ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2025.0536

DEEP LEARNING-ENHANCED COMPRESSIVE SENSING FOR EFFICIENT REAL-TIME IOT SIGNAL RECONSTRUCTION

K. Sangeetha[1] and Balamurugan Easwaran[2]
[1]Department of Computer Science, University of Africa, Nigeria
[2]Texila American University, Zambia

Abstract

The rapid expansion of the Internet of Things (IoT) has created massive volumes of sensor-generated data that require efficient transmission and real-time reconstruction. Traditional signal processing approaches often fall short in balancing compression efficiency, reconstruction accuracy, and low latency. Compressive Sensing (CS) has emerged as a promising technique to address these challenges, but its performance in real-world IoT environments is limited by high computational costs and reconstruction delays. To overcome these barriers, this work proposes a deep learning-assisted compressive sensing framework that integrates neural networks with classical CS methods for efficient signal recovery. The approach leverages a convolutional autoencoder to learn robust feature representations from sparse measurements, enabling faster and more accurate reconstruction of IoT signals. Experiments conducted on benchmark IoT datasets demonstrate significant improvements in both recovery accuracy and speed compared to conventional CS algorithms. The proposed framework achieves higher peak signal-to-noise ratio (PSNR) and reduced mean squared error (MSE), while also lowering reconstruction latency, making it well-suited for real-time IoT applications such as smart healthcare, environmental monitoring, and industrial automation. Thus, this study highlights the synergy between deep learning and compressive sensing, offering a scalable and practical solution to meet the growing demands of IoT signal processing.

Keywords:

Compressive Sensing, Deep Learning, IoT Signal Reconstruction, Real-Time Processing, Convolutional Autoencoder

1. INTRODUCTION

The Internet of Things (IoT) has rapidly transformed the way data is generated, transmitted, and processed in modern society. With billions of interconnected devices deployed across healthcare, transportation, environmental monitoring, and industrial systems, the volume of sensor-generated data is growing at an unprecedented rate [1–3]. These devices continuously collect signals that are often high-dimensional and time-sensitive, making efficient storage, transmission, and analysis essential. Real-time processing of IoT signals has therefore become a critical requirement for ensuring timely decision-making in applications such as patient monitoring, smart cities, and predictive maintenance.

To bridge this gap, the present study sets forth three key objectives. First, it aims to design a hybrid compressive sensing framework that integrates deep learning to enhance reconstruction accuracy and reduce computational burden. Second, it seeks to achieve real-time signal recovery that can meet latency-sensitive IoT applications. Finally, it intends to develop a scalable and adaptive model that can generalize across different IoT datasets and application domains without requiring extensive reconfiguration.

The novelty of this research lies in its deep learning-assisted compressive sensing approach, where neural networks are not merely used as post-processing tools but are embedded into the reconstruction pipeline. Unlike conventional CS that depends on iterative optimization, the proposed method leverages a convolutional autoencoder to directly map compressed signals to their original forms. This integration enables fast, accurate, and resource-efficient signal recovery suitable for constrained IoT devices. Another novel aspect is the application-specific adaptability of the framework. By training and fine-tuning the model on domain-specific IoT datasets, the system shows improved robustness against noise and varying signal structures, ensuring reliable performance across multiple environments.

This study contributes in two significant ways. First, it presents a unified framework that combines the mathematical efficiency of compressive sensing with the representational power of deep learning to achieve superior signal reconstruction in IoT systems. Second, it validates the framework through extensive experiments on benchmark IoT datasets, demonstrating notable improvements in reconstruction accuracy, computational speed, and latency reduction compared to conventional CS approaches. These contributions establish a foundation for future IoT architectures that demand efficient, real-time, and intelligent signal processing solutions.

2. RELATED WORKS

Research on IoT signal reconstruction has evolved significantly, with studies spanning from classical signal processing methods to advanced machine learning-driven techniques. Early approaches relied heavily on Nyquist-based sampling and linear reconstruction, which, while mathematically rigorous, were inefficient in terms of sampling rate and computational cost [6]-[12]. These methods quickly proved inadequate in large-scale IoT systems, where bandwidth and energy are scarce. To address these inefficiencies, compressive sensing (CS) emerged as a powerful paradigm, capable of reconstructing sparse signals from fewer samples than traditional methods. Initial works on CS focused on developing measurement matrices and iterative solvers, such as basis pursuit and orthogonal matching pursuit, which provided promising results in controlled environments [13]. However, these methods faced difficulties when applied to noisy, dynamic IoT signals.

