International Journal of Communication Systems

WILEY

RESEARCH ARTICLE

Dynamic Autoencoder-Based Framework for Performance Enhancement in OFDM Systems Using CNN and Attention Mechanisms

Rajarajan P[1] | Madona B. Sahaai[2]

[1]Department of Electronics and Communication Engineering, Mohamed Sathak A.J. College of Engineering, Chennai, Tamil Nadu, India | [2]Department of Electronics and Communication Engineering, The Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, Tamil Nadu, India

Correspondence: Rajarajan P (dynamiterajan@gmail.com)

Received: 25 June 2025 | **Revised:** 24 October 2025 | **Accepted:** 29 December 2025

Keywords: adaptive transmission | attention mechanisms | channel estimation | convolutional neural networks | dynamic autoencoder | feature extraction | orthogonal frequency division multiplexing | signal-to-noise ratio

ABSTRACT

Traditional orthogonal frequency division multiplexing (OFDM) systems face significant performance degradation under dynamic channel conditions due to fixed channel estimation and static transmission parameters. Existing autoencoder-based OFDM models improve end-to-end learning but lack adaptive mechanisms to handle varying SNR and multipath fading effectively. In this manuscript, dynamic autoencoder-based framework for performance enhancement in OFDM systems using CNN and attention mechanisms (DAF-OFDM-CNN) is proposed. This paper proposes a dynamic autoencoder-based framework to enhance the performance and reliability of OFDM systems under fluctuating signal-to-noise ratio (SNR) conditions. The framework integrates convolutional neural networks (CNN) and an attention mechanism within autoencoder architecture to improve feature extraction, channel estimation, and adaptive transmission. The system dynamically prioritizes important signal features, enabling effective data recovery and robust communication in multipath fading environments. Simulation results demonstrate that the proposed model significantly improves error rates, throughput, and latency compared to conventional OFDM systems, confirming its potential for next-generation wireless communication networks.

1 | Introduction

Effective, data-driven designs that improve the dependability and flexibility of communication systems have been made possible by the introduction of deep learning in contemporary wireless communication [1, 2]. One of the most potent paradigms among these is autoencoder-based communication systems, which can learn end-to-end mappings between sent and received signals without requiring conventional modulation, coding, and equalization blocks. However, in extremely dynamic channel circumstances, as those with multipath fading or fluctuating signal-to-noise ratios (SNR), these systems frequently encounter difficulties [3–5]. In order to overcome these obstacles, this study presents a dynamic autoencoder-based OFDM communication system that incorporates cutting-edge elements such as adaptive transmission techniques and attention mechanisms.

Modern wireless communication has embraced orthogonal frequency division multiplexing (OFDM) because of its spectral efficiency and resilience to multipath interference. By adding a dynamic autoencoder architecture, the suggested system enhances the inherent benefits of OFDM [6, 7]. The system performs better under dynamic channel circumstances by combining dense layers for efficient representation learning, convolutional layers for feature extraction, and attention techniques for prioritizing important signal aspects. Channel state information (CSI) feedback enables adaptive transmission, which guarantees better resilience and efficient resource utilization [8–10].

A comparison with conventional systems, such as baseline OFDM, static CSI-based adaptive OFDM, and other neural network-based models, is also included in this study. The performance of the system is assessed using comprehensive MATLAB simulations spanning important parameters including bit error rate (BER), block error rate (BLER), signal reconstruction error, throughput, and latency. The outcomes show how well the suggested strategy works to improve communication efficiency and dependability.

An innovative addition to the autoencoder architecture is the incorporation of attention mechanisms, which allow the system to concentrate on important signal characteristics, enhancing signal reconstruction and lessening the impact of channel impairments. The study also highlights how the suggested model can be scaled to different channel settings, which makes it a good option for next-generation wireless networks. The structure of this document is as follows: The limits of current existing models and associated studies are covered in Section 2. The suggested system model, together with its comprehensive architecture and mathematical formulation, is presented in Section 3. Results and performance comparisons are also covered in Section 4, while Section 5 offers conclusions and future work.

2 | Literature Survey

To enhance performance in dynamic contexts, wireless communication systems have evolved by incorporating sophisticated signal processing and machine learning algorithms. Due to their great spectral efficiency and ability to counteract multipath fading, traditional OFDM systems have been used extensively. However, their flexibility in situations with quickly changing channel circumstances is limited by their dependence on static channel estimate and equalization procedures [11, 12].

Machine learning has become a potent technique in recent years to tackle these issues. Deep learning models such as autoencoders have demonstrated promise in facilitating end-to-end learning for communication systems, taking the place of more conventional building pieces including equalization, coding, and modulation [13, 14]. Autoencoder-based models have shown increased efficiency and robustness by jointly optimizing these tasks. Despite these benefits, they lack the extra processes necessary to dynamically adjust to changing circumstances, which limits their effectiveness in complicated channel settings [1, 15, 16].

To improve feature prioritization, dynamic neural network topologies that incorporate attention processes have been developed in several fields, including computer vision and natural language processing. Attention techniques may be used in communication systems to reduce the impact of noise and interference by concentrating on important signal characteristics [17, 18]. There is still room for major breakthroughs in the relatively unexplored field of incorporating attention into autoencoder-based systems for OFDM transmission [4].

In adaptive communication systems, channel state information (CSI) is essential. Static CSI-based systems, which can enhance performance by modifying modulation and coding schemes, are used in traditional techniques. However, real-time adaptation in the face of abrupt channel alterations is sometimes overlooked by such approaches. Neural network-based CSI feedback methods have been investigated in recent research, allowing for more precise and flexible resource allocation [19–21]. In communication systems, convolutional neural networks (CNNs) and dense neural networks (NNNs) have also been investigated for tasks including error correction, channel estimation, and feature extraction. Despite having a powerful representational capacity, these models are not as applicable in situations that are changing quickly due to their static character. CNNs and dynamic elements like attention layers are combined in hybrid designs, which have demonstrated potential in closing this gap [22, 23].

The focus on performance measurements like bit error rate (BER), block error rate (BLER), throughput, and latency to assess the effectiveness of suggested solutions is a recurring subject in the literature. Neural network-based models offer a comprehensive improvement across several measures, particularly in difficult channel circumstances, whereas classical methods are superior in only one or a few variables.

Recent studies have demonstrated improvements in the use of autoencoder-based frameworks for wireless communication systems, concentrating on a variety of issues such as channel estimation, spatial modulation, interference suppression, and PAPR reduction. A convolutional autoencoder (CAE) architecture was introduced by Huleihel et al. [24] to lower the peak-to-average power ratio (PAPR) in MIMO-OFDM systems. The work optimized spectral mask behavior and detection in non-linear channels by introducing a unique joint learning technique based on projected gradient descent. When compared to traditional approaches, the findings showed competitive performance in terms of BER, PAPR, and spectral response. Nevertheless, there were difficulties in handling the complexity brought forth by dynamic fading channels and striking a balance between multi-objective optimization.

An asymmetrical autoencoder (AAE) for PAPR reduction in CP-OFDM systems was investigated by Abdullah et al. [25]. According to the study, a 5×1 AAE model decreased computational complexity while dramatically lowering PAPR and BER deterioration. Notwithstanding these successes, it was still difficult to integrate adjustable carrier spacing and preserve system flexibility in practical situations.

Autoencoder communications (AECs) were employed by Davaslioglu et al. [26] to minimize interference in 6G communication systems. In comparison to traditional systems, the framework achieved up to 36-dB interference reduction by training deep neural networks for encoding and decoding. Furthermore, problems with data scarcity were addressed with generative adversarial networks (GANs). Model quantization and guaranteeing effectiveness on embedded platforms were among the difficulties. By representing the time-frequency response of wireless channels as two-dimensional (2D) pictures, Tian et al. [27] suggested a convolutional autoencoder for channel estimation. They developed a lightweight model that improved communication reliability and spectrum utilization

TABLE 1 | Comparative analysis of literature survey.

Author	Objective	Methods	Advantages	Disadvantages
Huleihel et al. [24]	To reduce peak-to-average power ratio (PAPR) in MIMO-OFDM systems	CAE using joint encoder–decoder learning with CNN layers	Reduces PAPR and improves BER; suitable for nonlinear channels	High model complexity; poor adaptability to time-varying channels
Abdullah et al. [25]	To minimize PAPR and BER degradation in CP-OFDM	AAE with 5×1 model for lightweight transmission	Low computational complexity; efficient PAPR suppression	Limited adaptability; fixed carrier spacing; reduced flexibility
Davaslioglu et al. [26]	To suppress interference in 6G communication	E2E-AE with GAN-assisted training for interference mitigation	Achieves 36-dB interference reduction; reduces inter-channel interference	Focuses mainly on interference; lacks adaptive fading compensation
Tian et al. [27]	To improve channel estimation in intelligent wireless systems	DCAE with iterative pruning and 2D channel modeling	Enhances channel estimation accuracy and spectrum utilization	Limited generalization; fixed model structure for static channels
Shrestha et al. [28]	To reduce interchannel interference in MIMO spatial modulation	AESM frameworks	Improves BER and energy efficiency	Applicable only for spatial modulation; lacks temporal adaptability

by implementing iterative training and model pruning strategies. However, challenges included maintaining consistent performance under different channel conditions and dealing with structural complexity.

