DOI: 10.22413/ijastems/2026/v12/i1/5

Intelligent Energy Management and Power Optimization System for Electric Vehicles to Enhance Efficiency and Driving Range

S. Akhileswari, I. Preethi, M. Om Prakash Reddy, B. Kasi Viswanadh, K. Balanjaneulu
Electrical and Electronics Engineering, CBIT, Proddatur, India, 516360

Abstract*—As electric vehicle (EV) technology rapidly progresses, energy management techniques must also advance to implement smarter energy management systems beyond fixed-parameter control methods. This research outlines the development and subsequent implementation of an IoT integrated power saving system for EVs using an Arduino UNO microcontroller in partnership with a NodeMCU ESP8266 Wi-Fi module. The proposed system collects data from a temperature sensor, gas sensor, and voltage sensor to continuously determine the instantaneous operational capabilities of the vehicle's battery as well as evaluate the surrounding conditions of the vehicle. A DC motor driven by an L298N controller is used to model the load of the EV; while a relay module disconnects loads in the event of an emergency. The system includes an LCD, three LED's, and an active buzzer that provide real-time feedback to the user, while all sensor results are transmitted wirelessly to a cloud platform via the NodeMCU ESP8266 for remote monitoring and diagnostics. A potentiometer was implemented to allow for controlled variable voltage testing during prototype verification. The entire system is powered by a regulated 12-volt source. Experimental tests results demonstrate that the proposed solution results in increased energy efficiency, improved fault detection capabilities, and reliable remote monitoring capabilities when compared to traditional methods of energy management. Thus, the findings support the practicality of developing embedded-IoT solutions for next generation electric vehicle energy management systems.

Index Terms*— Electric Vehicles, Power Saving, Arduino, IoT, NodeMCU, Battery Management, Energy Efficiency, Sensor Monitoring.

I. INTRODUCTION

The use of electric vehicles (EVs) is changing the way we think about both personal and public transportation. EVs are powered solely by stored electric energy, so a great deal of emphasis is placed on the safe, effective management of stored power. Due to issues such as battery deterioration, thermal runs away, and gas-related dangers; there have been numerous concerns around safely and effectively using EVs. For those working on EVs, developing a single, functional and cost-effective embedded platform to address all three concerns at the same time continues to be an issue. Most traditional EV power management systems tend to utilize fixed thresholds, thereby depending primarily on the battery parameters in their operation and battery management unit (BMU). These types of systems are normally ideal under normal conditions, but they have difficulty operating when time, the lack of multi-parameter awareness creates a significant gap in safety and efficiency.

The combination of embedded control systems with Internet of Things (IoT) technology provides an opportunity to tackle this complex, multi-variable problem. When a microcontroller (e.g., Arduino) is connected to a cloud-based platform using Wi-Fi, it can acquire, process, and store sensor data, in real-time, while enabling anyone with access to the web to remotely view the data collected by the sensors. This type of remote monitoring and observation is very useful for fleet maintenance because it allows technicians to evaluate the condition of a vehicle without having to be in the same location as the vehicle. The system presented in this article employs an Arduino UNO-based microcontroller, a NodeMCU ESP8266-based IoT module, and three different types of sensors (temperature, gas concentration, battery voltage) to collect, process, and store sensor data. The microcontroller utilizes the collected data to make realtime decisions about varying the speed of an electric motor, switching the load from one power source to another, and generating an alert. In case of an abnormal reading received by the microcontroller from any of the three sensors, the microcontroller disconnects the load from the relay, sounds a buzzer, and displays a warning message on the LCD display, while simultaneously sending a log of the event to the cloud via the NodeMCU. The coordinated response of disconnecting the load, the buzzer sounding, the warning displayed on the LCD, and the NodeMCU logging the event constitutes the primary contribution of the proposed integration.

This paper is organized into seven sections, as follows: (i) Section II has a review of various prior work that has focused on operating EVs efficiently via the use of energy management strategies as well as by smartly monitoring energy evaluating energy management systems designed for use with EVs and IoT. Wangsupphaphol et al. (2023) reviewed existing architectures for energy management systems (EMS) for hybrid energy storage systems (HESS)

DOI: 10.22413/ijastems/2026/v12/i1/5

consisting of batteries and supercapacitors. They examined rule-based, fuzzy logic, and use with the IoT, (ii) Section III will identify limitations of traditional approaches, (iii) Section IV describes the proposed system design concept, including architecture and hardware components, (iv) Section V discusses software and control logic, (v) Section VI provides experimental results and an analysis of how these results compare to the prior studies, and (vi) Section VII discusses final conclusions and future research directions.

