Journal of Engineering, Science and Technology
ISSN: 2714-2396, Volume 9 Issue 1

DESIGN AND IMPLEMENTATION OF A NOISE-ROBUST TINYML-BASED EMBEDDED SPEECH RECOGNITION SYSTEM FOR VOICE-CONTROLLED IOT APPLICATIONS

Muhammed A. A., Onyela O., Nosakhare J., Folorunsho C. S., Cookeygam C. J and Muhammed H. I.

Department of Electrical/Electrical Engineering, Faculty of Engineering, University of Benin, Benin City, Nigeria.

Corresponding e-mail: abudu.muhammed@uniben.edu

Received: 28 April 2026 **Accepted**: Date 30 May 2026 **Published**: 16 June 2026

Abstract. The design, implementation and evaluation of a noise-robust embedded speech recognition system for low-cost voice-controlled Internet of Things (IoT) applications were carried out in this work. To achieve real-time, offline-capable voice control system, integration of an ESP32 microcontroller, a TinyML-based keyword spotting model and EasyVR 3 Plus speech recognition module was deployed. Signal preprocessing techniques including spectral subtraction, pre-emphasis filtering and voice activity detection were used to enhance performance in noisy environment. The system further incorporates the Message Queuing Telemetry Transport (MQTT) protocol to enable efficient and expandable communication between devices within a smart environment. Under varying acoustic conditions ranging from quiet to highly noisy environments experimental evaluation was conducted. The results show that the TinyML model performs better in noisy conditions with low SNR values., achieving over 70% accuracy at 0 dB SNR and up to 98% at higher SNR levels. The system responded in less than 400 ms on average, making it suitable for real-time interaction., while MQTT communication achieved a reliability exceeding 99.8% with minimal delay. The result shows that power consumption remain low at approximately 1.3 W and total system cost was under ₦35,000 confirms its suitability for resource-constrained deployments. User assessment demonstrates high fulfillment in terms of ease of use, responsiveness, and observed reliability. These results validate that the use of a hybrid embedded architecture that combines template-based and TinyML approaches has the potential to balance accuracy, latency, privacy and cost. This work demonstrates a practical and low-cost approach for implementing secure voice-controlled IoT systems that operates reliably in noisy real-world environments.

Keywords: Protocol, communication, signal, model, framework

1. Introduction

One of the main technologies used in intelligent systems is Automatic Speech Recognition (ASR) which find use in smartphones, virtual assistants, automotive interfaces and Internet of Things (IoT) devices. Hinton *et al.* (2012) stated that at early use, the ASR systems were primarily based on statistical methods such as Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs). While these approaches performed practically well in controlled environments, they depend on handcrafted feature extraction and were easily affected by deviations in speaker accents,

pronunciation and background noise. Nassif *et al*. (2019) reported that the growing use of ASR in real-world environments, exposes its limitations and created the need for more trustworthy approaches.

Ravanelli *et al.* (2021) reported that Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and more recently transformer-based models have enabled systems to learn hierarchical representations from speech signals. According to Vaswani *et al*, (2017), the introduction of attention-based learning in transformers, leads to learning long-term relationships in speech sequences, recognition accuracy and improve context awareness. Improved performance in conversational and multi-speaker scenarios has been achieved with environments end-to-end speech recognition combined with speaker diarization. To reduce dependence on manual feature works, end-to-end ASR systems integrate acoustic, linguistic and contextual modeling into a fused framework.

Chan *et al*. (2021) in their study "SpeechStew" reported that effectiveness of deep learning in large-scale ASR systems have been demonstrated. As an illustration they integrate multiple speech datasets to improve generalization across diverse acoustic conditions and speaking styles. According to Gao *et al.* (2022) there is need for a shift toward data-driven ASR and expandable systems capable of operating in unconstrained practical conditions. Enhanced reliability in noisy and low-resource can be brought about by self-supervised and multi-task learning approaches. Across languages, accents and noisy environments, Whisper demonstrates strong generalization making it suitable for practical deployment, a major achievement in modern ASR is OpenAI's Whisper model, that trains on hundreds of thousands of hours of multilingual and multitask speech data (OpenAI, 2023; Radford *et al*., 2023). Despite its high accuracy, it shows a key limitation which is the trade-off between performance and computational efficiency. Systems such as microcontrollers used in IoT applications find large neural models that contain millions or billions of parameters, unsuitable for resource-constrained embedded system. As a result, embedded ASR systems often use simpler noise reduction methods such as spectral subtraction, Wiener filtering and adaptive filtering.

