

University of Cincinnati

Date: 5/28/2023

I, Sunkanmi Oluwadare, hereby submit this original work as part of the requirements for the degree of Master of Science in Information Technology.

It is entitled:

A SMART AND SECURE CLOUD-INTEGRATED IoT FRAMEWORK FOR HEALTHCARE APPLICATIONS

Student's name: **Sunkanmi Oluwadare**

This work and its defense approved by:

Committee chair: Zaghoul Elsayed, Ph.D.

Committee member: Nelly Elsayed, Ph.D.

Committee member: M. Murat Ozer, Ph.D.



45602

**A SMART AND SECURE CLOUD-INTEGRATED IoT FRAMEWORK FOR
HEALTHCARE APPLICATIONS.**

A thesis submitted to the Graduate School of the University of Cincinnati in partial fulfillment of the requirements for the degree of

Master of Science

In the School of Information Technology of the College of Education, Criminal Justice, and Human Services

By

Sunkanmi Oluwadare

Thesis Advisor and Committee Chair: Zag ElSayed, Ph.D.

Committee Members: Nelly ElSayed, Ph.D.

Murat Ozer, Ph.D.

University of Cincinnati

2023

Abstract

This research proposes a cost-effective and reliable IoT-based system for real-time temperature monitoring using NodeMCU ESP8266 and ThingSpeak Cloud. The system can be deployed in healthcare settings, such as hospitals, clinics, and nursing homes, where accurate and reliable temperature monitoring is critical for patient care and comfort. The proposed system is capable of providing real-time environmental temperature readings and storing data on the cloud for remote access and analysis. In the proposed system, one ESP8266 is configured as an access point that wirelessly transmits temperature readings from a DS18B20 sensor to another ESP8266 configured as a client. The client ESP8266 sends the data to the ThingSpeak Cloud to display temperature readings and for analysis. The system can be integrated with medical alert systems to provide timely notifications to medical staff in case of any abnormalities. Overall, this system provides a cost-effective, reliable, and scalable solution for real-time temperature monitoring in healthcare settings.

Keywords

IoT, Cloud Computing, Sensors, Smart Healthcare, Security, Arduino.

Acknowledgements

I am deeply grateful to Dr. Zag ElSayed, my advisor, for the unwavering support and guidance throughout the completion of this thesis. His expertise, insightful feedback, and constructive corrections have been invaluable in shaping the direction and quality of this research.

I would also like to extend my heartfelt appreciation to the members of my thesis committee, Dr. Nelly ElSayed and Dr. Murat Ozer. Their valuable insights, thoughtful suggestions, and critical analysis have significantly contributed to the completion and refinement of this work. Their expertise and commitment to academic excellence have been inspiring and greatly appreciated.

Furthermore, I would like to express my sincere gratitude to my friends and family for their unwavering support, encouragement, and understanding throughout this challenging journey. Their belief in my abilities and their constant motivation have been instrumental in my perseverance and ultimate success.

Lastly, I am also grateful to the faculty and staff of the School of Information Technology, University of Cincinnati for providing a conducive academic environment, resources, and opportunities for intellectual growth. Their dedication to fostering a vibrant research community has greatly enriched my academic experience.

Contents

CHAPTER 1	1
Introduction	1
1.1 Significance	1
CHAPTER 2	3
Existing Work	3
CHAPTER 3	7
Methodology	7
3.1 System Set Up	8
3.2 Hardware Overview	8
3.3 Microcontroller	10
3.4 Sensor	12
3.5 Hardware Block Diagram	13
3.6 Software Overview	13
3.7 Arduino IDE (Integrated Development Environment)	14
3.8 ThingSpeak	15
3.9 System Security	16
3.10 Proposed Framework	17
CHAPTER 4	18
Experimental Procedure:	18
4.1 Experimental Results:	20
CHAPTER 5	23
5.1 Challenges and Future Direction	23
5.2 Conclusion	24
REFERENCES	25

List of Figures

Figure 1: Shows the Jumper wires used for hardware set up connections.....	8
Figure 2: Shows the 4.7k ohm resistor used for voltage stability.	9
Figure 3: Shows the ESP 8266 hardware set up with DS18B20.....	10
Figure 4: Pictorial Representation of the ESP 8266 used.	11
Figure 5: Pictorial Representation of the DS18B20.	12
Figure 6: Proposed System Framework set up.....	13
Figure 7: The Arduino IDE Interface	14
Figure 8: ThingSpeak Interface	15
Figure 9: Security set up viewed over the ESP8266 Client's web server.....	16
Figure 10: Represents Denies Access if Security Key is Invalid.....	19
Figure 11: Temperature readings displayed after decrypting with Security Key.	19
Figure 12: Displays Connection Established with ThingSpeak.	20
Figure 13: Temperature readings Charts in Celsius.....	21
Figure 14: Temperature readings Charts in Fahrenheit.	21

CHAPTER 1

Introduction

The Internet of Things (IoT) is the term that defines the network of devices, physical objects, vehicles, buildings, and other items which communicate with electronics, software sensors, and network connectivity that allows these objects to gather and exchange data. Christos et.al (2018). The Internet of Things utilizes the Internet as the foundation to enable intelligent communication between people and things. A large volume of data can be generated by sensors, networks, and applications when connected. According to Christos et al. (2018), the Internet of Things is made up of three main parts: The things, the networking that enables the communication, and the systems that send data across data points.

The Internet of Things, or IoT, has grown in popularity as these technologies are employed for a variety of applications, including communication, transportation, education, and commercial development. IoT introduced the concept of hyperconnectivity, which means that businesses and individuals may easily communicate with one another from faraway areas. (Tawalbeh et al., 2020).

IoT systems are reforming lives and activities by providing interconnectivity, ubiquitous computing resources, and automated control. Zheng Fang et al., (2021). Also, IoT systems allow 24/7 patient monitoring, increased visibility into health facts, collection of health records, detection of critical illness at the right time, remote monitoring, generation of statistical data, processing of data, and facilitating emergency services. Valanarasu, (2019). One of the most important benefits of IoT is its ability to gather and analyze massive amounts of data in real time. This data can assist organizations and individuals in making more informed decisions, improving efficiency, and lowering expenses. IoT devices, for example, can help organizations discover areas of energy waste and optimize energy usage to decrease costs by collecting data on energy consumption. They can also help in aiding real time monitoring in healthcare systems.

The combination of IoT and Cloud computing facilitates the advancement of storage capacity and computational resources that could impact or enable healthcare to offer valuable services. Cloud computing is defined as a network of computing services that promote ubiquitous access to networks

of computing assets and resources over the internet. Waleed et al. (2022). Cloud computing provides access to services such as storage, computing, networking, and applications over the Internet. According to *Christos et al. (2018)*, Cloud computing has some specific features, which include: **Storage over the Internet**, which describes the provision of storage technologies over network protocols to facilitate storage solution deployment. **Service over the Internet**, which describes rendering solutions through its great computing resources. **Applications over the Internet** describe applications that are offered to users to harness, access, or complete a task. **Energy Efficiency**, which describes offering its computational power by delivering more services for the same input while offering users to harness added advantages. **Computationally Capable**, which describes leveraging the computational resources the cloud provides to users.

However, with the advancement of technology, IoT presents various concerns and challenges, such as security, privacy, and interoperability. Because IoT devices are interconnected, they are vulnerable to cyber-attacks, and the data collected by these devices can be a target of cybercriminals. As a result, it is important that IoTs are built with security in mind and that proper measures are put in place to safeguard them from cyber threats.