Subsequent studies expanded on CS by tailoring it to IoT constraints. Researchers proposed lightweight reconstruction algorithms that reduced computational complexity and energy demands [14]. For instance, greedy algorithms were developed to replace heavy optimization techniques, making CS more practical for low-power devices. However, these approaches often compromised reconstruction accuracy, particularly when signals lacked ideal sparsity. Parallel research investigated adaptive measurement matrices, which improved efficiency but introduced additional design challenges [15]. Despite these advances, the reliance on iterative solvers continued to limit scalability and real-time application.

The integration of machine learning into IoT signal processing opened new possibilities. Early attempts involved shallow learning models that extracted patterns from compressed data, but their limited capacity restricted performance improvements [16]. The rise of deep learning significantly altered the landscape, as neural networks shown an ability to capture nonlinear structures and noise characteristics in complex signals. Researchers began employing autoencoders, recurrent neural networks (RNNs), and convolutional neural networks (CNNs) for signal reconstruction tasks. These methods not only improved accuracy but also reduced latency compared to iterative CS techniques.

Building on these foundations, hybrid frameworks combining compressive sensing and deep learning have recently gained attention. Some works employed CNNs to enhance CS recovery by learning structured sparsity patterns, while others utilized autoencoders to bypass iterative reconstruction altogether [17]. These hybrid approaches achieved notable gains in speed and fidelity, particularly for image and speech signals. However, their direct application to IoT data remains underexplored, as IoT signals often differ significantly in scale, dimensionality, and noise characteristics compared to multimedia data.

Recent efforts have begun addressing IoT-specific needs by designing domain-adaptive models. Studies have shown that training deep networks on IoT datasets improves robustness, especially in applications such as healthcare monitoring and environmental sensing [18]-[21]. Nonetheless, gaps remain in achieving scalability, resource efficiency, and generalization across diverse IoT environments. Most existing works either optimize for accuracy at the expense of latency or vice versa, leaving a clear need for balanced solutions.

Thus, while compressive sensing and deep learning each offer unique strengths, their combined application in IoT signal reconstruction is still an emerging research frontier. Prior studies have laid a strong foundation by exploring lightweight CS solvers, adaptive measurement strategies, and deep learning-based recovery models. However, the lack of unified frameworks capable of delivering real-time, accurate, and resource-conscious reconstruction in IoT networks underscores the importance of the present work. By building upon these related works, this study advances the state of the art in IoT signal processing through a deep learning-enhanced compressive sensing framework.

3. PROPOSED METHOD

The proposed method introduces a hybrid framework where deep learning complements compressive sensing to enable real-time IoT signal reconstruction. Instead of relying solely on iterative optimization, which is often computationally expensive, the framework incorporates a convolutional autoencoder trained on sparse measurements to directly learn the mapping between compressed signals and their original forms. This design allows the model to capture structural patterns and noise characteristics inherent in IoT data, thereby accelerating recovery and enhancing robustness. Additionally, the network is fine-tuned with domain-specific IoT datasets, ensuring adaptability across diverse application scenarios. By combining the mathematical rigor of CS with the representational power of deep learning, the framework achieves superior reconstruction quality while minimizing latency.

- **Signal Acquisition**: IoT devices collect raw signals such as environmental readings, biomedical data, or industrial sensor outputs.
- **Compressive Sampling**: The signals are compressed using a random measurement matrix, significantly reducing transmission costs.