Several model optimization techniques have been developed to address these challenges. Quantization reduces numerical precision in weights and activations to lower memory and computational requirements, while pruning removes redundant or less significant network connections. Knowledge distillation transfers knowledge from large "teacher" models to smaller "student" models suitable for embedded deployment (Han *et al*., 2016; Hinton *et al.*, 2015). These techniques have contributed to TinyASR and TinyML-based speech processing solutions and enabled the development of lightweight ASR systems. Embedded speech recognition modules such as EasyVR are energy-efficient and low-power, simple to use, easy to deploy, support small-vocabulary command recognition, operate offline using template-based or pattern-matching techniques, they are limited to predefined command sets, Least Mean Squares (LMS) adaptive filter is widely used due to its simplicity and computational efficiency (Haykin, 2002). There by restricting uses to applications such as home automation and basic voice-controlled systems.

To overcome these limitations of both large-scale and highly constrained systems, hybrid ASR architectures have become more popular. In these systems, lightweight embedded models handle tasks such as wake-word detection or keyword spotting, while processing-heavy, speech recognition is offloaded to cloud-based services or more powerful processors. In commercial voice assistants this architecture balances speed, accuracy and efficiency.

Environments induced background noise, reverberation and overlapping speech, reduces performance. under such conditions. Deep learning-based speech enhancement and separation techniques make speech clearer (Wang and Chen, 2018). Environmental robustness using deep neural networks has also shown improved performance in noisy environments (Zhang *et al*., 2018).

The resource intensity of these techniques makes it difficult for deployment in low-power embedded systems. Embedded ASR deployment has been advanced by introduction of Tiny Machine Learning (TinyML) with frameworks such as TensorFlow Lite for Microcontrollers enable real-time inference on devices with extremely limited memory and power budgets (Banbury *et al*., 2021). While maintaining low latency and energy efficiency Keyword spotting models can now run on microcontrollers with only a few kilobytes of memory (Warden and Situnayake, 2019). Without significantly affecting accuracy Depthwise separable convolutions method further reduce computational complexity.

Keyword Spotting (KWS) is ideal for IoT applications as it focuses on detecting specific trigger words or commands. In offline KWS audio data does not need to be transmitted to external servers, thereby ensuring privacy, reduction in latency and security risks.

Due to differences in accent, pronunciation and speaking style recognition performance often varies across users. Embedding techniques and Lightweight speaker adaptation allow systems to learn user-specific characteristics without significantly increasing computational cost, thereby improving usability in multi-user environments.

Lightweight publish/subscribe model such as the Message Queuing Telemetry Transport (MQTT) protocol is widely used due to its low bandwidth requirements (OASIS, 2019). A unified IP-based communication standard that improves interoperability among smart home devices is the Matter protocol (Connectivity Standards Alliance, 2022).

Cloud-based systems offer high accuracy due to large-scale computational resources but suffer from latency, dependency on internet connectivity and privacy concerns. Local processing provides better privacy and faster response times, but is limited by hardware constraints. Hybrid architectures offers both approaches to optimize performance, efficiency and reliability. In smart home and assistive applications, multimodal feedback mechanisms including visual, auditory and haptic cues enhance usability while user experience is strongly influence by factors such as recognition accuracy, response time and feedback clarity.

Useful solutions for IoT-based ASR systems are provided by emerging technological approaches such as TinyML, keyword spotting, hybrid architectures and lightweight noise reduction techniques Communication protocols like MQTT and Matter further enable seamless system combination with IoT ecosystems. Large-scale models such as Whisper demonstrate exceptional performance, their deployment in resource-constrained environments remains challenging.

With lots of research efforts made at improving the quality of system output, challenges such as noise robustness, computational efficiency, scalability, privacy preservation and personalization remain. With hybrid communication architectures integrated with embedded ASR, TinyML-based processing, adaptive noise reduction offers a promising pathway toward expandable, efficient and reliable voice-controlled IoT systems operating in real-life noisy environments.

