STM JOURNALS

Journal of
Control & Instrumentation

ISSN: 2229-6972 (Online)
ISSN: 2347-7237 (Print)
Volume 16, Issue 2, 2025
May–August
DOI (Journal): 10.37591/JoCI

https://journals.stmjournals.com/joci

Research JoCI

Revolutionizing School Schedules: An Arduino-Based Automatic Class Bell System with Real-Time Precision

Dhyvarkonda Udaykiran Tulshidas[1,*], Pranit Sunil Paul[1], Gone Yashasvi Prakash[1], IR. Kazi Kutubuddin Sayyad Liyakat[2]

Abstract

A sound that evokes nostalgia, signals transitions, and dictates the rhythm of academic life. However, the traditional reliance on manual bell ringing is prone to human error, inconsistencies, and inflexibility. This work explores the design and implementation of an Arduino-based automatic class bell system leveraging a Real-Time Clock (RTC) module to provide accurate and reliable bell timing. This system offers a significant upgrade over manual systems, streamlining school schedules and improving overall efficiency. Traditional methods of ringing the school bell often involve a designated individual manually activating the bell at scheduled times. This project presents the development of an Arduino-based automatic class bell system incorporating a DS3231 Real-Time Clock (RTC) module. The system aims to replace manual bell ringing, offering improved accuracy, consistency, and flexibility in managing school schedules. The RTC module provides precise timekeeping, while the Arduino microcontroller compares the current time with a pre-programmed schedule to trigger the bell. This automated system reduces human error, ensures timely bell ringing, and facilitates easy schedule modifications. The design, implementation, and potential benefits of this system are discussed, highlighting its potential for widespread adoption in educational institutions.

Keywords: Ringing bell, Arduino, real time clock, school, buzzer

INTRODUCTION

The familiar sound of the school bell, a rhythmic clang signaling the start and end of classes, has been a constant in education for generations. But behind that simple sound lies a logistical challenge: ensuring the bell rings at the precise moment, every day, year after year. While manual bell systems have served their purpose, the modern era calls for automation and precision. Enter the RTC, or Real-Time Clock, a small but powerful component that is revolutionizing how schools manage their time and, more importantly, their bell schedules [1–10].

*Author for Correspondence
Dhyvarkonda Udaykiran Tulshidas
E-mail: udaykirandhyvarkonda@gmail.com

[1]Students, Department of Electronics and Telecommunication Engineering, Brahmdevdada Mane Institute of Technology, Solapur, Maharashtra, India
[2]Professor and Head, Department of Electronics and Telecommunication Engineering, Brahmdevdada Mane Institute of Technology, Solapur, Maharashtra, India

Receiving Date: April 15, 2025
Accepted Date: April 29, 2025
Published Date: May 16, 2025

Citation: Dhyvarkonda Udaykiran Tulshidas, Pranit Sunil Paul, Gone Yashasvi Prakash, IR. Kazi Kutubuddin Sayyad Liyakat. Revolutionizing School Schedules: An Arduino-Based Automatic Class Bell System with Real-Time Precision. Journal of Control & Instrumentation. 2025; 16(2): 35-44p.

Traditional school bell systems often rely on manual operation, a responsibility typically delegated to a staff member. This approach is prone to several issues:

- *Human error:* Forgetfulness, distractions, or unexpected absences can lead to missed or incorrect bell times.
- *Inconsistency:* Minor variations in timing can disrupt the flow of classes and student schedules.
- *Inflexibility:* Adjusting the bell schedule for special events, holidays, or exams requires manual intervention and can be cumbersome.
- *Labor intensive:* Dedicating staff time to managing the bell system can be inefficient.

An RTC, as its name suggests, is an integrated circuit that keeps track of time, just like a wristwatch. However, its crucial advantage lies in its ability to retain the correct time even when power is removed, thanks to a small backup battery. This characteristic makes it ideal for automating tasks that require precise timing, like sounding the school bell [11–20]. Implementing an RTC-based bell system involves integrating the RTC with a microcontroller, a relay, and a power source for the bell. Here is a simplified breakdown:

1. *Timekeeping:* The RTC accurately maintains the current date and time.
2. *Programming:* The microcontroller is programmed with the desired bell schedule. This schedule defines the specific times for each bell ring throughout the day, week, and year.
3. *Comparison:* The microcontroller constantly compares the time from the RTC with the programmed bell schedule.
4. *Activation:* When the current time matches a scheduled bell time, the microcontroller activates the relay.
5. *Bell ringing:* The relay closes the circuit to the bell, causing it to ring.
6. *Automation:* This entire process happens automatically, ensuring consistent and precise bell times without any human intervention.

