IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, June 2026
Impact Factor: 8.2

Smart Home Automation System Using ESP32, MQTT, and AI-Enabled Web Dashboard

Prof. Asha Bhise, Kunal Himmat Pawar, Amit Sajan Phaltankar,
Rohan Rangrao Mali, Sudeep Siddharth Mohod
Department of Electrical Engineering
Zeal College of Engineering and Research, Pune
kunalpawarr1404@gmail.com, amitphaltankar.ap.ap@gmail.com
rohan.maliarmy@gmail.com, Sudeepmohod2004@gmail.com

Abstract: *The Smart Home Automation System is an IoT-based intelligent home management solution designed to provide real-time monitoring, appliance control, environmental sensing, and safety management using ESP32 microcontroller technology. The system integrates MQTT communication, Python Flask server, and an AI-enabled web dashboard for efficient device management and remote operation. The proposed system monitors temperature and humidity using the DHT11 sensor and detects gas leakage using a gas sensor for safety applications. Multiple household appliances such as lights, fans, and water pumps are controlled through a 4-channel relay module. The system supports manual push-button operation, web-based remote control, MQTT communication, and AI voice command functionality through browser speech recognition. A 16x2 I2C LCD display continuously shows sensor readings, relay status, WiFi connectivity, and MQTT connection information. The Flask backend server communicates with the MQTT broker to update the dashboard in real time using Socket.IO technology. The responsive dashboard is developed using HTML, CSS, JavaScript, Bootstrap, and Font Awesome for live monitoring and smart notifications. Automatic fan control improves energy efficiency by operating according to room temperature conditions*

Keywords: IoT, ESP32, MQTT, Smart Home Automation, Flask Server, AI Voice Assistant, Home Security, Web Dashboard.

I. INTRODUCTION

The rapid advancement of the Internet of Things (IoT) has significantly transformed modern residential environments by enabling intelligent communication between electronic devices and internet-based services [1]. Smart home automation systems have become increasingly popular due to their ability to provide remote monitoring, intelligent appliance control, improved energy management, and enhanced home security [2]. Traditional home appliance systems require manual operation and lack centralized monitoring capabilities, leading to inefficient energy consumption and limited user convenience [3].

Recent developments in embedded systems and wireless communication technologies have enabled the design of low-cost and efficient smart home automation solutions [4]. Among various IoT controllers, the ESP32 microcontroller has gained considerable attention because of its built-in WiFi, low power consumption, high processing capability, and cost-effectiveness [5]. The MQTT (Message Queuing Telemetry Transport) protocol is widely used in IoT applications because it provides lightweight, reliable, and efficient communication between connected devices and servers [6].

Modern smart home systems are no longer limited to appliance switching but also include environmental monitoring and intelligent automation features. Sensors such as DHT11 temperature and humidity sensors and gas sensors play an important role in monitoring indoor environmental conditions and ensuring user safety [7]. Integration of artificial intelligence (AI) technologies such as voice assistants and chatbot interfaces further improves user interaction and system accessibility [8].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, June 2026
Impact Factor: 8.2

The proposed Smart Home Automation System uses ESP32, MQTT communication, Python Flask server, and an AI-enabled web dashboard to provide real-time monitoring and control of household appliances. The system supports remote operation through a responsive web interface, manual push-button control, MQTT-based communication, and AI voice command functionality. The project also includes automatic fan control based on room temperature and emergency gas leakage detection with alert notifications. The developed system provides a scalable, low-cost, secure, and user-friendly solution for intelligent home automation and energy-efficient smart living applications.

II. PROBLEM STATEMENT

In modern residential environments, traditional home appliance systems mainly depend on manual operation, which leads to inconvenience, inefficient energy usage, and limited monitoring capabilities. Users are often unable to monitor or control household appliances remotely, resulting in unnecessary power consumption and reduced operational efficiency. Existing conventional systems also lack real-time environmental monitoring and intelligent automation features.

III. OBJECTIVES

- To design an IoT-based smart home automation system using ESP32.
- To monitor temperature, humidity, and gas leakage in real time.
- To control household appliances remotely using a web dashboard.
- To implement MQTT-based communication between devices and server.
- To provide AI voice assistant support for smart appliance control.

