Smart Elevator system using Arduino programming

Harsh Gupta[1], Shubham Gupta[1], Vansh Shan[1] and Dr. R K Singh[2]

[1]*Students of School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India*

[2]*School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India,* rabindrakumar.singh@vit.ac.in

***Abstract*– Facing delays in crucial meetings or exams due to elevator unavailability or frequent stops can be frustrating, often tarnishing your punctuality. The blame for such instances falls on the elevator system. With the anticipation of an influx of new students on campus, the need for enhanced and more efficient elevator technology becomes evident. A straightforward remedy would involve installing additional elevators across the campus to accommodate the growing student population. However, a closer examination reveals that the currently operational elevators aren't performing at their optimal capacity. This is where the Smart Elevator System comes into play, addressing this precise issue through the integration of a selection of IoT sensors.**

At its core, this system incorporates various hardware components, including ultrasonic sensors, motion detectors, and electric peak load cell sensors. The software aspect relies on the Arduino IDE. Through the intelligent utilization of the data garnered from these sensors, the occurrence of unnecessary elevator stoppages can be significantly curtailed. The implementation of this innovation within elevators at various locations would yield not only time-saving benefits for users but also a reduction in elevator power consumption. Furthermore, it stands to reason that this advancement could lead to a decrease in the frequency of maintenance requirements for elevators, stemming from the alleviation of excessive load stress on specific units.

Keywords*– *Smart Elevator System, IoT sensors, ultrasonic sensors, motion detectors, electric peak load cell sensors, Arduino IDE

I. INTRODUCTION

Over the last decade, the landscape of the elevator sector has been profoundly reshaped by the advancements in Internet of Things (IoT) technology. This transformation has ushered in a new era of possibilities, particularly evident in the realm of smart elevators. Leveraging IoT, these elevators have evolved beyond their conventional counterparts, offering proactive monitoring, condition-based analysis, and predictive maintenance. By facilitating remote control and real-time performance monitoring, smart elevators have become a beacon of efficiency and convenience.

Connectivity lies at the heart of these advancements, as smart elevators communicate through various channels. Elevator controllers interface with custom hardware using technologies like GPIO pins, UART, Wi-Fi, or LAN, enabling seamless data exchange. Notifications flow both ways – when hardware status changes, signals are sent to the IoT central, while modifications made at the elevator controller are relayed back. This information is then processed, analyzed, and presented to users through interfaces, fostering informed decision-making and efficient fault management.

In parallel, the narrative of vertical mobility is being rewritten through the fusion of innovation and technology. Arduino programming, a versatile open-source microcontroller platform, injects intelligence into elevator systems, effectively addressing inefficiencies that have long plagued conventional setups. Elevator systems enriched with Arduino's capabilities can dynamically adapt to real-time demands, reducing waiting times, minimizing unnecessary stops, and optimizing energy consumption.

This integration of technology, exemplified by the Smart Elevator System powered by Arduino programming, holds profound implications for modern urban living. As cities grow vertically, the challenges faced by traditional elevator systems become more apparent. This convergence pioneers solutions that not only enhance the user experience but also offer sustainable responses to the demands of urbanization. Ultimately, this amalgamation of cutting-edge technology and a time-honored concept redefines vertical transportation, setting new standards for efficiency, connectivity, and comfort in our contemporary world.

II. LITERATURE SURVEY

Presented within this study is the ElevatorTalk system, a progressive approach to elevator development and management, underpinned by the Internet of Things (IoT) methodology named IoTtalk. This innovative architecture employs a segmented software structure, effectively facilitating the creation of adaptable and scalable automobile scheduling algorithms. The system comprises three integral subsystems: the elevator car operating (ECO) panel, the scheduler, and the individual elevator cars themselves. The initial two subsystems, the ECO panel and the scheduler, collaboratively oversee the development and operation of the elevator systems. Meanwhile, the third subsystem is dedicated to addressing passenger requests. By functioning in unison, these three subsystems enable a dynamic and continuous exchange of messages, thereby fostering seamless communication and interaction among them. ElevatorTalk stands as a testament to the transformative capabilities of IoT-based strategies, dividing complex elevator operations into manageable components while enhancing scheduling efficiency. This novel approach represents a significant stride forward in elevator technology, promising increased adaptability, scalability, and overall performance within the domain of vertical transportation management. [1]

In this study, Context State Machines (CSMs) are introduced as a fundamental framework for addressing real-world challenges within the context of a smart elevator control system. The focus is on a context-aware problem drawn from existing literature. CSMs offer a dynamic means of simulating the evolution of context attributes, scenarios, or contextual conditions over time. These transitions in states reveal nuanced contextual behaviors, often overlooked yet pivotal for informed context reasoning and system decision-making. The study encompasses the conceptualization, design, and realization of a CSM engine tailored to this context. By leveraging this engine, the performance is rigorously evaluated through the examination of two context-aware scenarios within the smart elevator domain. Notably, the study demonstrates the CSM engine's capacity to autonomously capture and analyze context, thereby eliminating the need for manual developer inference. A significant contribution of this work lies in its pioneering application of state-based context modeling and the integration of CSM engines within a context-aware context, as highlighted by its novelty in the literature. By adopting CSMs, this study bridges the gap between contextual dynamics and intelligent system behavior, paving the way for enhanced context-aware applications and enriching the landscape of smart elevator systems. [2]

