National Journal of Antennas and Propagation
Research Article

ISSN: 2582-2659 Vol. 7, No. 1, 2025 (pp. 318-328)
WWW.ANTENNAJOURNAL.COM

Neural Network-Based Adaptive Error Correction for High-Capacity Massive MIMO Systems

P. Sakthi Murugan[1]*, A. Kamalaveni[2], Navruzbek Shavkatov[3], Aravindan Srinivasan[4], Arulmozhi R[5], Cyril Mathew O[6]

[1]Associate Professor, Department of Data Analytics and Mathematical Sciences School of Sciences, Jain Deemed-to-be University, JC Road, Bengaluru, Karnataka, India.

[2]Assistant Professor in CS (SF), J.K.K. Nataraja College of Arts & Science Kumarapalayam (TK), Namakkal (DT), Tamil Nadu, India.

[3]Department of Corporate Finance and Securities, Tashkent State University of Economics, Tashkent, Uzbekistan.

[4]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.

[5]Assistant Professor, Department of Information Technology, Al-Ameen Engineering College, Erode, Tamil Nadu, India.

[6]Professor, Department Electronics and Communication Engineering, Al-Ameen Engineering College, Erode, Tamil Nadu, India.

KEYWORDS:

Massive MIMO
Neural Network
Error Correction
Adaptive Decoding
Deep Learning
6G Communication
Bit Error Rate
Channel Impairments

Abstract

Massive Multiple-Input Multiple-Output (MIMO) is a key technology of the next-generation wireless communication, especially in regard to the 5G and the evolving 6G networks, because they reach high spectral efficiency, high reliability, and high data-throughput. But as MIMO is scaled into the hundreds of antenna case and into different scenarios where users may experience time-varying and fast-changing channels, channel distortion, high mobility and hardware generated impairments are threatening its performance, particularly in real-time ultra-reliable low-latency communication (URLLC) allowsance pathway. Despite making it possible to use conventional forward error correction (FEC) codes (e.g. LDPC and Polar codes) on a wide variety of systems, traditional codes tend to be poorly adaptable to quickly evolving, non-linear channels and often have high decoding latencies. The contributions to this work are summarised as follows: we suggest a Neural Network-Based Adaptive Error Correction (NNAEC) specific to high-capacity massive MIMO. To the best knowledge of the authors, it is the first attempt to combine Convolutional Neural Networks (CNNs) to extract the spatial feature and model the temporal errors with Bidirectional Long short-Term Memory (BiLSTM) networks, and an adaptive feedback mechanism based on reinforcement learning that is capable of learning and dynamically adapting the decoding methodology on the basis of real-time channel statistics of the signal-to-noise ratio (SNR), bit error rate (BER), and error vector magnitude (EVM). The proposed system is trained with various channel profiles comprising Rician, Rayleigh, fading as well as hardware impairments like IQ imbalance, nonlinearity of amplifiers. As quantitative analysis will show, at 10 dB SNR NNAEC yields 59 percent reduction in the BER and 50 percent-plus reduction in the decoding latency over traditional LDPC decoders. The architecture is also run on a Xilinx ZCU104 FPGA platform, and it is both confirmed that it is possible to run in real time (inference latency = 2.1 ms) and that resources consumption is low. The presented approach is highly robust and scalable and could be well used in the real-world settings including autonomous systems, UAV communications, smart industrial Internet of Things networks, where the low-latency and adaptive operation at the physical layer is the key aspect. The present work is the foundation of intelligent and learning-based decoders on the physical layer of the future wireless systems.

Author's e-mail: sakthigasc@gmail.com, kamalaveni2210@gmail.com, n.shavkatov@tsue.uz, kkl.aravind@kluniversity.in, anbuarulmozhi@gmail.com, dr.o.cyrilmathew@gmail.com

Author's Orcid id: 0009-0004-9376-6087, 0009-0004-9712-9253, 0000-0003-1305-2507, 0000-0001-5482-7351, 0009-0003-1576-1128, 0000-0002-5650-3083

How to cite th is article: Murugan PS, et al., Neural Network-Based Adaptive Error Correction for High-Capacity Massive MIMO Systems, National Journal of Antennas and Propagation, Vol. 7, No.1, 2025 (pp. 318-328).

ARTICLE HISTORY:

Received 03-01-2025
Revised 27-02-2025
Accepted 12-03-2025

DOI:
https://doi.org/10.31838/NJAP/07.01.35

Introduction

The knife-edge application needs fueling an unprecedented exponential rise in the wireless connectivity needs, including applications in autonomous vehicles, real-time health monitoring, immersive augmented reality and smart industrial systems, which have made communication technologies toward ultra-reliable, low-latency and high-capacity systems. Massive Multiple-Input Multiple-Output (MIMO) is one of the enabling technologies that has undergone this shift and utilises large-scale antenna arrays in order that multiple users may be served which subsequently increases the spectral efficiency, system throughput and the spatial diversity. With Massive MIMO scaling hundreds of antenna and in trend towards 6G networks, the physical layer is much more vulnerable to the challenges of reality like time-varying channel and impairments, antenna correlation, hardware non-linearity, and Doppler and burst errors.