Switching to an RTC-based bell system offers significant advantages:

- *Accuracy and reliability:* Eliminates human error and ensures consistent bell times, leading to a more organized school day.
- *Automation:* Frees up staff time and resources by automating the bell ringing process.
- *Flexibility:* Allows for easy adjustments to the bell schedule via software programming, accommodating special events and holidays with ease.
- *Cost-effectiveness:* The initial investment in hardware and programming is offset by the long-term cost savings from reduced labor and improved efficiency.
- *Remote control and monitoring:* Modern implementation allows to monitor and change the RTC schedule from anywhere through the internet.

While deploying an RTC-based bell system brings numerous benefits, several factors need consideration:

- *Hardware selection:* Choose a reliable RTC module and microcontroller suitable for the school's needs.
- *Programming:* Develop a robust and user-friendly software program for managing the bell schedule.
- *Power supply:* Ensure a stable power supply and backup battery for the RTC to maintain accurate timekeeping.
- *Wiring and installation:* Properly wire the RTC, microcontroller, relay, and bell system to ensure reliable operation.
- *Maintenance:* Regularly check the battery of the RTC and update the programming as needed.
- *Security:* Implement security measures to prevent unauthorized access to the system and prevent malicious schedules to be programmed.

The school bell, a seemingly simple instrument, plays a crucial role in structuring the educational day. By embracing technology and automation with RTC-based bell systems, schools can move beyond the limitations of manual operation, creating a more efficient, reliable, and organized learning environment for students and staff alike. As technology continues to evolve, the traditional school bell is ringing in a new era of precision and automation, powered by the humble yet powerful RTC [21–41].

THE ARDUINO-RTC SOLUTION: A STEP-BY-STEP OVERVIEW

The developed system utilizes the Arduino microcontroller, a versatile and affordable platform, as its central processing unit. The Real-Time Clock (RTC) module, specifically the DS3231, provides accurate timekeeping even when the main power supply is interrupted [42–55]. The system operates based on the following principles (Figure 1):

1. *Time acquisition:* The RTC module continuously tracks the current date and time with high precision, relying on a backup battery to maintain time even during power outages.

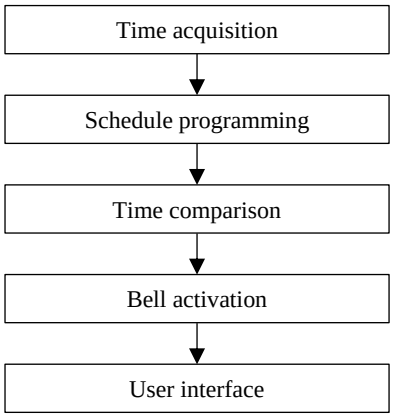

Figure 1. Steps of operating principle.

2. *Schedule programming:* A pre-defined schedule, consisting of bell ringing times, is programmed into the Arduino's memory. This schedule can be customized and modified as needed.
3. *Time comparison:* The Arduino continuously compares the current time obtained from the RTC module with the programmed schedule.
4. *Bell activation:* When the current time matches a scheduled bell ringing time, the Arduino activates a relay. This relay, in turn, controls the power supply to the bell, causing it to ring.
5. *User interface (optional):* The system can be further enhanced with an LCD display to show the current time and the next scheduled bell ringing time, providing visual feedback to users. Buttons can be added for easy schedule adjustments.

METHODOLOGY

In educational institutions, the class bell is a crucial element that signifies the beginning and end of classes, as well as breaks. A manual bell system can be inefficient and prone to errors. This study presents a solution to automate the class bell system using Arduino and a Real-Time Clock (RTC) module.

The Arduino-based automatic class bell system uses an Arduino board, a Real-Time Clock (RTC) module, a buzzer, and a power supply. The RTC module keeps track of time and triggers the buzzer at specified times, according to the pre-programmed class schedule. The Arduino board serves as the microcontroller, executing the program that controls the RTC and the buzzer. Figure 2 shows the proposed system using Arduino.

Components required are as:

1. *Arduino board:* We will use an Arduino Uno board for this project, but other boards like the Arduino Mega or Nano can also be used.
2. *Real-time clock (RTC) module:* We will use the DS1307 RTC module, which is a popular and affordable option. It can be connected to the Arduino board via the I2C protocol.
3. *Buzzer:* A simple active buzzer is sufficient for this project. It can be connected directly to one of the digital pins on the Arduino board.
4. *Power supply:* A stable 5 V power supply is required for the Arduino board and the RTC module. A USB cable or a battery pack can be used as a power source.