II. LITERATURE SURVEY

Growing studies have been dedicated towards developing smarter EV energy management systems; (Anonymous, 2025). These studies have highlighted significant advancements from working towards multi-objective optimisation frameworks (simultaneously considering battery degradation, energy use and vehicle dynamics) of energy management systems using more conventional methods for managing energy use. Moreover, a minor goal of the study was to develop a multi-objective energy management system and demonstrate the superiority of multi-objective optimisation methods vs. single-objective optimisation methods through measurable improvements. While this study was largely conducted to identify software optimisations, there was limited attention paid to multi-sensor integration at the hardware-level; this aspect should be considered when optimization-based control methods and determined that the combination of hybrid storage and advanced energy management systems (EMS) increases the useful life of batteries. They also identified that one major advantage of supercapacitors is that they can provide very high peak power, but the study did not look at real-time environmental sensing.

Anonymous (2025) examined the use of Internet of Things (IoT) technology as a means to optimize power factor at fast charging stations for electric vehicles (EVs). By utilizing continual measurements of load conditions with the capability to dynamically adjust reactive power compensation, this method improved grid reliability and reduced energy loss. The findings of this investigation demonstrated the potential for scalable IoT-enabled measurement solutions related to electric vehicle infrastructure, but the scope of this study was limited only to stationary charging and not mobile vehicle application.

Jagadeesh and Cariappa (2025) developed an advanced battery management system using IoT technology that allows for continuous monitoring of the battery's voltage, current, temperature, and state-of-charge (SOC). By additionally providing cloud-accessed data, this system will increase diagnostic capability and allow for earlier prediction of battery failure. The findings of these authors are very closely related to the current research project goals, and the hazards associated with gas leakage were not investigated by these authors.

Bannari Amman Institute of Technology (2024) introduced a deep learning algorithm for battery management of electric vehicles, which uses the Internet of Things sensors to provide real-time data to a trained neural network to identify anomalous behaviour and predict power consumption levels. While the initial findings were promising, deep learning algorithms require significant computational resources and, therefore, are predetermined limits. Although this approach is easily implemented and computationally inexpensive, it fails to recognize that real-world EV functionality is multidimensional. Generally unsuitable for low-cost embedded systems such as the Arduino. The author's current work takes a more practical solution between these two extremes by advising against using an adaptive threshold-based logic method that is implemented directly on the microcontroller.

Arévalo, Ochoa-Correa and Villa-Avila (2024) undertook a systematic review of the integration of artificial intelligence technology into electric vehicle energy management systems with a focus on the use of neural networks, reinforcement learning and fuzzy logic. The authors concluded that the use of AI technology in electric vehicle energy management systems improved the ability to adapt in real-time and perform multiple objectives, but also recognised the degree of complexity associated with embedded implementations of artificial intelligence technology. The system proposed in the present study draws from the information obtained in these systematic reviews, incorporates lightweight sensor fusion based logic and is intended for use in battery management systems operating on resource-limited hardware.

.

III. EXISTING METHODS AND LIMITATIONS

Traditional electric vehicle (EV) power management systems have provided dependable baselines for many years; however, their structural limitations are becoming more apparent with time as user orientation and safety expectations continue to grow. Most existing solutions rely on the concept of measuring either one or two parameters (primarily battery voltage and current) to establish fixed responses when those parameters exceed.

There are many limitations associated with traditional EV power management systems and one of the most critical is that they do not incorporate environmental sensing into their design. For example, a battery pack can operate within its voltage range yet still be in danger if the surrounding temperature exceeds normal limits or if a gas leak exists in the area around it. Since fixed-threshold controllers do not take into consideration secondary hazards, critical warning windows can be missed. While this type of oversight is usually harmless in a laboratory environment, it presents a significant safety concern when applied to actual vehicles.