The security of IoT is concerned with safeguarding connected networks and devices that transmit and share data on the Internet of Things. IoT in smart health care largely focuses on sensors, security, wearable devices, big data, emergency services, lab on a chip, monitors, and connectivity. The security requirements of IoT in healthcare include but are not limited to data confidentiality, data privacy, resiliency, location privacy, authentication, and many more. Valanarasu, (2019).

Security is a major requirement in the deployment of IoTs, there is a need to coordinate security between data centers and distributed IoT devices. The security mechanisms employed in the framework will determine the success of IoT applications. Some common security challenges faced by IoT application frameworks include Insecure communication between devices and the cloud, authentication, access control mechanisms, data breaches, and compliance. (Ammar et al., 2018)

The combination of IoT and cloud computing has facilitated the advancement of storage capacity and

computational resources that can impact or enable healthcare to offer valuable services. Cloud computing provides users with many benefits, including storage over the internet, service over the Internet, applications over the Internet, energy efficiency, and computational capabilities. Several challenges and difficulties have often been faced in the deployment of IoT and Cloud computing. One of the challenges faced is how software systems frameworks are designed, which affects the overall performance of the systems. Also, stabilization has also been identified as a problem, as data needs to be stabilized for interoperability. Waleed et al. (2022).

This paper aims to explore a secured framework for the integration of IoT and cloud computing to create a real-time room temperature monitoring system for healthcare using NodeMCU ESP8266 and ThingSpeak Cloud. The proposed system will allow for 24/7 patient monitoring, increased visibility into health facts, collection of health records, and detection of critical temperature data at the right time. The study will focus on developing a reliable and effective system that provides valuable services to the healthcare industry.

1.1 Significance

The significance of the research is aimed at the potential to transform how healthcare is delivered and managed by developing a smart and integrated framework that utilizes IoT devices and cloud computing to enhance patient outcomes and optimize resource utilization. Environmental temperature monitoring systems are important for patient comfort and safety because of the risk of dehydration, heat exhaustion and infection control. The study will contribute to the advancement of healthcare by providing a real-time environmental temperature monitoring system that enables remote monitoring, data collection, analysis, and timely detection of critical temperature conditions while prioritizing security by adding simple key encryption. With the advancement of connected devices, IoT devices have been prone to security breaches. The challenge posed by the homogenous nature of IoTs also has contributed largely to the magnitude of vulnerabilities found in a lot of IoT devices. (Tawalbeh et al., 2020).

When it comes to IoT devices and healthcare, security is a major problem. Data breaches can have serious implications, including the compromise of sensitive patient information, legal and ethical difficulties, and reputational damage to a healthcare company. As a result, the proposed framework

contains security mechanisms to protect the security of data collected and sent between devices and the cloud. By limiting illegal access and tampering, encryption techniques can protect data confidentiality, integrity, and availability.

Additionally, the study's framework will facilitate the deployment of ubiquitous computing resources and automated control, providing valuable insights into health facts and generating statistical data. The study's findings will be useful in developing future IoT-based healthcare solutions that could offer efficient and cost-effective services to patients and healthcare providers. By optimizing resources, healthcare can reduce operational costs and improve patient satisfaction by offering more tailored and personalized care.

CHAPTER 2

Existing Work

For the purpose of this paper, we studied and analyzed existing research in the field of the Internet of Things, Cloud computing, the benefits, and disadvantages in terms of security challenges, security frameworks, common features, existing solutions, and their integration with an emphasis on wireless sensors, and temperature sensor which can improve healthcare. The following paragraphs present the research works and papers that contributed to our study.

Christos et al. (2018) performed a survey of IoT and Cloud Computing with a focus on the security issues of both technologies. The research also shows the benefits of their integration which include eliminating the need for hardware equipment and providing computing logistical features as well as software capabilities. The survey also identified some of the security challenges of integration, it includes heterogeneity, performance, reliability, big data, and monitoring. Conclusively, the authors proposed an efficient security model based on AES & RSA encryption algorithms to handle the security challenges that have been identified with the integrations of IoTs and Cloud Computing.

Valanarasu, (2019) presented a smart and secure framework for healthcare systems using IoT and AI. The system was based on the basic network environment, application framework, logic structure, and data security. This integration and framework addressed the existing drawbacks of treatment, diagnosis, patient monitoring, and maintenance of hospital records. The system consists of an application and services layer that includes management applications and decision and health information applications. The paper contributed to the successful design of smart hospitals using IoT and AI and security, which contributes to IoT-based healthcare that works on surgical robots, wearable devices, and other technology devices.

Anuradha, M., et al. (2021) presented a security framework based on the authentication of IoT data in the cloud collected by IoT devices or sensors. They designed a cancer prediction system based on the Internet of Things to test the blood of patients to see whether it is normal or abnormal. The prediction is based on the results from IoT upon extracting the details of the results. However, for the storage of

the data collected, encryption is adopted to safeguard results in the cloud for quick reference to doctors or healthcare nurses and this is based on the AES algorithm.

Kishori et al. (2022) investigated the risks involved with the unprotected transmission of data in healthcare and encouraged the transmission of data remotely by effectively proposing a novel security framework that uses cryptographic techniques. A laboratory configuration of 2 different encryption at the IoT sensor level and two-stage decoding at the receiver level described as the physician's surgery operator was used to validate their method. The resulting methodology improved the security of data transmitted because many keys are involved in deciphering compared to using 1 key sequence for deciphering.

Waleed et al. (2022) introduced a data collection approach based on the evolution and integration of Cloud Computing with the Internet of Things and how cloud computing has influenced the Internet of Things, particularly in healthcare. The proposed framework can serve different applications, and the Amazon platform was identified as a cloud provider for the implementation. The sensor data will be stored in a highly scalable database in the cloud and a decision-making algorithm is used to forward decisions. The idea of the work is to propose a framework hosted in the Cloud that offers applications and uses wireless sensors in their infrastructure. However, the proposed framework did not discuss or take the security of this deployment into account.

(Saha & Majumdar, 2017) introduced an advanced IoT system that utilizes temperature sensors to monitor different points in a data center, with real-time data being accessible via a cloud-based dashboard. The system sends alerts to notify users when the temperature rises above safe levels, allowing for prompt action to be taken. The system employs a wireless sensor network that includes temperature sensors, an ESP8266, and a Wi-Fi router. The ESP8266 is connected to the Ubidots cloud through its API, which posts real-time temperature data to the cloud dashboard. When a high-temperature alert event is triggered, the cloud event management system generates alerts that can be configured beforehand through the platform's user interface.

(Zafar et al., 2018) describes the development of a real-time system for monitoring the environment that makes use of IoT technology, Arduino, and cloud services. The system has been designed to keep track of various environmental parameters like air quality, temperature, and humidity in real-time, and to send data to the cloud for analysis and visualization. The article presents the implementation of the monitoring system, which uses sensors to measure temperature and humidity, an Arduino UNO board,

a DHT11 sensor, and an ESP8266 Wi-Fi module to send the information to the ThingSpeak open IoT API service, where it is analyzed and stored. An Android app then retrieves the data from the cloud and shows it to the end users. The system is intended to continually monitor the environment and show both real-time data and graphical analyses of the surrounding conditions on the cloud.

(Hong et al., 2021) discussed the application of Arduino-based sensors for water quality monitoring, providing valuable insights into how this technology can contribute to ensuring safe and clean water for all. Wong Jun Hong and his team investigate the feasibility of using such a system, developing a prototype that includes a microcontroller and several attached sensors to gather relevant data from the environment. The collected data is then converted to digital output and presented in an understandable format on a portable laptop screen using appropriate software. The system was tested on-site at regular intervals and demonstrated reliable functionality, although it is reliant on human assistance and may be subject to inaccuracies in the collected data.