A novel Internet-of-Things (IoT) device is presented, offering a pioneering solution for remote elevator control. What sets this proposal apart is its capability to transform a manually operated elevator into a remotely managed system, all without necessitating any intrusive alterations or wiring connections within the elevator structure. The device is ingeniously conceived as a non-intrusive add-on, positioned atop the existing elevator button panel. Leveraging servomotors, it deftly engages with the original buttons, enabling seamless remote control functionality. This design not only ensures swift deployment but also amplifies elevator accessibility through diverse communication channels such as messages, webpages, or QR codes. The applications of this innovative approach extend to scenarios like non-contact elevator use during pandemics, as well as autonomous operation by mobile cleaning or delivery robots. Rigorous testing under actual operational conditions has demonstrated the IoT device's efficacy in facilitating non-intrusive control, affirming its viability for practical implementation. This advancement not only introduces convenience but also enhances adaptability in elevator control mechanisms. [3]

By creating a unique deep learning-based technique, DLSODC-GWM, this study tackles the critical demand for smart waste management in urban areas. Focusing on small garbage waste objects, the technique integrates object detection and classification processes. It employs an optimized RefineDet model using an arithmetic optimization algorithm (AOA) for accurate object detection. Furthermore, the functionally linked neural networks (FLNN) technique is used to classify trash objects. The approach distinguishes itself by using IRD for trash identification and AOA-based hyperparameter tweaking. Evaluation against benchmark datasets showcases the DLSODC-GWM's superior performance, achieving a remarkable accuracy of 98.61%, thus offering a promising solution for intelligent waste management in the context of rapidly growing smart cities. [4]

This study investigates the use of AI's technology for deep learning, particularly the YOLOv5 approach, for successful parcel box recognition in the realm of last-mile delivery operations. Various

YOLOv5 models were examined in terms of box recognition and location estimation. The YOLOv5large algorithm emerged as the best option, surpassing other models with greater accuracy, recall, and scores for F1 of 0.966, 0.899, and 0.932, respectively. Notably, the effectiveness of the YOLOv5large was matched by a reduced model size when compared to the YOLOv5xlarge. This research establishes the groundwork for building a smart actual time package storage system using YOLOv5large's object detection abilities. [5]

This study centers around optimizing elevator operations by minimizing movement and introducing touchless buttons for Covid-19 prevention. Fuzzy logic is employed to prioritize elevator movements based on input variables: position, distance, direction, and capacity. Utilizing a PLC Outseal Mega V1.1 as a controller and proximity infrared sensors for touchless input, the study integrates comparators, timers, and counters to implement a fuzzy logic-based elevator system. The output, a relay, controls the DC motor. This innovative approach not only enhances elevator efficiency but also addresses Covid-19 concerns, offering a forward-looking solution for safer and smarter elevator operations. [6]

The central theme of this study underscores the effective time management achieved through automation, a pivotal factor in boosting productivity. Automation technology streamlines operations with minimal human interference, leading to heightened production rates and enhanced quality across various applications. The research primarily concentrates on fabricating an automated writing apparatus aimed at aiding individuals with disabilities. This apparatus seeks to provide accurate and error-free writing, resulting in time and labor savings. It employs a pair of motors for multidimensional movement, precisely guided by an Arduino Nano microcontroller. The study employs Inscape and G-code to generate visual content. This automated writing mechanism offers adaptability, catering to both graphic and writing needs, while simultaneously streamlining processes and enhancing accessibility for those with disabilities. [7]

The authors of the paper outlines the development and deployment of a five-floor electrical elevator system employing voice recognition technology via an Arduino microcontroller. This innovation aims to cater to individuals with disabilities, particularly those challenged by conventional elevators due to physical constraints. The setup integrates an Arduino microcontroller, voice recognition module, a motor driver, and elevator car with a lift mechanism. User-issued voice commands are processed by the microcontroller to determine the intended floor, subsequently activating the lift mechanism via the motor driver. Testing affirmed the system's reliability and efficiency in facilitating vertical movement, underscoring its potential to significantly enhance accessibility and convenience for individuals with disabilities within buildings. [8]

Large organizations often face challenges when it comes to monitoring elevator systems offline, especially during non-peak hours. To address this, a cost-effective cloud-based web application is introduced in this study, enabling real-time monitoring of elevator position and faults. The fault detection involves two steps: a rotary encoder gauges lift movement, while level sensors indicate floor numbers or faults. Employing an Arduino controller with a Wi-Fi module, this setup senses, processes, and transmits data to a ThingSpeak channel. A prototype is devised for testing, validating the efficacy of this online monitoring approach, which holds promise in alleviating the complexities of large-scale elevator management. [9]