Components list:

- Arduino UNO board,
- RTC module (DS1307 or DS3231),
- Piezo buzzer,
- Breadboard and jumper wires,
- 10K resistor, and
- Power supply (9 V battery or USB cable).

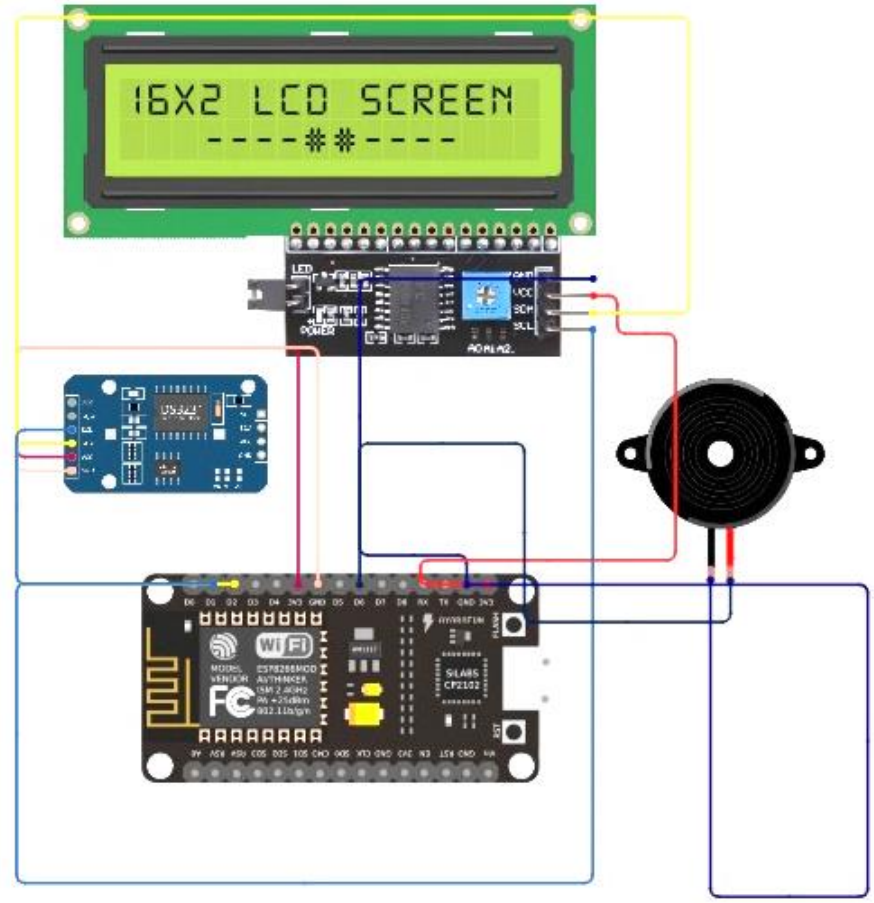

Figure 2. Proposed Bell system.

The wiring for this project is straightforward. Connect the VCC and GND pins of the RTC module to the 5 V and GND pins of the Arduino board, respectively. Connect the SDA and SCL pins of the RTC module to the corresponding pins on the Arduino board (A4 and A5 for the Uno). Finally, connect the positive terminal of the buzzer to one of the digital pins on the Arduino board (e.g., D9) and the negative terminal to the GND pin. The Arduino program for this project involves setting up the RTC module, defining the class schedule, and controlling the buzzer. The following steps outline the programming process:

1. *Install the RTC library:* Download and install the DS1307 RTC library for Arduino from the Arduino Library Manager or the official GitHub repository.
2. *Initialize the RTC module:* Use the RTC library functions to initialize the RTC module and set the current date and time.
3. *Define the class schedule:* Create a data structure or an array to store the class schedule, including the start and end times for each class and break.
4. *Monitor the RTC:* Continuously monitor the RTC module for time updates and compare the current time with the class schedule.
5. *Trigger the buzzer:* When the current time matches a class start or end time, trigger the buzzer for a predefined duration (e.g., 5 sec).

An Arduino-based automatic class bell system using a Real-Time Clock (RTC) module offers a reliable, efficient, and customizable solution for educational institutions. This system eliminates the need for manual intervention and reduces the chances of errors in the class bell schedule. With a few components and some programming, you can create a functional class bell system that can be easily adapted to various class schedules and timings.

