Design And Rationality Of Natural Gas Dispatching System Architecture Using Voice Control Instructions With Unique-Id For Improving Traceability

Hongtao Diao, Yu Li, Zengrui wang, Youyi Liang, and Yi Yang*

Pipe China Oil & Gas Control Center, Beijing, 100013 China

* *Corresponding author. E-mail: Yi_Yang97@outlook.com*

Received: Jul. 13, 2026; Accepted: Aug. 22, 2026

Natural gas dispatching (NgD) systems increasingly employ voice control instructions to improve operational efficiency and human–machine interaction. However, existing voice-command frameworks lack end-to-end command traceability, limiting failure localization and compromising operational reliability and cybersecurity. This study proposes a natural gas dispatching system architecture that integrates unique identifiers with voice control instructions to enhance command traceability, authentication, and secure execution. The proposed framework assigns a unique ID to each voice command and integrates Exponential Growth Spectral Subtraction (EG-SS) for industrial noise suppression, Gaussian Mixture Model (GMM)-based speech recognition, Grunwald–Letnikov Secure Hash Algorithm-3 (GL-SHA-3) for command integrity verification, Consensus Finite State Machine (C-FSM) for operational limit validation, and SSG-αSiLUGRU for intent recognition and attack detection within the SCADA environment. Experimental results using voice command data and the SCADA Cybersecurity Research dataset collected from a simulated SCADA environment demonstrate that the proposed framework achieves 98.945% intent recognition and attack detection accuracy, a 20.5 dB Noise Reduction Ratio, 32 dB Signal-to-Noise Ratio, 687 ms hash creation time, 654 ms hash verification time, and 98.46% state transition accuracy, outperforming conventional approaches. These findings confirm that the proposed architecture effectively enhances command integrity, enables precise failure localization, strengthens cybersecurity against malicious voice-command attacks, and improves the reliability, safety, and traceability of natural gas dispatching operations.

Keywords: Natural gas Dispatching (NgD) System, Voice Control Instructions (VCI), Traceability, Intent Recognition, Supervisory Control and Data Acquisition (SCADA).

http://dx.doi.org/10.6180/jase.202612_35.033

1. Introduction

Industrial automation enhances safe, efficient critical infrastructure operations, while SCADA enables continuous pipeline monitoring. Voice-based human–machine interaction further improves operational efficiency and minimizes manual intervention in natural gas dispatching systems. Voice Control Instructions (VCI) enable hands-free industrial human-machine interaction and safer equipment operation [1]. In natural gas dispatching (NgD), manual valve and compressor control may cause human errors and delayed responses during critical events [2]. Voice commands are also vulnerable to attacks such as dolphin attacks, which can trigger malicious actions, including emergency shutdowns and pipeline pressure manipulation [3]. Therefore, authentication is essential for secure voice-command execution [4]. Recent studies use VGG-16, LSTM,

and DNN-based methods for voice recognition and attack detection [5, 6]. However, existing approaches lack unique command identifiers for end-to-end traceability. Hence, this study proposes a uniquely identified voice-control architecture for enhanced authentication and traceability.

The traditional methodologies exhibit certain limitations, which are outlined below,

- Lack of unique command identifiers limits end-to-end traceability.
- Insufficient protection against unauthorized command injection threatens system integrity.
- Traditional ASR struggles with machinery, wind, and environmental noise.
- Broker failures can cause message loss or delay critical commands.

The proposed framework ensures command traceability through unique IDs, secures voice commands using GL-SHA-3 and SSG- α SiLUGRU attack detection, removes noise via EG-SS, and employs a local persistent SCADA queue to prevent message loss.

The novelty of the proposed framework lies in integrating unique-ID-based command traceability with hash-based integrity verification, intent recognition, operational limit validation, and attack detection into a unified architecture for natural gas dispatching systems.

1.1. Research Gap and Motivation

Existing studies lack end-to-end unique-ID traceability and unified integration of noise-robust recognition, hash verification, limit checking, and attack detection. This work integrates persistent unique-ID traceability, EG-SS, GL-SHA-3, C-FSM, and SSG- α SiLUGRU into a unified SCADA framework, enabling secure, reliable, auditable, end-to-end voice-command execution.

The rest of the paper is structured as follows: Sections 2 and 3 present the analysis of the existing works and the proposed methodology, respectively. Results and discussion are presented in Section 4. Section 5 presents the conclusion and future work.

2. related works

This section reviews existing research related to voice command systems in industrial environments, with a focus on three key areas: speech recognition and noise robustness, security and attack detection, and command traceability.

2.1. Speech Recognition and Noise Robustness

Studies have explored CNN-based voice command recognition, accelerometer-based speech reconstruction using STFT and Griffin-Lim, and hierarchical variational inference for zero-shot speech synthesis [7]. However, limitations remain in modeling speech sequences, capturing phoneme-specific features, and preserving semantic clarity.

2.2. Security and Attack Detection in Voice Systems

Voice-system security studies address liveness detection, adversarial attacks, and spoofing defense [8]. However, existing approaches primarily examine individual attack or detection mechanisms rather than integrated industrial control security.