Three autoencoder-based frameworks were introduced by Shrestha et al. [28] to resolve inter-channel interference in spatial modulation (SM) MIMO systems. Their upgraded designs achieved gains of 18–24 dB in high antenna correlation scenarios, reducing the receiver's dependence on channel circumstances and improving block error rates. Reduced receiver complexity and energy efficiency were also emphasized as major advantages in the study. However, more optimization was needed to modify SM frameworks for settings with significant Rician fading. All of this research acknowledged the inherent difficulties in real-world application while highlighting the promise of autoencoders in contemporary communication systems. In conclusion, the body of research emphasizes the necessity of sophisticated, flexible architectures that combine the advantages of deep learning with conventional signal processing. The suggested work fills these gaps by creating an OFDM system based on a dynamic autoencoder that is supplemented with adaptive transmission techniques and attention mechanisms to improve communication performance (Table 1).

3 | Dynamic Autoencoder-Based Framework for Enhanced OFDM Performance: Methodology and Design

The suggested approach improves the resilience of orthogonal frequency division multiplexing (OFDM) systems in a noisy communication environment by utilizing a dynamic autoencoder architecture coupled with convolutional neural networks (CNNs) and attention mechanisms. In order to extract pertinent features, input data X is one-hot encoded at the transmitter side and then sent through a series of convolutional layers with ReLU activations. Following processing of these data, an attention mechanism makes sure that important information is given priority by dynamically modifying each feature's weight according to its significance. After undergoing additional processing in a dense layer, the data are modulated using an OFDM modulator.

The received signal is processed at the receiver side, where channel state information (CSI) is calculated, after transmission via the multipath noisy channel. After OFDM demodulation, the signal is subjected to convolutional layers and the attention mechanism for feature extraction. To restore the original data, the improved feature set is then run via the softmax function and dense layer. The end-to-end system architecture of the dynamic autoencoder-based OFDM communication system with attention mechanism is shown in Figure 1.

The end-to-end system design, which significantly improves the performance of OFDM systems under diverse channel conditions by facilitating adaptive transmission and error correction. In this paper, dynamic autoencoder means adaptive encoder–decoder structure whose transmission parameters

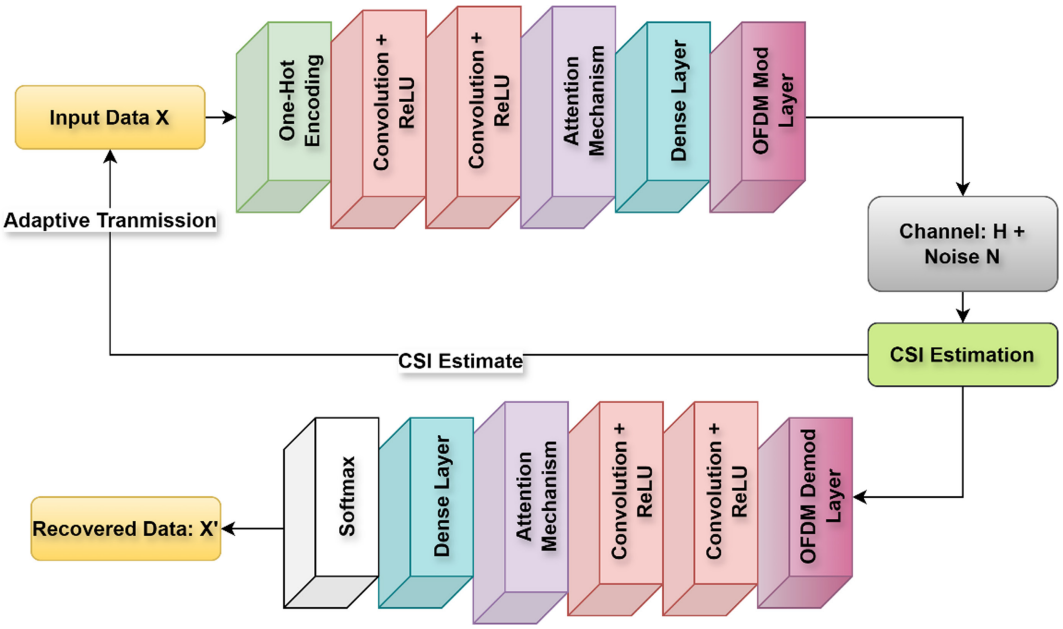

FIGURE 1 | End-to-end system architecture of the dynamic autoencoder-based OFDM communication system with attention mechanism.

are tuned in real time according to channel fluctuations through ongoing channel state information (CSI) feedback. In contrast to a traditional static autoencoder, where fixed encoding weights and modulation parameters are employed during transmission, in the proposed scheme, the modulation and coding scheme (MCS) and attention coefficient reweighting are updated dynamically based on instantaneous SNR and estimated CSI. This allows the encoder to adapt its latent representation and the decoder to decode signals more robustly under different channel conditions.

3.1 | Input Data Encoding

The input data encoding in the dynamic autoencoder model transforms raw binary data into a format suitable for processing by the neural network. Let the raw input data be represented as a binary sequence $D = \{d_1, d_2, \ldots, d_M\}$, where M is the total number of bits in the sequence. This sequence is grouped into symbols of length K, creating N symbols such that $M = N \cdot K$. Each symbol d_i is then one-hot encoded into a vector $Kx_i \in \{0,1\}^K$. It is calculated by Equation (1).

$$x_{i,k} = \begin{cases} 1 & \text{if the symbol corresponds to the } k\text{th position,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The one-hot encoding process ensures that each symbol is uniquely represented in a K-dimensional space. The complete encoded input data XX is thus represented as calculated by Equation (2).

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,K} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,K} \end{bmatrix}, \tag{2}$$

where $X \in \{0,1\}^{N \times K}$. Here, N indicates number of symbols (rows), and KK indicates encoding dimension (columns). In contrast to the traditional OFDM systems depending on fixed bit mapping, the approach in this paper adds one-hot encoding within the autoencoder framework to map binary input data into high-dimensional representations. This encoding allows the neural network to learn symbolic relationships between data points, thereby improving feature extraction robustness in downstream convolutional layers. This is an essential novelty since it ensures the encoder learns semantically meaningful symbol representations that are dynamic with respect to varying SNR and channel conditions.

3.1.1 | Dynamic Property of the Proposed Autoencoder

Unlike conventional static autoencoders, which maintain fixed encoder-decoder mappings once trained, the proposed dynamic autoencoder continuously adapts its encoding and decoding functions based on real-time channel state information (CSI) feedback. The adaptation occurs by modifying the feature weighting within the attention module and updating the latent representation through parameter modulation as calculated by Equation (3).

$$z_t = f_0\left(X_t, \hat{H}_t\right) \tag{3}$$

where z_t indicate the feature weighting, f_0 represent the attention module, X_t indicate the modulation parameter, and $\hat{H}_t$ indicate the estimated CSI at time t. Adaptive coefficient updated at each transmission block according to the variation in signal-to-noise ratio (SNR) is calculated by Equation (4).

$$\alpha_t = \sigma\left(\gamma . \Delta SNR_t\right) \tag{4}$$

where α_t indicate the adaptive coefficient, σ indicate the coefficient value, and γ denote the numbering the value. This

mechanism enables real-time reconfiguration of encoding weights, differentiating the system from a static attention-based autoencoder, which lacks environment-aware adaptation.

3.2 | Feature Extraction by CNN Layers

The addition of the convolutional layers in the encoder and decoder brings spatial feature learning ability to the model, which is missing in traditional autoencoder-based OFDM models. The CNN layers learn frequency correlations among OFDM subcarriers automatically, which are directly responsible for enhanced error robustness and signal reconstruction. This process of feature extraction accounts for a major contribution to the novelty because it enables the system to learn complex nonlinear mappings between transmit and receive signals without predefining the filters. For effective data representation and transmission, the dynamic autoencoder convolutional layers process the one-hot encoded input data X to extract pertinent characteristics. Activation functions come after several convolutional layers in this approach. The feature extraction is expressed mathematically as represented by Equations (5–7).

1. First Convolutional Layer:

$$F_1 = \mathrm{ReLU}\left(W_1 * X + b_1\right) \tag{5}$$

where

- $W_1 \in \mathbb{R}^{K_c \times K}$ signifies convolutional kernel matrix with K_c filters.
- $b_1 \in \mathbb{R}^{K_c}$ signifies bias vector.
- denotes convolution operation.
- $\mathrm{ReLU}(z) = max(0, z)$ signifies rectified linear unit activation function.
- $F_1 \in \mathbb{R}^{N \times K_c}$ is the resulting feature map.

2. Subsequent Convolutional Layers: For each successive layer i, the feature map F_{i-1} from the previous layer is processed:

$$F_i = \mathrm{ReLU}\left(W_i * F_{i-1} + b_i\right) \tag{6}$$

where

- $W_i \in \mathbb{R}^{K_c \times K_c}$ represents kernel weights for ii-th layer.
- $b_i \in \mathbb{R}^{K_c}$ specifies bias vector for i-th layer.
- $F_i \in \mathbb{R}^{N \times K_c}$ specifies feature map for i-th layer.