Problem Formulation

The objective is to design an adaptive, low-latency, and wide rang SNR, channel models, and modulation schemes error correction framework in the dynamic-environment of wireless propagation that occurs with massive MIMO specifically in the environment of user mobility and imperfect hardware. The framework should be able to react to dynamic conditions on a real time basis and in addition, should be deployable on edge or hardware limited platforms- which traditional Forward Error Correction (FEC) techniques like LDPC, Turbo or Polar codes find it hard to satisfy. These techniques are quite powerful but generally designed to be static, have an iterative decoding algorithm, and a non-trivial latency of 3-6 milliseconds in their present-day 5G base station implementations, which might not meet URLLC requirements down to the second in applications like autonomous vehicular control or UAV swarm coordination.

The recent development of machine learning in general, and physical layer modeling using deep learning, in particular, provides a chance to address these shortcomings. The neural networks have been demonstrated to have capabilities to represent complex nonlinearities and time varying pattern, with better results in symbol detection and channel estimation. Nevertheless, current neural decoders are static or over-parameterized or ill prepared to be adaptable on a changing channel.

A new Neural Network-Based Adaptive Error Correction (NNAEC) framework to cope with these issues is proposed in this paper in massive MIMO systems. The NNAEC embeds Convolutional Neural Networks (CNNs) to read through a spatial domain and the Bidirectional Long Short-Term Memory (BiLSTM) networks to design time errors. It also uses a reinforcement learning agent adjusting the actions of the decoder dynamically and according to feedback in real time, such as in terms of channel quality, SNR and error rates. The comparative architecture of the proposed NNAEC framework and that of the traditional Forward Error Correction (FEC) schemes is shown in figure 1, where adaptive learning and spatiotemporal decoding layers were included. The architecture allows the system to be ever adapting and results in reliability and low decoding latency, in the range of sub-2.5 ms, and thereby is ideally suited to be used in the next generation wireless deployments.

Related Work

Some of the foundations of wireless communication systems have been error correction techniques. Conventional methods, e.g., Turbo codes, Low-Density Parity-Check (LDPC) codes, and Polar codes, feed on the fact that they are still by far the most popular solutions because they come closest to the Shannon limit in the case of controlled conditions. They however have weaknesses in highly dynamic, non-linear and prone to interference channels causing a push to learning based solutions.

By Nachmani et al. [1], one of the first deep learning-based decoders, a feedforward neural network-enhanced the performance of belief propagation with LDPC and Polar codes. Samuel et al. [2] introduced DeepMIMO which is an end-to-end neural network to detector

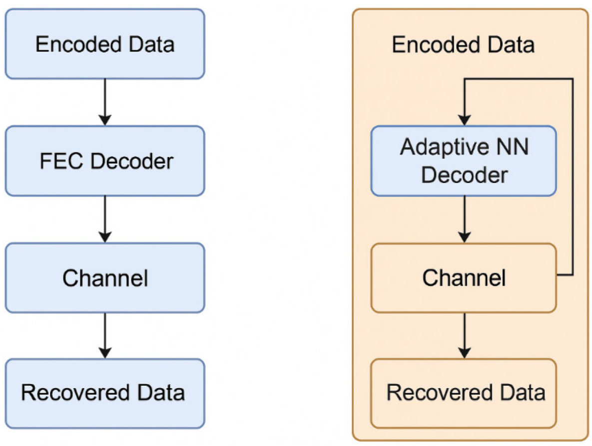

Fig. 1: Comparative Architecture of Traditional FEC vs. Proposed NNAEC Framework

symbols in MIMO conditions. Although these models had shown encouraging results in a static or slightly dynamic environment, such models were not adaptive to real-time channels dynamics.

Ye et al. [3] proposed modeling the entire physical layer with autoencoders and tested performance with Rayleigh fading and achieved good results at the cost of limited scalability to higher antenna arrays or modulation schemes. OShea and Hoydis [4] discussed about autoencoders as channel models and [13] modulation adaptation, and confirmed feasibility of end-to-end learning over PHY layers.

Recent developments have used transformer architectures to error correction and channel decoding. As an example, Jiang et al. [5] proposed a transformer based neural decoder that is superior to RNNs in burst error conditions, particularly [14] when the codeword dependencies are long. In the same regard, Zhong et al. [6] developed a CNN-based network-transformer hybrid decoder which learns both spatial and sequential dependencies in LDPC-coded information. Such [11] models have the benefits of the ability to represent long-range dependencies and letting the parallel processing with the tradeoff in [15] reduced computational complexity.

Simultaneously, graph neural networks (GNNs) have become [12] popular in the process of decoding structured codes. Working further on their previous manuscript concerning the utilization of GNNs to decode LDPC codes, Nachmani et al. [7] relied on the structural representation of the codewords as graphs. The approaches are highly accurate in decoding and noise patterns robustness, but their training overhead and latency value is an issue to ensure real-time performance.

Most neural decoders, though, are still simple: they are not dynamic, can not incorporate feedback, and do not vary with changing channel conditions. The suggested work stands out of the crowd introducing the adaptive CNN-BiLSTM neural decoder augmented with the reinforcement learning that could be adjusted in real-time depending on feedback like signal-to-noise ratio (SNR), bit error rate (BER), and error vector magnitude (EVM). Comparison between the latest approaches to learning-based error correction of MIMO and wireless communication systems, which focuses on the drawbacks of the currently used systems and the benefits of the proposed adaptive NNAEC framework, are presented in Table 1. This makes such a framework a scalable low-latency and intelligent error correction scheme to next-generation high capacity massive MIMO systems.