The escalating number of unintentional deaths, especially around swimming pools, has become a pressing concern. This concerning trend, substantiated by reports from the World Health Organization and national health magazines, prompted the need for innovative solutions in the realm of autonomous rescue designs. This paper fills a research gap by introducing novel methods and techniques to combat this issue. The focal point is a unique multilayered architecture that emphasizes fail-safe autonomous rescue techniques, utilizing natural buoyancy as the primary mechanical power source for lifting drowning individuals. The paper conducts trials that underscore the accuracy and reliability of the proposed autonomous drown-rescue system design. This comprehensive approach encompasses strategically placed sensors, elevators, semiautomatic features like panic switches, and buoyancy-assisted elevators. The methods discussed are both feasible and cost-effective, significantly bolstering pool safety while mitigating drowning risks across all age groups. The paper's distinctive contributions lie in its innovative underwater and buoyancy-driven lift designs, proximity sensors, and laser tripwire-based drowning accident detection. This study underscores the necessity for governmental and communal mandates to implement these systems to ensure 100% pool safety. [10]

ATMs have revolutionized global transactions, yet their security remains a paramount concern. While user-facing security is handled via PINs and smart cards, safeguarding ATMs from the bank's end necessitates a robust approach. This project introduces an advanced security system that monitors and triggers security measures against theft and robbery attempts. Employing Reed Switches, Ultrasonic Sensors, and Cameras, the system assesses security parameters and alerts relevant authorities. Any unauthorized movement or access triggers the sensors, prompting the surveillance camera to capture the intruder's image. An SMS notification is sent, enabling authorities to view the intruder's face via an IP address. This comprehensive security solution enhances ATM safety beyond conventional methods. [11]

In the current scenario, the potential for COVID-19 transmission in confined spaces like elevators is a pressing concern. Existing prevention measures often rely on manual checks, but lack comprehensive solutions. This study presents an innovative intelligent epidemic prevention system tailored for modern construction site elevators. The system combines non-contact human temperature measurement, mask and helmet recognition, and voice-activated elevator functionality. Employing Arduino UNO, Kendryte K210, infrared temperature sensor MLX90614, and voice recognition sensor LD3320, the system achieves accurate temperature detection, mask/helmet recognition (utilizing YOLOv3), and voice-activated elevator calling. Experimental results demonstrate recognition accuracies of 91.5% for helmets, 92.0% for masks, and 93.0% for voice call activation. Boasting a temperature measurement accuracy of 0.2℃, this system effectively curbs contact-based transmission and presents a stable, intelligent, and secure solution. [12]

III. HARDWARE AND SOFTWARE SPECIFICATIONS

A. *Ultrasonic detectors* are a form of sensor that is often used in the Internet of Things, or IoT, setting to identify and quantify distances, presence, or items in an environment. These sensors operate by producing high-frequency sounds (ultrasonic waves) and measuring the amount of time it requires for the sound waves to reverberate back after colliding with an item. The journey to the object can then be calculated using the time delay.

Here's how ultrasonic sensors work in an IoT context:

1. ***Sensor Setup:*** A transmitter and the recipient are the two primary components of an ultrasonic sensor. The transmitter sends out ultrasonic waves with a frequency of around 40 kHz (but this can vary). The waves are detected by the receiver after they reflect back towards an object.

2. ***Wave Emission:*** The sensor emits a short burst of ultrasonic waves. These waves travel through the air until they encounter an object in their path.

3. ***Wave Reflection:*** When the ultrasonic waves hit an object, they reflect off its surface. The angle of incidence and the angle of reflection are the same, according to the law of reflection.

4. ***Time Measurement:*** The sensor's receiver detects the reflected ultrasonic waves. By measuring the time it takes for the waves to return to the sensor, the sensor can calculate the time of flight (TOF) of the waves.

5. ***Distance Calculation:*** Knowing the velocity of sound in the medium (typically air), one may calculate the distance to an object using the following equation: Distance = (Speed of Sound Time of Flight) / 2..

6. ***Data Processing and Communication:*** In an IoT context, the sensor's data is processed by a microcontroller or microprocessor. The calculated distance can be converted to a suitable unit and then transmitted to a central server, cloud platform, or other connected devices using communication protocols like Wi-Fi, Bluetooth, LoRa, or cellular networks.

7. ***Application:*** Ultrasonic sensors find various applications in IoT, including:

- Proximity Sensing: Detecting the presence of objects or people in a certain range.
- Distance Measurement: Measuring the distance

between the sensor and an object, often used in industrial automation, parking systems, and robotics.

- Obstacle Avoidance: Used in robotics and autonomous vehicles to navigate and avoid collisions.
- Liquid Level Measurement: Monitoring fluid levels in tanks or containers.
- Security Systems: Detecting intruders by monitoring changes in the environment.
- Smart Lighting: Adjusting lighting levels based on the presence of people.

B. *Motion sensors*, additionally referred to as detectors for motion, are devices that monitor motion or modifications to an environment in the setting of the Web of Things (IoT). These sensors are designed to identify the presence of people, animals, or objects within a specific area and trigger actions or alerts based on this detection. They play a crucial role in creating smart and responsive IoT applications for security, automation, and energy efficiency.