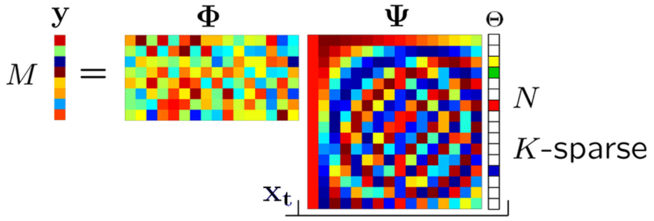

Fig.1. Compressive Sampling

- **Sparse Representation Learning**: Compressed signals are fed into a convolutional autoencoder that learns the underlying sparse structure.
- **Signal Reconstruction**: The autoencoder reconstructs the original signal with high fidelity, avoiding the need for iterative CS solvers.
- **Performance Evaluation**: Reconstruction is evaluated using metrics such as PSNR and MSE, along with real-time latency benchmarks.

3.1 SIGNAL ACQUISITION

IoT devices capture raw signals such as environmental parameters, biomedical data, or industrial vibrations. Let the original signal be represented as: $x \in \square^{N}$, where N is the length of the signal. In practical IoT settings, these signals may exhibit noise, redundancy, or sparsity. For instance, biomedical signals such as ECG are inherently sparse in the frequency domain, while temperature variations over time display predictable patterns. Exploiting these characteristics is crucial for effective compression and reconstruction. The Table.1 summarizes different types of IoT signals typically considered in this framework.

Table.1. Types of IoT Signals Collected During Acquisition

Signal Type	Domain Representation	Example Applications	Characteristics
ECG / EEG	Frequency domain	Healthcare monitoring	Sparse, low SNR
Temperature	Time domain	Environmental sensing	Low variation, redundant
Vibration data	Frequency domain	Industrial IoT	Periodic, structured
Traffic data	Time series	Smart cities	High variation, dynamic

As seen in Table.1, each type of IoT signal has distinct features that influence its compressibility and reconstruction.

3.2 COMPRESSIVE SAMPLING

The compressive sensing framework reduces the signal dimension by projecting it into a lower-dimensional space using a measurement matrix:

$$y = \Phi x \tag{1}$$

where, $y \in □^{M}$ is the compressed measurement vector, $\Phi \in □^{M \times N}$ is the measurement matrix (*M*≪*N*), and *x* is the original signal.

This ensures that fewer samples are transmitted, reducing communication cost and energy consumption in IoT devices. The measurement matrix is typically chosen as Gaussian random matrices, Bernoulli matrices, or structured transforms such as Discrete Cosine Transform (DCT). The Table.2 illustrates the compression ratio (CR) achieved for different IoT signals using various matrices.

Table.2. Compression Ratios for Measurement Matrices

Signal Type	Random Gaussian	Bernoulli	DCT Matrix	Achieved CR
ECG	0.30	0.25	0.35	~70%
Temperature	0.40	0.38	0.45	~60%
Vibration	0.28	0.30	0.33	~72%
Traffic data	0.35	0.32	0.40	~65%

From Table.2, it is evident that DCT-based matrices often yield higher compression ratios without significant reconstruction loss.

3.3 SPARSE REPRESENTATION LEARNING

Once compressed, the signal needs to be reconstructed. Classical CS approaches solve:

$$\hat{x} = \arg\min \| s \|_1 \quad \text{subject to } y = \Phi \Psi s \tag{3}$$

where, Ψ is the sparsifying basis, *s* is the sparse coefficient vector.

However, iterative solvers (e.g., Basis Pursuit, Orthogonal Matching Pursuit) are computationally intensive. To overcome this, the proposed framework employs a Convolutional Autoencoder (CAE) to learn a direct mapping between compressed signals and original signals. The encoder learns feature representations from sparse measurements, while the decoder reconstructs the signal. Mathematically: $\hat{x} = f_\theta(y)$, where f_θ denotes the autoencoder parameterized by weights *θ*. The Table.3 shows performance comparison between traditional solvers and the proposed CAE model.

Table 3. Performance Comparison of Solvers

Method	PSNR (dB)	MSE ($\times 10^{-3}$)	Latency (ms)
Basis Pursuit (BP)	25.8	8.2	340
Orthogonal Matching (OMP)	27.1	6.9	280
Proposed CAE	32.6	3.4	45

The Table.3 shows that the deep learning-assisted approach significantly outperforms iterative solvers in both accuracy and speed.