Software Development

The Arduino code is written using the Arduino IDE. Key aspects of the code include:

- *RTC library:* Utilizing a library designed for the DS3231 RTC module to efficiently read and manage time data.
- *Schedule management:* Implementing a data structure (e.g., an array or struct) to store the bell ringing schedule.
- *Time comparison algorithm:* Developing a robust algorithm to accurately compare the current time with the scheduled times.
- *Relay control:* Configuring the Arduino to control the relay module for activating the bell.
- *User interface code (optional):* Writing code to display current time and schedule information on the LCD and to handle button presses for schedule adjustments.

The first step is to program the Arduino to control the class bell using the RTC module. We will use the Arduino IDE to write and upload the code to the Arduino board.

Step 1: Install the RTC library: To use the RTC module with Arduino, we need to install the RTC library. Go to Sketch > Include Library > Manage Libraries, and search for "RTClib". Install the library created by Adafruit.

Step 2: Write the code: We will write the code to set the time, define the class periods, and control the class bell using the RTC module and the piezo buzzer. Here is the code:

```
#include <Wire.h>
#include <RTClib.h>

RTC_DS1307 rtc;

const int buzzer = 9;
const int numClasses = 5;
int classDurations[numClasses] = {45, 45, 45, 45, 60}; // class durations in minutes
int classBells[numClasses] = {1, 2, 3, 4, 5}; // class bell numbers
int classStartMinutes[numClasses] = {0, 45, 90, 135, 180}; // class start minutes
int currentClass = 0;

void setup() {
  Wire.begin();
  rtc.begin();

  if (! rtc.isrunning()) {
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }

  pinMode(buzzer, OUTPUT);
}

void loop() {
  DateTime now = rtc.now();
  int currentMinute = now.minute();

  if (currentMinute == classStartMinutes[currentClass]) {
    ringBell(classBells[currentClass]);
    currentClass++;
```



```
    if (currentClass >= numClasses) {
      currentClass = 0;
    }
  }

  delay(1000);
}

void ringBell(int bellNumber) {
  for (int i = 0; i < bellNumber; i++) {
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
    delay(500);
  }
}
```

Step 3: Upload the code: Connect the Arduino board to your computer using a USB cable. Select the correct board and port in the Tools menu of the Arduino IDE. Click the Upload button to upload the code to the Arduino board.

The RTC module keeps track of the time, even when the Arduino board is powered off. When the Arduino board is powered on, the setup function initializes the RTC module and sets the time if it is not running. The loop function checks the current minute and compares it to the class start minutes. When the current minute matches the class start minute, the ringBell function is called to ring the class bell.

The ringBell function uses a for loop to ring the bell for a specified number of times. The number of times the bell rings is determined by the class bell number. The delay function is used to control the duration of the bell sound.

In this study, we have explored how to create an Arduino-based automatic class bell system using a Real-Time Clock (RTC) module. This project is easy to build and can be customized to suit the needs of any institution. With this system, you can manage class schedules efficiently and effectively, improving the learning experience for students.

DISCUSSION

The resounding ring of the school bell is a familiar and sometimes dreaded sound. It dictates the rhythm of the day, signaling the start and end of classes, breaks, and ultimately, the end of the school day. But managing that bell, especially in schools without sophisticated infrastructure, can be a tedious and potentially unreliable manual task. Enter the Arduino-based automatic class bell system with a Real-Time Clock (RTC), a cost-effective and surprisingly powerful solution that is ringing in efficiency and accuracy for schools of all sizes. Traditionally, managing the school bell involves either a person manually ringing the bell or relying on a basic timer. Both methods have their drawbacks:

- *Human error:* Relying on a person introduces the possibility of human error. The bell might ring late, early, or even be forgotten entirely, disrupting the school schedule and potentially causing confusion.
- *Inconsistency:* Even with the best intentions, manually ringing a bell can be inconsistent. The timing might vary slightly each day, leading to minor disruptions.
- *Timer limitations:* Basic timers often lack flexibility. They struggle to handle variations in the schedule, such as shortened days, special events, or holidays. They might also require frequent resetting, which can be inconvenient.

- *Power outages:* Traditional timers reliant on mains power are susceptible to power outages, leading to the bell being silent when it is needed most.