(Jaber, Al-Mousawi, & Jasem, 2019) discusses the design of an IoT-based industrial environment monitoring and control system which monitors and controls industrial environments such as factories or labs. The DH T22 digital temperature and humidity sensor and the ESP8266 Wi-Fi module's antenna socket are just two of the primary components of the developed system that are discussed. The developed electronic system includes a control unit and a sensor unit. The control unit is responsible for processing data from the sensor unit and controlling the actuators based on predefined rules. The sensor unit collects data from various sensors and sends it to the control unit wirelessly using Wi-Fi technology. Overall, the study offers a practical strategy for putting IoT-based industrial environment monitoring and control systems into action.

(Perilla et al., 2018) describes the implementation of a fire safety and alert system that uses Arduino sensors with IoT integration to detect and sense various factors that change during a fire. The system includes four sensors, the Arduino GSM shield 2 modules, a GPS module, a buzzer, LED lights, LCD display panel, and relays. It logs all its activities and sends collected data to a server for future research and studies. The research does not provide specific information on how alerts are sent to fire-fighting facilities and building occupants or how the system can be customized for different types of buildings or environments, but it provides a conceptual framework and methodology for designing such a system.

Abd Jalil et al. (2021) present a comprehensive implementation of a vehicle ventilation system that uses the NodeMCU ESP8266 microcontroller as an IoT platform. The system is designed to provide remote monitoring and control of the vehicle's ventilation system through a mobile application. The implementation involves the use of NodeMCU ESP8266 microcontroller, which acts as the main controller for the system, and a set of sensors and actuators that enable it to control the ventilation system. The system is equipped with various widgets and libraries, including Firebase, which provides cloud services to the mobile application for remote monitoring and control. The system is designed to provide real-time monitoring of the vehicle's temperature and humidity, as well as the ventilation system's status, and enable the user to remotely control the ventilation system to maintain optimal conditions. To enhance the reliability of the system, a sound notification is incorporated into the design, which alerts the user in case of any malfunction or failure of the system. The implementation is an effective solution for improving the comfort and safety of passengers in vehicles, especially during extreme weather conditions.

CHAPTER 3

Methodology

This study's methodology involves a multi-approach that includes a literature review and an experimental technique for the implementation of the research. The first part of this study involves a thorough review of existing literature on IoT, Cloud Computing, and IoT applications in healthcare. The aim is to identify current challenges and opportunities in the field. Based on the findings, we integrated the ThingSpeak cloud with the ESP8266MOD sensor and a DS18B20 temperature sensor to create a real-time environmental temperature monitoring system for healthcare.

The experimental analysis involves extensive integration and generating meaningful data that could be useful in healthcare. The proposed system consists of two ESP8266s: one configured as an access point and the other as a client. The access point retrieves temperature readings from the DS18B20 sensor and wirelessly transmits them to the client ESP8266. If one of the access points stops working or is unable to read temperature readings, a light or LED notification will be activated to alert the user. The experiment further enhances the system by using a simple light weight encryption to secure data and uses a key to unlock the temperature readings before displaying them on the web server.

The proposed system can be deployed in healthcare settings such as hospitals, clinics, and nursing homes where temperature monitoring is critical for patient care and comfort. The system can also be integrated with medical alert systems to provide timely notifications to medical staff in case of any abnormalities.

3.1 System Set Up

3.2 Hardware Overview

The core of this system is based on two ESP8266 microcontrollers, with one acting as an access point and the other as a client. The access point is connected to a DSB1820 temperature sensor and is responsible for serving as a bridge between the sensor and the client. The client, in turn, retrieves temperature readings from the access point over Wi-Fi and sends the data to the for analysis.



Figure 1: Shows the Jumper wires used for hardware set up connections.

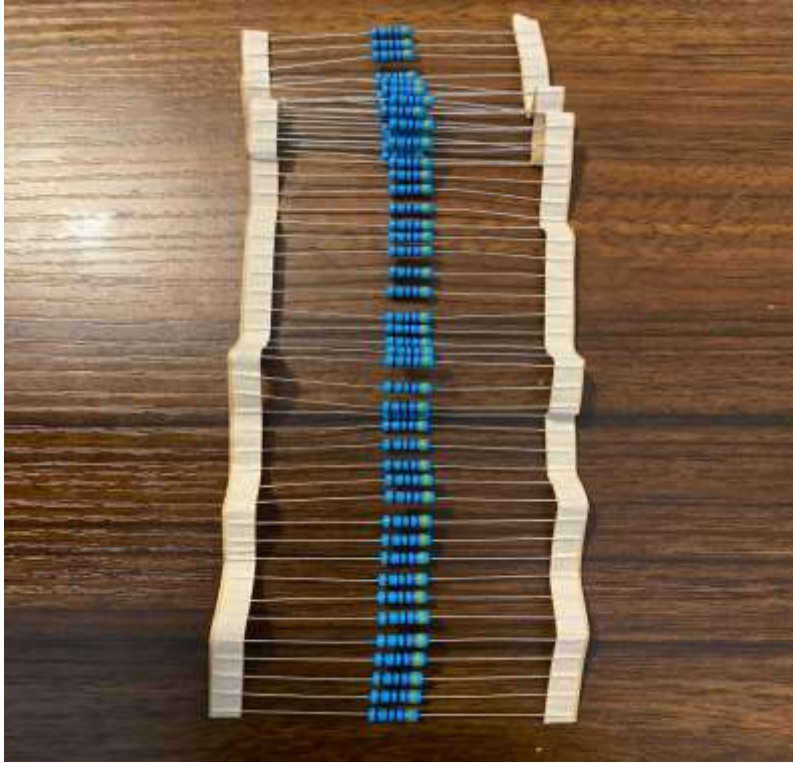


Figure 2: Shows the 4.7k ohm resistor used for voltage stability.

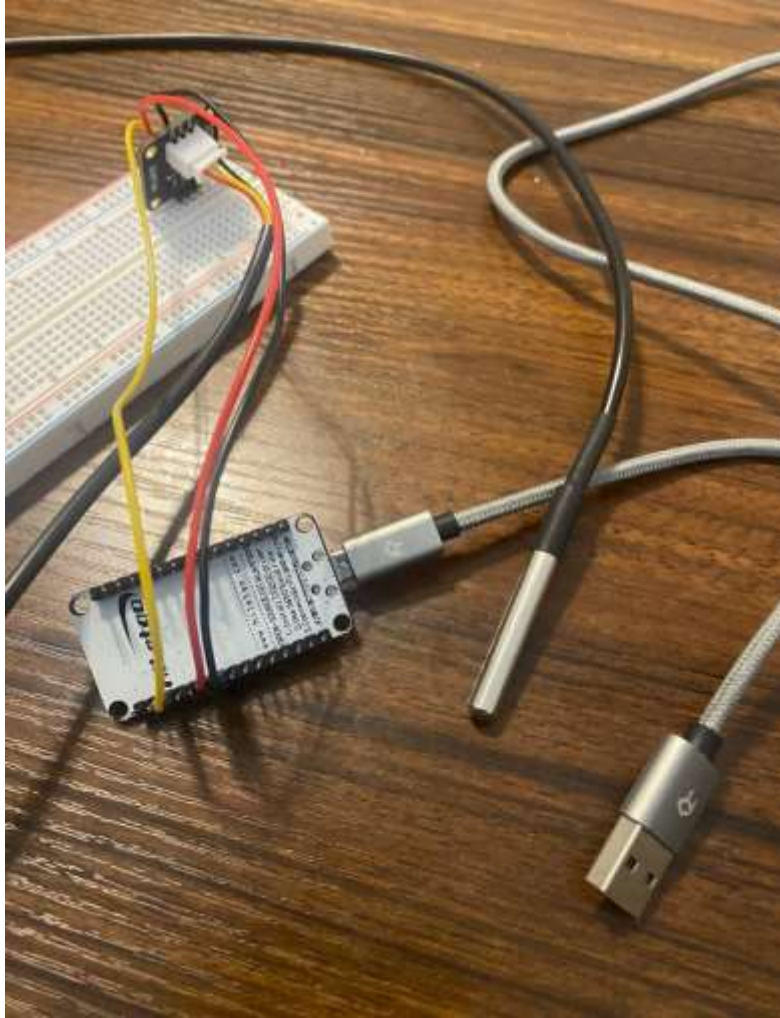


Figure 3: Shows the ESP 8266 hardware set up with DS18B20.