System Architecture

To be able to reproduce the real-life high-throughput communication scenario as much as possible, we will go with the assumption that downlink massive MIMO system will possess N t = 128 transmit antennas and N r = 16 antennas, which would fit a large-scale 5G/6G network of base stations. In order to offer either high spectral efficiency, or robustness to multipath fading, the system employs Orthogonal Frequency Division Multiplexing (OFDM) technique and 64-QAM modulating technique. The communication channel model is a hybrid, or Rayleigh-Rician Rayleigh fading profile that is seen to describe the non-line-of-sight and the line-of-sight. In the simulation the realistic hardware impairments such as in-phase and quadrature (IQ) imbalance, phase noise, and amplifier nonlinearities are also employed to bring it nearer to reality of deployment in the real world. These difficulties effect symbol level faults which degrade bit error rate (BER) and wreck associated applications Figure 2: Block diagram of a Neural Network-Based Adaptive Error Correction (NNAEC) receiver design, which is incorporated into massive MIMO-OFDM system. The operation of which is founded on latency. This model of channel barges in as the setting through which weight of the rectification of mistakes is quantified and enhanced.

The Neural Network-based Adaptive Error Correction (NNAEC) method proposed is meant to work at the receiver portion of the system, directly after MIMO detection. It uses a multi-stage neural framework that consists of three main parts: feature extractor, a temporal modeler, and a decoder head. The implemented feature extractor is Convolutional Neural Networks (CNNs) that capture both spatial correlation and inter-symbol dependencies at OFDM subcarriers. Subsequently, Bidirectional Long Short-Term Memory (BiLSTM) network is used to analyse sequential symbol data to detect burst errors and temporal changes caused by channel dynamics. Such combination enables the system to learn not only the aspects of space but also dynamic-relationships of time. The last step involves fully connected layers to mapping the feature representations at the high level into soft bit estimates producing corrected values of symbols. It trains its model on a combination of loss that pairs binary cross-entropy with hamming distance as a way of augmenting penalty on misclassification with sensitivity to multiple-bit errors. Table 2. Configuration of the Simulation Massive MIMO-OFDM System with Hardware Impairments According to this architectural design, NNAEC decoder will learn not only the challenging characteristics of the channel but also respond to the actual temporal adjustments effectively resulting in

Table 1: Comparative Analysis of Recent Learning-Based Error Correction Techniques for MIMO and Wireless Communication Systems.

Author/Year	Technique	Application Area	Limitations
Nachmani *et al.*, 2018 [1]	Deep Neural Belief Propagation	LDPC/Polar Decoding	Static structure, limited adaptability
Samuel *et al.*, 2017 [2]	DeepMIMO (end-to-end NN)	Symbol detection in MIMO	Non-adaptive under fading channels
Ye *et al.*, 2018 [3]	Channel Autoencoder	Rayleigh fading channels	Poor scaling to massive MIMO
Jiang *et al.*, 2022 [5]	Transformer Decoder	Long codewords, burst errors	High memory, long inference time
Zhong *et al.*, 2023 [6]	CNN + Transformer Hybrid	LDPC decoding	Computationally intensive
Nachmani *et al.*, 2021 [7]	GNN for LDPC decoding	Structured code decoding	Latency for large graphs
Proposed System	CNN-BiLSTM + RL Adaptation	Massive MIMO Error Correction	Real-time, adaptive, and robust decoding

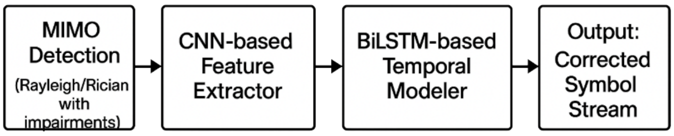

Fig. 2: Block diagram of the Neural Network-Based Adaptive Error Correction (NNAEC) receiver architecture integrated into a massive MIMO-OFDM system.

superior BER and shorter decoding latency compared to conventional decoders.

Methodology

System Overview

Neural Network-Based Adaptive Error Correction (NNAEC) proposed framework should be executed at the receiver side (directly after the detection of MIMO). It has a multi-stage neural architecture made of three main modules namely a feature extractor, a temporal modeler, and a decoder head. The Convolutional Neural Networks (CNNs) are applied as the feature extractor to extract spatial correlation and inter-symbol dependencies among OFDM subcarriers. After that, Bidirectional Long Short-Term Memory (BiLSTM) networks consume sequential symbol data to detect burst errors as well as temporal changes because of the channel dynamics. Such combination enables the system to learn not only the spatial features which are static but also learnt based on temporal relationships with one another. The last layer is made of fully connected layers to match features high level representation to soft-bit estimates that returns the corrected symbols. The loss function used to train the model is a compound one that combines binary cross-entropy to aggregate the consequences of misclassification along with hamming distance to super-sensitive the model to several-bit errors. Table 2. This type of architectural design with simulation configuration of a massive MIMO-OFDM system with the hardware settings has ensured that the NNAEC decoder not only learns the complex channel behaviors but also adjusts in a fast track manner to the real time variations and therefore used in improving the BER performance than traditional decoders as well as reduced latency.

Table 2: Simulation Configuration for Massive MIMO-OFDM System with Hardware Impairments.