Many earlier models of electric vehicle management systems lack remote monitoring and data collection functionality. Fleet managers and mechanics have always depended on manual inspections and periodic diagnostic procedures to evaluate vehicle health; therefore, this reactive approach often leads to delayed identification of issues, resulting in higher total repair costs. Remote monitoring also restricts the owner/operator from continuously observing the operational characteristics of the electric vehicles within the fleet, which eliminates the

DOI: 10.22413/ijastems/2026/v12/i1/5

opportunity for implementing predictive maintenance as a best practice in today's industrial environments.

Additionally, conventional electric vehicle management methodologies typically produce static control strategies. The response characteristics remain fixed regardless of fluctuating customer loads, driving conditions, and/or environmental variables since the calibration process is done at one point in time. Therefore, a much more intelligent solution is possible by allowing the electric drive trains to have some flexibility in performance based on multiple sensor inputs, rather than relying solely on one input. This proposed solution will provide a multi-sensor, adaptive, IoT-enabled architecture that will mitigate all of these problems.

IV. PROPOSED SYSTEM ARCHITECTURE

A. System Overview

In this project, an Arduino UNO board will be the central processing unit of the proposed power savings system. Three sensors will provide real-time data to the Arduino UNO: 1) a temperature sensor for thermal measurements, 2) an MQ-2 gas sensor for detecting combustible or toxic gas levels, and 3) a voltage sensor for measuring battery output. A potentiometer will also be integrated into the prototype test environment to emulate fluctuations in voltage. The Arduino will utilize the data from the three sensors to generate a set of adaptive control functions that will control the speed of the DC motor, state of the relay, messages sent to the LCD, activity of LEDs, and activity of the buzzer.

The Arduino is connected to a NodeMCU ESP8266 module via a serial interface. The Arduino will send retrieved values from each of the three sensors to the NodeMCU module on a periodic basis; the NodeMCU will then transfer to a cloud platform using the MQTT or HTTP protocol and over the local area Wi-Fi network. The cloud link allows users that are authorized to have access to live and historic readings of the data retrieved from the three sensors, at any time or place using their smartphone or web browser, regardless of the distance they are from the vehicle.

B. Hardware Components

Table I provides a summary of each hardware component utilized in the proposed power savings system, including its specifications, and functional roles.

TABLE I
HARDWARE COMPONENTS AND THEIR SPECIFICATIONS

S. No	Component	Specification	Function
1	Arduino UNO	ATmega328P, 5V	Central microcontroller
2	NodeMCU ESP8266	Wi-Fi enabled, 3.3V	IoT cloud data upload
3	DHT11 / LM35	0–50°C range	Temperature monitoring
4	MQ-2 Gas Sensor	Analog output, 5V	Gas leakage detection
5	Voltage Sensor	0–25V range	Battery voltage sensing
6	DC Motor + Driver	L298N, 12V	EV load simulation
7	Relay Module	5V, single channel	Safe load switching
8	LCD Display (16x2)	I2C interface	Real-time status display
9	Buzzer + LEDs	5V active	Alert and indication
10	12V Battery + Adapter	12V, 2A	System power supply

C. System Block Diagram Description

This layered model consists of sensing, processing, and actuation layers. Within the sensing layer, three types of sensors are used to monitor physical values, and the sensors send analog or digital signals to the input pins of the Arduino. Within the processing layer, the ATmega328P microcontroller on the Arduino compares the signals received to the adaptive thresholds set for the input signals and drives the outputs appropriately. The actuation layer includes the motor driver, relays, LCD display, LEDs, and buzzer.

V. SOFTWARE AND CONTROL LOGIC

A. Development Environment

All code for both the Arduino and NodeMCU devices are developed in the Arduino Integrated Development Environment (IDE) which is open source and is a C/C++ programming language with a huge library of available code to support many projects. Libraries used in the project to support the project include DHT sensor to read temperature, LiquidCrystal_I2C to communicate with the LCD display, ESP8266WiFi to allow NodeMCU to connect to WiFi, and PubSubClient to communicate with the cloud services. The code is compiled and uploaded to each board using the appropriate COM port selected within the IDE Tools menu.

B. Control Algorithm

At the beginning of each control loop execution cycle of the Arduino firmware, the system reads each of the three sensors. The temperature is checked against an upper safe limit of 60°C, the gas sensor is compared to a calibrated initial concentration and the voltage reading, via the potentiometer variable reference, is compared to both a minimum and maximum battery safe level.