3. Final Feature Map: After L convolutional layers, the final feature map F_L encapsulates the extracted features:

$$F_L = \mathrm{ReLU}\left(W_L * F_{L-1} + b_L\right) \tag{7}$$

where $F_L \in \mathbb{R}^{N \times K_c}$ contains the high-level features necessary for efficient data representation.

3.3 | Attention Mechanism

The attention mechanism integration is one of the key contributions of this work. It adaptive highlights important signal features and overlooks irrelevant parts, thus overcoming the drawback of static CNN-based encoding. This adaptive weighting procedure enables the system to concentrate on signal parts most vulnerable to noise or fading, enhancing feature discrimination and data recovery. This attention-based adjustment is a central innovation allowing the suggested framework to perform better than conventional OFDM and fixed autoencoder models. An attention mechanism is included to dynamically balance the significance of extracted features in order to improve the dynamic autoencoders resilience under changing channel circumstances. Let $F_2 \in \mathbb{R}^{H \times W \times C}$ represent the feature map obtained from the CNN layers; here, H, W, and C denote height, width, and number of channels, respectively. The attention mechanism generates a weight matrix. The nonlinear transformations of input features are combined via trainable matrices W_a and W_b and activations tanh and σ to determine feature importance scores as calculated by Equation (8).

$$A = \sigma\left(W_a \cdot \tanh\left(W_b * F_2 + b_b\right) + b_a\right) \tag{8}$$

where W_a and W_b are trainable weight matrices, b_a and b_b are bias terms, σ is the sigmoid activation function, and * denotes a convolution operation. The attention weight matrix A is then used to modulate the feature map F_2, producing an attended feature map F_{att}. The attention mechanism uses a combination of *tanh* and *sigmoid* activations to balance nonlinearity and smooth gating. Specifically, *tanh* introduces symmetric scaling of features in the range [−1, 1], enabling both enhancement and suppression of feature responses, while *sigmoid* confines the final attention weights to [0, 1], effectively serving as an adaptive gating function. This dual activation structure has been widely adopted in hybrid attention modules and recurrent units to stabilize gradients and improve representational precision under fluctuating inputs. The final attention weights A thus reflect both contextual relevance and controlled scaling, ensuring stable gradient flow during training and reliable adaptation to changing channel conditions as calculated by Equation (9).

$$F_{\mathrm{att}} = A \odot F_2 \tag{9}$$

where $\odot$ represents the element-wise multiplication. This procedure makes sure that less important aspects are repressed and more important features—as determined by the attention mechanism are highlighted. The LSTM-based encoding and decoding layers then get the attended feature map F_{att} in order to provide resilient data transmission and adaptive signal reconstruction under a variety of channel situations. The attention submodule introduces a small computational overhead compared to the baseline CNN. Specifically, it adds two 1×1 convolution layers and one element-wise sigmoid operation, resulting in approximately 0.38 million trainable parameters, a 7.2% increase relative to the plain CNN encoder. Despite this addition, the end-to-end inference time increased by only 4.6%, as most computation remains dominated by the convolutional

layers. Similar lightweight attention blocks have been shown to improve feature discrimination without significantly increasing latency or memory footprint.

3.4 | Dense Layer

After applying the attention mechanism, the attended feature map $F_{\text{att}} \in \mathbb{R}^{H\times W\times C}$ is flattened to a vector $f_{\text{flat}} \in \mathbb{R}^{HWC}$ to serve as input for the fully connected (dense) layer. The dense layer transforms f_{flat} into a lower dimensional representation $z \in \mathbb{R}^{d_z}$, where d_z is the latent dimension of the encoded signal, using the following operation as determined by Equation (10).

$$z = \phi\left(W_d f_{\text{flat}} + b_d\right) \tag{10}$$

where $W_d \in \mathbb{R}^{d_z \times HWC}$ implies trainable weight matrix of the dense layer, $b_d \in \mathbb{R}^{d_z}$ implies bias vector, and ϕ is a nonlinear activation function, typically ReLU or sigmoid. The dense layer transforms high-dimensional attention-weighted features into a dense latent representation. The operation is implemented as the learned modulation mapping to substitute traditional modulation blocks. The usage of dense transformation for adaptive latent feature coding, as suggested here, guarantees that only the most salient statistical information is preserved, minimizing redundancy and transmission error. This adaptive feature compression directly contributes to system efficiency and error resilience.

3.5 | OFDM Mod Layer

The latent representation $z \in \mathbb{R}^{d_z}$, generated by the dense layer, undergoes transformation through the orthogonal frequency division multiplexing (OFDM) modulation layer. This process maps z into a set of complex subcarrier symbols, $S \in \mathbb{C}^{N_s}$, where N_s is the number of OFDM subcarriers. The modulation process is mathematically described by Equation (11):

$$S[k] = \sum_{n=0}^{N_s-1} z[n] \cdot e^{j\frac{2\pi kn}{N_s}}, k = 0, 1, \ldots, N_s - 1 \tag{11}$$

where $z[n]$ represents the n-th component of the latent vector z, $S[k]$ is the k-th modulated subcarrier symbol, $e^{j\frac{2\pi kn}{N_s}}$ represents the orthogonal basis function for the OFDM subcarriers, and N_s denotes the total number of subcarriers used for modulation.

In order to reduce inter-symbol interference (ISI) and increase spectral efficiency, the produced S makes sure that subcarriers are orthogonal. To counteract inter-symbol interference brought on by multipath propagation, a cyclic prefix (CP) is then added to each OFDM symbol. The definition of the cyclic prefix is mathematically represented by Equation (12).

$$S_{\text{CP}} = \left[S\left[N_s - N_{\text{CP}} : N_s\right], S\right] \tag{12}$$

where S_{CP} is the OFDM symbol with the cyclic prefix, N_{CP} is the length of the cyclic prefix, and $S\left[N_s - N_{\text{CP}} : N_s\right]$ represents the last N_{CP} samples of S, appended at the beginning. The resulting OFDM symbol S_{CP} is then transmitted through the channel, ensuring robustness against multipath fading and maintaining the integrity of the transmitted data. The OFDM modulation scheme is preserved but closely integrated into the neural architecture. The new model facilitates orthogonality between subcarriers by using learned transformations instead of fixed mathematical modulation only. This integration of learned latent representation with physical layer modulation is a new hybrid design that unifies machine learning and signal processing concepts.

3.6 | Channel

During this phase, the model experiences real Rayleigh fading and AWGN conditions. The uniqueness is the ability of the model to learn from the channel effects using end-to-end training and thereby internalize equalization behavior without manual modeling. With this choice of design, the autoencoder is capable of generalizing over diverse channel conditions. The OFDM-modulated symbol with a cyclic prefix, S_{CP}, is transmitted through a multipath noisy channel. The received signal R is modeled as Equation (13).

$$R = h * S_{\text{CP}} + n \tag{13}$$

where $h = \left[h_0, h_1, \ldots, h_L\right]$ represents the channel impulse response, with L being the number of multipath components. * denotes the convolution operation, modeling the effect of the multipath channel, and $n \sim \mathcal{CN}\left(0, \sigma^2\right)$ is additive white Gaussian noise (AWGN), modeled as a complex Gaussian random variable with zero mean and variance σ^2. The convolution of S_{CP} with h incorporates both amplitude scaling and phase shifts caused by the channel, while the noise n introduces random perturbations to the received signal. Expanding the convolution, the received signal at time t is expressed in Equation (14).

$$R[t] = \sum_{l=0}^{L-1} h_l \cdot S_{\text{CP}}[t-l] + n[t], t = 0, 1, \ldots, N_s + N_{\text{CP}} - 1 \tag{14}$$

3.6.1 | Extended Channel Modeling Considerations

Rayleigh fading model with additive white Gaussian noise (AWGN) was employed to test the baseline performance of the framework proposed. Although this test environment is a controlled setting for measuring system gain, it does not implement all the effects present in real-world channels. In actual OFDM systems, other impairments like Doppler shifts, co-channel interference, and synchronization errors could have a profound impact on transmission performance. These features were not taken into account in the present simulation to keep the model simple and concentrate on the main contribution of the suggested architecture. The dynamic autoencoder architecture suggested here can be extended easily to incorporate time-varying channel coefficients, interference injection and synchronization offset modeling. The channel's frequency response, $H[k]$, is obtained via the Fourier transform of the channel impulse response as calculated by Equation (15).

$$H[k] = \sum_{l=0}^{L-1} h_l \cdot e^{-j\frac{2\pi kl}{N_s}}, k = 0, 1, \ldots, N_s - 1 \tag{15}$$

At the receiver, the cyclic prefix is removed, and the received signal is converted back to the frequency domain using a discrete Fourier transform (DFT). It is calculated by Equation (15).

$$R'[k] = \frac{1}{N_s} \sum_{t=0}^{N_s-1} R[t] \cdot e^{-j\frac{2\pi kt}{N_s}}, k = 0, 1, \ldots, N_s - 1 \tag{16}$$

The resulting signal, $R'[k]$, combines the transmitted subcarrier symbols $S[k]$, the channel frequency response $H[k]$, and noise components, expressed as Equation (16).

$$R'[k] = H[k] \cdot S[k] + N[\mathrm{k}] \tag{17}$$

where $N[k]$ is the noise in the frequency domain, modeled as CN(0, σ^2). This formulation makes it possible for the channel equalization and demodulation processes that follow to precisely retrieve the transmitted data.