2.3. Hardware and System-Level Approaches

Recent studies highlight uncertainty-aware methods for reliable industrial decision-making. Statistical fault diagnosis addresses uncertain conditions [9], while fuzzy and neutrosophic measures improve transformer defect diagnosis [10]. Optimization and adaptive control enhance energy management and load-frequency regulation [11, 12]. These advances motivate reliable and secure decision-making for natural gas dispatching systems. Existing studies separately address speech recognition, spoofing, authentication, and hardware security, whereas this framework integrates traceability, noise suppression, integrity verification, operational validation, and attack detection within SCADA.

Critical Review: Hardware-based solutions enhance physical security but lack software-level accountability, while system-level approaches remain conceptual and fail to provide concrete architectures for end-to-end command management.

2.4. Limitations of Existing Studies and Justification for the Proposed Framework

Existing studies address speech recognition, authentication, and attack detection separately, lacking end-to-end traceability. The proposed framework integrates unique-ID traceability, hash verification, limit validation, attack detection, and persistent queuing for secure execution.

3. Proposed methodology

Section 3 presents the proposed NgD framework for secure, traceable voice-command execution. As shown in Fig. 1, commands undergo recognition, preprocessing, feature extraction, intent recognition, GL-SHA-3 verification, C-FSM checking, and SSG- α SiLUGRU

The proposed framework systematically develops and evaluates its major components, including noise suppres-

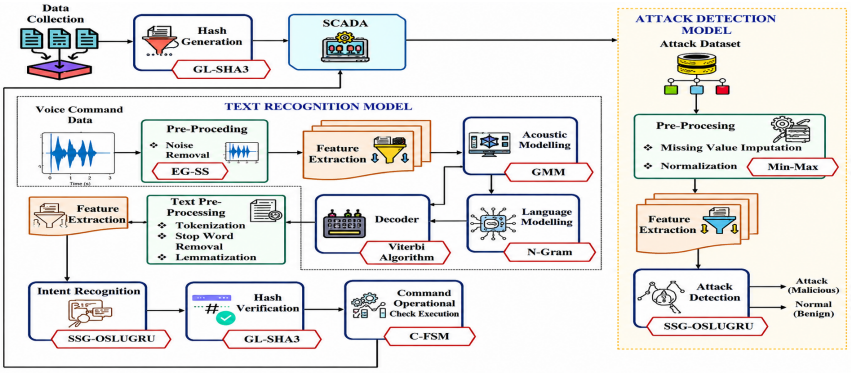


Fig. 1. Block diagram of the proposed framework.

sion, speech recognition, integrity verification, operational checking, and attack detection, using appropriate performance measures [13].

3.1. Data collection

Primarily, the set of commands (instructions) (g_q) from the human speakers related to NgD system is collected from the publicly available data sources.

$$g_q = \{g_1, g_2, \ldots\ldots g_Q\} \tag{1}$$

Where g_q denotes the Q voice command (instruction) collected for the natural gas dispatching (NgD) system, and (Q)represents the total number of collected voice commands.

3.2. Dataset Description

In this proposed work, a set of commands related to natural gas dispatching operations is manually collected from the publicly available data sources to support the design and rationality of the NgD system architecture. The MSVC dataset contains English commands from 20 volunteers across five classes: Lights On, Lights Off, Music On, Music Stop, and Next Song. Each speaker recorded 15 commands (14-15 after cleaning), using four sensors at three positions. Each utterance lasted 3 s, with WAV format and 5.9 h audio per sensor. The SCADA Cybersecurity Research dataset contains 48,466 network traffic records from simulated SCADA operations, with 80%$(38,773)$ for training and 20%$(9,693)$ for testing attack detection.

The collected datasets were randomly divided into training and testing subsets using an 80:20 ratio. The training set was used for model training and parameter optimization, while the remaining 20% was reserved exclusively for unbiased performance evaluation. The study utilized manually compiled public natural-gas voice commands, the public MSVC dataset for speech recognition, and the SCADA Cybersecurity Research dataset containing 48,466 simulated network traffic samples representing normal and attack classes. All datasets were publicly available for research purposes and originated from public or simulated sources.

3.3. Hash Generation

Next, the hash is generated for $(f\,(g_q))$ is generated using GL-SHA-3 for data integrity and command authentication. SHA-3 provides strong security, while the GL function reduces absorption-round complexity, improving computational efficiency. Table 1 presents the GL-SHA-3 implementation parameters, including Keccak settings, GL parameters, and round configuration, while Table 2 provides representative test vectors with input commands, unique IDs, and generated 256 -bit hashes for reproducibility.

Table 1. Implementation Parameters of the Proposed GL-SHA-3.

Parameter	Value
Keccak state size	1600 bits
Hash output length	256 bits
Rate (r)	1088 bits
Capacity (c)	512 bits
Padding scheme	SHA-3 pad10*1
GL fractional order (α)	0.5
GL time interval (Δt)	0.01
Standard Keccak-f rounds	24
GL-selected absorption rounds	12
Number of squeeze operations	1
Input encoding	UTF-8
Hash representation	64 hexadecimal characters

Step 1: Padding the input

Primarily, all bits (e) of the internal state $(x_{\text{int}}())$ are set to zero, and this zero state is denoted by(e^0). Next, $(f\,(g_q))$

Table 2. Test vectors for the proposed GL-SHA-3.