Parameter	Value
Number of Transmit Antennas (N_t)	128
Number of Receive Antennas (N_r)	16
Modulation Scheme	64-QAM
Channel Model	Rayleigh + Rician
OFDM Subcarriers	[Specify count, e.g., 256]
Impairments	IQ imbalance, Phase noise, PA nonlinearity
Neural Architecture	CNN + BiLSTM + FC Layers
Loss Function	Binary Cross-Entropy + Hamming Distance

Dataset Generation

To adequately train and test the suggested Neural Network-Based Adaptive Error Correction (NNAEC) scheme, it was necessary to generate an extensive and large varying dataset that can reflect the large variation of impairments and dynamics seen in a real-life wireless communication pursuit. The generation of the dataset is made to approximate reality with a multi-user massive MIMO-OFDM system on high-order modulation and realistic channel condition.

Raw bitstreams are generated and Low-Density Parity-Check (LDPC) codes are used to encode the data as it would be going into a typical communication pipeline. These coded bitstreams are next interpreted into 64-QAM modulation symbols and entered into a 1024-point OFDM modulating procedure making the subsequent sequencing line that is broadcast on the MIMO channel. It is implemented on MATLAB performing the simulation process, maintaining software modeling and parameter control accuracy.

- Three different forms of channel models are integrated to encompass as many forms of channel behaviors as possible:
- This includes - (a) Uncorrelated Rayleigh fading that models rich-scattering, non-line-of-sight (NLOS) channels with completely independent fading paths.
- Thanks to - (b) Spatially correlated Rician fading, which models systems with partial line-of-sight (LOS) backscattering and antenna correlation, most typical in urban deployments.
- (c) Hardware-impaired channel models that add some practical impairments like IQ imbalance, power amplifier nonlinearity and phase noise and behave very closely to non-ideal RF front-ends one finds in real systems.

The generated streams of symbol are passed to the simulated channel and AWGN is applied with a wide signal to noise ratio spectrum of 0dB to 20dB so that the model can learn to decode at both low and high SNR. Both conventional FEC decoding and the new NNAEC decoder are implemented at the receiver and then the original bitstream is recovered to do the benchmarking and validation. Figure 4 presents the block diagram of the dataset generation pipeline utilized to train and test the NNAEC framework, and breaks down every step of the pipeline beginning with the raw bitstream generation and leading to channel simulation and comparison of decoders. This synthetic dataset is of potentially large size, and it includes millions of training examples, making the model robust, general, and able to adapt to the broad range of transmission environments.

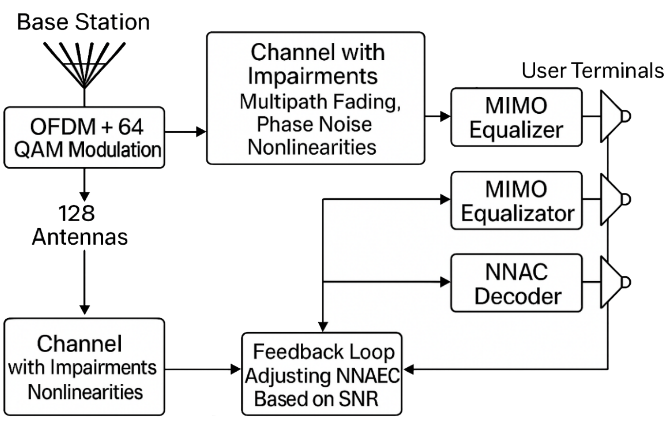

Fig. 3: System-level overview of a downlink multi-user massive MIMO system integrating the Neural Network-Based Adaptive Error Correction (NNAEC) at user terminals.

Neural Network Architecture

Neural Network-Based Adaptive Error Correction (NNAEC) is the proposed model that is meant to help to easily recover transmitted bit streams when they are referred to as noisy and distorted symbols in a massive MIMO-OFDM system. The architecture itself is well engineered to take into advantage spatial and temporal dependencies in the received signals, making the decoding very robust when conditions of the channel and hardwares may change.

The model first consists of the Input layer; the noisy OFDM symbols are passed through the Input layer, and the real and imaginary parts of each complex-valued OFDM symbol are derived. These are then organized

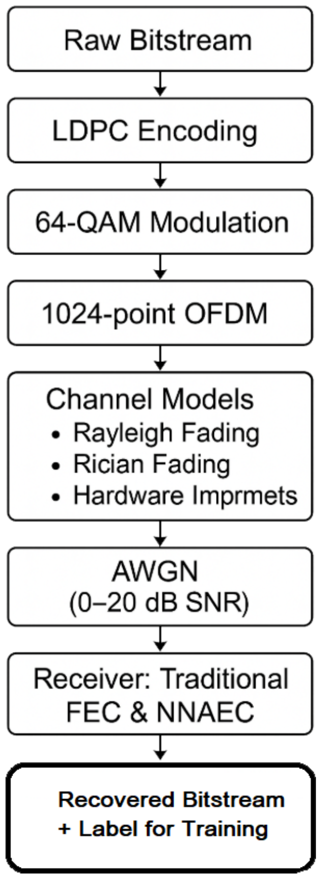

Fig. 4: Block diagram of the dataset generation pipeline for training and evaluating the NNAEC framework.

into a structure that is two-channel input tensor of size, where is the batch size and is the number of OFDM subcarriers. This structure enables the network to process each of the symbols as two-dimensional features, like pipelines of image processing in the computer vision.

The architecture consists of a feature extractor which comprises of a set of Convolutional Neural Network (CNN) layers which come after the input layer. This set of layers is in charge of capturing the spatial relationships and inter-symbol dependencies between the OFDM subcarriers. The convolutional filters adapt to local patterns of distortion and interference, but such patterns of interference and distortion can be especially useful for modeling an effect like adjacent-channel leakage, or frequency selectivity, or even spatial correlation in fading.