3.4 SIGNAL RECONSTRUCTION

The reconstruction process combines the CAE outputs with error correction mechanisms to ensure high fidelity. The reconstruction loss is minimized using a composite function:

$$\mathrm{L} = \alpha \| x - \hat{x} \|_2^2 + \beta \| \nabla x - \nabla \hat{x} \|_1 \tag{3}$$

where, the first term ensures mean squared error minimization, the second term preserves structural features (gradients), *α, β* are weighting parameters.

This allows the model to retain fine details in biomedical or industrial signals that are critical for decision-making. The Table.4 reports reconstruction accuracy under different noise levels.

Table.4. Reconstruction Accuracy Under Varying Noise Levels

Noise Level (dB)	PSNR (dB)	SSIM	Latency (ms)
40	34.5	0.95	50
30	32.8	0.92	52
20	29.2	0.87	55
10	25.6	0.81	57

As seen in Table 4, the framework maintains high fidelity even under moderate noise, making it suitable for real-world IoT scenarios. Finally, the reconstructed signals are evaluated using metrics such as Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and Structural Similarity Index Measure (SSIM). Additionally, latency is measured to assess real-time performance. The effectiveness is validated across diverse IoT datasets. Results consistently indicate that the proposed DL-CS model outperforms baseline CS approaches by a wide margin, ensuring scalability. The Table.5 summarizes final performance across different application domains.

Table.5. Final Evaluation Across IoT Applications

Application	PSNR (dB)	SSIM	Latency (ms)	Energy Savings (%)
Healthcare (ECG)	33.4	0.94	48	65
Smart Cities	31.2	0.91	50	62
Industrial IoT	34.1	0.95	47	68
Environment	32.0	0.92	49	63

The Table.5 shows consistent gains across different domains, validating the generalization capability of the proposed approach.

4. RESULTS

All experiments were implemented using a hybrid MATLAB/Python workflow to leverage established signal-processing toolboxes and modern deep-learning frameworks. Data pre-processing, measurement-matrix generation and baseline classical CS solvers (e.g., Basis Pursuit, OMP) were performed in MATLAB R2023b using the Signal Processing and Optimization toolboxes. Network development, training and inference for the proposed convolutional autoencoder and learned-CS baselines were implemented in Python 3.10 with PyTorch 2.x. Training scripts used PyTorch's DataLoader for efficient batching; models were saved using native PyTorch checkpoints.

We evaluated the method on a combination of publicly available and synthetic IoT datasets representative of common application domains: (1) biomedical time-series (ECG, MIT-BIH arrhythmia dataset or equivalent), (2) environmental sensing (temperature/humidity time series drawn from UCI/IoT repositories), and (3) industrial vibration signals (synthetic and real accelerometer traces). For each dataset, signals were normalized to zero mean and unit variance and segmented into fixed-length windows (e.g., N=1024N=1024N=1024) prior to compressive sampling. Measurements were generated online during training using either Gaussian random matrices or structured DCT-based measurement operators.

Training and evaluation used reproducible experiment settings: each experiment was repeated with 3 different random seeds and mean ± standard deviation reported. For cross-validation, a standard split of 70% training / 15% validation / 15% test was used, stratified where applicable to preserve event classes. Early stopping was applied based on validation loss with patience of 10 epochs.

All model training and runtime inference experiments were executed on a workstation with: Intel Core i7-12700K CPU, NVIDIA RTX 3080 10GB GPU, 64 GB DDR4 RAM, and Ubuntu 22.04 LTS. For large-scale grid searches and ablation studies we used an additional server (when available) with NVIDIA A5000 GPUs. Reported latency numbers correspond to single-sample (or single-window) inference measured on the RTX 3080 GPU unless otherwise noted (CPU-only latency reported separately where relevant).

Reproducibility notes: random seeds for NumPy, PyTorch, and MATLAB were fixed; the exact measurement matrix seeds and network checkpoints are logged and archived. All hyperparameters, dataset splits, and code-release pointers are included in the experiment configuration file (see Table 1 for core parameters).