3.3 Microcontroller

We chose the NodeMCU ESP8266 microcontroller which interfaces with other components as the central hardware component of our system for our systems for several reasons. The ESP8266 is a widely used Wi-Fi chip among developers since it is inexpensive and simple to incorporate into IoT applications. With this technology, sensors and cloud storage can communicate wirelessly, allowing for real-time monitoring of environmental variables. (Zafar et al., 2018).



Figure 4: Pictorial Representation of the ESP 8266 used.

The ESP8266's ability to connect to Wi-Fi networks is one of its main benefits. This enables it to connect to the internet and communicate with other networked devices. Depending on the application, the ESP8266 can function as either a station or an access point. The Arduino IDE (Integrated Development Environment) running on a laptop is used to program the ESP8266. It connects to the ESP's UART interface (Transmit and Receive pins) through a UART to USB converter. The C software is uploaded as a compiled binary to the ESP's flash memory. (Saha & Majumdar, 2017).

Generally, the ESP8266's flexibility, low cost, and ease of use make it a powerful and adaptable platform for creating IoT applications that require wireless communication. Its integration with cloud platforms enables remote monitoring and control of sensor devices, making it a valuable tool for a wide range of applications, including healthcare, environmental monitoring, and home automation.

3.4 Sensor

DS18B20 is the digital temperature sensor used for this research and it was chosen due to its high accuracy and reliability. The sensor is made by DALLAS and uses a 1-Wire bus interface. It has only one I/O port and can be configured to achieve temperature values from 9 bits to 12 bits. The sensor has an E2ROM and RAM, with the RAM utilized to store configuration numbers and temperature readings. (Shen et al.,2006). The DS18B20 has a 12-bit resolution by default with a temperature range of -55°C to $+125^{\circ}\text{C}$. Several sensors can connect to the same 1-Wire bus for distributed temperature detecting and process monitoring because of its unique silicon serial number. (Wu, Yan et al., 2011).



Figure 5: Pictorial Representation of the DS18B20.

3.5 Hardware Block Diagram

Figure 6 shows the hardware block architecture for our system. The figure displays how the hardware is set up and the system functions with DS18B20 providing real-time temperature readings to the microcontroller which sends these readings through the Wi-Fi module and sent to the ThingSpeak Cloud.

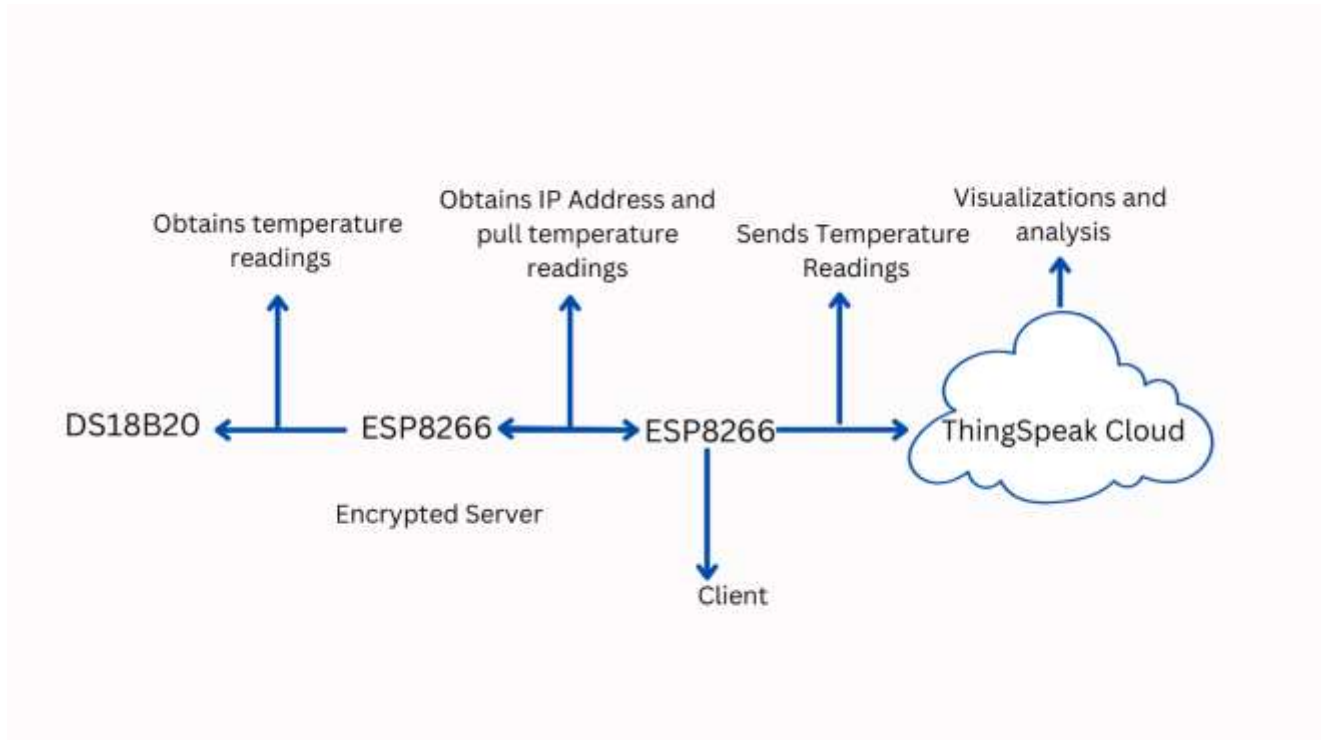


Figure 6: Proposed System Framework set up.

3.6 Software Overview

The core setup for the software components includes the Arduino IDE which was used as the development environment and the ThingSpeak cloud platform used for data collection, visualization, and analysis.

3.7 Arduino IDE (Integrated Development Environment)

The Arduino IDE was chosen for this research because of its ease of use which is crucial. It was used to write, compile and upload code to the ESP8266. It provides an excellent user-friendly platform to create programs for Arduino-compatible boards. The IDE firmware is written in C language. (Saha & Majumdar, 2017). With the IDE, users can write, compile and upload code to the microcontrollers board to control various electronic components and sensors. The IDE comes with a code editor, compiler, serial monitor, and firmware updater. It also includes a library manager that has different libraries, and pre-written code for various sensors, displays, and other components and can be integrated with other platforms like ThingSpeak and the Arduino Cloud.

Furthermore, the IDE has a vast community of developers who continuously update libraries and provide support. This community provides access to a wealth of information, making it easier to troubleshoot any issues encountered during the development process. Lastly, the availability of various libraries, including those for different sensors and components, also saved time and effort in writing code from scratch.