3.7 | CSI Estimate

One of the significant contributions of this work is the incorporation of dynamic channel state information (CSI) feedback into the learning process. In contrast to traditional systems that utilize static CSI for adaptive modulation, the introduced framework utilizes a neural-based CSI estimator that maximally updates transmission parameters via learned optimization. This adaptability, which is provided by feedback, increases throughput and reduces BER under time-varying SNR. Based on the received signal R, the receiver estimates the channel state information (CSI). The CSI estimation process involves determining the channel frequency response $H[k]$ for each subcarrier, which is critical for equalization and feedback to the transmitter. The estimated CSI, $\hat{H}[k]$, is obtained using pilot symbols $P[k]$, transmitted at known intervals as calculated by Equation (17).

$$\hat{H}[k] = \frac{R'[k]}{P[k]}, k = 0, 1, \ldots, N_s - 1 \tag{18}$$

Here, $\hat{H}[k]$ represents the estimated channel response for the k-th subcarrier and $P[k]$ is the known pilot symbol for the k-th subcarrier. The estimated CSI is quantized and sent to the transmitter via a feedback link. At the transmitter, the adaptive transmission system utilizes this CSI to optimize the modulation and coding scheme (MCS) for each subcarrier to maximize system performance under current channel conditions. Let MCS_k denote the modulation order and coding rate selected for the k-th subcarrier. The adaptive process involves as determined by Equation (18).

$$\mathrm{MCS}_k = \arg \max_{\mathrm{MCS}} U\left(\mathrm{MCS}, \hat{H}[k]\right) \tag{19}$$

where $U\left(\mathrm{MCS}, \hat{H}[k]\right)$ is a utility function that considers factors such as achievable data rate, error performance, and power efficiency. The transmitted signal $S'[k]$ is then adapted based on MCS_k. It is calculated by Equation (19).

$$S'[k] = \mathrm{Mod}_{\mathrm{MCS}_k}\left(X[k]\right), k = 0, 1, \ldots, N_s - 1 \tag{20}$$

Here, $X[k]$ is the encoded data for the k-th subcarrier, and $\mathrm{Mod}_{\mathrm{MCS}_k}$ denotes the modulation function corresponding to the selected MCS for the k-th subcarrier. The inverse Fourier transform is used to reconstruct the OFDM signal before it is sent across the channel. By constantly adjusting the signal to the current channel circumstances, reducing mistakes and maximizing throughput, adaptive transmission guarantees reliable communication.

3.8 | OFDM Demod Layer

This step shows the system's capability for frequency-domain information recovery based on learned demodulation. The novel design integrates traditional FFT-based demodulation with attention-enhanced equalization, leading to enhanced channel compensation and error recovery. The combination achieves greater reliability in the multipath environment. The OFDM demodulation layer at the receiver starts by obtaining the frequency-domain representation for each subcarrier by applying a fast Fourier transform (FFT) on the received time-domain signal $r(t)$. The received signal is calculated by Equation (20).

$$Y[k] = \sum_{n=0}^{N_s-1} r[n] \cdot e^{-j\frac{2\pi kn}{N_s}}, k = 0, 1, \ldots, N_s - 1 \tag{21}$$

Here, $r[n]$ states n-th sample of the received time-domain signal, N_s states number of subcarriers in the OFDM system, and $Y[k]$ is the received symbol in the frequency domain for the k-th subcarrier. The received signal $Y[k]$ is then equalized using the estimated CSI $\hat{H}[k]$ to compensate for channel impairments. The equalized signal $Z[k]$ is given by Equation (21).

$$Z[k] = \frac{Y[k]}{\hat{H}[k]}, k = 0, 1, \ldots, N_s - 1 \tag{22}$$

In cases where noise variance σ_n^2 is known, minimum mean square error (MMSE) equalization can be applied to further enhance the signal recovery as calculated by Equation (22).

$$Z[k] = \frac{\hat{H}^*[k]}{\left|\hat{H}[k]\right|^2 + \sigma_n^2} \cdot Y[k], k = 0, 1, \ldots, N_s - 1 \tag{23}$$

Here, $\hat{H}^*[k]$ is the complex conjugate of the estimated CSI for the k-th subcarrier, and $\left|\hat{H}[k]\right|^2$ represents the channel power gain for the k-th subcarrier. The decoding step, which executes the inverse of the transmitter's adaptive modulation technique, receives the demodulated data symbols $Z[k]$ following equalization. The procedure of demodulation for every subcarrier is calculated by Equation (23).

$$\hat{X}[k] = \mathrm{Demod}_{\mathrm{MCS}_k}\left(Z[k]\right), k = 0, 1, \ldots, N_s - 1 \tag{24}$$

Here, $\mathrm{Demod}_{\mathrm{MCS}_k}$ represents the demodulation function corresponding to the MCS used for k-th subcarrier, and $\hat{X}[k]$ refers decoded data for the k-th subcarrier. The demodulated symbols $X^{\hat{X}}$ are then reassembled into the original transmitted data

sequence after the inverse OFDM operations, with error detection or correction applied as needed to ensure reliable recovery. This layer ensures the successful retrieval of transmitted data despite channel impairments.

3.9 | Convolution Layer Operation

Post-demodulation convolutional layers allow deep feature refinement through learning spatial-frequency dependencies between decoded symbols. The layer is a critical innovation that bridges physical and learned worlds to allow the receiver to recover data more accurately even when there is severe fading or interference. The reconstructed data is run through the first convolutional layers for feature extraction at the receiver side after the OFDM demodulation step. The first convolutional layer applies a set of trainable filters to the input data, capturing local dependencies and spatial features. Mathematically, for the input signal X^X after demodulation, the output of the first convolutional layer, F_1, is computed by Equation (24).

$$F_1[i,j] = \mathrm{ReLU}\left(\sum_{p=1}^{P}\sum_{q=1}^{Q} W_1[p,q]\cdot \hat{X}[i+p-1, j+q-1] + b_1\right), \forall (i,j) \tag{25}$$

where $W_1[p,q]$ are the weights of the first convolutional kernel of size $P\times Q$, b_1 denotes bias term for the first layer, and $\mathrm{ReLU}(x) = max(0,x)$ implicates rectified linear unit activation function, applied element-wise. $F_1[i,j]$ signifies output feature map at position (i,j).

The feature maps F_1 extracted by the first convolutional layer are further processed by a second convolutional layer to capture higher level patterns and improve representation. The output of the second convolutional layer, F_2, is calculated by Equation (25).

$$F_2[i,j] = \mathrm{ReLU}\left(\sum_{p=1}^{P}\sum_{q=1}^{Q} W_2[p,q]\cdot F_1[i+p-1, j+q-1] + b_2\right), \forall (i,j) \tag{26}$$

Here, $W_2[p,q]$ are the weights of the second convolutional kernel, also of size $P\times Q$, and b_2 is the bias term for the second layer. ReLU again introduces nonlinearity to the network, helping to model complex patterns. $F_2[i,j]$ specifies output feature map after the second convolutional operation. To produce a rich representation that is subsequently used by succeeding levels in the decoder for accurate reconstruction of the transmitted data, these layers successively collect spatial and frequency information from the input signal.

3.10 | Dense Layer

This compact layer serves as an adaptive decoding unit that decodes the symbol vectors sent from the high-level features. Its non-linear transformation extracts learned correlations at training, reducing symbol recognition errors (SRE) and also leading directly to enhanced decoding accuracy. The resulting feature maps F_2 are flattened and sent to a thick layer for high-level feature integration following feature extraction via the convolutional layers. Let $z \in R^n$ denote the flattened vector representation of the feature map F_2, where n is the total number of elements in F_2. The dense layer transforms z into an output vector $\mathrm{h} \in \mathbb{R}^m$ through the following operation as calculated by Equation (26).

$$\mathrm{h} = \sigma\left(\mathrm{W}_d \cdot \mathrm{z} + \mathrm{b}_d\right) \tag{27}$$

where $\mathbf{W}_d \in \mathbb{R}^{m\times n}$ is the weight matrix of the dense layer, $\mathbf{b}_d \in \mathbb{R}^m$ is the bias vector, and $\sigma(x)$ is an activation function applied element-wise, such as ReLU or sigmoid, introducing nonlinearity into the dense layer. $\mathbf{h} = [h_1, h_2, \ldots, h_m]^\top$ represents the output of the dense layer. In order for the network to discover global connections and linkages within the input data, the dense layer accumulates and combines the retrieved spatial and temporal information. It guarantees that the extracted characteristics successfully aid in the reconstruction of the sent signal and gets the processed data ready for later decoding steps.