Input Command	Unique ID	Generated Hash (256-bit)
Open valve 01	VCI-0001	8f2a1c6d9b4e7a3f5c1d8e2b6a9f0c4d7e3b5a1c9f6d2e8b4a 7c0f3d6e9b2a5
Close valve 02	VCI-0002	3a7e9c1f5b2d8a6e4c0f7b1d9e3a5c8f6d2b4a7e1c9f3d5b8a 0e6c2f4d7b9a1
Increase pressure to 50 bar	VCI-0003	b4d8f2a7c1e9b5d3f6a0c8e2b7d4f1a9c5e3b8d6f2a7c0e4b9 d1f5a8c3e6b2

is padded to a specific length to prevent specific vulnerabilities.

$$\mathfrak{I}^{\text{padd}} = f\left(g_q\right) \oplus \varepsilon^{\circ} \tag{2}$$

Where, I^{natd} denotes the padded input message, $f\left(g_q\right)$ represents the unique ID-assigned voice command, $\oplus$ denotes the bitwise XOR operation, and ε^0 represents the zero-padded state with values in the range $(0, 1)$.

Step 2: Absorption Phase

Further, $\left(\mathfrak{I}^{\text{pad}}\right)$ is divided into blocks $\left(\overline{\mathfrak{I}}_s^{\text{padd}}\right)$ and it is defined as,

$$\overline{\mathfrak{I}}_s^{\text{padd}} = \left\{\overline{\mathfrak{I}}_1^{\text{padd}}, \overline{\mathfrak{I}}_2^{\text{padd}}, \ldots\ldots \overline{\mathfrak{I}}_S^{\text{padd}}\right\} \tag{3}$$

Where, $\bar{I}_s^{\text{rald}}$ denotes the s^{th} padded input block, and S represents the total number of padded input blocks. The output of the absorption phase ($\ddot{\tau}$)is expressed as,

$$\ddot{\tau} = \dot{X}, \alpha^{kp}\left(\sum \overline{\mathfrak{I}}_s^{\text{padd}}\right) \otimes \breve{\wp} \tag{4}$$

Where $\oplus$ denotes XOR, Keccak represents permutation, $\oplus$ denotes GL operation, and the Grunwald-Letnikov function determines absorption rounds.

Step 3: Squeezing Phase

Afterward, in the squeezing phase, the permutation function is applied to ($\ddot{\tau}$) for generating a new shuffled state $\left(\delta^{\text{shuff}}\right)$.

$$\delta^{\text{shuff}} = \alpha^{kp} \times \ddot{\tau} \tag{5}$$

Where, δ^{shuff} is the shuffled state, α^{kp} the Keccak permutation, and $\tau^{"}$ the absorption output. Hash uses $(\bar{r}_1)$ from (δ^{shuff}).

$$\chi^{hh} = \bar{r}_1\left(\delta^{\text{shuff}}\right) \tag{6}$$

Where, x^{ht} is the generated hash, r_1 is the output rate, δ^{shuff} is the shuffled state, and $f\left(g_q\right)$ denotes the uniquely ID-assigned voice command.

3.4. Text Recognition Model

Voice commands are recognized through data collection, preprocessing, feature extraction, acoustic modeling, language modeling, and decoding.

3.4.1. Voice Command Data

Initially, the voice command (V_l) data is collected from the Multi-Sensor Voice Command (MSVC) dataset. It is initialized as follows,

$$V_l = \{V_1, V_2 \ldots\ldots V_L\} \tag{7}$$

Where, (L)indicates the maximum number of (V_l). Then, a unique ID$(\dot{f})$ is assigned to (V_l) and it is denoted by $\left(V_l^{\dot{f}}\right)$.

3.4.2. Pre-processing

Machinery, wind, and environmental noise are removed using EG-SS, which adjusts the spectral floor to preserve speech and reduce distortion.

Initially, $\left(V_l^{\dot{f}}\right)$ is divided into (L) number of frames $\left(\tilde{V}_l^{\dot{f}}\right)$, and a window function $\left(\ell^{wf}\right)$ is applied to each frame.

$$\dot{V}_l = \sum \tilde{V}_l^{\dot{f}} \cdot e^{wf} \tag{8}$$

Where, $\left(\dot{V}_l\right)$ indicates the windowed frame. Then, the spectrum of $\left(\dot{V}_l\right)$ is estimated by applying Fast Fourier Transform (FFT) as follows,

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, k = 0, 1, \ldots, N-1 \tag{9}$$

where $x(n)$ denotes the windowed speech frame, $X(k)$ is the frequency-domain representation of the speech signal, N denotes the frame length, k represents the frequency-bin index, and $j = \sqrt{-1}$.

3.4.3. Feature Extraction

Significant features, including MFCC, spectrogram, glottal features, spectral tilt, slope, and cepstral peak prominence, are extracted from $(\breve{V})$ denoted as $\left(\breve{V}_{\text{ex}}\right)$.