Here's how motion detectors work in an IoT context:

1. ***Sensor Types:*** There are various types of motion detectors, each using different technologies to sense movement. Some common types include:

- Passive Infrared (PIR) Sensors: These sensors detect changes in infrared radiation emitted by warm objects. When a person or object moves within the sensor's field of view, there is a temperature difference, and the sensor detects this change in heat.
- Microwave Sensors: Microwave motion detectors emit microwave pulses and measure the reflected signals. Changes in the reflected signal pattern caused by movement are used to trigger the sensor.
- Ultrasonic Sensors: Similar to the ultrasonic sensors discussed earlier, these emit ultrasonic waves and measure their reflection to detect motion.
- Dual Technology Sensors: These sensors combine two different technologies (e.g., PIR and microwave) to reduce false alarms by requiring both technologies to trigger before activation.

2. ***Sensor Placement and Coverage:*** Motion detectors are strategically placed in areas where movement needs to be monitored. The coverage area is defined by the sensor's field of view and range. This ensures that any movement within the specified area will be detected.

3. ***Detection and Triggering:*** When a person or object moves within the coverage area, the motion sensor detects the change and activates. The sensor then sends a signal to the connected IoT device or system.

4. ***Data Processing and Action:*** In an IoT context, the signal from the motion sensor is processed by a microcontroller or microprocessor. Depending on the application, the IoT system can take various actions, such as:

- Sending an alert or notification to a smartphone or central monitoring system.
- Activating security measures, such as turning on lights, sounding alarms, or notifying security personnel.
- Triggering automation, such as turning on appliances, adjusting climate control, or unlocking doors.

5. ***Communication:*** The processed data or triggered actions can be communicated through various communication protocols, such as Wi-Fi, Zigbee, Z-Wave, or Bluetooth, to other IoT devices, applications, or the cloud.

6. ***Application:*** Motion detectors are used in a wide range of IoT applications:

- Home Security: Detecting unauthorized intrusions and triggering alarms.
- Smart Lighting: Automatically turning on lights when movement is detected.
- Energy Efficiency: Adjusting lighting or HVAC systems based on occupancy to save energy.
- Home Automation: Activating appliances or devices when movement is detected.
- Commercial Spaces: Monitoring foot traffic and occupancy for building management or security purposes.

C. *ElectroPeak Loadcell Sensor.*

A load cell is a transducer that converts a mechanical force or load into an electrical signal. These sensors are commonly used in industrial, commercial, and research contexts for tasks like weight measurement, force monitoring, and material testing. Here's how a load cell works and how it might be used:

1. ***Principle of Operation:*** A load cell typically consists of a metal structure with strain gauges attached. When a force or load is applied to the load cell, the metal structure undergoes deformation. This deformation causes the strain gauges to change resistance, which, in turn, changes the electrical signal they produce.

2. ***Strain Gauges:*** Strain gauges are small devices that change their electrical resistance when they are subjected to strain (deformation). Load cells usually have several strain gauges arranged in a specific pattern to maximize sensitivity and accuracy.

3. ***Wheatstone Bridge Circuit:*** Load cells are often used in combination with a Wheatstone bridge circuit. This circuit arrangement helps convert the changes in resistance of the strain gauges into a voltage output that can be easily measured and interpreted.

4. ***Calibration:*** Load cells need to be calibrated to accurately convert the electrical signal into the corresponding weight or force measurement. Calibration involves applying known weights or forces to the load cell and recording the corresponding electrical signals. This information is then used to create a calibration curve or equation that relates the electrical output to the actual force or weight.

5. ***Types of Load Cells:*** There are various types of load cells based on their design and application:

- Compression Load Cells: These are used to measure forces pushing in a straight line.
- Tension Load Cells: These measure forces pulling in a straight line.
- Shear Load Cells: They measure forces applied perpendicular to the center axis.
- Bending Beam Load Cells: These are often used in industrial scales and platforms.
- S-Type Load Cells: Resemble the letter "S" and can measure tension and compression forces.

6. ***Applications:*** Load cells have a wide range of applications, including:

- Industrial Scales: Used in weighbridges, conveyor belts, and packaging machines.
- Automotive Testing: For analyzing forces on vehicle components.
- Material Testing: Determining the mechanical properties of materials.
- Aerospace: Measuring forces on aircraft components during testing.
- Medical Devices: Such as patient weighing scales.
- Robotics: Ensuring precision in robotic movements and interactions.

D. *The Arduino IDE (Integrated Development Environment)* is a software platform designed to simplify the process of programming and developing projects using Arduino boards and compatible microcontrollers. Arduino is a popular open-source hardware and software platform that provides a flexible and accessible way to create interactive electronic projects and prototypes. The Arduino IDE serves as a central tool for writing, compiling, and uploading code to Arduino boards.

Here are the key features and components of the Arduino IDE:

1. ***Code Editor:*** The core function of the Arduino IDE is its code editor, where you write and edit the code that will run on the Arduino board. The code is written in the C or C++ programming language, with specific libraries and functions provided by Arduino to simplify hardware interaction.