In the event that one of the three sensors exceeds their respective thresholds, the Arduino immediately energizes the relay to remove power from the DC motor load and illuminates the LED associated with that sensor, activates the buzzer for a specified alert period/length of time and displays a descriptive fault message on the LCD. If all three sensors are within the acceptable safe limits, the DC motor is able to run at full speed, the green LED remains lit, and the LCD will show the current value from all 3 sensors.

DOI: 10.22413/ijastems/2026/v12/i1/5

Every five seconds, the NodeMCU sends a formatted data packet to the cloud endpoint, regardless of whether or not there is any fault detected. The priority order of fault conditions is one of the most important design features of the system. The highest priority fault is gas leakage because it is a safety hazard that needs immediate attention. The next priority order would be temperature exceedance and then voltage deviation. This means that the system response will always be proportional to the
severity of the found anomaly and will not treat all faults the same.

C. Cloud Integration

The NodeMCU will connect to the local Wi-Fi access point with the credentials held in the firmware. Once connected, the NodeMCU will subscribe to an MQTT broker or make an HTTP GET/POST request to send sensor data from the NodeMCU to the cloud dashboard. The cloud platform will provide visualisation of the provided time series graphs so that the operator can see trends and anomalies over time of the temperature, gas concentration and
voltage levels. Push notifications can also be configured on the cloud side to alert users when the sensor values reported from the NodeMCU exceed user-defined thresholds; therefore, adding another level of remote safety management.

VI. RESULT AND DISCUSSION

The prototype was put together. Tested in a lab with a technician watching to make sure everything works like it should. The sensors on the unit were sending signals fine so the motor kept running for a long time at full speed. It also lit up a light and showed the voltage, gas and temperature on a screen. Every five seconds the NodeMCU sent the sensor information to the cloud over the internet. Updated the cloud dashboard really fast. The NodeMCU was doing a job of sending the sensor signal data to the cloud and the cloud dashboard was getting updated almost right away with the new sensor signal data, from the NodeMCU.

A thermal test was conducted by gradually increasing the temperature around the sensor using a heat source. When the temperature crossed 60 degrees Celsius, the Arduino responded within one control cycle (approximately 500 milliseconds), disconnecting the motor via the relay, activating the buzzer, switching to the red warning LED, and displaying a temperature fault message on the LCD. The NodeMCU simultaneously transmitted the fault state readings to the cloud, where they appeared as a spike in the temperature graph.

In a simulation of a gas leak, the MQ-2 gas sensor was tested using LPG. The output of the sensor increased dramatically, resulting in the sensor activating a top-priority fault. The relay opened within a single control cycle, and the buzzer made a distinctive tone that sounded different from the sound used for temperature alerts. The sensor was able to recover automatically when the concentration of gas decreased below the set point level, which transient faults without human intervention.

There were two tests run with a potentiometer adjusted throughout the entire voltage range to determine if there are potential voltage anomalies. The first test simulated a voltage lower than the minimum allowed voltage and initiated a low voltage warning on the LCD by decreasing the load on the motor. The second test simulated a voltage greater than the maximum allowed voltage and produced an over voltage alert. Both tests provided the appropriate notification of under voltage or over voltage condition on the cloud-based dashboard. Table II presents a comparison of the proposed system to a typical EV Power Management System (PMS).

TABLE II
Comprehensive feature set comparison of Conventional PMS to Proposed PMS for EV.

Parameter	**Conventional PMS**	**Proposed PMS**
Monitoring Parameter	Voltage / Current only	Voltage, Temperature, and Gas
Control Strategy	Static / Pre-defined	Adaptive / Real-time
Remote Monitoring	Not Available	Cloud Based/ NodeMCU IoT
Fault Detection	Delayed / Manual	Immediate / Automated
Energy Efficiency	Moderate	High (adaptive control)
Cost	Higher (dedicated units)	Low (embedded + IoT)

The advantages of the proposed system over each of the 3 evaluated characteristics are clearly demonstrated. The additional multi-sensor monitoring allows for the fast detection of dangerous conditions that are not detectable by traditional voltage-only controllers. The addition of a cloud-based connectivity layer gives the system an entire new level of remote visibility, which traditional designs lack. The cost structure for the prototype, based on commercially available embedded modules/sensors is substantially less than that of a dedicated proprietary battery management unit.