3.4.4. Acoustic Modelling

Based on $\left(\breve{V}_{ex}\right)$, GMM estimates the phoneme sequence by probabilistically mapping acoustic features to phonemes. The number of Gaussian distributions $\left(\Psi_{gd}\right)$ and parameters (μ), (Φ^{co}), and $(\ddot{\xi})$ are initialized.

- *Expectation Step* $(E-Step)$

  Next, the E-step is performed for estimating the responsibility $(\hat{P})$ that each feature in $\left(\widetilde{V}_{ex}\right)$ belongs to $\left(\Psi_{gd}\right)$.

$$\hat{P} = \mu + \Phi^{co} + \ddot{\xi}\left(\sum \check{V}_{ex} \otimes \Psi_{gd}\right) \tag{10}$$

- *Maximization Step* $(M-Step)$

  In the M-step, parameters are updated iteratively to maximize likelihood. After convergence, the updated parameters yield the final phoneme sequence, $\left(A_{ph}\right)$.

3.4.5. Language Modelling

Further, based on $\left(A_{ph}\right)$, the N-gram model resolves voice-command ambiguity. First, $\left(A_{ph}\right)$ is tokenized into (τ^{tok}), then n -grams (v) are generated and conditional probability (P^{cd}) is determined as:

$$P\left(w_i \mid w_{i-(N-1)}, \ldots, w_{i-1}\right) = \frac{C\left(w_{i-(N-1),\ldots,}w_i\right)}{C\left(w_{i-(N-1)}, \ldots, w_{i-1}\right)} \tag{11}$$

where $C(\cdot)$ denotes the frequency count of the corresponding N -gram, and $P(\cdot)$ represents the conditional probability of predicting the current word from the preceding $N-1$ words.

3.4.6. Decoder

Next, based on $\left(A_{ph}\right)$ and $(\tilde{\chi})$,the Viterbi algorithm recognizes the corresponding text by determining the most probable hidden-state sequence.

$$\check{O} = A_{ph} \otimes \tilde{\chi} \tag{12}$$

Then, the initial Viterbi probability (δ^{vit})is evaluated to determine the consecutive sequence of the text.

$$\delta^{vit} = \gamma^{\circ}(\ddot{g}) \times E_{\ddot{g}\cdot}(\check{O}) \tag{13}$$

Where, (γ°), indicates the starting probability of the random state $(\ddot{g})$, $\left(E_{\ddot{g}\cdot}\right)$ denotes the emission probability of the initial $(\check{O})$.

3.5. Text Pre-Processing

Furthermore, (R_{txt}) is pre-processed by tokenization into (R_{txt}), stop-word removal producing $\left(\widetilde{\mathrm{R}}_{\mathrm{txt}}\right)$,and lemmatisation, yielding the base-form output (Ψ_{lmm}).

3.6. Feature Extraction

Here, the significant features like Bag of words, N-grams, Character n-grams, term frequencyinverse document frequency, word count, and All Caps Ratio (ACR) are extracted from (ψ_{lmm}), and the extracted features are denoted by $\left(\widetilde{D}_z\right)$.

$$\widetilde{D}_z = \left\{\widetilde{D}_1, \widetilde{D}_2 \ldots\ldots \widetilde{D}_{lm}\right\} \tag{14}$$

Where, (lm) indicates the total number of $\left(\widetilde{D}_z\right)$.

3.7. Intent recognition

In this section, based on $\left(\widetilde{D}_z\right)$, SSG- α SiLUGRU recognizes voice intent; SG reduces overfitting, while αSiLU mitigates vanishing gradients. Architecture shown in Fig. 2.

Table 3 summarizes the SSG- α SiLUGRU implementation and training configuration, including model dimensions, layer structure, activation, initialization, optimizer, learning rate, loss function, batch size, epochs, regularization, stopping criterion, random seed, and trainable parameters.

Table 3. Implementation and Training Parameters of SSG-α SiLUGRU.

Parameter	Value
Input / spatial dimension	50 / 64
GRU layers / hidden units	2 / 128
Output classes	2
Activation / α	αSiLU/0.01
Initialization	Sine-Gordon
Optimizer / learning rate	Adam / 0.001
Loss function	Cross-Entropy
Batch size / epochs	32 / 100
Regularization	L2 ($\lambda = 0.0001$)
Stopping criterion	Validation loss; patience = 10
Random seed	42
Trainable parameters	176,322

- *Spatial Feature Conversion Block*

  Primarily, the spatial features are captured from $\left(\widetilde{D}_z\right)$ for modelling the spatial relationship effectively.

$$\dot{J} = \sum \widetilde{D}_z \oplus b \tag{15}$$

  Where, (j) signifies the spatial feature with respect to $\left(\widetilde{D}_z\right)$ and (b) indicates the spatial structure of $\left(\widetilde{D}_z\right)$.

- *GRU Layer*

  Next, (j) enters the GRU, comprising reset (ψ^{re}) and update (ψ^{up}) gates; α SiLU prevents vanishing gradients. The reset gate uses $(\dot{J})$ and $(\bar{h}_{t-1})$.