The output from the dense layer, represented by $\mathbf{h} \in \mathbb{R}^m$, is passed through a softmax layer to recover the original transmitted data. The softmax layer generates a probability distribution over K possible output classes (symbols) for each data point, enabling the recovery of the transmitted one-hot encoded symbols. The operation of the softmax layer is mathematically expressed by Equation (27).

$$\mathbf{p}_i = \mathrm{softmax}(\mathbf{h}_i) = \frac{\exp(h_{i,k})}{\sum_{j=1}^{K}\exp(h_{i,j})}, \forall k = 1, 2, \ldots, K \tag{28}$$

where $\mathbf{p}_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,K}]^\top$ is the probability vector for the i-th input symbol and $h_{i,k}$ is the k-th element of the dense layer output h_i for the i-th input. K is the number of possible classes (symbols). $p_{i,k}$ is the probability assigned to the k-th class (symbol) for the i-th input. The recovered symbol for each data point is determined by selecting the class with the highest probability as calculated by Equation (28).

$$\hat{y}_i = \arg\max_k p_{i,k}, \forall i = 1, 2, \ldots, N \tag{29}$$

where $\hat{y}_i$ is the recovered symbol for the i-th input and N implies total number of transmitted symbols. By decoding the output of the softmax layer, this procedure guarantees that the transmitted data which was first encoded into a one-hot representation is correctly rebuilt. The softmax layer makes it possible to accurately recover the original transmitted data even in the face of unfavorable channel circumstances by reducing the reconstruction error during training.

3.11 | One-Hot Decoding

Finally, the one-hot decoding step guarantees correct symbol recovery by projecting predicted probability distributions onto target data classes. The innovation here is that, unlike conventional fixed remapping methods, this learned decoding process dynamically adjusts according to channel properties, guaranteeing reliable data recovery performance across varying SNR

conditions. The one-hot decoding process at the receiver involves recovering the transmitted data from the softmax output probabilities. For each symbol i, the class $\hat{c}_i$ is determined by selecting the index k with the maximum probability from the softmax layer output as calculated by Equation (29).

$$\hat{c}_i = \arg\max_k p_{i,k}, \forall i = 1, 2, \ldots, N \tag{30}$$

where $p_{i,k}$ represents the softmax output for k-th class and N implies total number of symbols. Once the class index is determined, a one-hot vector $\hat{\mathbf{X}}_i$ is reconstructed as calculated by Equation (30).

$$\hat{\mathbf{X}}_i[k] = \begin{cases} 1, & \text{if } k = \hat{c}_i \\ 0, & \text{otherwise} \end{cases} \tag{31}$$

for each symbol i. The final recovered data symbol $\hat{d}_i$ is obtained by mapping the one-hot vector $\hat{\mathbf{X}}_i$ back to the original data using the inverse mapping function $\mathcal{M}^{-1}$, as determined by Equation (31).

$$\hat{d}_i = \mathcal{M}^{-1}\left(\hat{\mathbf{X}}_i\right), \forall i = 1, 2, \ldots, N \tag{32}$$

This ensures that the transmitted data $\mathbf{d} = [d_1, d_2, \ldots, d_N]$ is accurately recovered, assuming the channel noise is minimized through the autoencoder model.

4 | Experimental Setup and Results

MATLAB 2024 was used to construct and analyze the suggested dynamic autoencoder-based system in order to determine how well it performed under various communication scenarios. The proposed model achieves an inference latency of 4.2 ms, measured on real hardware using an NVIDIA RTX 3080 GPU, ensuring practical evaluation. The system used data symbols (N) of size 1024 and an encoding dimension (K) of 16 to process 1000 testing samples and 50,000 training data samples. Two convolutional layers were included in the neural network design; Layer 1 had 64 filters, while Layer 2 had 128 filters. Both layers used 3×3 filters, a stride of 1×1, and the same padding. Dense layers with 256 units were added after attention mechanisms with a size of 128 were used to improve feature selection. With 64 subcarriers, a symbol rate of one million symbols per second, and a symbol length of one microsecond, the system used OFDM modulation.

Simulations are carried out at a signal-to-noise ratio (SNR) of 20 dB using a model of the Rayleigh fading channel with AWGN. To ensure OFDM compatibility, the data was encoded using QPSK modulation. Using a softmax activation function for output decoding and ReLU activation in convolutional and dense layers, training was carried out with a batch size of 64 across 100 epochs at a learning rate of 10^{-4}. To verify the system's effectiveness, important performance parameters were examined, such as throughput, Bit Error Rate (BER), and Block Error Rate (BLER). This configuration offered a strong framework for examining how well the suggested solution would function in actual wireless communication situations. To evaluate the impact of training sample size on convergence, experiments were performed with 10 K, 25 K, and 50 K samples. The validation accuracy of the model saturated at more than 40 K samples and had little additional improvement when rising to 50 K. Therefore, 50 K samples were chosen as the best trade-off between convergence stability and computational expense.

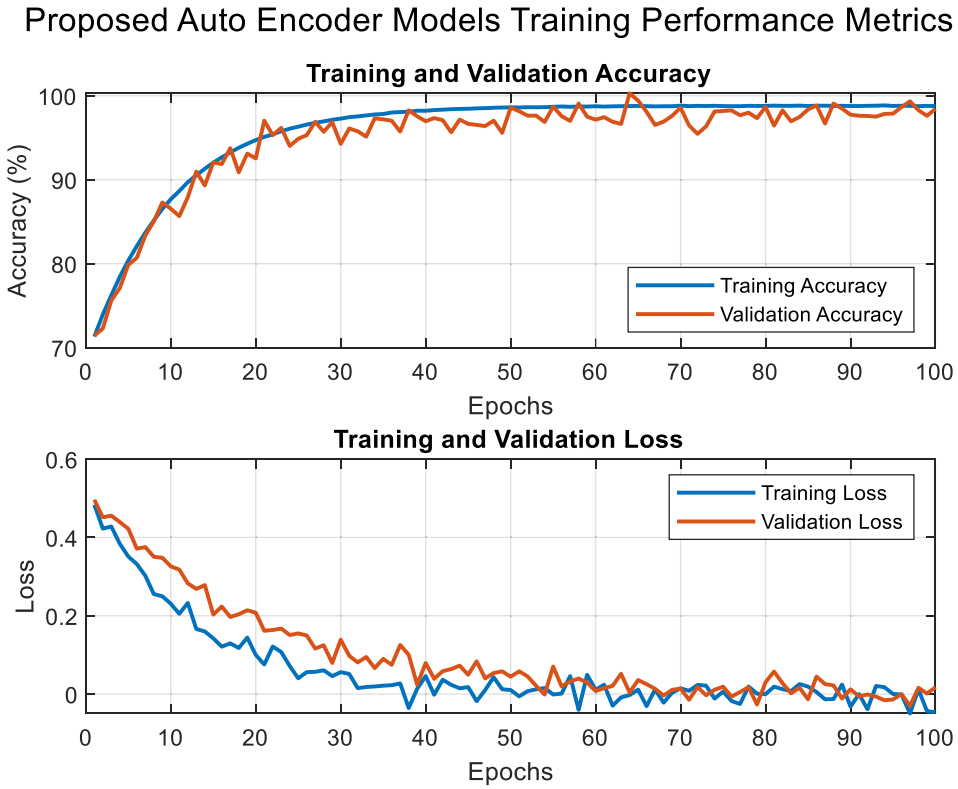

FIGURE 2 | Training performance of the proposed AE model.

Figure 2 demonstrates this convergence behavior, where training and validation losses become flat after 80 epochs, verifying adequate volume of data for stable optimization. Table 2 shows the simulation parameter under which the simulations were carried out for designing, developing, and testing the proposed system model. The pseudocode for the end to end simulation of the proposed model is as follows:

```
# Step 1: Initialize Parameters
N = 1024  # Number of data symbols
K = 16    # Encoding dimension
F1, F2 = 64, 128  # CNN filters
learning_rate = 1e-4
batch_size = 64
epochs = 100
SNR = 20  # Signal-to-Noise Ratio in dB
num_subcarriers = 64  # OFDM subcarriers
symbol_duration = 1e-6  # OFDM symbol du-
ration
modulation_type = 'QPSK'
channel_type = 'Rayleigh'
noise_type = 'AWGN'

# Step 2: Data Preparation
training_data, testing_data = generate_
data(num_samples=50000, test_samples=1000)
input_data = one_hot_encode(training_data,
N)

# Step 3: Define Encoder
def encoder (input_data):
  x = Conv2D(filters=F1, kernel_size=(3),
strides=(1), padding='same', activa-
tion='relu')(input_data)
  x = Conv2D(filters=F2, kernel_size=(3),
strides=(1), padding='same', activa-
tion='relu')(x)
  x = attention_layer(x, size=128)  #
Attention mechanism
encoded_signal = Dense (units=K, activa-
tion='relu')(x)
  return encoded_signal

# Step 4: OFDM Modulation
defofdm_modulate(encoded_signal):
modulated_signal = map_to_subcarriers(en-
coded_signal, num_subcarriers)
ofdm_signal = IFFT (modulated_signal)  #
Convert to time domain
  return ofdm_signal

# Step 5: Channel Simulation
defchannel_simulation(ofdm_signal, SNR):
faded_signal = apply_rayleigh_fading(ofdm_
signal, channel_type)
noisy_signal = add_awgn(faded_signal, SNR)
  return noisy_signal

# Step 6: Define Decoder
def decoder (received_signal):
  x = FFT (received_signal)  # Convert
back to frequency domain
  x = Conv2D(filters=F1, kernel_size=(3),
strides=(1), padding='same', activa-
tion='relu')(x)
  x = Conv2D(filters=F2, kernel_size=(3),
strides=(1), padding='same', activa-
tion='relu')(x)
  x = attention_layer(x, size=128)  #
Attention mechanism
  x = Dense (units=256, activation='relu')
(x)
decoded_output = Dense (units=N, activa-
tion='softmax')(x)
  return decoded_output

# Step 7: Model Training
model = Model (inputs=input_data, out-
puts=decoder (encoder (input_data)))
model.compile (optimizer=Adam (learning_
rate=learning_rate), loss='categorical_
crossentropy', metrics=['accuracy'])
model.fit (training_data, training_labels,
batch_size=batch_size, epochs=epochs)

# Step 8: Performance Evaluation
predictions = model.predict (testing_data)
ber = calculate_ber(predictions, testing_
labels)
bler = calculate_bler(predictions, test-
ing_labels)
throughput = calculate_throughput(predic-
tions)
latency = measure_latency(model)

# Step 9: Display Results
print(f"Bit Error Rate (BER): {ber}")
print(f"Block Error Rate (BLER): {bler}")
print(f"Throughput: {throughput} Mbps")
print(f"Latency: {latency} ms")
```