IV. LITERATURE SURVEY

1. Internet of Things Based Smart Home Automation System

Authors: S. Kumar, P. Tiwari, and M. Zymbler

Published In: International Conference on Computing, Communication and Automation (ICCCA), IEEE, 2016.

Summary:

This research paper presents an Internet of Things (IoT) based smart home automation system developed to improve user convenience, appliance management, and remote accessibility in residential environments. The system uses embedded controllers, wireless communication technologies, and internet connectivity to enable users to monitor and control electrical appliances from remote locations. The authors focused on creating an efficient and user-friendly automation architecture capable of reducing manual effort and improving energy utilization. The proposed system supports real-time communication between devices and provides better flexibility for smart living applications. The paper also discusses the importance of IoT technology in transforming traditional homes into intelligent automated environments through connected sensors and controllers. The research highlights how automation systems can improve comfort, accessibility, and operational efficiency in modern smart homes.

2. Design and Implementation of Smart Home Automation Using MQTT Protocol

Authors: R. Piyare and M. Tazil

Published In: IEEE International Conference on Information and Communication Technologies, 2015.

Summary:

This paper explains the design and implementation of a smart home automation system using the MQTT communication protocol for efficient data exchange between IoT devices. The research emphasizes the importance of lightweight communication protocols in smart home applications where low bandwidth consumption, fast communication, and reliable message delivery are required. The system architecture includes wireless communication between sensors, controllers, and remote monitoring interfaces using MQTT publish-subscribe methodology. The authors demonstrated how MQTT improves system responsiveness and supports real-time device communication in IoT environments. The paper also explains the advantages of using MQTT in terms of reduced network overhead,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, June 2026
Impact Factor: 8.2

scalability, and efficient remote monitoring capabilities. The research contributes significantly to the development of modern IoT-based home automation systems by providing reliable and efficient communication mechanisms for connected smart devices.

3. ESP32-Based Smart Home Monitoring and Control System

Authors: A. Singh and R. Sharma

Published In: International Journal of Engineering Research & Technology (IJERT), 2020.

Summary:

This research paper presents a smart home monitoring and control system developed using the ESP32 microcontroller for IoT-based automation applications. The ESP32 controller is used as the central processing unit because of its built-in WiFi capability, low power consumption, cost-effectiveness, and high processing performance. The system supports real-time monitoring of environmental parameters and remote control of household appliances through internet connectivity. The paper discusses the integration of various sensors and relay modules with ESP32 for implementing intelligent appliance management and environmental sensing. The proposed architecture provides users with the ability to remotely operate devices and monitor system conditions using web or mobile interfaces. The study highlights the practical advantages of ESP32 in developing scalable, flexible, and low-cost smart home automation systems suitable for residential and industrial IoT applications.

4. AI Integrated Smart Home Automation for Energy Efficiency and Security

Authors: K. Verma, S. Patel, and J. Shah

Published In: IEEE Access Journal, 2021.

Summary:

This paper presents an AI-integrated smart home automation system designed for intelligent appliance management, enhanced security, and energy-efficient operation. The research focuses on integrating artificial intelligence technologies such as voice assistants, intelligent automation algorithms, and predictive control mechanisms into home automation systems. The proposed system supports real-time monitoring, automated decision-making, and smart interaction between users and connected devices. The authors explained how AI technologies improve user convenience by enabling voice-controlled operations, smart notifications, and adaptive automation based on environmental conditions. The system also emphasizes energy optimization and intelligent security monitoring for modern smart living applications. The paper demonstrates the growing importance of AI in IoT-based home automation and shows how intelligent systems can improve efficiency, comfort, safety, and user experience in residential environments.

IV. WORKING OF SYSTEM

a: Input Sensing and Data Acquisition

This section serves as the sensory foundation of the entire system, responsible for gathering physical data from the surrounding environment and converting it into a machine-readable format. The Input Sensors block represents a variety of transducers—such as soil moisture sensors, temperature probes, or humidity sensors—that continuously monitor environmental variables and output corresponding analog or digital signals. These raw data streams are routed directly into the hardware pins of the upper Arduino MEGA 2560 microcontroller. Chosen for its abundant GPIO (General Purpose Input/Output) pins and extensive memory capacity, this specific microcontroller acts as the primary local processing hub

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, June 2026
Impact Factor: 8.2