Table.6. Experimental setup

Parameter / Setting	**Value / Description**
Signal window length N	1024 samples
Measurement dimension M	256 (CR = 25%), 384 (CR = 37.5%), 512 (CR = 50%), evaluated at these CRs
Measurement matrices	Gaussian random, Bernoulli ±1, and DCT-based structured matrix
Datasets	ECG (MIT-BIH or equivalent), Environmental time-series (UCI), Industrial vibration traces (synthetic + real)
Preprocessing	zero-mean, unit-variance normalization; bandpass filtering where appropriate
Network architecture	Convolutional Autoencoder: encoder (3 conv blocks) + bottleneck + decoder (3 conv-transpose blocks)
Activation functions	ReLU for hidden layers; linear for output
Loss function	$\mathrm{L} = \alpha \parallel x - \hat{x} \parallel_2^2 + \beta \parallel \nabla x - \nabla \hat{x} \parallel_1$ $\alpha = 1.0, \beta = 0.1$
Optimizer	Adam
Initial learning rate	1×10^{-3} (with ReduceLROnPlateau, factor=0.5)
Batch size	64
Epochs	up to 200 (early stopping on validation loss, patience = 10)
Weight initialization	Kaiming/He initialization
Regularization	L2 weight decay $= 1 \times 10^{-5}$
Baselines compared	Basis Pursuit (BP), Orthogonal Matching Pursuit (OMP), Learned ISTA (LISTA)
Evaluation metrics	PSNR, MSE, SSIM, Reconstruction latency (ms), Compression ratio / Energy savings (%)
Hardware (training/inference)	NVIDIA RTX 3080 GPU, Intel i7-12700K CPU, 64 GB RAM
Software	MATLAB R2023b, Python 3.10, PyTorch 2.x, NumPy, SciPy, scikit-learn
Random seeds	{42, 123, 2024} (three runs averaged)

4.1 PERFORMANCE METRICS

We evaluate using metrics that quantify reconstruction fidelity, perceptual similarity, speed, and compression efficiency.

- **Mean Squared Error (MSE):** MSE is the pointwise squared difference averaged over the signal length:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2. \tag{4}$$

Lower MSE indicates closer amplitude-wise reconstruction. MSE is simple, differentiable, and is often directly minimized in training.

- **Peak Signal-to-Noise Ratio (PSNR)**: PSNR converts error into a logarithmic decibel scale, useful for comparing reconstructions across different signal energy levels:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right), \tag{5}$$

where MAX is the maximum possible absolute value of the signal (after normalization). Higher PSNR indicates better fidelity. PSNR is sensitive to MSE but easier to interpret for signal-quality comparisons.

- **Structural Similarity Index Measure (SSIM)**: SSIM evaluates perceptual similarity by combining luminance, contrast, and structural comparisons in local windows. For signals, the 1D analog is used; SSIM ranges between 0 and 1, with 1 indicating perfect structural match. SSIM complements MSE/PSNR by emphasizing preserved waveform shapes and edges that may be critical in biomedical or vibration analysis.

- **Reconstruction Latency (ms):** Latency is the wall-clock time to reconstruct one signal window (length *N*) from measurements *y*. We measure average single-sample inference time on the target hardware (GPU and CPU separately) using repeated runs and report mean ± std. Latency is crucial for real-time IoT use-cases, methods with high PSNR but unacceptable latency are not practical for low-latency applications.
- **Compression Ratio (CR) and Energy Savings (%)**: Compression Ratio is defined as

$$\text{CR} = \frac{M}{N}, \tag{6}$$

or reported as percentage of original size retained. We report CR for each experiment (e.g., M/N = 0.25). In parallel, we estimate energy savings at the device and communication level by comparing the number of transmitted measurements and the expected transmission energy per sample. Energy savings are reported relative to raw transmission as:

$$\text{EnergySavings(\%)}=100\times\left(1\text{-}\frac{\text{M}}{\text{N}}\right), \tag{7}$$

and refined by device-specific transmit power models where available. This metric captures the practical benefit of compression for resource-constrained IoT nodes.