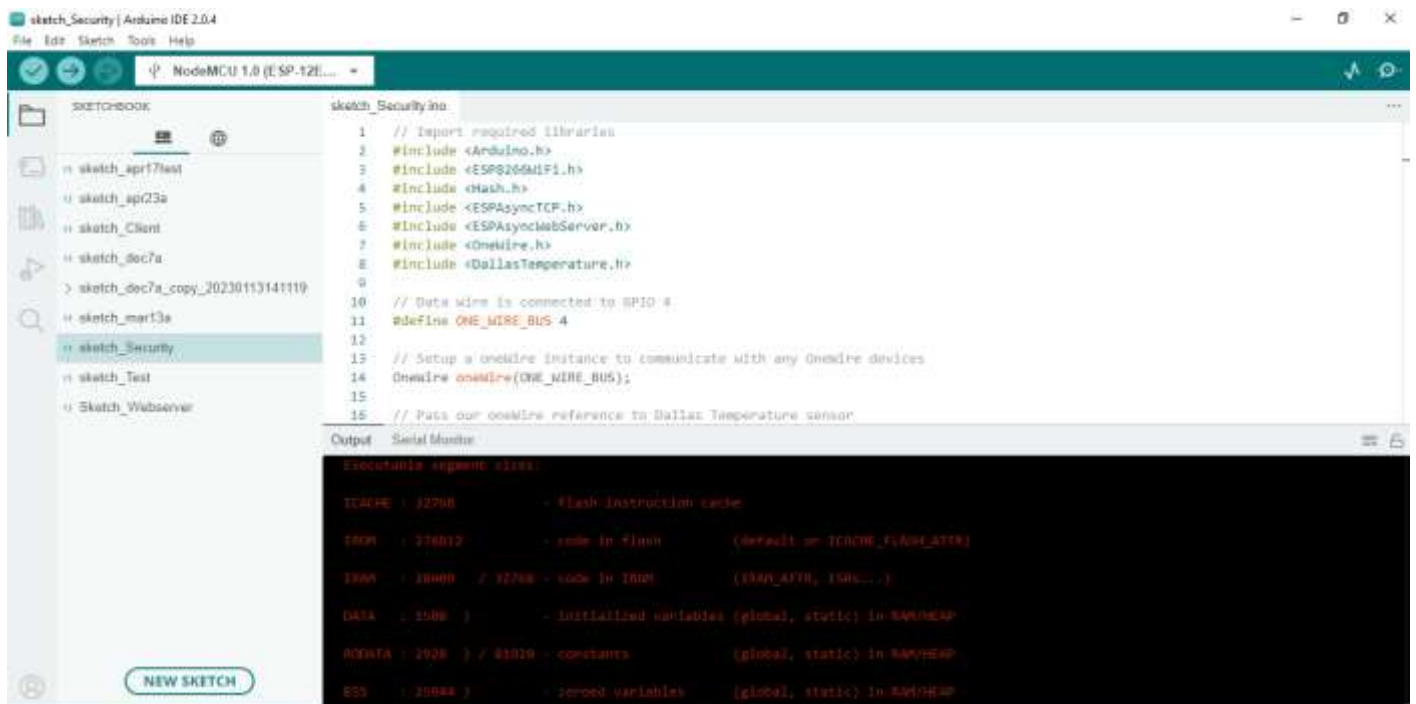


Figure 7: The Arduino IDE Interface

3.8 ThingSpeak

ThingSpeak was chosen as the preferred platform for data visualization and analysis because of its versatility and ease of use. ThingSpeak is an open-source Internet of Things (IoT) platform that allows users to collect, visualize, and analyze sensor data from different sources in real time. It was originally launched in 2010 by ioBridge. The platform's main component is the Channel which receives and stores data sent from different devices. The Channel contains up to eight fields and can be made public or private depending on users' preference. (Nettikadan & Raj, 2018). The IoT platform provides an easy-to-use web interface for the visualization and analysis of data and also built-in support for MATLAB analytics, allowing users to perform advanced data analysis and visualization.

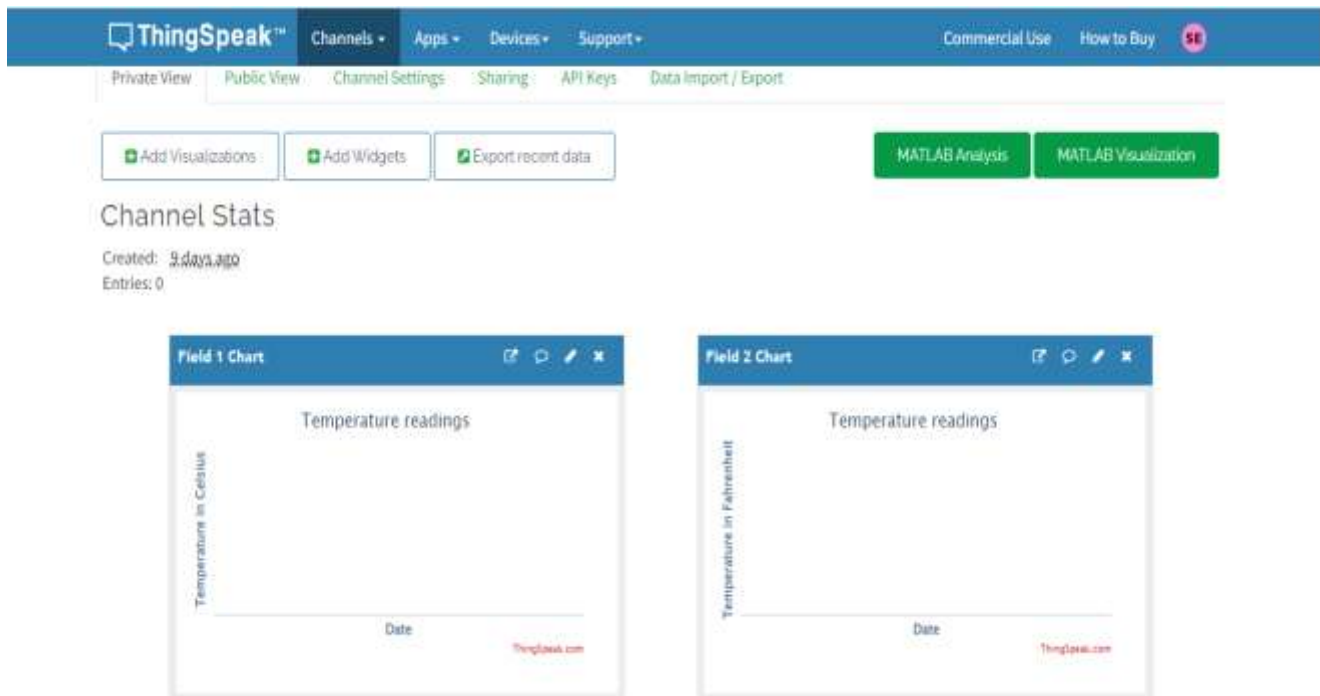


Figure 8: ThingSpeak Interface

3.9 System Security

For this research, we used a simple XOR encryption function to encrypt and decrypt temperature values before sending them over to the web server. This is a symmetric-key encryption algorithm that uses a constant key to perform the encryption and decryption. The key is stored in the `xorConstKey` variable. The system requires the users to input the key before the temperature readings are decrypted and displayed. The choice of a lightweight cryptography method is based on performance and fast mathematical operation with minimal computational resources needed, which makes it ideal for embedded devices or low-power sensors. (Thakor et al., 2020).