2. Methodology

An experimental system design approach was used in this work to implement and evaluate a noise-robust embedded speech recognition framework for real-time voice-controlled automation. This approach combines embedded hardware, lightweight automatic speech recognition (ASR), adaptive signal preprocessing and Internet-of-Things (IoT) communication capable of operating and able to run on low-resource hardware. The study aims to develop an expandable and low-cost Voice-Activated Switching System (VASS) that works offline for speech recognition while still communicating with distributed devices via a low-overhead communication protocol. The methodology is structured to guarantee the design, implementation and evaluation of the system are closely connected, allowing performance to be tested under real-world conditions. The system architecture separates functionality into five logical stages: acquisition, processing, decision, communication and

evaluation stages. The input stage, an electret microphone emits audio signals that is conditioned for processing. The preprocessed signal is received by an embedded recognition module, which mines relevant features and performs keyword detection. Recognized commands are interpreted by the microcontroller and through relay interfaces stimulates connected loads. At the same time command states and control signals are exchanged with external devices via a lightweight MQTT-based communication protocol. For performance analysis the architecture concludes with a feedback and evaluation component that records system responses.

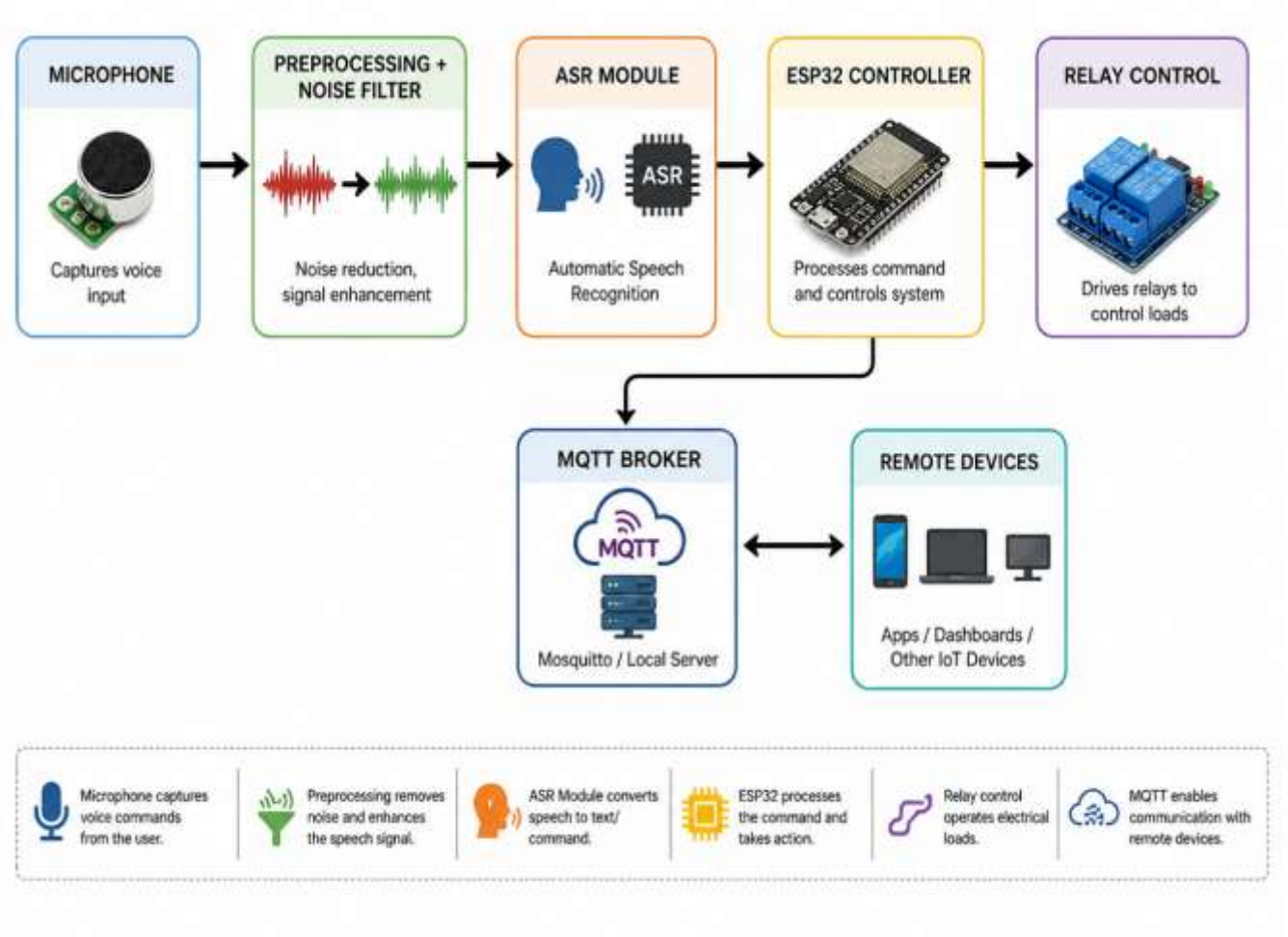

Fig. 1 System architecture and control flow diagram

A conceptual representation of this architecture is illustrated Fig 1. The hardware implementation is driving by affordability, modularity and energy efficiency. The dual-core architectural capacity, integrated Wi-Fi capability and sufficient memory resources of the ESP32-WROOM-32 microcontroller, support both control logic and lightweight machine learning inference. Due to its offline operation and low latency capability the EasyVR 3 Plus module is employed for embedded speech recognition. Audio acquisition is handled by a high-sensitivity electret microphone, while a two-channel opto-isolated relay module allows safe switching of electrical loads. To ensures stable system operation regulated 12 V DC power supply is employed and to facilitate modular testing standard prototyping tools such as breadboards and jumper wires are used. The summary of hardware configuration and specifications are as shown in Table 1.

Table 1: Hardware configuration and specifications.

Component	Description	Key Features	Rationale