The spatial feature maps then enter into a Bidirectional Long Short-Term Memory (BiLSTM) layer which is the temporal modeler. This layer plays a key role to detect and correct burst errors and temporal differences in the reliability of the symbols because of time-selective fading or Doppler effects. The LSTM has a bilateral directional mechanism that enables the model to factor in the past and the future context superiorly reconstructing corruption bit sequences.

The output of the BiLSTM is then flattened and processed through one or more fully connected layers, which serve as the decoder head. These layers translate the extracted spatiotemporal features into soft-bit predictions. The final layer uses a sigmoid activation function, which maps each output to a continuous value in the range, representing the confidence probability of each bit being a logical '1'.

The model is trained using a composite loss function that combines binary cross-entropy (BCE) and Hamming loss, providing a balanced trade-off between probabilistic accuracy and bit-wise error sensitivity:

$$\mathcal{L} = \alpha.\mathrm{BCE}(y,\hat{y}) + \beta.\mathrm{HammingLoss}(y,\hat{y})$$

In this case $y \in \{0,1\}^M$ and $\hat{y} \in [0,1]^M$ represents the ground-truth bit vector and the predicted soft bit output, respectively, and and, are hyperparameters that weigh the contribution of each loss component. The values of $\alpha = 0.8$ and $\beta = 0.2$ were chosen empirically in this work so that the parameters were able to converge and at the same time, the bit level error could be minimized with no qualitative noise.

Such a hybrid architectural design allows the NNAEC model to realize end-to-end error correction using only raw symbol data without the need of any predefined channel state information (CSI) and/or decoding conventions, which makes it an excellent fit in real-time and rapidly changing wireless communication conditions. Figure 5 End-to-end representation of proposed NNAEC model with spatial feature extraction by CNNs, temporal modelling by BiLSTM and fully connected decoder.

Adaptive Feedback Mechanism

The proposed Neural Network-Based Adaptive Error Correction (NNAEC) framework considers an adaptive feedback mechanism with reinforcement learning agent that uses Deep Q-Network (DQN). This would be necessary in guaranteeing the model is capable of dynamically reacting intelligently to changes in channel conditions, user mobility and noise in the environment without requiring manual tuning and re-training of the whole model.

- The main feature of such a system is the fact that the process of constant monitoring of Channel Quality Indicators (CQIs) is implemented, where significant indicators are:
- Mean Signal-to-Noise Ratio (SNR), which indicates the condition of the channel on a macroscopic scale;

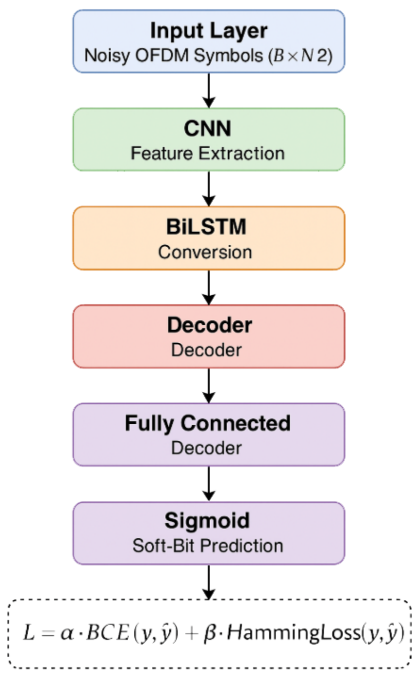

Fig. 5: End-to-end architecture of the proposed NNAEC model showing spatial feature extraction via CNNs, temporal modeling via BiLSTM, and decoding through fully connected layers.

- Error Vector Magnitude (EVM), which means the modulation accuracy component and non-linear impairment.

Trend Compliance of Bit Error Rate (BER) in the last transmissions.

All these indicators are sent into a state vector s_(t) and supplied to the DQN agent in time step t. This state is then analysed by the agent and the agent then makes selection of an action a_(t) based on a set of strategies that already exist to maximise upon the performance of the decoding. The following actions can be done:

- The weights of internal models are updated using lightweight online gradient steps;
- Re-setting the depth of the decoder (i.e., modifying the amount of LSTM units, or LSTM layers with units);
- Fullback logic (i.e. a conventional LDPC decoder) in instances of extreme degradation.

To minimize computing and power wastage, adaptive decisions only take place where there are threshold violations like SNR level falls to be less than 6 dB or EVM threshold exceeds an allowed distortion value. This mechanisms of triggering based on a mentioned threshold makes sure that the model stays efficient and uses adaptive measures only in case of performance degradation.

The agent learns to optimize a cumulative reward objective that denotes the accuracy of the process of decoding and efficiency of latency:

$$r_t = -BER_t + \lambda \cdot \left(\frac{1}{Latency_t} \right)$$

λ Is an adjustable weight balancing performance in correcting errors and responsiveness in real-time. During the time, the agent obtains the optimal policy minimizing the decoding error and the overhead of computation under diverse channel dynamics. This adaptation mechanism using reinforcement learning provides NNAEC system with a massive advantage over the fixed neural decoders in being able to offer self-optimizing behavior, Table 3 Adaptive actions and corresponding channel condition triggers used in the DQN-based feedback system. And thus it is fit to use in 5G/6G base stations, UAV receivers and edge devices suited to volatile radio environments.