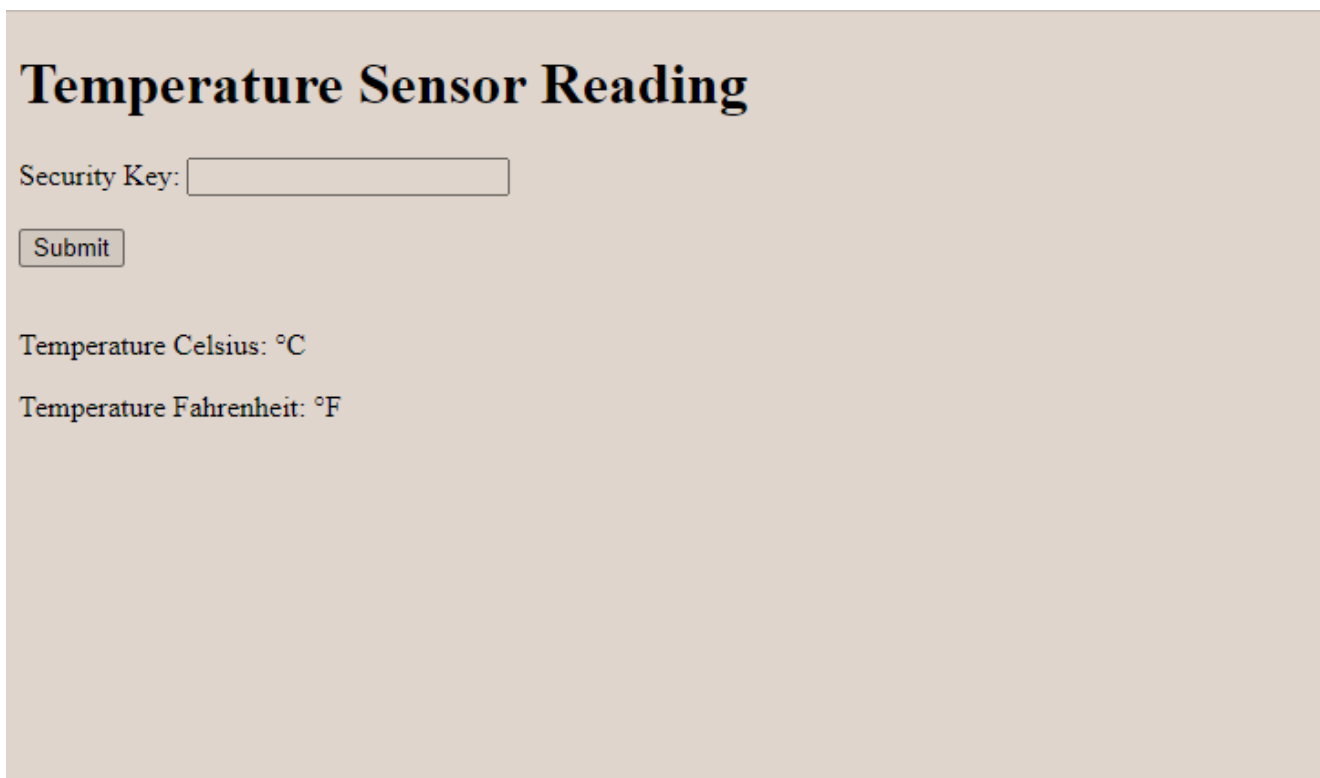


Figure 9: Security set up viewed over the ESP8266 Client's web server.

3.10 Proposed Framework

The proposed system framework consists of four main parts: The hardware, the software components, the Security system, and the Cloud environment. The framework is designed to implement temperature monitoring and data transmission system while ensuring security is in place. The hardware component includes the physical components necessary to implement the system, such as the sensors, microcontrollers, and wireless communication modules. The software component includes the code necessary to control the hardware and process the data. Security is put in place by simple encryption to prevent temperature readings from being assessed by unauthorized users. Temperature findings and readings are sent to the cloud for visualization and analysis. The proposed system framework has several benefits. Firstly, the use of ESP8266 modules and open-source software makes the system cost-effective and accessible. Secondly, the system is designed to operate wirelessly, eliminating the need for complex wiring, and allowing for easy installation and use. Thirdly, the use of XOR encryption ensures that temperature data is securely transmitted and protected from unauthorized access. Overall, this system provides a cost-effective, reliable, and scalable solution for real-time temperature monitoring in healthcare settings.

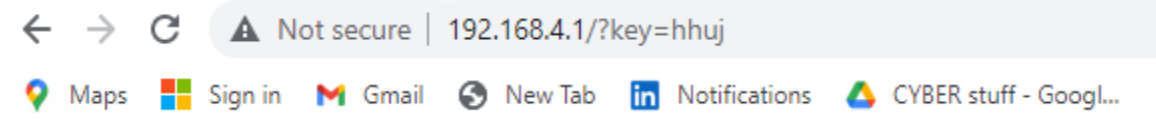
CHAPTER 4

Experimental Procedure:

The experimental analysis aimed to investigate the efficiency of the proposed system framework in retrieving and wirelessly transmitting data in real-time while securing it with XOR encryption. To conduct the experiment, the following components were used: two ESP8266s, a DS18B20 temperature sensor, a breadboard, jumper wires, a USB cable, and a laptop. The ESP8266s were programmed using the Arduino IDE, and the system code was tested using a serial monitor and uploaded to both ESP8266s.

The DS18B20 temperature sensor was connected to the access point ESP8266, and the temperature readings were retrieved using the OneWire and DallasTemperature libraries. The retrieved temperature values were encrypted using the XOR encryption algorithm before being transmitted wirelessly to the client ESP8266. To ensure the system operates efficiently and effectively, the framework includes a number of critical features. Firstly, the access point ESP8266 has been configured to activate an LED notification when it is unable to retrieve temperature readings. This feature ensures that users are promptly alerted to any issues that arise with the system. Additionally, a timer has been set up to send temperature readings from the access point ESP8266 to the client ESP8266 every ten seconds. This feature ensures that temperature readings are consistently transmitted in real-time, allowing for immediate analysis and visualization.

The system was tested by varying the distance between the access point and client ESP8266s, and the performance of the system was evaluated by analyzing the data retrieved from the sensors. The experiment was repeated several times to ensure that the results obtained were consistent and reliable.



Error: Invalid key

Figure 10: Represents Denies Access if Security Key is Invalid.