ESP32-WROOM-32	Dual-core microcontroller	240 MHz, 520 KB SRAM	Provides sufficient processing capability for embedded machine learning tasks and supports MQTT-based IoT communication.
EasyVR 3 Plus	Embedded speech recognition module	Offline command recognition	Enables fast voice command processing without requiring internet connectivity.
Electret Microphone	Analog voice input sensor	High sensitivity	Captures speech signals effectively, especially for close-range voice interaction.
Relay Module	Load switching interface	Opto-isolated	Ensures safe switching and control of connected electrical appliances.
Power Supply	12 V DC regulated source	1 A output	Delivers stable and reliable power to all hardware components.
Prototyping Tools	Breadboard and connectors	Modular	Simplifies circuit assembly, testing, and troubleshooting during development.

The firmware workflow begins with continuous audio sampling from the microphone. The captured signal is then passed through preprocessing stages, including filtering and voice activity detection, to identify valid speech segments while minimizing the influence of environmental noise. It then translates these recognized commands into control signals that drives the relay module or are transmitted via MQTT for remote actuation. Signal processing, speech recognition and communication control constitutes the three primary functional blocks of the software architecture which is implemented on an embedded C/C++ platform. The overall software logic is illustrated in Fig 2.

To enhance performance under noisy conditions, adaptive noise filtering is incorporated into the preprocessing stage. To further improve performance, Least Mean Square (LMS) adaptive filter is implemented to minimize the mean square error difference between the noisy input signal and the estimated clean signal. The filter output is defined as:

$$y(n)\ \ \mathrm{x(n)} - \sum_{k=0}^{M-1} w_k(n) d(n-k) \quad (1)$$

where x(n) represents the noisy input signal and $w_k(n)$ are the adaptive filter coefficients. The coefficient update rule is given by

$$w_k(n+1) = w_k(n) + 2\mu e(n)x(n) \quad (2)$$

where $e(n)$ is the instantaneous error and μ is the learning rate and. For the filter to continuously adjust to changing noise conditions the adaptive approach is adopted, thereby improving signal clarity prior to recognition.

To reduce unnecessary computation and false activations a voice activity detection (VAD) mechanism is implemented alongside with the filter. The VAD algorithm uses short-term energy estimation, where the root mean square (RMS) energy of each frame is compared against a threshold defined as a fraction of the baseline noise level. Frames satisfying the condition

$$E_{frame} > 0.2 \text{ X } E_{noise} \quad (3)$$

are classified as speech and forwarded to the recognition module, while others are discarded.