An Arduino-based automatic class bell system, integrated with an RTC, offers a far superior and more reliable alternative. Here is why it is a smart choice:

- *Accuracy and reliability:* The RTC module ensures highly accurate timekeeping, even when the Arduino is powered off. This guarantees the bell rings precisely on schedule every time, eliminating the risk of human error.
- *Customizable schedules:* The Arduino is programmable, allowing for highly customizable bell schedules. Different schedules can be programmed for weekdays, weekends (for after-school activities), special events, shortened days, and holidays. This flexibility ensures the bell system adapts to the school's needs.
- *Simple programming and maintenance:* The Arduino IDE is relatively easy to learn and use. This allows school staff to easily modify the bell schedule as needed. Once programmed, the system requires minimal maintenance.
- *Cost-effectiveness:* Compared to expensive, proprietary bell systems, an Arduino-based solution is significantly more affordable. The required components (Arduino, RTC module, relay, bell) are readily available and relatively inexpensive.
- *Power outage resilience:* Many RTC modules include a battery backup, ensuring the time is maintained even during a power outage. This means the bell system will continue to function as scheduled, preventing disruptions.
- *Open source and adaptable:* The open-source nature of Arduino means the system can be further customized and expanded. You could integrate it with a web interface for remote control, add a display to show the current schedule, or even incorporate a PA system announcement before the bell rings.

The core components of an Arduino-based automatic class bell system are:

1. *Arduino microcontroller:* The brains of the system, responsible for running the program and controlling the other components.
2. *RTC module:* Keeps accurate time and date, even when the Arduino is powered off.
3. *Relay:* Acts as a switch, allowing the Arduino to turn the bell on and off.
4. *Bell:* The device that produces the actual ringing sound.

The Arduino program works by constantly checking the current time from the RTC module. When the current time matches a scheduled bell ringing time, the Arduino activates the relay, which in turn closes the circuit and rings the bell.

CONCLUSION

The Arduino-based automatic class bell system, combined with the precision of the RTC module, offers a cost-effective and reliable solution for managing school schedules. Its ease of implementation, flexibility, and accuracy make it a significant improvement over traditional manual systems. By automating the bell ringing process, schools can enhance efficiency, minimize distractions, and ensure a smooth and predictable academic environment. The potential for future enhancements further solidifies its position as a valuable tool for modern educational institutions. The successful development and implementation of the Arduino-based automatic class bell system incorporating the DS3231 RTC module demonstrates a viable alternative to traditional manual methods. The system's accuracy in timekeeping, programmed schedule adherence, and reliable bell activation showcase its potential to improve efficiency and reduce errors in managing school schedules. This project highlights the benefits of using microcontroller-based solutions for automating routine tasks in educational settings. Further research and development could focus on exploring cloud integration, remote management capabilities, and advanced scheduling features to further enhance the system's functionality and applicability in diverse educational contexts.