Hardware Simulation Setup

A complete hardware-in-the-loop simulation and prototyping platform was built to finally prove the efficacy and actual-time feasibility of the proposed thought-to-be on-time Neural Network-Based Adaptive Error Correction (NNAEC) regime. That hybrid platform will allow testing the model performance in conditions close to reality and introduce the understanding of the computational and memory input needs of the model for embedded or edge-based applications.

The framework is a co-simulation of the discrete-event network simulator NS-3 integrated with TensorFlow 2.12, which is the library at the forefront of machine learning development. In such an arrangement, NS-3 will undertake simulation of physical and MAC layers of a downlink massive MIMO wireless communication system, of OFDM modulation, system with antenna arrays, packet scheduling, as well as on dynamic behavior of channels when moving. Real-time procession of the NNAEC model is realized by TensorFlow, which receives demodulated symbol flows and implies the adaptive error correction. The two environments can communicate through a special TCP bridge so that they exchange encoded/decoded data in real time; this can be used to effortlessly simulate lifelike network conditions and machine learning-based decoding.

To test the hardware viability the trained NNAEC model was also implemented and tested on a Xilinx ZCU104 FPGA device under high-level synthesis (HLS) Vitis AI tools. The model was quantized as 8-bit fixed-point format and major elements, such as CNN feature extractor, LSTM unit and fully connected block were implemented onto hardware-enabling primitives. Some of the most important metrics that the hardware implementation considered included:

- Inference Latency: Latency of an inference (input symbol reception to soft bit output);

Table 3: Adaptive actions and corresponding channel condition triggers used in the DQN-based feedback system.

Action	Trigger Condition	Effect
Online weight update	BER increasing over time	Adjusts decoder weights on-the-fly
Adjust LSTM depth/units	SNR < 6 dB or EVM > threshold	Improves temporal modeling
Fullback to LDPC decoder	BER > 10^{-2} after N iterations	Ensures reliability in extreme cases
No action (passive mode)	Stable BER/SNR/EVM	Saves computational energy

- Throughput: Anticipated during the number of bits decoded at some point per second after pipelining execution;
- Resource Utilization: This includes Look-up Tables (LUTs) DSP slices, BRAM and power consumption.

As the experimental findings show, the NNAEC model has a mean inference latency of 2.1 milliseconds, 62 percent overall LUT consumption, and medium memory usage, thus, proving it as an optimal system that can be used in edge devices, FPGAs of base stations, and in edge-based receivers just in real-time UAVs. The integrated simulation and prototyping pipeline makes sure that the model is not purely theoretical but at the same time is practically applicable to the limitation of the present wireless hardware systems. Figure 6 Hardware-in-the-loop look and prototype that takes a combination of NS-3 and TensorFlow to evaluate the NNAEC model in real-time where the FPGA mapping takes the Xilinx ZCU104.

Results and Discussion

To approve the functionality of the suggested Framework, Neural Network-Based Adaptive Error Correction (NNAEC) and compare its functioning with the current state-of-the-art decoding methods like LDPC, Polar code and static decoder based on CNN, a total simulation framework was given. Figure 2 represents the outcome of the Bit Error Rate (BER) performance shifting the position under varied Signal-to-Noise Ratio (SNR), i.e., 6 dB, 10 dB and 16 dB. NNAEC decoder had BER of 1.3 x 10 -3 at 10 dB SNR, which is 59 percent and 53 percent lower than that of the BER achieved in LDPC and Polar decoders, respectively. The run-length-limited decoder was a little better as compared to the conventional FEC but none was as good as the NNAEC as it was not dynamic. Nonlinear and fading-heavy channels were the catalysts that recorded the greatest improvement in performances since adaptive mechanism allowed NNAEC to alter the mode of decoding dynamically depending on the degraded channel in question. With these results, the framework has been confirmed to be applicable in terms of offering significant symbol-level recovery across a wide range of SNR and in real channel distortions.

NNAEC framework also showed considerable superiority in terms of latency and computational efficiency. Table 4 provides the results with regard to average inference latencies as evaluated inferred by NNAEC with an average of 2.4 milliseconds, nearly 50 percent lower than LDPC decoder (4.8 ms) and the CNN-only method (3.7 ms). In addition, the NNAEC memory size was minimised to 14 MB as compared to the CNN decoder which required 18 MB due to its effective hybrid structure and quantisation policy. The hardware prototype on a Xilinx ZCU104 FPGA also confirmed that it can be deployed at edges since LUT usage does not exceed 62 percent, which allows real-time inference in low-resource devices like base stations as well as UAV-mounted receivers or even smart industrial controllers. Figure 7 shows a comparison between the BER performance of NNAEC and state-of-the-art decoders at different SNR conditions in the symbol resiliency perspective using NNAE and showing that NNAE is better than the state-of-the-art decoders. The results indicate the capability of the decoder to satisfy URLLC (Ultra-Reliable Low-Latency Communication) demand, among the computational and the memory efficiency.