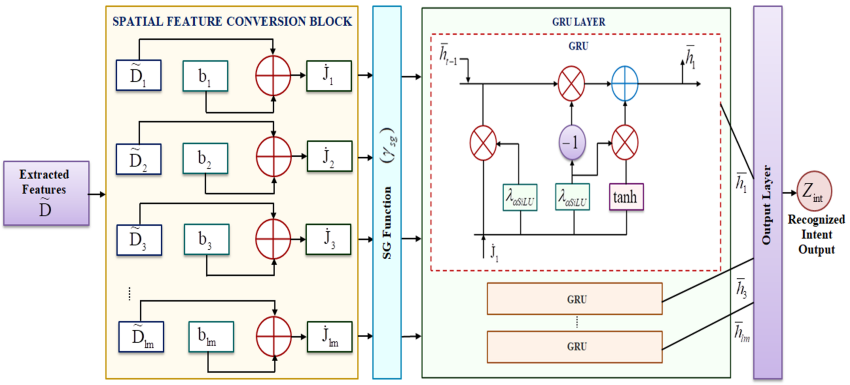


Fig. 2. Architecture of the proposed SSG- α SiLUGRU classifier.

$$\psi^{re} = \sum \chi_{\alpha silu}\left(w^{re} \times \left(\dot{J}, \bar{h}_{t-1}\right) + b^{re}\right) \tag{16}$$

Where, (w^{re}) indicates the weight parameter of (ψ^{re}) belonging to (γ_{sg}), (b^{re}) is the bias parameter of (ψ^{re}), $(\ddot{f})$ denotes the scaling parameter, and $(\tilde{u})$ denotes the weighted sum of inputs to (ψ^{re}).

- *Output Layer*

  Afterward, $(\bar{h})$ from the GRU layer is processed in the output layer to obtain the intent recognized output(Z_{int} ()).

$$Z^{\circ} \sum \left(w^{fc} \times (\bar{h}) + b^{fc}\right)_{int} \tag{17}$$

Where (S°) denotes softmax activation, $\left(w^{fc}\right)$ represents output-layer weights associated with (γ_{sg}), and $\left(b^{fc}\right)$ denotes output-layer bias.

3.8. Hash Verification

Using the process in Section 3.2, GL-SHA-3 generates a new hash $\left(Z^{hh}\right)$ for $(Z_{\text{int}}())$ preventing false command-data injection. The normalised recognised command together with its unique ID is processed using the same GL-SHA-3 hashing procedure applied during enrollment. The generated hash is then compared with the stored hash to verify command integrity, ensuring that identical inputs are used during both hash generation and verification.

3.9. Command Operational Check Limit

The equipment COL($\Im_{\text{equip}}$) is verified using C-FSM to ensure operational safety. Possible states $(\bar{S})$ in (Z_{int} (O) are evaluated, and state diagram (ε^{st}) models transitions using consensustheorem guard conditions.

$$\check{\beta} = \overleftrightarrow{E}\hat{Y} + \overleftrightarrow{E}'Z \tag{18}$$

Where, $(\bar{S}_{\text{tr}})$ denotes state transitions of $(\Im_{\text{equip}})$; $\left((\overleftrightarrow{E}), (\hat{Y}), (Z)\right.$ are variables in (ε^{st}), with $(\overleftrightarrow{E}\hat{Y})$ as AND, $\left(\overleftrightarrow{E'}\right)$ as NOT, and $\left(\overleftrightarrow{E'}Z\right)$ as OR.

3.10. Attack Detection Model

Before executing the command, the attack detection model is trained by collecting network traffic data, pre-processing, feature extraction, and attack detection using SSG- α SiLUGRU.

- *Attack Dataset*

  Here, the network traffic data (a'') is collected from the SCADA Cybersecurity Research dataset.

$$a'' = \left\{a''_1, a''_2 \ldots\ldots\ldots a''_{yy}\right\} \tag{19}$$

Where, (yy) indicates the maximum number of (a'').

- *Pre − processing*

  Moreover, (a'') is pre-processed to standardize and to improve the quality of data. The steps involved in pre-processing are described further.

- *Missing Value Imputation*

  First, the missing values in (a'') are imputed by determining the mean of the available data, thereby handling the incomplete information. The missing value imputed data is denoted by $(\dot{A})$.

- Normalization

  Then, $(\dot{A})$ is normalized using min-max normalization technique to standardize the data.

$$\dot{A}_{\text{norm}} = \frac{\dot{A} - \min(\dot{A})}{\max(\dot{A}) - \min(\dot{A})} \tag{20}$$

Where, $(\dot{A}_{\text{norm}})$ indicates the normalized data of $(\dot{A})$, $\min(\dot{A})$ and $\max(\dot{A})$ denotes the minimum and maximum values of $(\dot{A})$, respectively.

- *Feature Extraction*

  Furthermore, the essential features like Bag of words, N-grams, character n-grams, term frequency-inverse document frequency, word count, punctuation count, All Caps Ratio (ACR), and rate of commands are extracted from $(\dot{A}_{\text{norm}})$, and the extracted features are denoted by $(\check{A})$.