REFERENCES

1. Raut A, Mali M, TM, KKS. Bagasse Level Monitoring System. International Journal of Trend in Scientific Research and Development (IJTSRD). 2018 Apr 30; 2(3): 1657–9.
2. Shinde VB, Wandre SS. Solar photovoltaic water pumping system for irrigation: A review. Afr J Agric Res. 2015 May 28; 10(22): 2267–73.
3. Mulani AO, Bang AV, Birajadar GB, Deshmukh AB, Jadhav HM, Liyakat KK. IoT Based Air, Water, and Soil Monitoring System for Pomegranate Farming. Ann Agri-Bio Res. 2024; 29(2): 71–86.
4. Parihar B, Kiran A, Valaboju S, Rashid SZ, Liyakat KK, DR AS. Enhancing Data Security in Distributed Systems Using Homomorphic Encryption and Secure Computation Techniques. In ITM Web of Conferences, EDP Sciences. 2025; 76: 02010.
5. Liu H, Crespo RG, Martínez OS. Enhancing privacy and data security across healthcare applications using blockchain and distributed ledger concepts. Inhealthcare. 2020;8:243. doi:10.3390/healthcare8030243. PMID: 32751325.
6. Liyakat KK, Khadake SB, Tamboli DA, Sawant VA, HM M, Sathe S. AI-Driven-IoT (AIIoT) Based Decision-Making-KSK Approach in Drones for Climate Change Study. In 2024 IEEE 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS). 2024 Dec 12; 1735–1744.
7. Prasad KR, Karanam SR, Ganesh D, Liyakat KK, Talasila V, Purushotham P. AI in public-private partnership for IT infrastructure development. J High Technol Manag Res. 2024 May 1; 35(1): 100496.
8. Liyakat KK. Detecting malicious nodes in IoT networks using machine learning and artificial neural networks. In 2023 IEEE International Conference on Emerging Smart Computing and Informatics (ESCI). 2023 Mar 1; 1–5.
9. Kasat K, Shaikh N, Rayabharapu VK, Nayak M, Liyakat KK. Implementation and recognition of waste management system with mobility solution in smart cities using Internet of Things. In 2023 IEEE Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS). 2023 Aug 23; 1661–1665.
10. Liyakat KK. Smart Grid Energy Saving Technique Using Machine Learning. Journal of Instrumentation Technology & Innovation (JoITI). 2022; 16(3): 1–10.
11. Kazi K. AI-driven IoT (AIIoT) in healthcare monitoring. In: Nguyen TVT, Vo NTM, editors. Using Traditional Design Methods to Enhance AI-Driven Decision Making. Pennsylvania, United States: IGI Global Scientific Publishing; 2024. p. 77–101. doi:10.4018/979-8-3693-0639-0.ch003.
12. Kazi K. Modelling and simulation of electric vehicle for performance analysis: BEV and HEV electrical vehicle implementation using Simulink for E-mobility ecosystems. In: L D, Nagpal N, Kassarwani N, Varthanan G V, Siano P, editors. E-Mobility in Electrical Energy Systems for Sustainability. Pennsylvania, United States: IGI Global; 2024. p. 295–320. doi:10.4018/979-8-3693-2611-4.ch014.
13. Kazi KSL, Shinde SS, Nerkar PM, Kazi SS, Kazi VS. Machine learning for brand protection: A review of a proactive defense mechanism. In: Khan MI, Haq MAU, editors. Avoiding Ad Fraud and Supporting Brand Safety: Programmatic Advertising Solutions. Pennsylvania, United States: IGI Global Scientific Publishing; 2025. p. 175–220. doi:10.4018/979-8-3693-7041-4.ch007.
14. Tarnanidis TK, Papachristou E, Karypidis M, Manda VK, editors. Sustainable Practices in the Fashion and Retail Industry. Pennsylvania, United States: IGI Global; 2025.
15. Kazi KSL. AI-driven IoT (AIIoT)-based decision-making system for high BP patient healthcare monitoring: KSK1 approach for BP patient healthcare monitoring. In: Mzili T, Arya AK, Pamucar D, Shaheen M, editors. Optimization, Machine Learning, and Fuzzy Logic: Theory, Algorithms, and Applications. Pennsylvania, United States: IGI Global Scientific Publishing; 2025. p. 71–102. doi:10.4018/979-8-3693-7352-1.ch003.
16. Kazi KSL. Advancing towards sustainable energy with hydrogen solutions: Adaptation and challenges. In: Özsungur F, Semsari MC, Bayraktar HK, editors. Geopolitical Landscapes of Renewable Energy and Urban Growth. Pennsylvania, United States: IGI Global Scientific Publishing; 2025. p. 357–394. doi:10.4018/979-8-3693-8814-3.ch013.