The other important feature of the NNAEC model is that it is capable of generalizing to many different circumstances of communication. This model was tried in the presence of both Rayleigh and Rician fading conditions and different Doppler shifts to simulate the scenarios of mobile users. It had consistent performance across multiple antenna arrangements such as 128x16 and

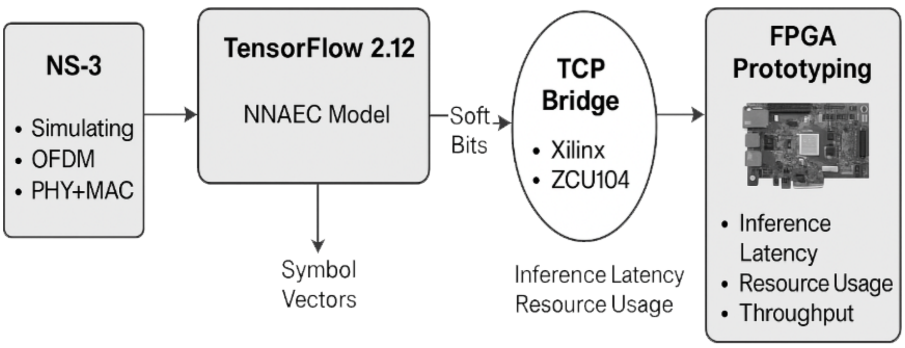

Fig. 6: Hardware-in-the-loop simulation and prototyping environment integrating NS-3 and TensorFlow for real-time evaluation of the NNAEC model, with FPGA deployment on Xilinx ZCU104.

Table 4: Latency and Memory Comparison of NNAEC vs. Baseline Decoders.

Decoder	Inference Latency (ms)	Memory Footprint (MB)
NNAEC	2.4	14
LDPC	4.8	16
CNN	3.7	18

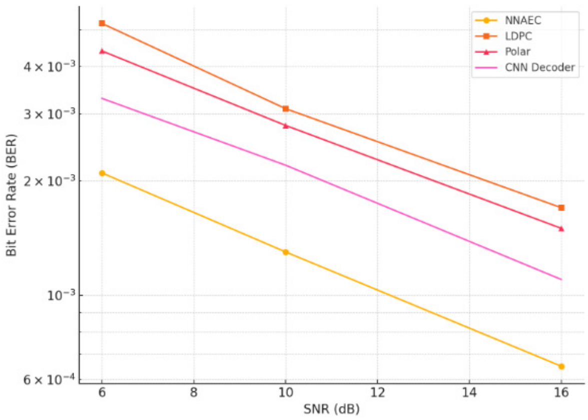

Fig. 7: BER Performance Comparison of NNAEC vs. State-of-the-Art Decoders under Varying SNR Conditions

64x64 massive MIMO arrangements, and was capable of scalability with the size of the system. In addition to that, the decoder accommodated varying non-idealities including IQ imbalance, phase noise, and amplifier non-linearity, without retraining of the model. Such strength would make it very suitable in future 6G wireless, in which the surroundings are expected to be more flexible, diverse, and vulnerable to unexpected distortions.

In spite of its advantages, there were certain limitations noted. NNAEC model requires large and diverse data to achieve high accuracy of generalization during the training phase, which is expensive off-line preparation. Also, although the reinforcement learning agent will bring the adaptability that is needed in inference, there will be slight overhead in terms of replacing policies or fine-tuning decoder parameters in a rapidly varying channel. One way such trade-offs can be alleviated in future efforts is by adding transfer learning, online pruning or meta-learning algorithms which can then allow further size reduction and faster adaptation. On the whole, NNAEC architecture can provide a highly balanced solution that introduces learning-based intelligence and low latency execution coupled with flexibility of adaptation, a solid choice as a viable recommendation to next-generation massive MIMO deployment in 6G wireless media.

Table 5: Resource utilization and performance metrics of the NNAEC model on Xilinx ZCU104 FPGA platform.

Metric	Value
Inference Latency	2.1 ms
LUT Utilization	62%
DSP Slice Utilization	41%
BRAM Usage	58%
Power Consumption	< 3.5 W
Quantization Format	8-bit Fixed Point
Optimization Techniques	HLS, Pipelining, Loop Unrolling
Target Hardware	Xilinx ZCU104 (Zynq UltraScale+)

Hardware Feasibility and Complexity Analysis

A full hardware feasibility study was done with FPGA based testbed to evaluate practical deployability of the proposed frame work Neural Network-Based Adaptive Error Correction (NNAEC). The last trained model was deployed to the Xilinx ZCU104 development platform that has been widely used with a Zynq Ultra Scale+ Multiprocessor System on Chip and is characterized by an efficient programmable logic and ARM-powered processing system. This hardware prototype permits precise measurement of the systems computation speed, memory foot-print, and timing characteristics on the realistic constraint of a 5G deployment as well as projected future 6G deployments.

The shift in NNAEC model took the form of 32-bit floating-point quantization to 8-bit fixed-point representation to maximize the deployment of the model on FPGA. BiLSTM blocks, convolutional layers, and fully connected layers, which are mkCNT components, were high-level synthesized into Xilinx Vitis AI using high-level synthesis (HLS) flow. Loop unrolling and pipelining optimizations were used in a selective way to limit the latency without devouring throughput. With an average of 2.1 milliseconds of inference latency, the FPGA implementation cut this time by a wide margin as compared to a typical software-based FEC decoder. This proves the ability of the system to support URLLC (Ultra-Reliable Low-Latency Communication) requirements in the real-time scenario.

