doi: 10.70729/se21807152139.

[5] Manonmani, S. Hemanath, G. Unesh, and P. Vijayarangan, "Received signal strength indicator (RSSI) based car theft detection system," Deleted Journal, vol. 91, no. 4, Apr. 2022, doi: 10.37896/pd91.4/91453.

[6] S. R. S. Ramzan, M. J. I. M. J. Iqbal, F. A. F. Arslan, S. N. S. Nawaz, N. J. C. N. J. Chauhdary, and M. R. M. Ramzan, "Smart and secure vehicle parking system to avoid theft using deep image recognition," Journal of Innovative Computing and Emerging Technologies, vol. 4, no. 1, Mar. 2024, doi: 10.56536/jicet.v4i1.105.

[7] E. Leka, L. Lamani, and K. Hamzallari, “A framework solution for vehicle theft detection by integrating NFC with a Blockchain-Based system,” TEM Journal, pp. 2056–2063, Nov. 2023, doi: 10.18421/tem124-16.

[8] N. A. Zulkifli and S. M. Shah, “Three-level car security system with GPS tracker using IoT,” Journal of Electronic Voltage and Application, vol. 4, no. 1, Oct. 2023, doi: 10.30880/jeva.2023.04.01.004.

[9] Q. B. Truong, T.-L. Tran, T. T.-K. Nguyen, and H. C. Nguyen, “Face and number plate recognition for car anti-theft,” CTU Journal of Innovation and Sustainable Development, vol. 15, no. ISDS, pp. 110–118, Oct. 2023, doi: 10.22144/ctujoisd.2023.041.

[10] M. H. Abdurrahman, H. A. Darwito, and A. Saleh, “Face recognition system for prevention of car theft with Haar Cascade and local binary pattern histogram using Raspberry Pi,” Emitter International Journal of Engineering Technology, pp. 407–425, Dec. 2020, doi: 10.24003/emitter.v8i2.534.

[11] “Advisory on Modus Operandi (MO) of Organized Crime Groups (OCG) through auto loans,” Aug. 26, 2021. https://www.bsp.gov.ph/Regulations/Issuances/2021/M-2021-047.pdf (accessed Jun. 13, 2025).

[12] J. K. Scroggins et al., “Identifying stigmatizing and positive/preferred language in obstetric clinical notes using natural language processing,” Journal of the American Medical Informatics Association, Nov. 2024, doi: 10.1093/jamia/ocae290.

[13] M. Al-Alshaqi, D. B. Rawat, and C. Liu, “Ensemble techniques for robust fake news detection: integrating transformers, natural language processing, and machine learning,” Sensors, vol. 24, no. 18, p. 6062, Sep. 2024, doi: 10.3390/s24186062.

[14] B. K. Sospeter and W. Odoyo, “AI-Based Phishing Attack Detection and Prevention using Natural Language Processing (NLP),” International Conference on Information Technology and Security, vol. 5, no. 1, pp. 597–602, Dec. 2024, doi: 10.32664/ic-itechs.v5i1.1590.

[15] J. I. Park, J. W. Park, K. Zhang, and D. Kim, “Advancing equity in breast cancer care: natural language processing for analysing treatment outcomes in under-represented populations,” BMJ Health & Care Informatics, vol. 31, no. 1, p. e100966, Jul. 2024, doi: 10.1136/bmjhci-2023-100966.

[16] T. Li and X. Dai, “Financial Risk Prediction and Management using Machine Learning and Natural Language Processing,” International Journal of Advanced Computer Science and Applications, vol. 15, no. 6, Jan. 2024, doi: 10.14569/ijacsa.2024.0150623.

[17] H. Wang, Y. Li, S. A. Khan, and Y. Luo, “Prediction of breast cancer distant recurrence using natural language processing and knowledge-guided convolutional neural network,” Artificial Intelligence in Medicine, vol. 110, p. 101977, Nov. 2020, doi: 10.1016/j.artmed.2020.101977.

[18] D. J. Lemas et al., “Classifying early infant feeding status from clinical notes using natural language processing and machine learning,” Scientific Reports, vol. 14, no. 1, Apr. 2024, doi: 10.1038/s41598-024-58299-x.

[19] R. Tanamal, N. Minoque, T. Wiradinata, Y. Soekamto, and T. Ratih, “House price prediction model using random forest in Surabaya City,” TEM Journal, pp. 126–132, Feb. 2023, doi: 10.18421/tem121-17.

[20] S. Babu and B. Thomas, “A survey on air pollutant PM2.5 prediction using random forest model,” Environmental Health Engineering and Management, vol. 10, no. 2, pp. 157–163, May 2023, doi: 10.34172/ehem.2023.18.

[21] D. Dansana, S. G. K. Patro, B. K. Mishra, V. Prasad, A. Razak, and A. W. Wodajo, “Analyzing the impact of loan features on bank loan prediction using Random Forest algorithm,” Engineering Reports, vol. 6, no. 2, Jun. 2023, doi: 10.1002/eng2.12707.

[22] Z. Tarek et al., “Soil erosion status prediction using a novel random forest model optimized by random search method,” Sustainability, vol. 15, no. 9, p. 7114, Apr. 2023, doi: 10.3390/su15097114.

[23] S. Rasheed, G. K. Kumar, D. M. Rani, M. V. V. P. Kantipudi, and A. M, “Heart disease prediction using GridSearchCV and Random Forest,” EAI Endorsed Transactions on Pervasive Health and Technology, vol. 10, Mar. 2024, doi: 10.4108/eetpht.10.5523.

[24] M. N. V. S. S. Abhiram, T. Sudheer, Y. S. R. Tej, and A. N. V. K. Swarupa, “Brain stroke prediction using random Forest algorithm,” International Journal on Science and Technology (IJSAT), vol. 16, no. 1, Mar. 2025, doi: 10.71097/ijsat.v16.i1.3059.

[25] O. Shobayo, O. Zachariah, M. O. Odusami, and B. Ogunleye, “Prediction of Stroke Disease with Demographic and Behavioural Data Using Random Forest Algorithm,” Analytics, vol. 2, no. 3, pp. 604–617, Aug. 2023, doi: 10.3390/analytics2030034.

[26] A. Nandedkar, C. Gadve, C. Gowda, S. Deep, and R. Mulla, “Athlete performance prediction using Random Forest,” International Journal for Research in Applied Science and Engineering Technology, vol. 12, no. 5, pp. 2370–2377, May 2024, doi: 10.22214/ijraset.2024.62112.

[27] R. Meenal, P. A. Michael, D. Pamela, and E. Rajasekaran, “Weather prediction using random forest machine learning model,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 22, no. 2, p. 1208, May 2021, doi: 10.11591/ijeecs.v22.i2.pp1208-1215.