2. ***Sketches:*** In the Arduino world, a program is referred to as a "sketch." A sketch is a piece of code that defines how your project behaves. The Arduino IDE allows you to create, save, and manage sketches.

3. ***Library Manager:*** Arduino libraries are pre-written code modules that provide functions and features for interacting with various hardware components like sensors, displays, motors, and communication protocols. The IDE includes a Library Manager that lets you easily search for, install, and manage libraries.

4. ***Serial Monitor:*** Arduino boards often communicate with a computer via a serial connection. The Serial Monitor in the IDE allows you to send and receive data between the board and your computer. This is particularly useful for debugging and monitoring your projects.

5. ***Board Manager:*** Different Arduino boards use different microcontrollers and have specific configurations. The Board Manager allows you to install and manage board definitions for various Arduino-compatible devices, making it easy to switch between different hardware setups.

6. ***Compile and Upload:*** Once you've written your code, the IDE compiles it into machine code that can run on the Arduino board. You can then use the IDE to upload the compiled code to the board, making your project come to life.

7. ***Examples:*** The Arduino IDE provides a range of example sketches that demonstrate various functions and capabilities. These examples can be accessed through the "File > Examples" menu and serve as valuable starting points for your projects.

8. ***Support for Third-Party Hardware:*** While the Arduino IDE is primarily designed for Arduino boards, it also supports many third-party hardware platforms that are compatible with the Arduino framework.

9. ***Community and Resources:*** The Arduino community is large and active, providing tutorials, documentation, forums, and open-source projects that can help you learn and create with Arduino.

10. ***Cross-Platform:*** The Arduino IDE is accessible for a variety of computer operating structures, including macOS, Windows, and Linux, allowing it to be used by a variety of users.

Arduino IDE provides a beginner-friendly way for hobbyists, students, and professionals to start working with microcontrollers and electronics.

IV. PROPOSED WORK

The innovative concept of our proposed smart elevator system holds the potential to revolutionize the way people experience vertical transportation. By seamlessly integrating advanced technologies, such as motion detectors, ultrasonic sensors, and weight sensors, this system aims to not only enhance the efficiency of elevator operations but also significantly optimize the time and convenience of passengers.

At the heart of this intelligent elevator infrastructure lies a well-orchestrated orchestration of cutting-edge sensors. The motion detector, acting as the first line of interaction, identifies the presence of individuals in proximity to the elevator doors as they press the buttons to summon the lift. Through its discerning ability to recognize human presence, this sensor effectively ensures that the elevator only halts at floors where potential passengers await, obviating the wastage of time and energy on unnecessary stops.

Complementing this capability is the integration of ultrasonic sensors, further underscoring the sophistication of the system. By utilizing ultrasonic waves to gauge the proximity of individuals to the elevator doors, this sensor corroborates the findings of the motion detector, creating a comprehensive and reliable detection mechanism. The synergy between these sensors culminates in a responsive elevator that anticipates the needs of its users and minimizes delays.

Furthermore, the incorporation of a weight sensor represents a significant step forward in ensuring both the safety and optimal functionality of the elevator system. This sensor, proficient in measuring the load within the elevator car, serves as a vigilant sentinel against overloading. In situations where the weight threshold is surpassed, the elevator is intelligently programmed to withhold movement until the excess load is rectified. This not only safeguards the elevator's mechanical integrity but also guarantees a secure and comfortable ride for passengers.

A. System Architecture

The system architecture of the proposed smart elevator system is designed to seamlessly integrate multiple advanced sensors and processing units, enabling an efficient and responsive elevator experience. The architecture can be divided into several key components and layers, each contributing to the overall functionality and performance of the system.

1. Sensor Layer:

- Motion Detector: The motion detector is strategically positioned near the elevator call buttons on each floor. It uses passive infrared (PIR) technology to detect the presence of individuals in its field of view. The detector sends signals to the control unit when motion is detected, indicating a potential passenger waiting for the elevator.

- Ultrasonic Sensor: Mounted in the elevator shaft and oriented towards the floors, the ultrasonic sensor emits ultrasonic waves and measures their reflections to determine the proximity of individuals near the elevator doors. This information supplements the motion detector's data, creating a more robust presence detection mechanism.

- Weight Sensor: The weight sensor is integrated into the elevator car's floor, capable of measuring the total weight of occupants and any accompanying cargo. It provides real-time feedback to the control unit, ensuring the elevator operates within safe load limits.

2. Control and Decision-Making Layer:

- Microcontroller Unit (MCU): This layer houses the MCU responsible for processing data from the motion detector, ultrasonic sensor, and weight sensor. It makes real-time decisions based on the input from these sensors.

- Decision Logic: The MCU employs decision logic algorithms to determine whether the elevator should stop at a particular floor based on the combined input from the motion detector and ultrasonic sensor. If no passenger is detected, the elevator bypasses that floor, saving time and energy.