17. Kazi KSL. Machine learning-based pomegranate disease detection and treatment. In: Haq MZU, Ali I, editors. Revolutionizing Pest Management for Sustainable Agriculture. Pennsylvania, United States: IGI Global; 2024. p. 469–498. doi:10.4018/979-8-3693-3061-6.ch019.
18. Garcia MB, de Almeida RP, editors. Transformative Approaches to Patient Literacy and Healthcare Innovation. Pennsylvania, United States: IGI Global; 2024.
19. Kazi KSL. IoT driven by machine learning (MLIoT) for the retail apparel sector. In: Tarnanidis TK, Papachristou E, Karypidis M, Ismyrlis V, editors. Driving Green Marketing in Fashion and Retail. Pennsylvania, United States: IGI Global; 2024. p. 63–81. doi:10.4018/979-8-3693-3049-4.ch004.
20. Kazi KSL. Artificial intelligence (AI)-driven IoT (AIIoT)-based agriculture automation. In: Satapathy S, Muduli K, editors. Optimization, Machine Learning, and Fuzzy Logic: Theory, Algorithms, and Applications. Pennsylvania, United States: IGI Global; 2024. p. 72–94. doi:10.4018/979-8-3693-3583-3.ch005.
21. Kazi KSL. Machine learning (ML)-based Braille Lippi characters and numbers detection and announcement system for blind children in learning. In: Sart G, editor. Social Reflections of Human-Computer Interaction in Education, Management, and Economics. Pennsylvania, United States: IGI Global; 2024. p. 16–39. doi:10.4018/979-8-3693-3033-3.ch002.
22. Kazi KSL. Machine Learning-Driven Internet of Medical Things (ML-IoMT)-Based Healthcare Monitoring System. In: Soufiene BO, Chakraborty C, editors. Responsible AI for Digital Health and Medical Analytics. Pennsylvania, United States: IGI Global; 2024. p. 49–86. doi:10.4018/979-8-3693-6294-5.ch003.
23. Kazi KSL. Transformation of Agriculture Effectuated by Artificial Intelligence-Driven Internet of Things (AIIoT). In: Garwi J, Dzingirai M, Masengu R, editors. Integrating Agriculture, Green Marketing Strategies, and Artificial Intelligence. Pennsylvania, United States: IGI Global; 2024. p. 449–484. doi:10.4018/979-8-3693-6468-0.ch015.
24. Sayyad Liyakat KK, Konnur RG. Vehicle Health Monitoring System (VHMS) by Employing IoT and Sensors. Grenze International Journal of Engineering & Technology (GIJET). 2024 Jun 15; 10: 5367–5374.
25. Liyakat KK, Kutubuddin K. A Novel Approach on ML based Palmistry. In15th International Conference on Advances in Computing, Control, ad Telecommunication Technologies, ACT 2024. 2024 Jun 15; 2.
26. Liyakat KK, Kutubuddin K. IoT based boiler health monitoring for sugar industries. In 15th International Conference on Advances in Computing, Control, ad Telecommunication Technologies, ACT 2024. 2024 Jun 15; 2: 5178–5185.
27. Keerthana R, Bhagyalakshmi K, Papinaidu M, Liyakat KK. Machine Learning Based Risk Assessment for Financial Management in Big Data IoT Credit. Machine Learning Based Risk Assessment for Financial Management in Big Data IoT Credit (November 15, 2024). 2024 Nov 15.
28. Saraswat D, Bhattacharya P, Verma A, Prasad VK, Tanwar S, Sharma G, Bokoro PN, Sharma R. Explainable AI for healthcare 5.0: Opportunities and challenges. IEEE Access. 2022;10:84486–84517. doi:10.1109/ACCESS.2022.3197671.
29. Kazi KSL. Machine learning (ML)-based Braille Lippi characters and numbers detection and announcement system for blind children in learning. In: Sart G, editor. Social Reflections of Human-Computer Interaction in Education, Management, and Economics. Pennsylvania, United States: IGI Global; 2024. p. 16–39. doi:10.4018/979-8-3693-3033-3.ch002.
30. Liyakat KK. Machine learning approach using artificial neural networks to detect malicious nodes in IoT networks. In International Conference on Machine Learning, IoT and Big Data. Singapore: Springer Nature Singapore; 2023 Mar 10; 123–134.
31. Wohiduzzaman K, Ullah R, Sharkar A. Design and development of a low cost smart electronic school bell. Int J Latest Technol Eng Manag Appl Sci. 2024;13(10):7–11. doi:10.51583/IJLTEMAS.2024.131002.
32. Kazi KSL. IoT Technologies for the Intelligent Dairy Industry: A New Challenge. In: Thandekkattu SG, Vajjhala NR, editors. Designing Sustainable Internet of Things Solutions for Smart Industries. Pennsylvania, United States: IGI Global; 2024. p. 321–350. doi:10.4018/979-8-3693-5498-8.ch012.