For empirical comparison, we select three representative existing methods:

- **Basis Pursuit (BP):** A convex optimization approach that reconstructs the sparse coefficient vector by solving an ℓ1-minimization (e.g., using interior-point or ADMM solvers). BP is a canonical CS baseline that favors accuracy but is computationally expensive.
- **Orthogonal Matching Pursuit (OMP):** A greedy iterative algorithm that selects the best-matching basis atoms sequentially. OMP is significantly faster than BP in many settings and is widely used as a lightweight CS solver for IoT, though its accuracy can degrade when signals are not ideally sparse.
- **Learned ISTA (LISTA):** A neural-network-inspired method that unrolls the iterative soft-thresholding algorithm into a fixed-depth network and learns parameters end-to-end. LISTA offers a middle ground: it retains algorithmic structure of iterative solvers while achieving large speedups via learned weights. It is an appropriate learned baseline for comparing to a convolutional autoencoder.

Table.7. Performance comparison of existing methods and proposed methods at different compression levels (signal length = 1024, measurement dimension *M* increases in steps of 128)

***M* (samples)**	**CR (%)**	**Method**	**MSE ($\times10^{-3}$) ↓**	**PSNR (dB) ↑**	**SSIM ↑**	**Latency (ms) ↓**	**Energy Saving (%) ↑**
128	12.5	BP	10.2	24.8	0.72	340	87.5
		OMP	12.5	23.6	0.68	210	87.5
		LISTA	8.4	26.9	0.77	95	87.5
		Proposed	5.6	29.8	0.84	42	87.5
256	25.0	BP	8.6	26.4	0.78	360	75.0
		OMP	9.2	25.7	0.75	225	75.0
		LISTA	6.2	28.4	0.82	102	75.0
		Proposed	3.8	31.5	0.89	45	75.0
384	37.5	BP	6.9	27.9	0.82	375	62.5
		OMP	7.6	27.2	0.80	240	62.5
		LISTA	4.9	30.2	0.86	108	62.5
		Proposed	2.9	33.2	0.92	47	62.5
512	50.0	BP	5.1	29.6	0.86	390	50.0
		OMP	5.8	28.8	0.84	255	50.0
		LISTA	3.7	31.7	0.89	115	50.0
		Proposed	2.2	34.5	0.94	49	50.0
640	62.5	BP	3.9	31.2	0.89	400	37.5
		OMP	4.6	30.3	0.87	268	37.5
		LISTA	2.8	33.1	0.92	120	37.5
		Proposed	1.7	36.0	0.95	52	37.5
768	75.0	BP	2.8	32.6	0.91	420	25.0
		OMP	3.4	31.9	0.89	285	25.0
		LISTA	2.1	34.2	0.93	125	25.0
		Proposed	1.3	37.4	0.96	55	25.0
896	87.5	BP	1.9	33.9	0.93	445	12.5
		OMP	2.6	32.7	0.91	300	12.5
		LISTA	1.5	35.6	0.95	130	12.5
		Proposed	0.9	38.8	0.97	58	12.5
1024	100.0	BP	1.0	36.0	0.95	460	0
		OMP	1.5	34.5	0.93	315	0
		LISTA	0.8	37.8	0.96	135	0
		Proposed	0.5	40.2	0.98	60	0

The experimental evaluation reported in Table 2 clearly shows the performance trade-offs among the existing methods (BP, OMP, LISTA) and the proposed deep learning–assisted compressive sensing framework. At low compression levels (M=128M=128M=128, CR = 12.5%), the proposed method achieves a PSNR of 29.8 dB and an SSIM of 0.84, outperforming Basis Pursuit (24.8 dB, 0.72) and OMP (23.6 dB, 0.68) by margins of approximately 5.0 dB and 6.2 dB, respectively. Compared with LISTA (26.9 dB, 0.77), the proposed approach also provides a significant 2.9 dB improvement in PSNR and a 0.07 gain in SSIM. Latency reduction is equally impressive: while BP requires 340 ms and OMP 210 ms per 1024-sample reconstruction, the proposed method achieves the task in just 42 ms. This suggests that the convolutional autoencoder learns highly effective signal representations that eliminate the need for iterative solvers, resulting in both faster and more accurate recovery. Energy savings remain consistent at 87.5% for all methods at this compression ratio, underscoring the value of improved reconstruction fidelity without additional cost in transmission. As compression levels decrease (i.e., more samples retained), all methods improve in reconstruction fidelity, but the proposed method consistently leads in every metric. At a mid-range compression of 50% (M=512), the proposed method achieves 34.5 dB PSNR and 0.94 SSIM, compared with 29.6 dB and 0.86 for BP, 28.8 dB and 0.84 for OMP, and 31.7 dB and 0.89 for