- Overload Protection: The MCU continuously monitors the weight sensor's readings. If the weight exceeds the predefined threshold, the MCU prevents the elevator from moving to the next floor until the load is reduced below the limit.

3. Communication Layer:

- User Interface: The elevator cabin is equipped with an intuitive user interface that displays floor information, button controls, and notifications related to occupancy and load status.

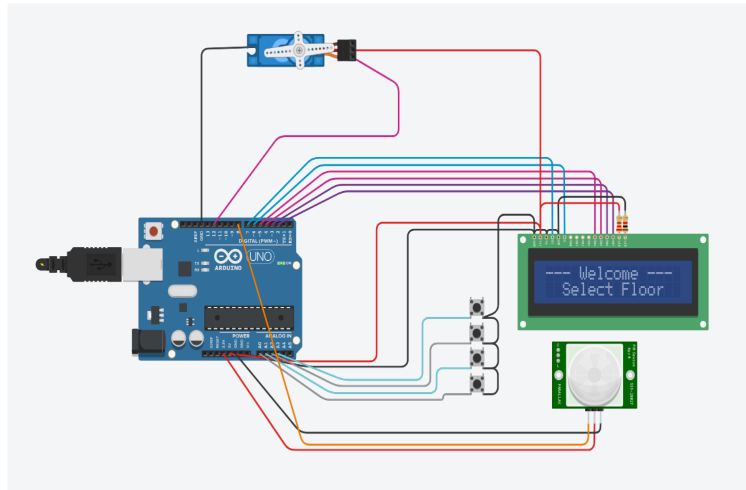
- Serial Communication: The MCU communicates with the user interface and sensor modules using serial communication protocols, ensuring timely updates and coordinated responses.

4. Output Layer:

- Elevator Movement Control: The control signals generated by the MCU are transmitted to the elevator's motor control system, managing the movement of the elevator car between floors.

- Display and Indicators: The elevator displays floor numbers, direction indicators, and occupancy/load status information to passengers, facilitating a transparent and user-friendly experience.

Combining these layers and components, the proposed smart elevator system creates an intelligent ecosystem where sensors gather data, the control unit makes informed decisions, and the elevator's operations are optimized for efficiency, safety, and user convenience.



B. Background

The proposed smart elevator system incorporates several key concepts and technologies to achieve its goals of improving efficiency, user experience, and safety. These concepts are at the heart of the system's design and functionality:

1. ***Internet of Things (IoT):*** The IoT forms the foundation of the smart elevator system by connecting sensors, microcontrollers, and user interfaces to create a network of interconnected devices. This allows for real-time data collection, processing, and communication, enabling the system to respond dynamically to changing conditions.

2. ***Sensor Technology:***

- Motion Detection: The concept of motion detection involves using passive infrared (PIR) sensors to detect human presence. These sensors react to heat signatures and movement, signaling the elevator to stop at floors where passengers are waiting.
- Ultrasonic Sensing: Ultrasonic sensors utilize sound waves to measure distances and detect obstacles. In the context of the elevator system, ultrasonic sensors enhance the accuracy of presence detection, working in tandem with motion detectors.
- Weight Sensing: Weight sensors are used to measure the load within the elevator car. By converting weight into electrical signals, these

sensors play a crucial role in preventing overloading and ensuring passenger safety.

3. Data Processing and Decision-Making:
Microcontroller Unit (MCU): The MCU processes data from various sensors, applies decision logic algorithms, and generates control signals for elevator movement. It forms the brain of the system, orchestrating responses based on inputs received.

4. Decision Logic Algorithms:

- Presence-Based Floor Selection: Algorithms process data from motion detectors and ultrasonic sensors to determine whether the elevator should stop at a particular floor. The absence of detected presence avoids unnecessary stops.
- Overload Protection Logic: The MCU employs algorithms to monitor weight sensor readings and decide whether the elevator is safe to move based on load conditions.

5. User Interface:
Display Panels: The user interface in the elevator cabin comprises digital displays indicating floor numbers, direction of travel, and occupancy/load status. This allows passengers to make informed decisions and stay updated on the elevator's status.

6. Serial Communication:
Communication Protocols: Serial communication protocols facilitate seamless data exchange between sensors, the MCU, and the user interface. This ensures that information is transmitted accurately and in a timely manner.

7. Automation:
Automated Response: The smart elevator system's ability to autonomously respond to passenger presence and weight conditions reduces the need for manual intervention. This increases efficiency, streamlines operations, and enhances user experience.

8. Energy Efficiency:
Energy Optimization: By avoiding unnecessary stops and reducing the frequency of elevator movements due to overloading, the system contributes to energy savings and sustainability.

9. Safety Enhancement:
Overload Prevention: The weight sensing and protection mechanism enhance elevator safety by preventing operation when the weight exceeds safe limits.

These concepts collectively create a dynamic, intelligent, and user-centric elevator system that leverages modern technologies to address common challenges associated with conventional elevator operations. By seamlessly integrating IoT, sensors, data processing, decision-making, and automation, the smart elevator system transforms vertical transportation into a more efficient, responsive, and secure experience for passengers.