33. Liyakat KKS. Heart Health Monitoring Using IoT and Machine Learning Methods. In: Shaik A, editor. AI-Powered Advances in Pharmacology. Pennsylvania, United States: IGI Global; 2024. p. 257–282. doi:10.4018/979-8-3693-3212-2.ch010.
34. Kazi KSL. AI-Driven-IoT (AIIoT)-Based Decision Making in Kidney Diseases Patient Healthcare Monitoring: KSK Approach for Kidney Monitoring. In: Polat LÖ, Polat O, editors. AI-Driven Innovation in Healthcare Data Analytics. Pennsylvania, United States: IGI Global; 2024. p. 277–306. doi:10.4018/979-8-3693-7277-7.ch009.
35. Liyakat KK. Machine learning approach using artificial neural networks to detect malicious nodes in IoT networks. In International Conference on Machine Learning, IoT and Big Data. Singapore: Springer Nature Singapore; 2023 Mar 10; 123–134.
36. Biradar PK, Pardeshi YS, Bagwan IA, Kazi TI, Ganji SS. Remotely Operated Video Enhanced Receiver. Int Adv Res Sci Commun Technol. 2025 Jan; 5(2): 696–707.
37. Kazi KSL, Mahant MA. Machine Learning-Driven Internet of Things (MLIoT)-Based Healthcare Monitoring System. In: Wickramasinghe N, editor. Digitalization and the Transformation of the Healthcare Sector. Pennsylvania, United States: IGI Global; 2024. p. 205–236. doi:10.4018/979-8-3693-9641-4.ch007.
38. Mulani AO, Liyakat KK, Warade NS, Patil A, Kolte MT, Kinage K, Rana M, Salunkhe SS, Jadhav VS, Nagrale M. WITHDRAWL–Administrative Duplicate Publication: ML-powered Internet of Medical Things Structure for Heart Disease Prediction. *J Pharmacol Pharmacother*. 2025; 16(2): 243.
39. Odnala S, Shanthy R, Bharathi B, Pandey C, Rachapalli A, Liyakat KK. Artificial Intelligence and Cloud-Enabled E-Vehicle Design with Wireless Sensor Integration. Available at SSRN 5107242. 2024 Nov 15.
40. Neeraja P, Kumar RG, Kumar MS, Liyakat KK, Vani MS. DL-based somnolence detection for improved driver safety and alertness monitoring. In 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT). 2024 Feb 9; 5: 589–594.
41. Sayyad Liyakat KK, Magadum PK. Machine Learning for Predicting Wind Turbine Output Power in Wind Energy Conversion Systems. Grenze International Journal of Engineering & Technology (GIJET). 2024 Jan 22; 10: 2074–2080.
42. Nerkar PM, Dhaware BU, Liyakat KS. Predictive data analytics framework based on heart healthcare system (HHS) using machine learning. J Adv Zool. 2023; 44(2): 3673–3686.
43. Kutubuddin KA, Nerkar PR, Sultanabanu KA. IoT-based skin health monitoring system. Int J Biol Pharm Allied Sci. 2024; 13(11): 5937–50.
44. Liyakat KK, Khadake SB, Chounde AB, Suryagan AA, HM M, Khadatare MR. AI-Driven-IoT (AIIoT) Based Decision Making System for High-Blood Pressure Patient Healthcare Monitoring. In 2024 IEEE International Conference on Sustainable Communication Networks and Application (ICSCNA). 2024 Dec 11; 96–102.
45. Zoha A, Qadir J, Abbasi QH. AI-Powered IoT for Intelligent Systems and Smart Applications. Front Commun Netw. 2022 Jul 7; 3: 959303.
46. Spagnolo P. Advancements in Precision Agriculture: Integrating AI and IoT for Smart Crop Monitoring and Management. European Journal of Crop Science and Technology. 2025 Mar 6; 1(01): 21–9. P-ISSN 3051–0139 en E-ISSN 3051-0147.
47. Kazi KSL. KK Approach to Increase Resilience in Internet of Things: A T-Cell Security Concept. In: Darwish D, Charan K, editors. Analyzing Privacy and Security Difficulties in Social Media: New Challenges and Solutions. Pennsylvania, United States: IGI Global; 2024. p. 87–120. doi:10.4018/979-8-3693-9491-5.ch005.
48. Uprety A, Rawat DB, Sadler B. Human Immune System Inspired Security for Federated Learning-Empowered Internet of Things. ACM Trans Internet Things. 2025; 6(2): 11(18p).
49. Kazi KSL. Healthcare monitoring system driven by machine learning and Internet of medical things (MLIoMT). In: Kumar VV, Katina PF, Zhao J, editors. Convergence of Internet of Medical Things (IoMT) and Generative AI. Pennsylvania, United States: IGI Global Scientific Publishing; 2025. p. 385–416. doi:10.4018/979-8-3693-6180-1.ch016.

50. Poudel YK, Pudasaini RK, Kandel A, Karki NR. Battery, Super Capacitor-Based Hybrid Energy Storage with PV for Islanded DC Microgrid. Arch Adv Eng Sci. 2025; 3(2): 100–110.
51. Ribeiro C, Gil J, Bento X, Figueiredo M, Rosa J, Antonio R, Ferrari L, Assuncao P. VIEXPAND – Industrial Video for Expanded Vision in Remote Operations. Int J Ind Eng Manag. 2024;15:291–303. doi:10.24867/IJIEM-2024-4-364.
52. Kazi L, Sayyad K. ChatGPT: An automated teacher's guide to learning. In: Bansal R, editor. AI Algorithms and ChatGPT for Student Engagement in Online Learning. Hershey (PA): IGI Global; 2024. p. 1–20. doi:10.4018/979-8-3693-4268-8.ch001.
53. Gulzar B, Ahmad Sofi S, Sholla S. Cognitive transformation in personal IoT: Pioneering intelligent automation. Cyber-Physical Systems (CPS). 2025 Apr 3; 11(2): 183–240.
54. Chandane ER, Patil A, Bokephode P, Shete A, Kulkarni A, Rajwani D. Landmine Detecting Robot. Int Adv Res Sci Commun Technol. 2025 Jan; 5(1): 208–219.
55. Kazi KS. Transformation of Agriculture Effectuated by Artificial Intelligence-Driven Internet of Things (AIIoT). In: Integrating Agriculture, Green Marketing Strategies, and Artificial Intelligence. IGI Global Scientific Publishing; 2025; 449–484.