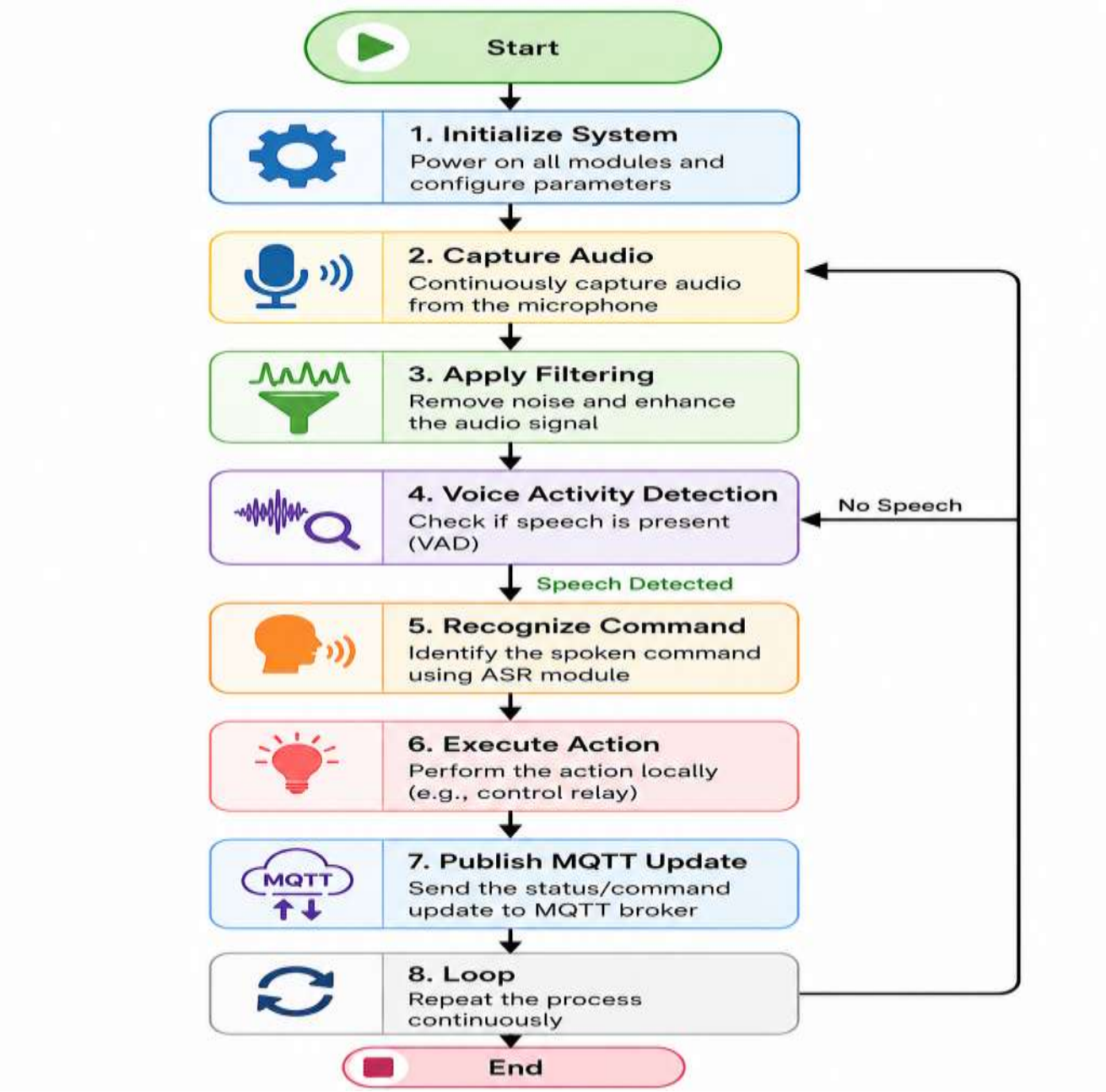

Fig. 2 Operational Flowchart of the Embedded Voice-Activated Switching System (VASS)

The speech recognition component is implemented using a hybrid approach, the EasyVR module provides template-based recognition for a predefined set of commands, while for comparative evaluation a TinyML-based keyword spotter is deployed on the ESP32. Each command is trained with multiple repetitions to improve reliability through template averaging, these commands are organized into two groups: speaker-independent commands such as "ON" and "OFF," and speaker-dependent commands such as "LIGHTS" and "FAN.". The TinyML model is trained using Mel-spectrogram features extracted from 8 kHz audio signals. The model is quantized to 8-bit integer precision, resulting in a compact size of approximately 42 KB, making it suitable for embedded deployment. While maintaining high accuracy the use of quantization significantly reduces memory requirements and inference latency. The MQTT protocol ensures communication between the embedded system and external devices while Mosquitto broker is deployed on a local server and the ESP32 acts as both a publisher and subscriber. Device states are published to the topic.

$$vass/device1/state$$

while control commands are received via

$$vass/device1/cmd$$

A Quality of Service (QoS) level of 1 is used to ensure reliable message delivery. To ensure that communication remains secure even in distributed environments security is enforced through authentication and optional Transport Layer Security (TLS) encryption,

Three noise environments are considered: a quiet laboratory setting, a moderately noisy domestic environment and a high-noise scenario with strong background interference. The corresponding signal-to-noise ratios (SNRs) are approximately 35 dB, 15 dB and 5 dB, respectively. Performance is evaluated using metrics including recognition accuracy, latency, false activation rate and user satisfaction.