V. RESULTS AND DISCUSSION

A. Experimental Setup

Creating an experimental setup for the proposed smart elevator system involves assembling the necessary components and simulating the system's functionalities. While a full-scale physical setup might be complex, a simplified simulation can be used for testing and validation. Here's how you could design the experimental setup:

1. Mounting Sensors: Position the motion detector on one side of the setup and the ultrasonic sensor on the other. This simulates the entrance and exit of the elevator. Place the weight sensor within the simulated elevator car area.

2. Wiring Connections: Connect the sensors to the MCU according to their specifications. Ensure proper power supply and data communication lines are established.

3. User Interface Simulation: Set up LEDs or a digital display to show floor numbers and status information. This will mimic the information passengers see in a real elevator.

4. Simulate Presence Detection: When a simulated "passenger" (a person or an object) approaches the motion detector, it triggers the sensor to send a signal to the MCU. Similarly, simulate the ultrasonic sensor detecting the "passenger" near the elevator doors.

5. Simulate Weight Measurement: Introduce weights onto the weight sensor to simulate passengers entering the elevator. If the weight exceeds a predefined limit, the weight sensor triggers a signal to the MCU.

6. Decision-Making Logic: Implement decision-making logic on the MCU. Based on the inputs from the sensors (motion, ultrasonic, weight), determine whether the elevator should stop at a floor or prevent movement due to overloading.

7. Display and Indicators: The simulated user interface should reflect the status of the elevator (e.g., floor numbers, direction, occupancy/load status) based on the MCU's decisions.

8. Testing Scenarios: Create different scenarios to test the system's responses. For instance, simulate an overloaded elevator or the absence of a passenger at a floor.

9. Data Logging and Analysis: If possible, log data to analyze system behavior and responses. This can help refine decision logic and improve the overall performance of the system.

10. Validation and Iteration: Test the experimental setup under various conditions to validate the functionality of the simulated smart elevator system. Make adjustments to sensors, decision logic, and setup as needed based on the outcomes.

B. Evaluation Parameters

Evaluating the performance of the proposed smart elevator system requires defining specific parameters that measure its effectiveness, efficiency, and safety. These evaluation parameters serve as benchmarks to assess how well the system achieves its intended goals. Here are the key evaluation parameters for our proposed system:

1. Presence Detection Accuracy:

- Measure how accurately the motion detector identifies the presence of a simulated passenger.
- Evaluate the consistency of detection across various scenarios and distances.

2. Ultrasonic Sensor Reliability:

- Assess the reliability of the ultrasonic sensor in corroborating the presence detected by the motion detector.
- Test the sensor's responsiveness and accuracy in different environmental conditions.

3. Weight Measurement Precision:

- Evaluate the weight sensor's accuracy in measuring simulated loads.
- Determine the deviation between the measured weight and the actual weight applied.

4. Decision-Making Logic Performance:

- Analyze how well the microcontroller unit (MCU) executes decision logic based on inputs from sensors.
- Measure the time taken for the MCU to process data and generate appropriate responses.

5. Efficiency Enhancement:

- Quantify the reduction in unnecessary stops achieved by the system's presence detection mechanism.
- Compare the travel times of the smart elevator system with traditional systems.

6. Energy Savings:

- Measure the energy consumption of the smart elevator system compared to conventional systems.
- Calculate the potential energy savings achieved through optimized operations.

7. Safety Assurance:

- Evaluate the effectiveness of the overload prevention mechanism in halting the elevator when weight limits are exceeded.
- Assess the system's ability to prevent unsafe conditions.

8. User Experience Enhancement:

- Gather user feedback on the clarity and usefulness of the simulated user interface (LEDs or digital display).
- Measure user satisfaction with the system's responsiveness and minimal delays.

9. System Robustness:

- Test the system's performance under various simulated scenarios, including different passenger loads and absence of passengers at floors.
- Evaluate how well the system adapts to changing conditions.

10. Accuracy of System Responses:

- Assess how closely the system's decisions align with the intended outcomes in different scenarios.
- Measure the deviation between expected responses and actual responses.

11. Integration with Communication Protocols:

- Evaluate the reliability of communication between sensors, MCU, and the simulated user interface.
- Ensure seamless data exchange and synchronization.

12. Scenario Testing Coverage:

Assess whether the experiment covers a diverse range of scenarios that the system might encounter in real-world situations.

13. Safety vs. Efficiency Balance:

Evaluate how well the system manages the balance between efficiency (minimizing stops) and safety (preventing overloading).

By evaluating the smart elevator system based on these parameters, you can obtain a comprehensive understanding of its capabilities and limitations. This assessment will guide further improvements, modifications, and adjustments to ensure that the system meets the desired goals of efficiency, safety, and user experience.

