

# Comparative Performance Analysis of Deep Learning-Based Channel Estimation for 5G Massive MIMO Systems: DNN, CNN, and LSTM Architecture

Rajarajan P

Department of Electronics and Communication Engineering,  
Hindustan College of Engineering and Technology  
Coimbatore, 641032, Tamil Nadu, India.

**Abstract** - Acquiring accurate Channel State Information (CSI) is arguably the most fundamental challenge in making 5G Massive MIMO live up to its spectral efficiency promise. Classical approaches—Least Squares (LS) and Linear Minimum Mean Square Error (LMMSE)—have well-understood behaviour, but they struggle to simultaneously satisfy the accuracy, latency, and robustness demands of dense urban 5G deployments where channels are non-linear and fast-fading. Deep Learning (DL) has emerged as a compelling way forward, yet the literature tends to evaluate each neural architecture in isolation, making it hard to draw practical conclusions about which one to actually deploy. This paper addresses that gap. We carry out the first unified, side-by-side experimental comparison of three representative DL architectures—Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks—trained and tested under identical conditions using the Robust 5G Dataset. We assess each model across Normalized Mean Square Error (NMSE) over a 35 dB SNR range, convergence speed, parameter count, and real-time inference latency. The CNN stands out as the best overall option, hitting a test NMSE of  $-22.15$  dB with 72.4 million parameters. The LSTM demands  $4.8\times$  more parameters for only a marginal accuracy gain. The DNN, despite being the lightest model at just 9.2 M parameters, still delivers a solid  $-18.42$  dB with sub-3 ms inference—a strong baseline by any measure. Taken together, these results offer engineers a practical decision guide for choosing the right DL estimator in real-time 5G deployments.

**Keywords** - 5G Networks, Deep Learning, Channel Estimation, Massive MIMO, OFDM, Convolutional Neural Network

## I. INTRODUCTION

The global rollout of 5G New Radio (NR) has fundamentally changed what we expect from wireless physical layers. Massive MIMO—systems with tens to hundreds of antennas simultaneously serving many users—sits at the heart of 5G's capacity story (Gao et al. 2016; Björnson et al. 2016). But that story only holds if the base station (BS) knows the channel accurately, in real time, for every user. Without precise Channel State Information (CSI), the beamforming gains that justify deploying such large antenna arrays simply evaporate, and the system falls back to single-antenna performance (Alkhateeb et al. 2014).

In OFDM-based 5G, channel estimation has traditionally relied on pilots: known reference symbols scattered across the

time-frequency resource grid, from which the receiver infers the channel everywhere else. The simplest approach—Least Squares (LS)—works without any prior knowledge of channel statistics, but it is fundamentally noise-limited, with accuracy degrading roughly as  $1/\text{SNR}$ . The Linear Minimum Mean Square Error (LMMSE) estimator does much better by weighting pilot observations against the channel covariance matrix. The catch is that computing and inverting that covariance matrix costs  $\mathcal{O}(N_p^3)$  operations in the number of pilots  $N_p$ , which becomes prohibitively slow when you have thousands of antenna elements (Soltani et al. 2019).

Deep learning has offered a genuinely different way of thinking about this problem (Jiang et al. 2019). Instead of imposing a hand-crafted statistical model, a neural network learns the relationship between noisy pilot observations and the true channel directly from data. Non-linear fading, inter-carrier interference, hardware imperfections—things that rigid parametric models handle poorly—end up being absorbed into the network's weights (Ye et al. 2018). Three architectural families have attracted particular attention:

1. **Deep Neural Networks (DNNs):** Fully-connected feed-forward networks that take the flattened pilot observation as input and directly regress the full channel response. They are simple and fast, though they do not explicitly exploit spatial structure.
2. **Convolutional Neural Networks (CNNs):** 1D convolutional layers naturally capture correlations between neighbouring subcarriers, effectively performing learned interpolation across the frequency axis (He et al. 2020).
3. **Long Short-Term Memory networks (LSTMs):** Recurrent architectures that treat the sequence of subcarrier observations as a time series, capturing long-range frequency correlations and temporal variation across OFDM symbols (Wen et al. 2018).