- *Attack Detection*

  SSG- α SiLUGRU detects attacks in incoming commands, blocking malicious inputs and executing legitimate commands within the NgD system.

3.11. Assumptions

The proposed framework assumes English voice commands and validation using MATLAB simulations and laboratory datasets. The current study does not include multilingual commands, diverse English accents, real natural gas plant deployment, or extensive evaluation under severe industrial noise and multiple cyber-attack scenarios.

4. Results and discussions

In this section, the performance of the design and rationality of the NgD system architecture using Voice Control Instructions is demonstrated by comparing it with several conventional works. The implementation of the proposed work is conducted in the working platform of MATLAB.

4.1. Performance Analysis

In this section, the performance of the proposed work is regarding various performance metrics.

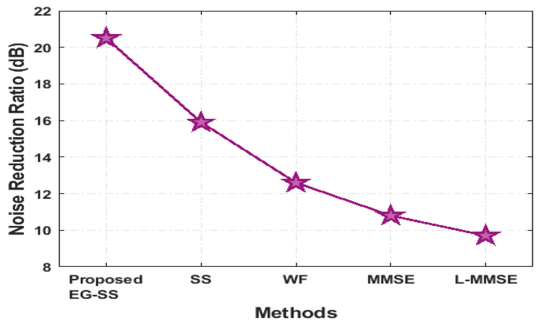
(a) Performance evaluation of noise removal

In this section, the performance of the proposed EG-SS is compared with the existing methods, such as the SS, Wiener Filter (WF), Minimum Mean-Square Error (MMSE), and Log-MMSE Estimator (L-MMSE).

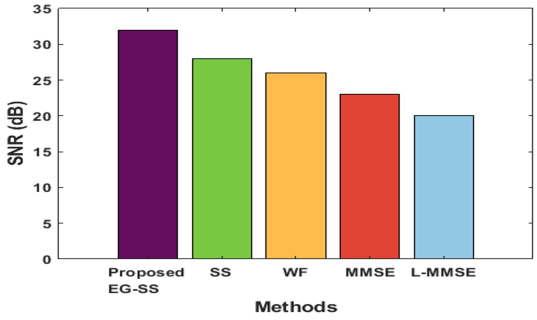
As shown in Fig. 3(a–b), EG-SS achieves 20.5 dB NRR and 32 dB SNR, outperforming existing noise-removal methods.

(b)Performance analysis for hash generation and verification

The performance of the proposed GL-SHA-3 is analyzed and compared with secure hash algorithms, including SHA-3, SHA-256, BLAKE2, and BLAKE3, under identical experimental conditions.



(a)



(b)

Fig. 3. Noise removal performance based on (a) NRR and (b) SNR.

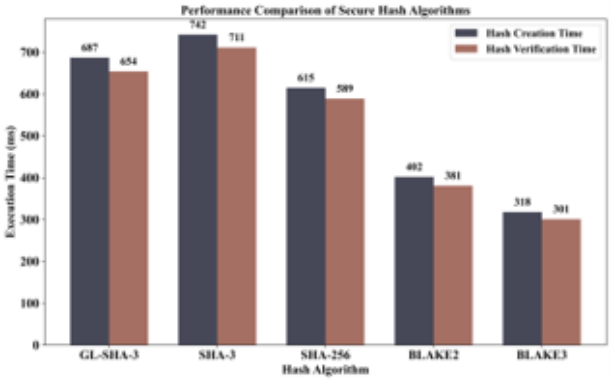


Fig. 4. Hash creation and verification time comparison.

As shown in Fig. 4, the proposed GL-SHA-3 achieves a hash creation time of 687 ms and a verification time of 654 ms, demonstrating lower execution time than SHA-3 and SHA-256, while providing a comparable performance with BLAKE2 and BLAKE3.

(c) Performance validation for Command Operational limit check

The proposed C-FSM is compared with FSM, Extended Finite State Machine (EFSM), Mealy Machine (MM), and Probabilistic Finite Automaton (PFM) in Table 4.

As shown in Table 4, C-FSM achieves 98.46% state-

Table 4. Test vectors for the proposed GL-SHA-3.

Methods	State Transition Accuracy (%)	False Rejection Ratio (%)	Limit Violation Detection Rate (%)
Proposed CFSM	98.46	0.64	96.7
FSM	95.61	0.96	93.4
EFSM	93.45	1.68	90.56
MM	90.678	2.67	89.45
PFM	88.612	3.96	87.98

transition accuracy, 96.7% violation detection, and 0.64% false rejection.

(d) Performance analysis for anomaly classification

The proposed SSG- α SiLUGRU is validated against conventional SGRU, GRU, RNN, and LSTM models for attack-detection performance.

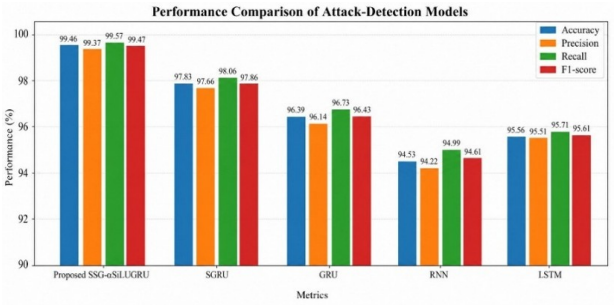


Fig. 5. Performance comparison of attack-detection models.