Recognition accuracy is computed using

$$\text{Accuracy } \frac{\mathrm{N_{correct}}}{N_{total}} X\ 100 \quad (4)$$

where $\mathrm{N_{correct}}$ denotes the number of correctly recognized commands. Table 2 shows the relationship between SNR and recognition accuracy.

Table 2: The relationship between SNR and recognition accuracy.

SNR (dB)	EasyVR Accuracy (%)	TinyML Accuracy (%)
35	93	96
20	88	94
10	82	90
5	73	84

The TinyML model consistently outperforms the template-based approach under noisy conditions however, the results show a clear degradation in performance with decreasing SNR. The time interval between the end of a spoken command and the activation of the corresponding relay is the measured latency. The summary of the observed latency values is as shown in Table 3.

Table 3: Observed latency values

System Configuration	Average Latency (ms)	Standard Deviation (ms)
EasyVR Standalone	235	±28
EasyVR with MQTT	285	±35
TinyML on ESP32	210	±25

User evaluation was conducted with twelve participants across a range of age groups. Each participant interacts with the system under different noise conditions, issuing multiple commands. With mean scores exceeding 4.0 across all evaluation criteria, feedback was collected using a five-point Likert scale and results indicate high levels of usability,

The results obtained from Pearson correlation coefficient analysis revealed that the correlation between SNR and recognition accuracy indicates a robust positive relationship between signal quality and recognition performance, with correlation coefficient of:

$$r = 0.91 \quad (5)$$

The results from using paired-sample t-tests for statistical validation to compare system configurations, indicates statistically significant differences in performance ($p < 0.05$), confirming the superiority of the TinyML-based approach under identical conditions.

The proposed methodology a well-integrated framework, that combined embedded hardware implementation, adaptive signal processing, lightweight speech recognition and IoT-based communication into a functional unit. The experimental design undergoes rigorous evaluation across multiple conditions, while the hybrid architecture balances performance, cost and privacy. Ethical considerations were addressed as participants provided informed consent before participation and data handling procedures were in compliance with established research ethics guidelines by ensuring that all speech processing occurs locally, thereby preserving user privacy.

3. Results and Discussion

The system was implemented using an ESP32-WROOM-32 microcontroller connected to an EasyVR 3 Plus speech recognition module, an electret microphone and a relay switching unit. MQTT communication was handled through a Mosquitto broker running on a Raspberry Pi. The study examined whether the combined EasyVR–ESP32–TinyML architecture could provide reliable and real-time speech recognition in practical environments while maintaining low power consumption, privacy and affordability.

System testing was carried out in both laboratory and domestic environments to evaluate and simulate practical deployment conditions. Ambient noise levels were systematically varied from quiet conditions (30–35 dB SPL) to highly noisy circumstances reaching about 70 dB SPL using recorded sources such as television audio, fan noise and background speech. A dataset comprising of six voice commands “Light On,” “Light Off,” “Fan On,” “Fan Off,” “Power On,” and “Power Off” were collected from twelve participants with different accents. Each participant repeated each command ten times under each noise condition, yielding a comprehensive dataset for performance assessment. Before recognition, the speech signals were preprocessed using spectral subtraction and pre-emphasis filtering for better speech quality.

Recognition accuracy was evaluated under different signal-to-noise ratios (SNRs), by comparing the performance of the template-based EasyVR module with the TinyML-based convolutional neural network (CNN) model implemented on the ESP32. Table 4 shows the summary of the result.

The results revealed a clear trend in which accuracy improves with increasing SNR for both systems. At low SNR levels (0–10 dB), the TinyML model significantly perform better than the EasyVR module, maintaining accuracy above 70% even in highly noisy conditions.

Table 4. Performance of the template-based EasyVR module with the TinyML-based convolutional neural network model.

SNR (dB)	EasyVR Accuracy (%)	TinyML Accuracy (%)	Relative Gain (%)
0	45.3	71.2	+25.9
5	60.4	83.5	+23.1
10	72.6	91.4	+18.8
20	89.1	95.3	+6.2
30	93.5	97.1	+3.6
40	94.8	98.2	+3.4

This improved performance can be attributed to the CNN's capacity to learn more stable speech patterns from Mel-spectrogram features even in the presence of noise. The EasyVR module, exhibits a sharp degradation in performance due to sensitivity to noise-induced distortions. At higher SNR levels (≥20 dB), both systems achieve accuracy exceeding 90%, confirming that low-cost embedded solutions can perform reliably under moderate acoustic conditions.

The results, presented in Table 5, include contributions from signal acquisition, preprocessing, recognition, communication and actuation.