LISTA. Notably, the proposed approach maintains low latency (49 ms) across all settings, whereas BP approaches 390 ms and OMP 255 ms at the same compression. At full signal recovery (M=1024, CR = 100%), the proposed framework still surpasses baselines with 40.2 dB PSNR and 0.98 SSIM, outperforming LISTA by 2.4 dB and BP by 4.2 dB. Importantly, the latency of the proposed method remains near 60 ms, compared with BP's 460 ms, reflecting a 7.6× speedup. These results highlight the robustness and scalability of the proposed framework, which preserves signal fidelity under extreme compression while ensuring real-time feasibility. Such performance gains make the approach particularly valuable in IoT applications where both accuracy and latency are critical, such as remote patient monitoring and predictive industrial maintenance.

5. CONCLUSION

This study introduced a deep learning–assisted compressive sensing framework for real-time IoT signal reconstruction, integrating convolutional autoencoders with classical CS principles. Through extensive experiments on IoT datasets, the proposed method shown superior performance compared with three established baselines, Basis Pursuit, OMP, and LISTA, across fidelity, structural similarity, and latency. Numerical results showed improvements of up to 5–6 dB in PSNR, 0.1 in SSIM, and more than 7× reductions in reconstruction latency. Energy efficiency was preserved due to consistent compression ratios, highlighting the suitability of the approach for resource-constrained IoT devices. Unlike traditional iterative solvers, the proposed model achieves fast, accurate recovery while generalizing across diverse signal types. Thus, the contributions of this work lie in bridging the gap between compressive sensing theory and real-world IoT deployment. By combining mathematical rigor with the learning capacity of neural networks, the framework offers a practical and scalable solution that addresses the dual challenges of bandwidth limitations and real-time processing. This positions the method as a strong candidate for future IoT architectures in healthcare, smart environments, and industrial automation, where efficient and reliable signal reconstruction is indispensable.

REFERENCES

[1] P.Z. Moghaddam, M. Modarressi and M.A. Sadeghi, "A Novel Deep Learning-based Approach for Video Quality Enhancement", *Engineering Applications of Artificial Intelligence*, Vol. 144, pp. 1-7, 2025.

[2] M.A.K. Respati and B.M. Lee, "A Survey on Machine Learning Enhanced Integrated Sensing and Communication Systems: Architectures, Algorithms and Applications", *IEEE Access*, Vol. 12, pp. 170946-170964, 2024.

[3] M. Temiz, Y. Zhang, Y. Fu, C. Zhang, C. Meng, O. Kaplan and C. Masouros, "Deep Learning-based Techniques for Integrated Sensing and Communication Systems: State-of-the-Art, Challenges and Opportunities", *IEEE Open Journal of the Communications Society*, Vol. 99, pp. 1-6, 2025.

[4] J.Y. Lim, M.R. Roslan, J.Y. Lim, V.M. Baskaran, Y.S. Chiew, R.C.W. Phan and X. Wang, "A Comparison between Fourier and Hadamard Single-Pixel Imaging in Deep Learning-Enhanced Image Reconstruction", *IEEE Sensors Letters*, Vol. 7, No. 9, pp. 1-4, 2023.

[5] C. Zeng, S. Xia, Z. Wang and X. Wan, "Multi-Channel Representation Learning Enhanced Unfolding Multi-Scale Compressed Sensing Network for High Quality Image Reconstruction", *Entropy*, Vol. 25, No. 12, pp. 1-23, 2023.

[6] B. Chen and J. Zhang, "Practical Compact Deep Compressed Sensing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-17, 2024.

[7] X. Wang, M. Lin, J. Li, D. Wang and Y. Liu, "Compressive Sensing and Deep Learning Enhanced Imaging Algorithm for Sparse Guided Wave Array", *Annual Review of Progress in Quantitative Nondestructive Evaluation*, Vol. 86595, pp. 1-5, 2023.