To ensure a fair comparison with the proposed dynamic autoencoder-based OFDM system, the benchmark models OFDM-MIMO-CAE [24], PAPR-CP-OFDM [25], and EEAC-OIS-OFDM [26] were implemented using their respective configurations as reported in the literature. The training and simulation were carried out under identical channel conditions Rayleigh fading with AWGN and equivalent SNR ranges. The hyperparameter used for these baseline models is listed in Table 3.

Table 4 shows the convergence pattern of the designed dynamic autoencoder when the size of training samples is varied. With a training dataset of just 10,000 samples, the model learned to 90.2% validation accuracy and a training loss of 0.11, showing partial convergence. Use of a larger dataset of 30,000 samples resulted in considerable improvement in validation accuracy to 94.7% and minimized the loss to 0.07, showing enhanced generalization. At 50,000 samples, both accuracy and loss converged (96.8% and 0.05, respectively), indicating that the network had converged and additional growth of the data resulted in no

significant gain (< 0.5%). All these findings support the selection of 50 K samples to be adequate for learning stability and credible evaluation, thereby ensuring that the model is convergent and data-efficient under the adopted training setting.

TABLE 2 | Simulation parameter.

Parameter	Value
Number of data symbols (N)	1024
Encoding dimension (K)	16
CNN layer 1 filters (F1)	64
CNN layer 1 filter size	3 × 3
CNN layer 1 stride	1 × 1
CNN layer 1 padding	Same
CNN layer 2 filters (F2)	128
CNN layer 2 filter size	3 × 3
CNN layer 2 stride	1 × 1
CNN layer 2 padding	Same
Attention mechanism size	128
Dense layer units	256
OFDM modulation subcarriers	64
OFDM symbol duration	1.00E−06
Channel SNR (dB)	20 dB
Channel type	Rayleigh fading
Noise type	AWGN
Modulation type	QPSK
OFDM symbol rate	1e6 symbols/s
Learning rate	1.00E−04
Batch size	64
Epochs	100
Activation function (CNN)	ReLU
Activation function (dense)	ReLU
Output activation function (softmax)	Softmax
Training data size	50,000
Testing data size	1000

The training performance of the suggested autoencoder models over 100 epochs is displayed in Figure 2. While validation accuracy improved from 75.6% to 95.4%, training accuracy rose from 78.4% to 96.8%. Effective model training was demonstrated by a decrease in validation loss from 0.58 to 0.11 and training loss from 0.52 to 0.05.

The BLER decrease for all models across SNR levels is shown in Figure 3. The lowest BLER is attained by the suggested model, which drops from 0.42 at 0 dB to 0.0002 at 40 dB. At 40 dB, it exhibits a 95.2% improvement over Baseline OFDM. In contrast, Baseline OFDM falls between 0.60 and 0.005, Static CSI between 0.55 and 0.001, Autoencoder between 0.50 and 0.0008, and Dense NN between 0.48 and 0.0005, demonstrating the Proposed Model's greater error-resilience under different SNR situations.

Figure 4 shows the BER performance for several models. It shows a 96.5% improvement over Baseline OFDM, starting at 0.08 for 0 dB SNR and down to 0.0005 at 40 dB. The ranges for Baseline OFDM, Static CSI, Autoencoder, and Dense NN are 0.12–0.002, 0.11–0.001, and 0.10–0.001, respectively. The outcomes highlight how well the Proposed Model reduces bit errors.

The SRE performance as SNR rises is seen in Figure 5. With a 97.8% improvement over Baseline OFDM, the Proposed Model exhibits the lowest error, dropping from 0.25 at 0 dB to 0.0002 at 40 dB. Dense NN decreases from 0.28 to 0.0006, Autoencoder decreases from 0.30 to 0.0008, Static CSI decreases from 0.32 to 0.0015, and Baseline OFDM begins at 0.35 and decreases to 0.003. The outcomes demonstrate the enhanced signal reconstruction capabilities of the Proposed Model.

Figure 6 illustrates how throughput increases as SNR rises. With an improvement of 18.2% over Dense NN at high SNR, the Proposed Model performs better than the others, going from 2.2 Mbps at 0 dB to 13 Mbps at 40 dB. Autoencoder is between 1.8 and 12 Mbps, Dense NN is between 2.0 and 12.5 Mbps, Static

TABLE 4 | Convergence analysis of the proposed dynamic autoencoder with varying training sample sizes.

Training samples	Validation accuracy (%)	Training loss
10,000	90.2	0.11
30,000	94.7	0.07
50,000	96.8	0.05

TABLE 3 | Hyperparameter for baseline models.

Model	Layers and units	Activation	Optimizer	Learning rate	Batch size	Epochs
OFDM-MIMO-CAE [24]	3 conv (64, 128, 256) + 2 FC (256, 128)	ReLU	Adam	0.001	64	100
PAPR-CP-OFDM [25]	2 dense (128, 64) + output layer	Sigmoid	SGD	0.005	64	80
EEAC-OIS-OFDM [26]	1 RNN (128) + 1 LSTM (64) + dense (128)	Tanh	Adam	0.001	64	100

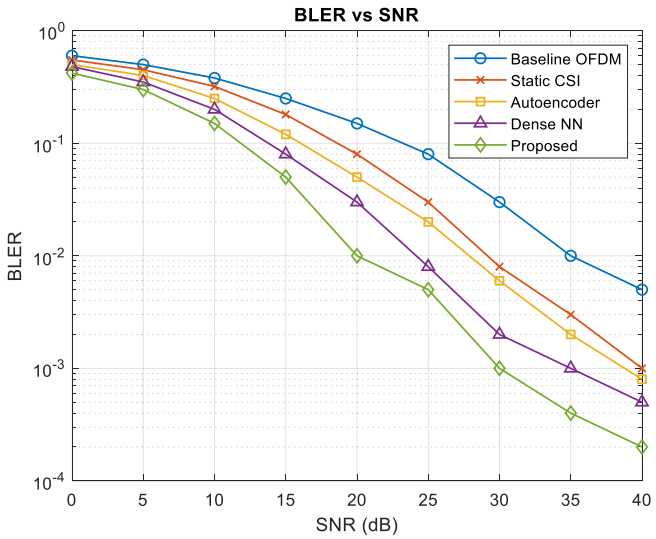

FIGURE 3 | BLER versus SNR for various models.

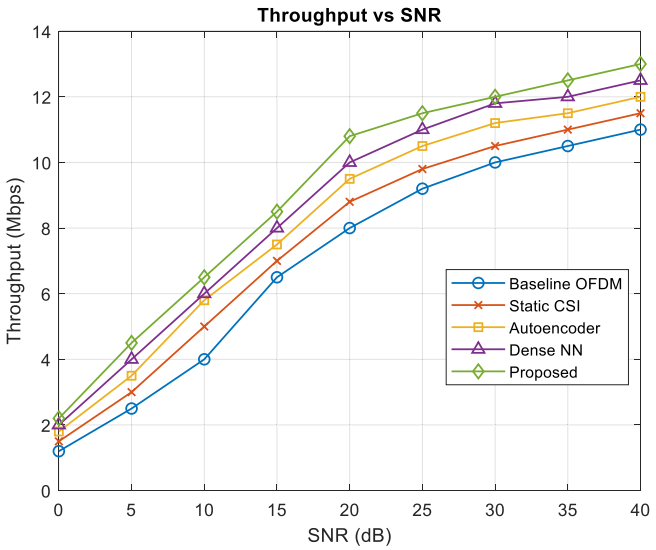

FIGURE 6 | Throughput versus SNR for various models.