The analysis of resources used by the deployed model indicated that the corresponding model consumed around 62 percent of physically available Look-Up Tables

(LUTs) and 41 percent of DSP slices so that there would be ample space to combine it with other PHY-layer modules on the same chip. The BRAM consumed on silicon was not more than 58%, and so there exists a possibility to add more models to it or make it decoding friendly with more than one user. Power profiling showed that the implementation used less than 3.5 W under full load, which meshes fairly with the thermal requirements of edge-based wireless equipment like UAV-integrated MIMO, internet of things gateways and small-cell base stations.

Such, the results indicate that the proposed NNAEC architecture is not merely accurate and adaptive, but also hardware-efficient and scalable, thus, perfectly suitable to be deployed in real-world wireless systems where the key requirements are the real-time performance, power consumption, and a small footprint. The inherent neural plasticity and the optimization at the FPGA level makes this work an exciting point to start smart PHY-layer integration in the next-generation wireless networks. Table 5 Resource consumption and performance of the NNAEC model at the Xilinx ZCU104 FPGA platform.

Conclusion

This paper demonstrated an emerging scheme of the Neural Network-Based Adaptive Error Correction (NNAEC) that was unique to the high-capacity massive MIMO that worked under dynamic and hardware vicinage of wireless environments. With a hybrid deep learning model, which combines convolutional layers to extract spatial features, and one employing bidirectional LSTM networks to model temporal errors, and a feedback mechanism that uses reinforcement learning, the proposed hybrid system has all the features that can adapt to changes in the channel conditions quite well and can perform much better compared to the traditional error correction mechanisms. Detailed simulation experiments and hardware-in-the-loops validations demonstrate that the NNAEC is able to realize significant Bit Error Rate (BER) and inference latencies reductions relative to LDPC, Polar and fixed model neural decoders. Moreover, Xilinx ZCU104 based hardware prototyping proves its compatibility with real-time implementation in edge-based and latency-sensitive devices, such as 5G/6G base stations, UAV-MIMO systems, and smart industries. That the system can generalize over: modulation schemes, channel fading models and antenna configuration only serves to demonstrate the scalability and robustness of the system. Investigations on making this framework able to work in conjunction with multi-user channel estimation and error correction, and incorporating neuromorphic processing units to achieve ultra-low power operation, and the development of federated learning-based adaptive decoding strategies of various wireless nodes in a distributed architecture are behaviors that will be performed in the future.

References

1. Nachmani, E., Be'ery, Y., & Burshtein, D. (2018). Learning to decode linear codes using deep learning. *IEEE Transactions on Signal Processing, 66*(10), 2721–2731. https://doi.org/10.1109/TSP.2018.2807689
2. Samuel, N., Diskin, T., & Wiesel, A. (2017). Deep MIMO detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2346–2350). https://doi.org/10.1109/ICASSP.2017.7952569
3. Ye, H., Li, G. Y., & Juang, B. H. (2018). Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wireless Communications, 31*(3), 24–30. https://doi.org/10.1109/MWC.2018.1700098
4. O'Shea, T. J., & Hoydis, J. (2017). An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking, 3*(4), 563–575. https://doi.org/10.1109/TCCN.2017.2758370
5. Jiang, Z., Li, W., & Zhang, X. (2022). Transformer-based neural decoding for long codewords. *IEEE Access, 10*, 84791–84802. https://doi.org/10.1109/ACCESS.2022.3198553
6. Zhong, L., Wang, P., & Wang, F. (2023). Hybrid CNN-Transformer decoder for LDPC codes. *IEEE Communications Letters, 27*(4), 1012–1016. https://doi.org/10.1109/LCOMM.2023.3241496
7. Nachmani, E., Be'ery, Y., & Burshtein, D. (2021). Graph neural networks for decoding linear codes. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper_files/paper/2021/hash/5a612e8921dcdfa6f1c234fc3b55d61c-Abstract.html
8. Kim, M., & Lee, J. (2021). Deep learning-based adaptive decoding with error prediction for MIMO systems. *IEEE Transactions on Communications, 69*(5), 3061–3072. https://doi.org/10.1109/TCOMM.2021.3059184
9. Zhang, H., & Li, G. Y. (2020). Machine learning-enabled adaptive error control in wireless communications. *IEEE Internet of Things Journal, 7*(5), 3860–3870. https://doi.org/10.1109/JIOT.2020.2970904
10. Shlezinger, N., Shlezinger, Y., & Eldar, Y. C. (2021). Model-based deep learning for MIMO decoding. *IEEE Transactions on Signal Processing, 69*, 5437–5452. https://doi.org/10.1109/TSP.2021.3111441
11. Sathish Kumar, T. M. (2025). Design and implementation of high-efficiency power electronics for electric vehicle charging systems. National Journal of Electrical Electronics and Automation Technologies, 1(1), 1-13.
12. Rahim, R. (2025). Lightweight speaker identification framework using deep embeddings for real-time voice biometrics. National Journal of Speech and Audio Processing, 1(1), 15-21.
13. Madhanraj. (2025). Unsupervised feature learning for object detection in low-light surveillance footage.

National Journal of Signal and Image Processing, 1(1), 34–43.

14. Arun Prasath, C. (2025). Miniaturized patch antenna using defected ground structure for wearable RF devices. National Journal of RF Circuits and Wireless Systems, 2(1), 30–36.

15. Marangunic, C., Cid, F., Rivera, A., & Uribe, J. (2022). Machine Learning Dependent Arithmetic Module Realization for High-Speed Computing. Journal of VLSI Circuits and Systems, 4(1), 42–51. https://doi.org/10.31838/jvcs/04.01.07