VII. CONCLUSION

The objective of this paper has been to present an IoT integrated power saving solution for electric vehicles that integrates multi-parameter sensing with adaptive control technique and real-time cloud monitoring. The use of an Arduino UNO microcontroller enables it to continuously analyze data from temperature, gas and voltage sensors to determine how to operate the motor, switch the load on/off and generate alerts. The sensor data is transmitted to an IoT cloud platform via NodeMCU ESP8266, allowing remote

DOI: 10.22413/ijastems/2026/v12/i1/5

access to the data and analysis of historical trends, which traditional systems do not provide.

The prototype test of the system has shown that it reliably detects and responds to thermal fault conditions, gas leaks, and battery voltage anomalies within one control cycle. The System has a fault handling mechanism based on established priorities so the most serious condition will give the greatest speed and degree of respondmg. Compared to fixed threshold conventional architectures, the proposed solution will have multi-parameter awareness, adaptive control and automated fault detection, all based on a cost-effective embedded solution and having an ability to provide remote access to data.

The next phase of development for the system will look at exposing ml-based predictive fault detection algorithms on edge devices prior to threshold violations or predictive failures. Also, incorporating state-of-charge estimation models with regenerative braking control signals will help to, further, enhance the energy efficiency of the overall system. For the system to scale and result in real vehicle deployment a weather-rated enclosure, CAN bus type communication from the vehicles' onboard diagnostic port and relevant automotive safety standard compliance testing will need to be completed before any type of certification will be issued.

ACKNOWLEDGEMENT

The authors thank the Department of Electrical and Electronic Engineering for the laboratory facilities and ongoing assistance provided during the course of this project. The authors also acknowledge the project guide's substantial contribution to the technical direction of the project and recognize those who assisted in the evaluation and testing of the prototype.

REFERENCE

[1] Author Unknown (2025). Design of an Intelligent Energy Management System for Electric Vehicles - A Comprehensive Review. Energy Informatics.

[2] Wangsuphaphol, A., et al. (2023). Energy Management Systems for Electric Vehicles with Battery/Super Capacitor Applications - A Comprehensive Review. Journal of Energy Management and Sustainable Systems, 3(2), 210-229.

[3] Gundu, Srinivasa Rao, et al. "The Dynamic Computational Model and the New Era of Cloud Computation Using Microsoft Azure." SN Computer Science, vol. 1, no. 5, Sept. 2020, p. 264. DOI.org (Crossref), https://doi.org/10.1007/s42979-020-00276-y.

[4] Author Unknown (2025). A Fast Charging Station for Electric Vehicles with Power Factor Capability via the Internet of Things. This is a paper that was published in the Journal of Umm Al-Qura University for Engineering and Architecture. The Journal of Umm Al-Qura University, for Engineering and Architecture published this paper about Electric Vehicles and the Internet of Things.

[5] Jagadeesh, N.; Cariappa, S. (2025). An IoT-Based Battery Management System International Journal of Engineering Research and Technology (IJERT).

[6] Panem, Charanarur, et al. "Polynomials in Error Detection and Correction in Data Communication System." Coding Theory, edited by Sudhakar Radhakrishnan and Muhammad Sarfraz, IntechOpen, 2020. DOI.org (Crossref), https://doi.org/10.5772/intechopen.86160.

[7] Bannari Amman Institute of Technology (2024). A Power Management System with IoT Integration Utilizing Deep Learning and an Electric Vehicle Battery Management System. SAE Technical Paper 2024-28-0085.

[8] Arevalo, P.; Ochoa-Correa, D.; Villa-Avila, E. (2024). AI-Integrated Energy Management Systems for Electric Vehicles - A Comprehensive Review. World Electric Vehicle Journal, 15(2).

[9] Microchip Technology Inc. (2023). ATmega328P - Microcontroller Data Sheet. Microchip Technology Inc.

[10] Espressif Systems (2023). ESP8266 Technical Reference Manual. Espressif.

[11] Texas Instruments (2022). L298N Dual Full-Bridge Driver Datasheet. Texas Instruments.