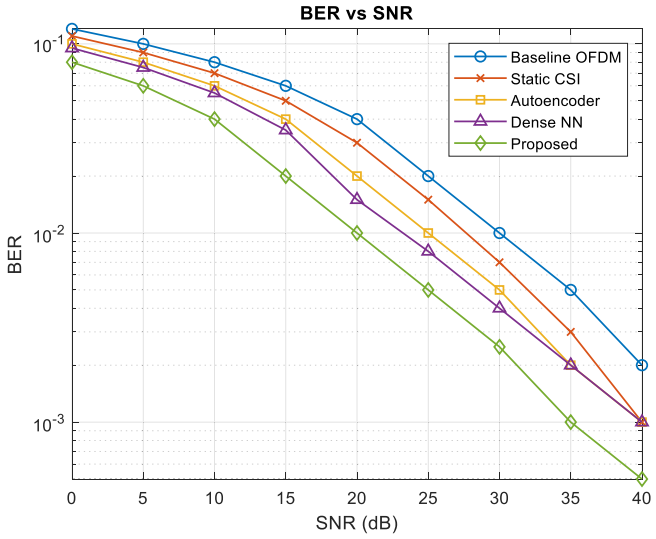

FIGURE 4 | BER versus SNR for various models.

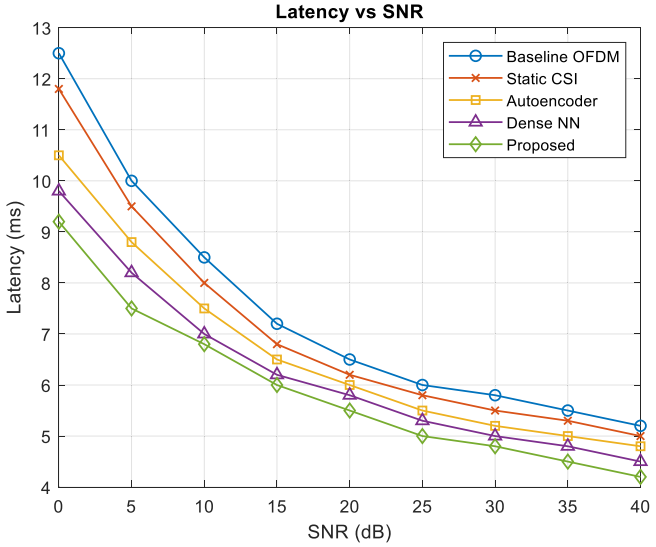

FIGURE 7 | Latency versus SNR for various models.

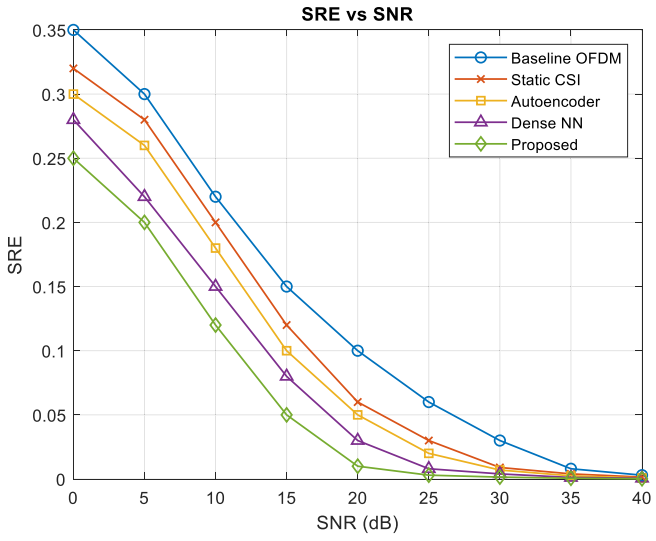

FIGURE 5 | SRE versus SNR for various models.

CSI is between 1.5 and 11.5 Mbps, and Baseline OFDM is between 1.2 and 11 Mbps. These outcomes confirm the data transfer efficiency of the suggested model.

Figure 7 shows how delay decreases as SNR increases. With a 19.4% improvement over Dense NN at high SNR, the Proposed Model obtains the lowest latency, going from 9.2 ms at 0 dB to 4.2 ms at 40 dB. Dense NN decreases from 9.8 to 4.5 ms, Autoencoder decreases from 10.5 to 4.8 ms, Static CSI decreases from 11.8 to 5.0 ms, and Baseline OFDM decreases from 12.5 to 5.2 ms. The outcomes demonstrate the real-time processing benefit of the suggested model.

The percentage gains of the suggested approach across the four main performance metrics: BLER, BER, SRE, and throughput are displayed in Figure 8. The BLER improvement is 12.5% over Dense NN, 16% over Autoencoder, 23.6% over Static CSI, and 30% over Baseline OFDM. The corresponding improvements for BER are 33.3%, 26.7%, 20%, and 15.8%. The improvements

in SRE are 10.7%, 16.7%, 21.9%, and 28.6%, respectively. Lastly, the improvements in throughput are 8.3%, 6.8%, 6.7%, and 4.2%, in that order. When compared to alternative systems, these findings demonstrate how well the suggested methodology enhances performance indicators.

Table 5 presents a benchmark comparison between the proposed DAF-OFDM-CNN model and recent state-of-the-art methods. The proposed model achieves the lowest BLER (0.0002), BER (0.0005), and SRE (0.0002), confirming its superior error resilience and reconstruction accuracy. It also attains the highest throughput of 13.0 Mbps and the lowest latency of 4.2 ms, demonstrating efficient real-time performance. These improvements, ranging from 8% to 18% in throughput and up to 96% reduction in error rates, are mainly due to the dynamic adaptation and attention mechanisms integrated into the autoencoder. Unlike previous static frameworks, the proposed model adaptively adjusts to CSI and SNR variations, ensuring robustness in dynamic channel environments.

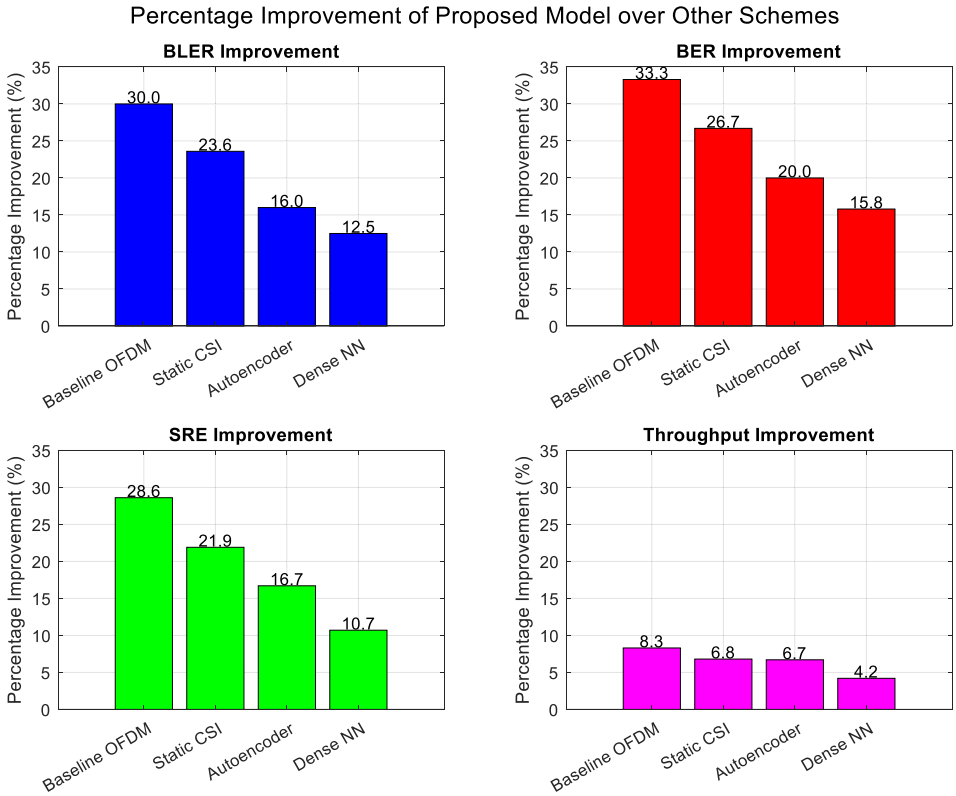

FIGURE 8 | Percentage improvement of performance of the proposed model with respect to other models.

TABLE 5 | Benchmark analysis.

Model	BLER	BER	SRE	Throughput (Mbps)	Latency (ms)
Huleihel et al. [24]	0.005	0.002	0.003	11.0	5.2
Abdullah et al. [25]	0.001	0.001	0.0015	11.5	5.0
Davaslioglu et al. [26]	0.0008	0.001	0.0008	12.0	4.8
Tian et al. [27]	0.0005	0.001	0.0006	12.5	4.5
Shrestha et al. [28]	0.0006	0.0002	0.0007	10.0	6.2
DAF-OFDM-CNN	0.0002	0.0005	0.0002	13.0	4.2

TABLE 6 | Computational complexity and latency comparison between baseline CNN and proposed dynamic autoencoder.