As shown in Table 5 and Fig. 5, and Figure 6 SSG- α SiLUGRU achieves 99.464% accuracy, 99.368% precision, 99.571% recall, and 99.469% F1-score, with consistent attack-detection performance.

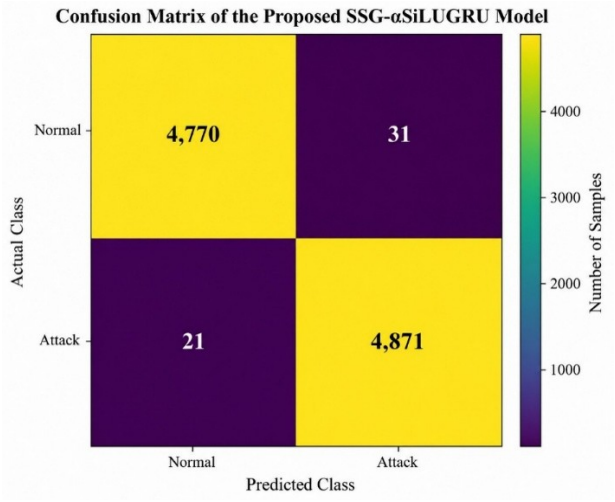


Fig. 6. Confusion matrix of the proposed SSG- α SiLUGRU model for attack detection.

(e) Handling of Unrecognized or Out-of-Vocabulary Commands

Commands outside the predefined SCADA lexicon are marked "Command Not Understood," uniquely logged, and blocked from verification and execution.

4.2. Comparative analysis

The system-level comparison provides qualitative context using representative voice-spoofing studies, acknowledging differences in datasets, attack definitions, devices, and experimental protocols that limit direct accuracy comparisons. Table 6 compares existing works with the proposed framework for voice spoofing detection and command traceability in NgD systems.

Table 6 compares methodological differences with representative studies. Accuracy values are contextual due to differing datasets and protocols, while emphasis is placed on the proposed framework's unique-ID-based end-to-end command traceability.

5. Conclusion

This paper presents a natural gas dispatching architecture integrating voice control with unique identifiers for end-to-end command traceability. The proposed approach addresses failure localization in voice-based industrial control by embedding unique IDs throughout the processing pipeline, covering command enrollment, hash generation, authentication, and execution. The key contributions of this work are summarized as follows:

- EG-SS: Reduces industrial acoustic noise, achieving 20.5 dB NRR and 32 dB SNR.
- C-FSM: Performs operational limit checking with 98.46% STA, 96.7% LVDR, and 0.64% FRR.
- GL-SHA-3: Verifies command integrity with 687 ms minimum hash creation and 654 ms verification times.
- SSG- α SiLUGRU achieves 98.945% accuracy, 98.641% precision, 98.6477% recall, 98.316% F-measure, 98.654% TPR, 98.649% TNR, 0.00654 FPR , and 0.0065 FNR.

Table 5. Performance analysis of the proposed and prevailing methods.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	TPR (%)	TNR (%)	FPR (%)	FNR (%)
SSGα SiLUGRU	**99.464**	**99.368**	**99.571**	**99.469**	**99.571**	**99.354**	**0.646**	**0.429**
SGRU	97.833	97.659	98.058	97.858	98.058	97.604	2.396	1.942
GRU	96.389	96.141	96.730	96.434	96.730	96.042	3.958	3.270
RNN	94.532	94.223	94.993	94.606	94.993	94.062	5.938	5.007
LSTM	95.564	95.513	95.708	95.610	95.708	95.417	4.583	4.292

Table 6. Comparative analysis of the proposed and prevailing works.

Work	Objective	Technique	Accuracy
[14]	Voice spoofing attack detection	Adversarial speaker regularization with joint loss	93.57%
[15]	Voice-system attack detection	DNN	92%
Proposed	Secure NgD voice control with unique command IDs and traceability	SSG- α SiLUGRU	98.945%

Experimental results show that the proposed framework improves noise suppression, command integrity, operational validation, and intent recognition, enabling secure, reliable, and traceable voice-command execution in natural gas dispatching systems. Future work will focus on four directions: developing a real-time scheduler to prioritize concurrent commands by urgency; designing distributed unique-ID generation to prevent collisions; introducing semantic normalization or semantic-preserving hashing to address ASR variations; and extending the Consensus Finite State Machine with adaptive guard conditions using short-horizon prediction models such as ARIMA.

Additionally, lightweight model-compression techniques will be investigated to enable efficient edge deployment in resource-constrained industrial environments while maintaining reliable, secure, and responsive natural gas dispatching operations. The framework was evaluated using English commands, MATLAB simulations, and laboratory datasets. Future work will address multilingual and accent-diverse speech, real natural gas environments, mechanical and industrial noise, dynamic conditions, and diverse cyber-attack scenarios for improved robustness and applicability.