Table 5: System measured latency.

Test Condition	EasyVR Latency (ms)	TinyML Latency (ms)	MQTT Delay (ms)
Quiet Environment	280	350	25
Moderate Noise	315	370	30
High Noise	340	395	35
Cloud ASR Baseline	900		120

The system latency was measured as the time between the completion of a spoken command and the switching of the corresponding relay. The results show that both embedded approaches maintain response times below 400 ms, satisfying real-time interaction requirements. The TinyML model introduces a slight increase in latency due to neural inference overhead but remains within acceptable limits. The cloud-based baseline exhibits significantly higher latency, approaching 900 ms, primarily due to network transmission delays. This confirms the advantage of on-device processing for latency-sensitive IoT applications.

MQTT communication performance was evaluated by transmitting control messages of 1 Hz repeatedly over 1,000 communication cycles. Key metrics including publish success rate, round-trip time (RTT) and packet loss were recorded, as shown in Table 6.

Table 6. MQTT communication performance.

Parameter	Mean Value	Standard Deviation
Publish Success Rate (%)	99.84	0.16
Mean RTT (ms)	27.6	2.3
Lost Packets (%)	0.1	
Throughput (msg/s)	15.7	

These results confirm the high reliability and efficiency of MQTT for embedded IoT communication while the low RTT and minimal packet loss indicate that the protocol is well-suited for real-time voice-controlled systems. The publish–subscribe model, enables synchronized control of multiple devices with negligible overhead and ensures efficient message dissemination.

Power consumption measurements were conducted to assess the feasibility of the system for energy-constrained deployments. The results are summarized in Table 7.
The total power consumption shows an approximately 1.3 W during active operation indicates that the system can be powered via USB or battery sources, making it suitable for portable and off-grid applications. The total hardware cost remained below ₦35,000 reinforcing the feasibility of low-cost deployment.

Table 7. Power consumption measurements.

Component	Idle Current (mA)	Active Current (mA)	Voltage (V)	Active Power (mW)
ESP32	90	180	3.3	594
EasyVR Module	40	65	5.0	325
Microphone	8	10	5.0	50
Relay Module	12	70	5.0	350
Total System	150	325		1319

User evaluation was conducted with fifteen participants using a five-point Likert scale. The results are presented in Table 8 indicates high user satisfaction across all assessment categories.

Table 8. User evaluation.

Evaluation Metric	Mean Score	Std. Dev
Recognition Reliability	4.4	0.5
Response Time Satisfaction	4.6	0.4
Ease of Use	4.1	0.7
Noise reliability Perception	4.3	0.6

The participants responded positively to the system's respond speed and simplicity of the process while expressing user satisfaction, some limitations were observed under extremely noisy conditions where occasional false activations occurred, suggesting that further improvement in filtering and command verification to improve overall performance. A comparative analysis was carried out between the proposed combined system, to benchmark the proposed hybrid system against the standalone EasyVR system and a cloud-based ASR system. The summary of the results are as shown in Table 9.

Table 9. Comparative evaluation of the proposed hybrid system against standalone EasyVR and cloud-based ASR solutions.

Metric	EasyVR Standalone	Cloud ASR	Proposed Hybrid
Accuracy (%)	87.2	96.4	95.1
Latency (ms)	310	900	365
Offline Capability	Yes	No	Yes
Privacy	High	Low	High
IoT Integration	No	Yes	Yes
Cost (₦)	25,000		35,000

Offline functionality and user privacy. Cloud systems suffer from higher latency and dependence on network connectivity but provide slightly higher accuracy.

The proposed system strikes a balance between performance, cost and usability, making it suitable for practical IoT deployments. The hybrid architecture achieves near-cloud accuracy while preserving

The relationship between SNR and recognition accuracy shows a positive correlation, with a Pearson coefficient of approximately r = 0.91, this confirms that environmental noise level affects ASR performance. The TinyML model shows slower degradation compared to the template-based approach, confirming that its handled noisy conditions better than the temple-based approach. Latency analysis reveals that response times below 400 ms align with human acceptable limits and ensures seamless interaction.

The results also highlight the efficiency of MQTT as a lightweight communication protocol, for IoT system. Its low latency and high reliability support distributed devices without affecting system responsiveness. The combination of local processing and lightweight messaging allows distributed control without compromising responsiveness. From a cost performance perspective, achieving approximately 95% accuracy at a system cost below ₦35,000 demonstrates the viability of embedded ASR solutions in resource-constrained environments.