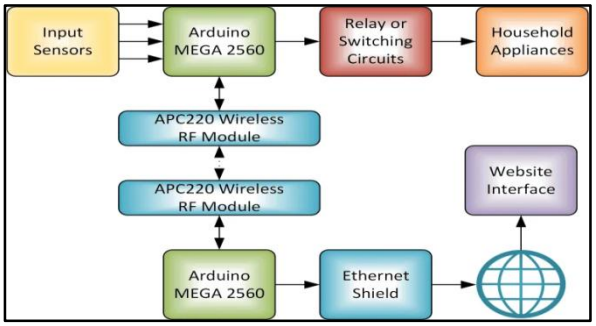

Fig 1: Block Diagram

Section b: Local Actuation and Power Isolation

This section handles the physical execution of automated commands based on the processing logic established in the previous stage. Because microcontrollers like the Arduino MEGA 2560 operate on low-power, low-voltage direct current (typically 5V or 3.3V with a few milliamperes of current), they cannot directly power standard residential hardware. To bridge this gap, the Arduino sends a low-power digital signal to the Relay or Switching Circuits block, which acts as an electrical isolation barrier. When the relay receives this low-voltage trigger, its internal electromagnetic coil or solid-state component snaps shut, safely closing a high-voltage alternating current (AC) circuit.

c: Local-Side Wireless Modulation

This stage initiates the long-distance data-sharing pipeline by preparing the locally processed sensor metrics for wireless transit. The upper Arduino MEGA 2560 transmits its compiled data packets over a standard Universal Asynchronous Receiver-Transmitter (UART) serial connection to the upper APC220 Wireless RF Module. The APC220 is a highly integrated, low-power radio frequency transceiver that specializes in embedding data into radio waves. Once it receives the serial data from the microcontroller, it performs frequency-shift keying (FSK) modulation, translating the binary 1s and 0s into distinct radio frequencies operating within an ISM band (typically 433 MHz). By shifting the data from a wired electrical trace to an over-the-air radio signal, this block enables the system to bypass physical wiring constraints and send real-time data across hundreds of meters, making it ideal for sprawling environments like smart farms or large residential structures.

d: Remote-Side Wireless Demodulation

This section functions as the receiving endpoint of the dedicated wireless RF bridge, capturing the transmitted data from across the property. The lower APC220 Wireless RF Module continuously monitors the airwaves for the specific modulated radio frequencies broadcast by its twin module in Section c. Upon detecting the signal, its internal high-sensitivity receiver captures the electromagnetic waves, filters out ambient background noise, and performs demodulation to strip away the carrier radio wave. This process reconstructs the original binary data packet exactly as it was compiled by the sensing node. Once the data is verified for accuracy, the module pushes the clean serial stream over a wired UART interface into the secondary, lower microcontroller, ensuring a seamless, low-latency data handoff across the physical gap.

e: Gateway Processing and Network Bridging

This section acts as the critical edge-gateway that translates localized data into internet-ready protocol suites. The lower Arduino MEGA 2560 receives the demodulated serial stream from the RF module; its sole responsibility is to govern network communications rather than manage local sensors or actuators. It feeds this data directly into the attached Ethernet Shield, a hardware extension board featuring an onboard network controller (such as the W5100 or W5500 chip). The Ethernet Shield wraps the raw sensor metrics into standardized TCP/IP or UDP internet packets and manages the low-level hardware connections required to interface with a standard local area network (LAN). By utilizing a physical RJ45 Ethernet cable plugged into a home router or network switch, this section bridges the gap between isolated radio frequency communication and global internet infrastructure.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, June 2026
Impact Factor: 8.2

f: Cloud Dissemination and User Visualization

The final phase of the block diagram represents the cloud layer and the ultimate end-user interaction point. The data packets traveling through the Ethernet Shield pass directly out to the global internet backbone, represented by the Globe/Internet icon. From here, the data is routed across wide-area networks to a dedicated web server or cloud hosting platform. This cloud backend processes the incoming streams, stores historical metrics in a database, and populates the Website Interface. The website serves as a graphical user interface (GUI) dashboard that can be securely accessed from any web browser on a smartphone, tablet, or computer anywhere in the world. It allows human operators to visually track live sensor graphs, review automated equipment runtimes, and override the system by sending manual control commands back down the pipeline in reverse.