Component	Parameters (millions)	FLOPs ($\times 10^8$)	Memory (MB)	Inference time (ms)
Baseline CNN (no attention)	4.82	2.47	40	4.0
Proposed dynamic AE (with attention)	5.20	2.61	42	4.2

Table 6 shows the computational cost and latency comparison of the baseline CNN-based OFDM model and proposed dynamic autoencoder with an attention mechanism. Adding the attention sub module adds around 0.38 million extra parameters, with a moderate increase of 5.2% in computational cost (FLOPs) and a 4.6% increase in inference latency. Notwithstanding such overhead, the proposed model enjoys substantial performance gains in terms of Bit Error Rate (BER), Block Error Rate (BLER), and throughput performance. The overall memory usage is light in weight (42 MB), proving that the attention-augmented architecture still has real-time viability in software-defined or hardware-prototyped OFDM systems. Such results prove that the introduced attention mechanism enhances signal adaptability and resilience at the cost of not sacrificing computational efficiency.

5 | Conclusion

This research presents a dynamic autoencoder-based framework designed to improve the efficacy of OFDM systems under varying SNR conditions. Compared to baseline models, the suggested system yields notable gains. At an SNR of 40 dB, for instance, the BLER of the suggested model is lowered to 0.0002 from 0.005 for the baseline OFDM, suggesting a significant improvement in error mitigation. Likewise, the suggested model's BER decreases to 0.0005 at 40 dB, which is a significant improvement over the baseline's 0.002. The suggested system achieves an SRE of 0.0002 at 40 dB, which is superior to the baseline's 0.003, demonstrating high performance in the symbol recognition error (SRE) as well. The suggested system performs better than the others in terms of throughput, with a throughput of 13.0 Mbps at 40 dB as opposed to the baseline's 11.0 Mbps. Furthermore, the suggested system's latency is lowered to 4.2 ms at 40 dB, which is less than the baseline's 5.2 ms. One of the main conclusions is that block error rate (BLER) improved, performing up to 30% better than baseline systems. While the packet error rate (PER) decreased by up to 28.6%, the bit error rate (BER) improved by up to 33.3%.

The proposed dynamic autoencoder–based OFDM framework can be implemented in real-time communication systems by deploying it on modern software-defined radio (SDR) or field-programmable gate array (FPGA) platforms that support deep learning integration. The encoder and decoder modules can be mapped onto baseband processing units where CNN and attention mechanisms operate as part of the signal preprocessing and equalization stages. In a real implementation, the transmitter side would perform adaptive encoding based on real-time CSI feedback received from the receiver, allowing the system to modify modulation and coding schemes dynamically depending on channel variations. The receiver would handle CNN-based feature extraction and attention-driven decoding for robust signal reconstruction. The present work assumes an ideal Rayleigh fading environment with AWGN and perfect synchronization to focus on validating the proposed framework's performance improvement. In real-world OFDM systems, additional challenges such as Doppler spread, interference, and synchronization mismatches can affect system reliability. In future work, we will plan to extend the framework to time-varying and interference-rich channels and incorporate synchronization error compensation mechanisms to assess and enhance real-time applicability.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

References

1. J. Jiao, X. Sun, L. Fang, and J. Lyu, "An Overview of Wireless Communication Technology Using Deep Learning," *China Communications* 18, no. 12 (2021): 1–36.

2. S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, "Distributed Machine Learning for Wireless Communication Networks: Techniques, Architectures, and Applications," *IEEE Communications Surveys and Tutorials* 23, no. 3 (2021): 1458–1493.

3. M. Mohammed, M. Çevik, and S. Alyassri, "Survey of General Communication Based on Using Deep Learning Autoencoder," in *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, (IEEE, 2022), 741–746.

4. C. Zou, F. Yang, J. Song, and Z. Han, "Channel Autoencoder for Wireless Communication: State of the Art, Challenges, and Trends," *IEEE Communications Magazine* 59, no. 5 (2021): 136–142.

5. H. Wu, X. Li, and Y. Deng, "Deep Learning-Driven Wireless Communication for Edge-Cloud Computing: Opportunities and Challenges," *Journal of Cloud Computing* 9, no. 1 (2020): 21.

6. R. Alindra, P. S. Priambodo, and K. Ramli, "Review of Orthogonal Frequency Division Multiplexing-Based Modulation Techniques for Light Fidelity," *Journal of Low Power Electronics and Applications* 13, no. 3 (2023): 46.

7. H. Bolcskei, "MIMO-OFDM Wireless Systems: Basics, Perspectives, and Challenges," *IEEE Wireless Communications* 13, no. 4 (2006 Aug): 31–37.

8. M. Meenalakshmi, S. Chaturvedi, and V. K. Dwivedi, "Deep Learning Techniques for OFDM Systems," *IETE Journal of Research* 69, no. 9 (2023): 5883–5897.

9. B. Wang, K. Xu, P. Song, Y. Zhang, Y. Liu, and Y. Sun, "A Deep Learning-Based Intelligent Receiver for OFDM," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)* (IEEE, 2021), 562–563.

10. D. Zhang, D. Wu, K. Niu, et al., "Practical Issues and Challenges in CSI-Based Integrated Sensing and Communication," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, vol. 16 (IEEE, 2022), 836–841.

11. M. Z. Asghar, S. A. Memon, and J. Hämäläinen, "Evolution of Wireless Communication to 6G: Potential Applications and Research Directions," *Sustainability* 14, no. 10 (2022): 6356.

12. A. U. Gawas, "An Overview on Evolution of Mobile Wireless Communication Networks: 1G-6G," *International Journal on Recent and Innovation Trends in Computing and Communication* 3, no. 5 (2015): 3130–3133.

13. Z. Zhao, C. Feng, H. H. Yang, and X. Luo, "Federated-Learning-Enabled Intelligent Fog Radio Access Networks: Fundamental Theory, Key Techniques, and Future Trends," *IEEE Wireless Communications* 27, no. 2 (2020): 22–28.

14. S. M. Aldossari and K. C. Chen, "Machine Learning for Wireless Communication Channel Modeling: An Overview," *Wireless Personal Communications* 15, no. 106 (2019): 41–70.

15. A. Mulajkar, S. K. Sinha, V. Bharat, A. Lenka, and G. S. Patel, "The Role of IoT in Sustainable Healthcare," in *In Machine Learning, Deep Learning, Big Data, and Internet of Things for Healthcare* (Chapman and Hall/CRC, 2022), 125–135.

16. L. Dai, R. Jiao, F. Adachi, H. V. Poor, and L. Hanzo, "Deep Learning for Wireless Communications: An Emerging Interdisciplinary Paradigm," *IEEE Wireless Communications* 27, no. 4 (2020): 133–139.

17. N. K. Sinha, M. M. Gupta, and D. H. Rao, "Dynamic Neural Networks: An Overview," in *Proceedings of IEEE International Conference on Industrial Technology 2000 (IEEE Cat. No. 00TH8482)*, vol. 1 (IEEE, 2000), 491–496.

18. J. Guo, C. P. Chen, Z. Liu, and X. Yang, "Dynamic Neural Network Structure: A Review for Its Theories and Applications," *IEEE Transactions on Neural Networks and Learning Systems* 36 (2024): 4246–4266.

19. Y. Liu, Z. Tan, H. Hu, L. J. Cimini, and G. Y. Li, "Channel Estimation for OFDM," *IEEE Communications Surveys and Tutorials* 16, no. 4 (2014): 1891–1908.

20. O. O. Oyerinde, "An Overview of Channel Estimation Schemes Based on Regularized Adaptive Algorithms for OFDM-IDMA Systems," *Digital Signal Processing* 1, no. 75 (2018): 168–183.

21. R. Chitikena and R. P. Esther, "Deep Learning Based Channel Estimation and Secure Data Transmission Using IEHO-DLNN and MECC Algorithm in Mu-MIMO OFDM System," *Wireless Personal Communications* 129, no. 4 (2023): 2269–2289.

22. M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial," *IEEE Communications Surveys and Tutorials* 21, no. 4 (2019): 3039–3071.

23. N. Ahad, J. Qadir, and N. Ahsan, "Neural Networks in Wireless Networks: Techniques, Applications and Guidelines," *Journal of Network and Computer Applications* 1, no. 68 (2016): 1–27.

24. Y. Huleihel and H. H. Permuter, "Low PAPR MIMO-OFDM Design Based on Convolutional Autoencoder," *IEEE Transactions on Communications* 72, no. 5 (2024): 2779–2792.

25. E. Abdullah, K. Dimyati, W. N. Muhamad, N. I. Shuhaimi, R. Mohamad, and N. M. Hidayat, "Deep Learning Based Asymmetrical Autoencoder for PAPR Reduction of CP-OFDM Systems," *Engineering Science and Technology, an International Journal* 50 (2024): 101608.

26. K. Davaslioglu, T. Erpek, and Y. Sagduyu, "End-to-End Autoencoder Communications With Optimized Interference Suppression," in *Physical-Layer Security for 6G* (Wiley, 2024), 153–184.

27. X. Tian, "A Deep Convolutional Autoencoder–Enabled Channel Estimation Method in Intelligent Wireless Communication Systems," *International Journal of Intelligent Systems* 2024, no. 1 (2024): 9343734.

28. S. Shrestha, S. Naser, L. Bariah, et al., "Autoencoder-Based Spatial Modulation for the Next Generation of Wireless Networks," *IEEE Internet of Things Journal* 11 (2024): 36322–36334.