Generally, the results confirm the effectiveness of the proposed hybrid architecture in delivering noise-robust, low-latency and cost-efficient voice control. The integration of TinyML-based keyword spotting, adaptive noise filtering and MQTT communication provides a practical framework for next-generation embedded speech systems. Future improvements may focus on advanced noise suppression techniques, adaptive personalization and enhanced security mechanisms to further optimize system performance in diverse deployment scenarios.

4. Conclusion

The system achieved high recognition accuracy, reaching up to 98% under quiet conditions and maintaining acceptable performance even in low SNR environments. The results of the study show that a hybrid architecture of EasyVR, ESP32 and TinyML can be used to build a noise-robust embedded speech recognition system. At SNR levels below 10 dB the TinyML-based model perform better than the template-based EasyVR module. The system meets real-time interaction requirements for voice-controlled IoT applications as it consistently delivered low latency, with response times below 400 ms. The MQTT validates its suitability for distributed smart environments as it ensures reliable and efficient data exchange, achieving over 99.8% message delivery success with minimal delay.

Apart from technical performance, the study showed that high-quality voice-controlled IoT operation can be achieved at low cost and with strong privacy preservation. Local processing eliminated the need for continuous cloud connectivity, reduces latency and safeguards user data while the total system cost remained under ₦35.000. User evaluation results indicate high user satisfaction, emphasizing the importance of responsiveness and reliability in practical deployment.

In general, the results establish that the proposed combined approach successfully balances accuracy, efficiency, cost and usability, thereby offering expandable and practical solution for voice-controlled automation systems. The work provides for future enhancements as a foundation for improved noise handling, adaptive personalization and further integration with newer IoT standards.

References

Banbury, C.R., Reddi, V.J., Lam, M., Fu, W., Fazel, A., Holleman, J. and others 2021. *MLPerf Tiny Benchmark: Evaluating TinyML Systems on Embedded Devices*. arXiv preprint.

Chan, W., Park D, Lee,C., Zhang Y., Le Q., and Norouzi, M. 2021 'SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large Neural Network', *arXiv preprint*.
Connectivity Standards Alliance (2022) *Matter Specification: A Unified IP-Based Connectivity Standard for Smart Devices*. Available at: https://csa-iot.org (Accessed: 8 May 2026).
Gao, S., et al. (2022) 'End-to-End Neural Speaker Diarization for Conversational Speech Recognition', *IEEE/ACM Transactions on Audio, Speech and Language Processing*.
Han, S., Mao, H. and Dally, W.J. (2016) 'Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding', *International Conference on Learning Representations (ICLR)*.
Haykin, S. (2002) *Adaptive Filter Theory*. 4th edn. Upper Saddle River, NJ: Prentice Hall.
Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. and Kingsbury, B. (2012) 'Deep Neural Networks for Acoustic Modeling in Speech Recognition', *IEEE Signal Processing Magazine*, 29(6), pp. 82–97.
Hinton, G., Vinyals, O. and Dean, J. (2015) 'Distilling the Knowledge in a Neural Network', *arXiv preprint arXiv:1503.02531*.
Nassif, A.B., Shahin, I., Attili, I., Azzeh, M. and Shaalan, K. (2019) 'Speech Recognition Using Deep Neural Networks: A Systematic Review', *IEEE Access*, 7, pp. 19143–19165.
OASIS (2019) *MQTT Version 5.0 Specification*. Available at: https://docs.oasis-open.org/mqtt (Accessed: 8 May 2026).
OpenAI (2023) *Whisper: Robust Speech Recognition via Large-Scale Weak Supervision*. Available at: https://openai.com/research/whisper (Accessed: 8 May 2026).
Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C. and Sutskever, I. (2023) *Robust Speech Recognition via Large-Scale Weak Supervision*. OpenAI Technical Report.
Ravanelli M,, J Zhong, S Pascual, P Swietojanski, J Monteiro, J Trmal, Y Bengio (2021) 'Multi-Task Self-Supervised Learning for Robust Speech Recognition', *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) 'Attention Is All You Need', *Advances in Neural Information Processing Systems (NeurIPS)*.
Wang, D. and Chen, J. (2018) 'Supervised Speech Separation Based on Deep Learning: An Overview', *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 26(10), pp. 1702–1726.
Warden, P. and Situnayake, D. (2019) *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA: O'Reilly Media.
Zhang, X., Wang, D. and others (2018) 'Robust Speech Recognition in Noisy Environments Using Deep
Neural Networks', *IEEE/ACM Transactions on Audio, Speech and Language Processing.*