Future work

In the future, a real-time prioritization scheduler will be incorporated into the proposed framework to classify and queue voice commands based on operational urgency, ensuring that low-importance instructions do not delay emergency actions.

Declarations

Funding

Authors did not receive any funding.

Conflicts of interests

Authors do not have any conflicts.

Data Availability Statement

The data generated and analyzed during the current study are available from the author Yi Yang upon reasonable request but are not yet publicly available due to ongoing research.

Code availability

Not applicable.

Authors' Contributions

Hongtao Diao, Yu Li, Zengrui wang is responsible for designing the framework, analyzing the performance, validating the results, and writing the article. Youyi Liang, Yi Yang is responsible for collecting the information required for the framework, provision of software, critical review, and administering the process.

References

[1] M. Norda, C. Engel, J. Rennies, J. E. Appell, S. C. Lange, and A. Hahn, (2023) *"Evaluating the Efficiency of Voice Control as Human Machine Interface in Production"* **IEEE Transactions on Automation Science and Engineering *21***(3): 4817–4828. DOI: 10.1109/TASE.2023.3302951.

[2] H. Kwon, D. Park, and O. Jo, (2024) *"Silent-Hidden-Voice Attack on Speech Recognition System"* **IEEE Access *12***: 173010–173019. DOI: 10.1109/ACCESS.2022.3181194.

[3] X. Ji, G. Zhang, X. Li, G. Qu, X. Cheng, and W. Xu, (2024) *"Detecting Inaudible Voice Commands via Acoustic Attenuation by Multi-Channel Microphones"* **IEEE Transactions on Dependable and Secure Computing**: DOI: 10.1109/TDSC.2024.3355117.

[4] P. Cheng and U. Roedig, (2022) *"Personal Voice Assistant Security and Privacy—A Survey"* **Proceedings of the IEEE *110***(4): 476–507. DOI: 10.1109/JPROC.2022.3153167.

[5] J. Guan, L. Pan, C. Wang, S. Yu, L. Gao, and X. Zheng, (2023) *"Trustworthy Sensor Fusion Against Inaudible Command Attacks in Advanced Driver-Assistance Systems"* **IEEE Internet of Things Journal *10***(19): 17254–17264. DOI: 10.1109/JIOT.2023.3275931.

[6] Z. Sun, J. Zhao, F. Guo, Y. Chen, and L. Ju, (2024) *"CommanderUAP: A Practical and Transferable Universal Adversarial Attacks on Speech Recognition Models"* **Cybersecurity 7**(1): 38. DOI: 10.1186/s42400-024-00218-8.

[7] H. Cao, H. Jiang, D. Liu, R. Wang, G. Min, J. Liu, and J. C. Lui, (2022) *"LiveProbe: Exploring Continuous Voice Liveness Detection via Phonemic Energy Response Patterns"* **IEEE Internet of Things Journal *10***(8): 7215–7228. DOI: 10.1109/JIOT.2022.3228819.

[8] S. Wang, Z. Zhang, G. Zhu, X. Zhang, Y. Zhou, and J. Huang, (2022) *"Query-Efficient Adversarial Attack with Low Perturbation Against End-to-End Speech Recognition Systems"* **IEEE Transactions on Information Forensics and Security *18***: 351–364. DOI: 10.1109/TIFS.2022.3222963.

[9] A. R. Abbasi, (2025) *"Statistical Techniques in Power Systems Fault Diagnostic: Classifications, Challenges, and Strategic Recommendations"* **Electric Power Systems Research *239***: 111279. DOI: 10.1016/j.epsr.2024.111279.

[10] A. R. Abbasi and C. Parkash, (2025) *"Innovative Diagnosis of Transformer Winding Defects Using Fuzzy and Neutrosophic Cross Entropy Measures"* **Advanced Engineering Informatics *65***: 103196. DOI: 10.1016/j.aei.2025.103196.

[11] M. Zadehbagheri and A. R. Abbasi, (2023) *"Energy Cost Optimization in Distribution Network Considering Hybrid Electric Vehicle and Photovoltaic Using Modified Whale Optimization Algorithm"* **The Journal of Supercomputing *79***(13): 14427–14456. DOI: 10.1007/s11227-023-05214-2.

[12] J. Ansari, M. Homayounzade, and A. R. Abbasi, (2025) *"Innovative Load Frequency Control: Integrating Adaptive Backstepping and Disturbance Observers"* **IEEE Access *13***: 53673–53693. DOI: 10.1109/ACCESS.2025.3554141.

[13] H. K. Risan, F. M. Serhan, and A. A. Al-Azzawi, (2024) *"Management of a Typical Experiment in Engineering and Science"* **AIP Conference Proceedings *2864***(1): 050001. DOI: 10.1063/5.0186079.

[14] A. Kassis and U. Hengartner, (2021) *"Practical Attacks on Voice Spoofing Countermeasures"* **arXiv**: DOI: 10.48550/arXiv.2107.14642. eprint: 2107.14642.

[15] Y. Wang, H. Guo, and Q. Yan, (2022) *"GhostTalk: Interactive Attack on Smartphone Voice System Through Power Line"* **arXiv**: DOI: 10.14722/ndss.2022.24254. eprint: 2202.02585.