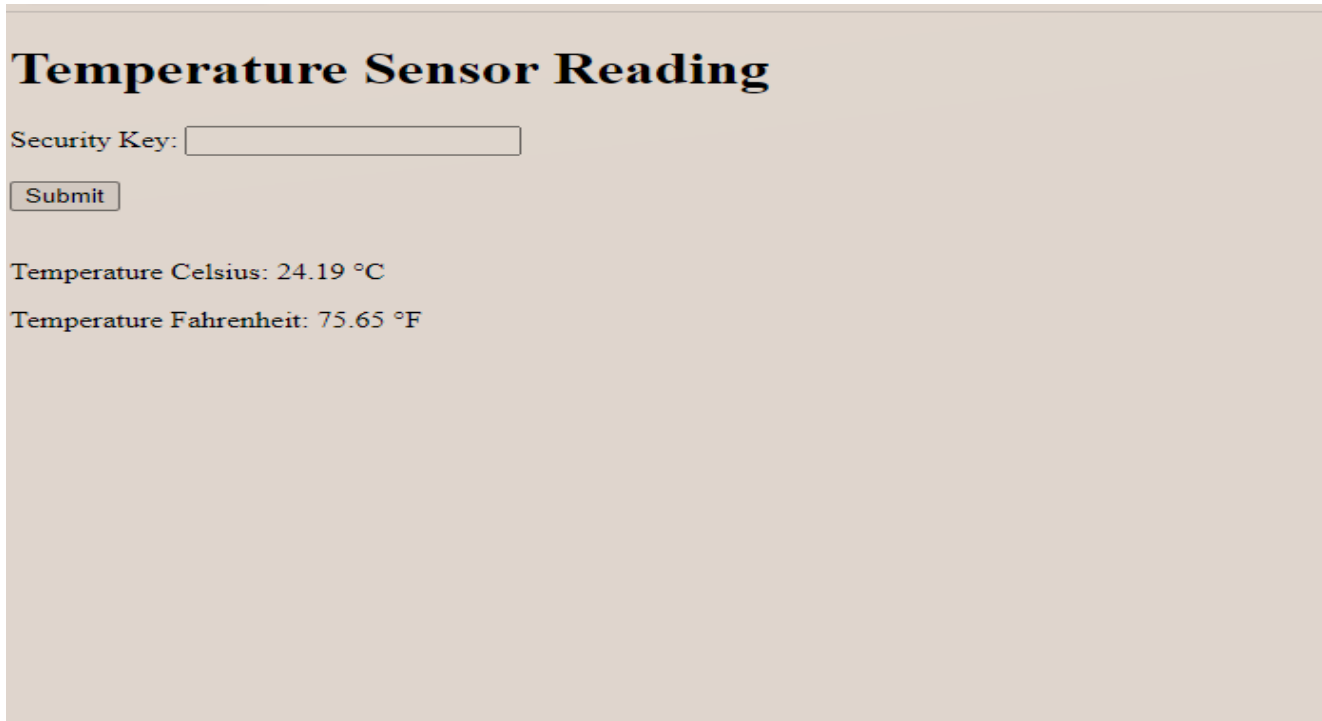
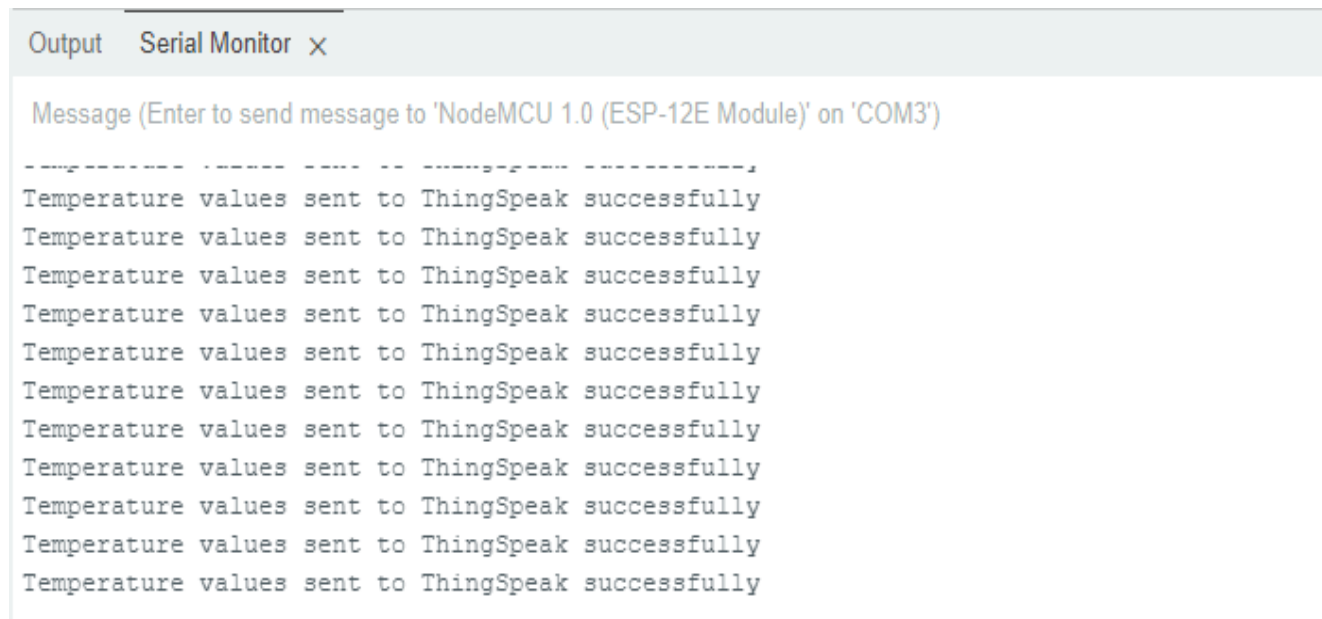


Figure 11: Temperature readings displayed after decrypting with Security Key.

4.1 Experimental Results:

The experimental results showed that the proposed system framework was able to retrieve temperature readings from the DS18B20 sensor, encrypt the data using XOR encryption, and wirelessly transmit it to the client ESP8266 in real time. The LED notification feature in the access point ESP8266 was effective in promptly alerting users to any issues with the system, ensuring that the system operates efficiently and effectively. Additionally, the timer feature in the system allowed temperature readings to be transmitted in real-time every ten seconds, allowing for immediate analysis and visualization.

The data obtained from the sensors was uploaded to the ThingSpeak cloud platform, which provided a range of visualization tools, including charts, graphs, and gauges, that could be customized to suit the needs of the user. This allowed for remote monitoring and control of the system, providing a convenient and efficient way to collect and analyze data from the sensors.



```
Output  Serial Monitor x
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM3')
-----
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
Temperature values sent to ThingSpeak successfully
```

Figure 12: Displays Connection Established with ThingSpeak.

Channel Stats

Created: [.15 days ago](#)
Last entry: [less than a minute ago](#)
Entries: 73

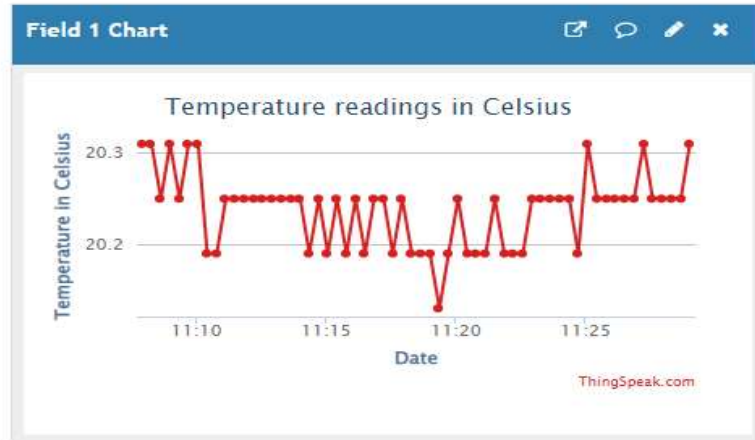


Figure 13: Temperature readings Charts in Celsius.

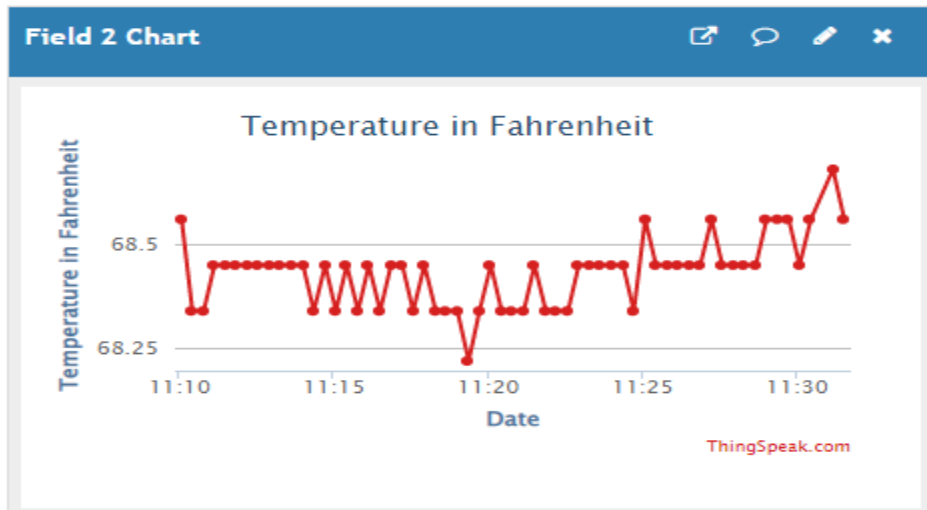


Figure 14: Temperature readings Charts in Fahrenheit.