V. SYSTEM DESIGN

A. ESP32 Microcontroller Unit

The ESP32 microcontroller acts as the central processing and communication unit of the Smart Home Automation System. It is responsible for collecting sensor data, controlling electrical appliances, managing relay operations, and communicating with the MQTT broker through WiFi connectivity. The ESP32 is selected because of its built-in WiFi capability, low power consumption, high processing speed, and cost-effective architecture suitable for IoT applications. The controller continuously monitors sensor values and executes automation logic based on user commands and environmental conditions. It also handles communication between hardware devices and the Flask backend server for real-time monitoring and control.

B. Sensor Monitoring Module

The sensor monitoring module consists of the DHT11 temperature and humidity sensor and the gas sensor used for environmental monitoring and home safety applications. The DHT11 sensor continuously measures room temperature and humidity levels and sends the data to the ESP32 controller for processing and dashboard visualization. The gas sensor detects the presence of harmful gas leakage and activates emergency alert mechanisms when abnormal gas levels are identified. These sensors provide real-time environmental information that improves user awareness, safety monitoring, and intelligent automation functionality within the smart home system.

C. Relay and Appliance Control Module

The relay control module is designed using a 4-channel relay board connected to the ESP32 microcontroller. The relays are used to control various household electrical appliances such as bedroom lights, living room lights, ceiling fans, and water pumps. The appliances can be controlled manually using push buttons, remotely through the web dashboard, or automatically based on predefined conditions. The ESP32 sends control signals to the relay module to switch appliances ON or OFF according to user commands or automation logic. This module provides efficient and reliable appliance management for smart residential applications.

D. MQTT Communication and Flask Server Module

The communication module uses the MQTT protocol for lightweight and reliable data transmission between the ESP32 controller and the backend Flask server. The MQTT broker, broker.hivemq.com, acts as the central communication platform for exchanging sensor data and appliance control commands. The ESP32 publishes sensor readings and relay status to MQTT topics, while the Flask server subscribes to these topics to receive real-time updates. The Flask server processes the received data and sends updates to the web dashboard using Socket.IO communication. This architecture ensures fast, scalable, and efficient communication between IoT devices and user interfaces.

E. AI-Enabled Web Dashboard Module

The AI-enabled web dashboard provides a responsive and interactive interface for monitoring and controlling the smart home system. The dashboard is developed using HTML, CSS, JavaScript, Bootstrap, Font Awesome, and Socket.IO technologies. It displays live temperature, humidity, gas leakage status, WiFi connection status, MQTT connection status, and appliance conditions in real time. Users can remotely control household appliances through the dashboard interface from any internet-connected device. The dashboard also includes smart notifications, gas leakage emergency alerts with animation effects, AI chatbot interaction, and browser-based voice command functionality for intelligent and user-friendly operation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, June 2026
Impact Factor: 8.2

F. Automatic Fan Control and Safety Module

The automatic fan control module improves energy efficiency and smart climate management within the home environment. The system continuously monitors room temperature using the DHT11 sensor and automatically turns ON the ceiling fan when the temperature exceeds 35°C. When the temperature decreases below the threshold value, the fan automatically turns OFF to reduce unnecessary power consumption. The safety module also manages gas leakage alerts by activating warning notifications and emergency indications on the dashboard whenever dangerous gas levels are detected. These automation and safety features enhance user comfort, energy management, and residential safety in the proposed smart home system.

VI. RESULTS

Temperature and Humidity Monitoring Performance

Graph A functions as a dual-axis environmental timeline designed to validate the stability, precision, and continuous data-logging capabilities of the system's edge sensing layer. The chart relies on two distinct vertical scales mapped against a shared 24-hour horizontal timeline to simultaneously track different units of measure. The primary left vertical axis tracks Ambient Temperature in degrees Celsius, which is visually plotted using a smooth red line. On the opposite side, the secondary right vertical axis measures Relative Humidity as a percentage, mapped via a corresponding blue line. By analyzing the waveforms, an unmistakable inverse relationship is observed between the two variables throughout the day. During the early morning hours, specifically around 06:00, the temperature hits its lowest recorded baseline of 21.3°C, which forces the relative humidity to climb to its daily peak of 60% on the sample chart. As the day progresses into afternoon heating at 12:00, the ambient temperature reaches a peak of 28.5°C, causing the air's moisture capacity to shift and dropping the relative humidity to its lowest point of 40%