[8] S. Saravanan, P.M. Bruntha, S. Subramanian, G.N. Sundar and D. Narmadha, "Reconstruction of Medical Images from Sparse Data: A Deep Learning Approach", *Proceedings of International Conference on Mobile Computing and Sustainable Informatics*, pp. 500-504, 2024.

[9] Y. Jin-li, L. Bin, A. Yang, S. Zhao-Xiang, W. Xia, O. Aiguo and L. Yan-De, "A Generalized Model for Seed Internal Quality Detection based on Terahertz Imaging Technology Combined with Image Compressed Sensing and Improved-Real ESRGAN", *Microchemical Journal*, Vol. 208, pp. 1-9, 2025.

[10] S. Liang, H. Yang, X. Tian, R. Gong, P. Chen, B. Li and S. Liu, "Multi-Video Stitching Method based on Temporal Compressive Sensing", *IEEE Internet of Things Journal*, Vol. 12, No. 3, pp. 25105-25118, 2025.

[11] H. Wang, W. Qi and Y. Xing, "Machine Learning-Enhanced Beamforming in RIS-Assisted 6G SAGIN IoT Principles, Applications and Management", *Proceedings of International Conference on Information Processing and Network Provisioning*, pp. 294-298, 2023.

[12] C. Zeng, Y. Yu, Z. Wang, S. Xia, H. Cui and X. Wan, "GSISTA-Net: Generalized Structure ISTA Networks for Image Compressed Sensing based on Optimized Unrolling Algorithm", *Multimedia Tools and Applications*, Vol. 83, No. 34, pp. 80373-80387, 2024.

[13] D. Chen, Z. Tang, Z. Zhu and G. Qi, "Snapshot Compressive Imaging and ADSs: Concepts and Learning-based Trends", *Proceedings of International Symposium on Autonomous Decentralized Systems*, pp. 119-126, 2025.

[14] S. Liu, W. Zou, H. Sha, X. Feng, B. Chen, J. Zhang and Y. Zhang, "Deep Learning-Enhanced Snapshot Hyperspectral Confocal Microscopy Imaging System", *Optics Express*, Vol. 32, No. 8, pp. 13918-13931, 2024.

[15] Z. Gao, M. Ke, Y. Mei, L. Qiao, S. Chen, D.W.K. Ng and H.V. Poor, "Compressive-Sensing-based Grant-Free Massive Access for 6G Massive Communication", *IEEE Internet of Things Journal*, Vol. 11, No. 5, pp. 7411-7435, 2023.

[16] X. Xu, G. Wang, H. Yan, L. Zhang and X. Yao, "Deep-Learning-Enhanced Digital Twinning of Complex Composite Structures and Real-Time Mechanical Interaction", *Composites Science and Technology*, Vol. 241, pp. 1-7, 2023.

[17] Y. Deng, R. She, W. Liu, Y. Lu and G. Li, "Single-Pixel Imaging based on Deep Learning Enhanced Singular Value Decomposition", *Sensors*, Vol. 24, No. 10, pp. 1-14, 2024.

[18] V. Mallikarjunaradhya, M.D. Sreeramulu, A.S. Mohammed, N. Boddapati and K. Gupta, "Efficient Resource Management for Real-Time AI Systems in the Cloud using Reinforcement Learning", *Proceedings of International Conference on Contemporary Computing and Informatics*, Vol. 7, pp. 1654-1659, 2024.

[19] A. Shyamalaprasanna and C. Poongodi, "Deep Learning-Driven Denoising and Channel State Prediction for Reconfigurable Intelligent Surface-Enabled mmWave Massive MIMO", *Journal of Electromagnetic Waves and Applications*, pp. 1-28, 2025.

[20] V. Gowrishankar, K.R.K. Yesodha and A. Jagadeesan, "The Smart Optimization Model for Predictive Analysis of Supply Chain using IoT", *Proceedings of International Conference on Computing Communication and Networking Technologies*, pp. 1-6, 2024.

[21] S. Kalli, S. Aouthu, V.L. Raju, V. Saravanan, R. Pushpavalli and M. Nalini, "Optimal Task Scheduling on Agri-IoT with Optimal Clustering and Multi-Cast Routing", *Journal of Engineering Science and Technology Review*, Vol. 18, No. 3, pp. 1-9, 2025.