The experimental results provide meaningful data that could be useful in healthcare, where temperature monitoring is critical. The experimental results obtained from this study provide a good knowledge of the benefits of integrating IoT with cloud computing, particularly for healthcare, and may serve as a basis for future research in this area.

In conclusion, the experimental results demonstrated the efficiency and effectiveness of the proposed system framework in retrieving and wirelessly transmitting data in real time while securing it with XOR encryption. The results obtained may have important implications for healthcare and other fields where real-time data collection and analysis are critical. Further research may be necessary to explore the full potential of this technology and its applications in different fields.

CHAPTER 5

5.1 Challenges and Future Direction

The challenges encountered for the proposed system framework for monitoring and transmitting temperature data wirelessly and securely in real-time includes limitations related to the capability of the ESP8266 sensor and the use of XOR encryption for system security. These limitations give rise to specific challenges and future directions that can guide further research and development.

One of the challenges is enhancing the sensor capabilities beyond the limitations of the ESP8266 used in this research. Future research should focus on integrating advanced sensors with improved precision, wider temperature ranges, and additional environmental measurements. By incorporating these sensors, the system can achieve more comprehensive and accurate temperature monitoring.

Another challenge lies in improving the security algorithms employed within the system. While XOR encryption provides a basic level of security, future efforts should explore the adoption of more robust encryption algorithms such as AES or RSA. These advanced cryptographic algorithms can significantly enhance the security of the system, ensuring the confidentiality and integrity of the transmitted temperature data. However, it is essential to strike a balance between enhanced security and system efficiency. This requires optimizing the encryption algorithms, data transmission protocols, and power management techniques to maintain efficient operation without sacrificing performance or draining excessive power.

Lastly, to extract more robust meaningful insights from the collected temperature data, advanced data analytics techniques should be leveraged. Future directions should involve the application of machine learning algorithms, data mining techniques, and anomaly detection methods. These techniques can help identify patterns, trends, and anomalies in the data, enabling proactive decision-making and system optimization.

5.2 Conclusion

In conclusion, the proposed system framework presented in this study offers a reliable and effective solution for wirelessly monitoring and transmitting temperature data in real-time. By utilizing ESP8266 modules and open-source software, the system ensures accessibility and cost-effectiveness without compromising functionality.

The incorporation of key features, including LED notifications and real-time transmission, enhances the operational efficiency and effectiveness of the system. These features enable prompt notifications and instantaneous data transmission, ensuring timely response and decision-making. Moreover, the inclusion of lightweight encryption strikes a balance between the need for security and the demand for resource efficiency, compliance, and compatibility. This encryption mechanism safeguards the transmitted data while minimizing resource consumption, making it suitable for resource-constrained environments.

The proposed framework also offers a centralized management system, allowing for the efficient administration and monitoring of devices and data. This centralized approach facilitates real-time issue identification and resolution, enhancing the overall system performance and reliability.

This system holds great promise for applications in critical domains such as healthcare, where continuous environmental temperature monitoring is of utmost importance. The real-time temperature data provided by the system can offer valuable insights into the behavior and performance of the monitored system, aiding in timely interventions and proactive decision-making.

Overall, the proposed framework presents a comprehensive solution that combines accessibility, cost-effectiveness, security, and centralized management. It represents a significant advancement in the field of wireless temperature monitoring and has the potential to positively impact various industries and sectors requiring continuous and reliable temperature monitoring capabilities.

REFERENCES

- Abd Jalil, A. M., Mohamad, R., Anas, N. M., Kassim, M., & Suliman, S. I. (2021). Implementation of vehicle ventilation system using NodeMCU ESP8266 for remote monitoring. *Bulletin of Electrical Engineering and Informatics*, 10(1), 327-336.
- Anuradha, M., Jayasankar, T., Prakash, N. B., Sikkandar, M. Y., Hemalakshmi, G. R., Bharatiraja, C., & Britto, A. S. F. (2021). IoT enabled cancer prediction system to enhance the authentication and security using cloud computing. *Microprocessors and Microsystems*, 80, 103301.
- Ammar, M., Russello, G., & Crispo, B. (2018). Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications*, 38, 8-27.
- Fang, Z., Fu, H., Gu, T., Qian, Z., Jaeger, T., Hu, P., & Mohapatra, P. (2021). A model checking-based security analysis framework for IoT systems. *High-Confidence Computing*, 1(1), 100004.
- Hong, W. J., Shamsuddin, N., Abas, E., Apong, R. A., Masri, Z., Suhaimi, H., ... & Noh, M. N. A. (2021). Water quality monitoring with arduino based sensors. *Environments*, 8(1), 6.
- Hussein, W. N., Hussain, H. N., & Humod, I. M. (2022). A proposed framework for healthcare based on cloud computing and IoT applications. *Materials Today: Proceedings*, 60, 1835-1839.
- Jaber, A. A., Al-Mousawi, F. K. I., & Jasem, H. S. (2019). Internet of things based industrial environment monitoring and control: a design approach. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(6), 4657-4667.
- Kasat, K., Rani, D. L., Khan, B., Kirubakaran, M. K., & Malathi, P. (2022). A novel security framework for healthcare data through IOT sensors. *Measurement: Sensors*, 24, 100535.
- Nettikadan, D., & Raj, S. (2018). Smart community monitoring system using ThingSpeak IoT platform. *International Journal of Applied Engineering Research*, 13(17), 13402-13408.
- Perilla, F. S., Villanueva Jr, G. R., Cacanindin, N. M., & Palaoag, T. D. (2018, February). Fire safety and alert system using arduino sensors with IoT integration. In *Proceedings of the 2018 7th International Conference on Software and Computer Applications* (pp. 199-203).

- Saha, S., & Majumdar, A. (2017, March). Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system. In *2017 Devices for Integrated Circuit (DevIC)* (pp. 307-310). IEEE.
- Shen, H., Fu, J., & Chen, Z. (2006, November). Embedded system of temperature testing based on DS18B20. In *2006 International Technology and Innovation Conference (ITIC 2006)* (pp. 2223-2226). IET.
- Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. *Future Generation Computer Systems*, 78, 964-975.
- Tawalbeh, L. A., Muheidat, F., Tawalbeh, M., & Quwaider, M. (2020). IoT Privacy and security: Challenges and solutions. *Applied Sciences*, 10(12), 4102.
- Thakor, V. A., Razzaque, M. A., & Khandaker, M. R. (2020). Lightweight cryptography for IoT: A state-of-the-art. *arXiv preprint arXiv:2006.13813*.
- Valanarasu, M. R. (2019). Smart and secure IoT and AI integration framework for hospital environment. *Journal of ISMAC*, 1(03), 172-179.
- Wu, Y. X., Liu, D., & Kuang, X. H. (2011). A temperature detecting system based on DS18B20. *Advanced Materials Research*, 328, 1806-1809.
- Zafar, S., Miraj, G., Baloch, R., Murtaza, D., & Arshad, K. (2018). An IoT based real-time environmental monitoring system using Arduino and cloud service. *Engineering, Technology & Applied Science Research*, 8(4), 3238-3242.