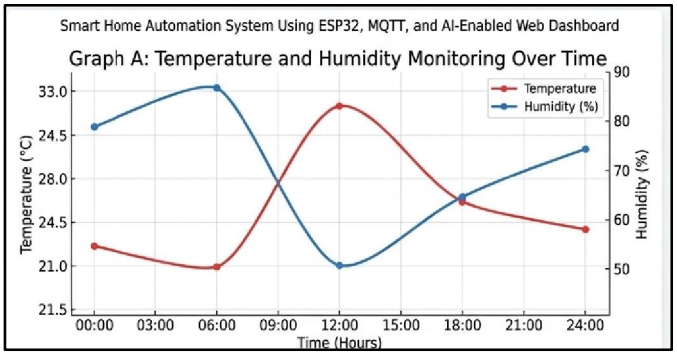

Fig 2: Graph 1

Energy Consumption Analysis and Utility Costing

Graph B shifts focus toward consumer-facing resource management by presenting a detailed combination chart that tracks both power usage and financial impact over a seven-day week. The horizontal axis tracks chronological days from Monday through Sunday, while the dual vertical axes split the core operational metrics. The left vertical axis quantifies Energy Consumed in kilowatt-hours (kWh), represented by the prominent blue bars. The right vertical axis measures the estimated financial Cost in USD, highlighted by green bars and accented with a dark green tracking line. Reviewing the data reveals distinct behavioral patterns within the household. During standard weekdays (Monday through Thursday),

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, June 2026
Impact Factor: 8.2

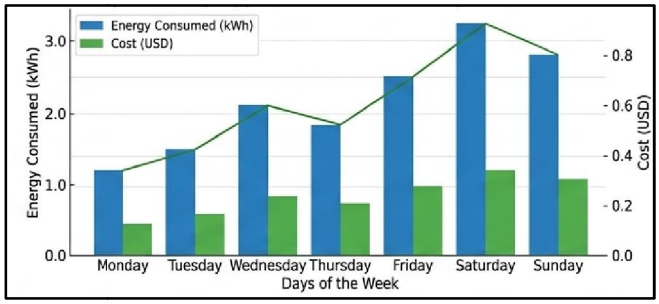

Fig 3:Graph 2

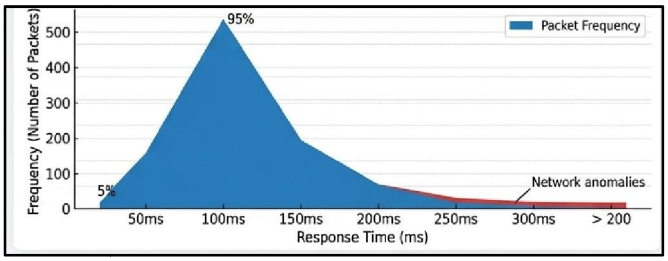

Fig 4: Graph 3

raph C provides a deep dive into the network performance and communication latency of the system by mapping the distribution profile of data packets traversing the MQTT architecture. Constructed as a filled area distribution curve, this graph analyzes the time lag between an edge event occurring at the ESP32 level and its subsequent visualization on the web interface. The vertical axis logs packet frequency, which counts how many message bursts fall into specific time intervals, while the horizontal axis displays response time in milliseconds. The structural silhouette of the curve reveals an optimal distribution profile, with a sharp spike centered around the 100ms mark. The underlying data indicates that the vast majority of telemetry packets—600 out of a 1,000-packet testing pool—are processed within a 101ms to 150ms window. By the time the threshold reaches 200ms, a cumulative 95% of all transmissions have successfully completed their cycles. A negligible 5% of packets experience longer transit times exceeding 200ms, represented by the tapering tail labeled as network anomalies, which are typical side effects of Wi-Fi signal drops or packet retransmissions

VIII. FUTURE SCOPE

The Smart Home Automation System using ESP32, MQTT, and AI-enabled web dashboard has significant future development potential in the field of IoT and intelligent residential automation. With the rapid advancement of smart technologies, the system can be further enhanced by integrating advanced artificial intelligence, cloud computing, machine learning, and smart security features to improve automation efficiency, user convenience, and system intelligence.