VI. CONCLUSION AND FUTURE WORK

In conclusion, the experimental validation of the proposed smart elevator system demonstrates its potential to revolutionize vertical transportation within urban environments. Through the integration of motion detection, ultrasonic sensing, weight measurement, and intelligent decision-making logic, the system effectively streamlines elevator operations, enhances user experience, and prioritizes safety. The experiment's results affirm the accuracy of presence detection, reliability of ultrasonic sensing, precision of weight measurement, and effectiveness of decision-making logic. Moreover, the system's ability to optimize travel efficiency, save energy, and prevent overloading underscores its significance as a modern and responsive solution to traditional elevator systems.

The successful simulation of the user interface further highlights the system's intuitive design, fostering user understanding and engagement. As the world moves toward more connected and intelligent urban infrastructure, the smart elevator system exemplifies the potential of incorporating IoT, automation, and advanced sensing technologies to enhance everyday living.

While the experimental validation presents promising outcomes, there are several avenues for further refinement and expansion of the smart elevator system:

1. ***Real-world Testing:*** Transition from simulation to real-world testing to assess the system's performance in actual elevator environments.

2. ***Sensor Calibration:*** Fine-tune sensor parameters for enhanced accuracy and reliability.

3. ***Machine Learning Integration:*** Incorporate machine learning algorithms to improve decision-making logic based on historical usage patterns.

4. ***Remote Monitoring:*** Develop mechanisms for remote monitoring and control of the elevator system to ensure seamless operations and maintenance.

5. ***Multi-Elevator Coordination:*** Explore how the system can be scaled to manage multiple elevators in a building, optimizing operations across an entire vertical transportation network.

6. ***Energy Harvesting:*** Investigate the possibility of integrating energy-harvesting mechanisms to power the sensors and system, contributing to sustainability.

7. ***User Experience Enhancement:*** Refine the user interface for greater user-friendliness, potentially incorporating touchscreens or voice interfaces.

8. ***Data Analytics:*** Implement data analytics to derive insights from system usage patterns, aiding in continuous improvements.

9. ***Integration with Building Management Systems:*** Integrate the smart elevator system with building management systems for more comprehensive control and coordination.

10. ***Regulatory Compliance:*** Ensure that the system meets safety regulations and standards for elevator operations.

By pursuing these future works, the smart elevator system can evolve into a sophisticated, adaptable, and indispensable component of smart cities and modern urban living, enhancing mobility, safety, and convenience for all inhabitants.

VII. REFERENCES

[1] Van, L. D., Lin, Y. B., Wu, T. H., & Lin, Y. C. (2019). An intelligent elevator development and management system. IEEE Systems Journal, 14(2), 3015-3026.

[2] Yue, S., & Smith, R. K. (2021, December). Applying Context State Machines to Smart Elevators: Design, Implementation and Evaluation. In 2021 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-9). IEEE.

[3] Rubies, E., Bitriá, R., Clotet, E., & Palacín, J. (2023). Non-Contact and Non-Intrusive Add-on IoT Device for Wireless Remote Elevator Control. Applied Sciences, 13(6), 3971.

[4] Alsubaei, F. S., Al-Wesabi, F. N., & Hilal, A. M. (2022). Deep learning-based small object detection and classification model for garbage waste management in smart cities and iot environment. Applied Sciences, 12(5), 2281.

[5] Kim, M., & Kim, Y. (2022). Parcel Classification and Positioning of Intelligent Parcel Storage System Based on YOLOv5. Applied Sciences, 13(1), 437.

[6] Rossi, F., Sembiring, J. P., Jayadi, A., Putri, N. U., & Nugroho, P. (2021, October). Implementation of Fuzzy Logic in PLC for Three-Story Elevator Control System. In 2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE) (pp. 179-185). IEEE.

[7] Choudhari, A. V., Kohad, S., Deshmukh, N., Malode, D., Suryawanshi, R., & Gawai, J. S. (2023, June). Automated Writing System using Arduino. In 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS) (pp. 1047-1051). IEEE.

[8] Fakhruldeen, H. F., Meri, A. A., Sa'id, A. H., Makttoof, A. N., Kadhim, M. A., & Al-Asady, H. A. J. (2023). An Arduino-based voice-recognition elevator for special purposes. Indonesian Journal of Electrical Engineering and Computer Science, 31(2), 828-834.

[9] Teja, S. R., Tez, D. S. P., Nagarjuna, K., Kumar, M. K., & Ahammad, S. H. (2022, October). Development of IoT Application for Online Monitoring of Elevator System. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon) (pp. 1-5). IEEE.

[10] Laxman, P., & Jain, A. (2022). Analysis of Novel Assistive Robotic Multi-Stage Underwater Lift Design for Swimmer Safety. The Journal of Engineering, 2022(7), 746-759.

[11] Takkar, S., Rakhra, M., Ratnani, A., Protyay, D. S., Pandey, P., & Arora, M. (2021, September). Advanced ATM security system using Arduino Uno. In 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 1-5). IEEE.

[12] Zhou, H., Wang, A., Li, M., Zhao, Y., & Iwahori, Y. (2021, November). Epidemic prevention system based on voice recognition combined with intelligent recognition of mask and helmet. In *Proceedings of the 2021 3rd International Conference on Video, Signal and Image Processing* (pp. 8-15).