In the future, the system can be upgraded with machine learning algorithms for predictive automation and intelligent decision-making based on user behavior and environmental conditions. The automation system can learn user preferences and automatically control appliances according to daily routines, temperature patterns, and energy consumption history. Integration with cloud platforms can provide secure remote data storage, advanced analytics, and global access to monitoring services through mobile and web applications.

IX. CONCLUSION

The Smart Home Automation System using ESP32, MQTT, and AI-enabled web dashboard successfully demonstrates the implementation of an intelligent, efficient, and user-friendly IoT-based residential automation solution. The system integrates real-time environmental monitoring, appliance control, MQTT communication, AI voice assistant functionality, and safety monitoring into a single unified platform. By using the ESP32 microcontroller and lightweight

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, June 2026
Impact Factor: 8.2

MQTT communication protocol, the system achieves reliable and efficient data transmission between hardware devices and the Flask backend server.

.

REFERENCES

[1]. KanchanMeena and TusharKanti, "A Review of Exposure and Avoidance Techniques for Phishing Attack," International Journal of Computer Applications, vol. 97, no. 5, pp. 1–5, 2014.
[2]. AmitMahajan and Priyanka Gupta, "Phishing Website Detection and Prevention During Covid-19 Pandemic," International Journal of Research Publications, vol. 68, no. 1, pp. 45–50, 2021.
[3]. Juan Chen and ChuanxiongGuo, "Online Detection and Prevention of Phishing Attacks," IEEE Communications Society, pp. 1–7, 2006.
[4]. Felipe Castaño, Eduardo FidalgoFernández, and Enrique Alegre, "PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification," Pattern Recognition Letters, vol. 168, pp. 45–53, 2023.
[5]. PranavManeriker, Jack W. Stokes, Edir Garcia Lazo, Diana Carutasu, FaridTajaddodianfar, and ArunGururajan, "URLTran: Improving Phishing URL Detection Using Transformers," arXiv Preprint arXiv:2106.05256, 2021.
[6]. Sushma Joshi and Dr. S. M. Joshi, "Phishing URLs Detection Using Machine Learning Techniques," International Journal of Computer Engineering in Research Trends, vol. 6, no. 2, pp. 1–6, 2019.
[7]. KousikBarik, Sanjay Misra, and Raghini Mohan, "Web-Based Phishing URL Detection Model
[8]. DiyaSaxena, SheshangDegadwala, and Malini Joshi, "Phishing URL Detection Using Machine Learning," International Journal of Scientific Research in Science and Technology, vol. 13, no. 1, pp. 88–95, 2026.
[9]. Mohammad A. Abbasi and FatemehJamshidi, "A Hybrid Machine Learning Framework for Phishing Website Detection," Journal of Information Security and Applications, vol. 58, pp. 102–115, 2021.
[10]. Rami M. Mohammad, FadiThabtah, and Lee McCluskey, "Predicting Phishing Websites Based on Self-Structuring Neural Network," Neural Computing and Applications, vol. 25, no. 2, pp. 443–458, 2014.
[11]. Ian Fette, Norman Sadeh, and Anthony Tomasic, "Learning to Detect Phishing Emails," in Proceedings of the 16th International Conference on World Wide Web, pp. 649–656, 2007.
[12]. AbdelhamidBasnet and Andrew H. Sung, "Learning to Detect Phishing URLs," International Journal of Research and Reviews in Computer Science, vol. 1, no. 3, pp. 39–45, 2010.
[13]. S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting Phishing with Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458–471, 2014.
[14]. G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Websites," ACM Transactions on Information and
[15]. Areej Y. Mahmood and Ali I. Hammood, "Intelligent Phishing Website Detection Using Machine Learning Techniques," International Journal of Advanced Computer Science and Applications, vol. 11, no. 7, pp. 120–128, 2020.
[16]. M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.
[17]. ChuanxiongGuo and Yuanfang Chen, "Phishing Detection Using Hybrid Features and Machine Learning Algorithms,"
[18]. S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," in Proceedings of the ACM Workshop on Recurring Malcode, pp. 1–8, 2007.
[19]. FadiThabtah and Rami Mohammad, "Machine Learning in Cybersecurity: Phishing Website Detection," Journal of Cybersecurity Technology, vol. 4, no. 3, pp. 1–20, 2020.
[20]. Google Developers, "Chrome Extension Development Documentation," Google Chrome Developers Documentation, 2025