Despite the wealth of work on each of these individually, controlled comparisons that pit all three against each other under the same conditions are rare. A base-station engineer trying to choose between a DNN and an LSTM is largely left guessing—and that is the gap this work aims to close. A DNN may be perfectly adequate for a low-mobility indoor hotspot; an LSTM might be worth the overhead in a vehicular scenario. But making that call requires a unified empirical

study, not three separate benchmarks with different datasets and evaluation protocols.

**Contributions.** This paper makes the following specific contributions:

- A unified benchmark of DNN, CNN, and LSTM for 5G MIMO channel estimation on the Robust\_5G\_Dataset, with identical training and evaluation protocols for all three models.
- A multi-dimensional comparison covering estimation accuracy (NMSE), training convergence, model complexity (parameter count and FLOP estimate), and real-time inference latency.
- An ablation study on pilot density, showing how the gap between structured architectures (CNN/LSTM) and DNN widens as pilots become sparser.
- Scenario-specific deployment guidelines that map each architecture to concrete 5G use cases—indoor, urban macro, and vehicular.

The paper is organised as follows. Section 2 reviews related work. Section 3 sets up the 5G MIMO system model. Section 4 describes the three architectures and training protocol. Section 5 covers the experimental setup and dataset. Section 6 presents and analyses the results. Section 7 discusses deployment implications, and Section 8 concludes.

## II. RELATED WORK

### A. Classical Channel Estimation

Pilot-assisted channel estimation has been studied for decades, and the LS and LMMSE estimators (Ye et al. 2018) remain the standard references against which newer methods are measured. Early interpolation schemes—linear, spline, and DFT-based—helped reduce pilot overhead by inferring channel values between pilot positions, but their accuracy was always tied to how well the underlying channel model matched reality. Blind and semi-blind methods pushed further by eliminating pilots entirely, though at the cost of high computational complexity and poor behaviour in rapidly time-varying channels.

### B. DNN-Based Estimation

The landmark work of Ye et al. (Ye et al. 2018) showed that a feed-forward network trained on simulated OFDM data could beat LS and come close to MMSE performance in static channels. Later work extended DNNs to multi-antenna systems and introduced data augmentation and domain adaptation to bridge the inevitable gap between simulation and real channels.

### C. Convolutional Approaches

Soltani et al. (Soltani et al. 2019) took a different angle, framing channel estimation as an image super-resolution problem and exploiting the spatial structure of the pilot grid through deep residual CNNs. Their approach achieved near-MMSE performance at a fraction of the cost of explicit covariance inversion. He et al. (He et al. 2020) extended this idea to beamspace massive MIMO, demonstrating that locally-connected convolutional layers effectively capture the angular-domain sparsity typical of millimetre-wave channels.

### D. Recurrent and Sequential Architectures

The sequential nature of OFDM subcarriers naturally motivates recurrent models. LSTM-based estimators have demonstrated a clear edge over CNNs in high-Doppler scenarios, where correlations span multiple OFDM symbols rather than just neighbouring subcarriers. The trade-off is significant, though: LSTM networks carry a heavy parameter overhead—four coupled gate matrices per layer—which limits their appeal in power-constrained edge deployments.

### E. Where This Work Fits

Each of the works above advances the state of the art, but they do so by benchmarking a proposed architecture either against LS/MMSE baselines or a single alternative DL model. To our knowledge, no prior study conducts a controlled three-way DNN vs. CNN vs. LSTM comparison under matched conditions across the full set of metrics we consider here: NMSE, inference latency, convergence behaviour, and pilot density sensitivity. Filling that gap is the central motivation for this work.

## III. SYSTEM MODEL AND PROBLEM FORMULATION

### A. 5G MIMO-OFDM Signal Model

We consider a 5G NR downlink in which a BS with  $M$  antennas communicate with a single-antenna UE over an OFDM system with  $N$  subcarriers and cyclic prefix length  $N_{cp}$ . After cyclic prefix removal and DFT processing, the received signal in the frequency domain is:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{N}$$

where  $\mathbf{Y} \in \mathbb{C}^{M \times N}$  is the received signal matrix,  $\mathbf{H} \in \mathbb{C}^{M \times N}$  is the frequency-domain channel matrix whose  $(m, n)$ -th entry represents the channel gain between the  $m$ -th BS antenna and the UE on subcarrier  $n$ ,  $\mathbf{X} \in \mathbb{C}^{N \times N}$  is a diagonal matrix of transmitted pilot symbols on the pilot subcarrier set  $\mathcal{P} \subset \{1, \dots, N\}$ , and  $\mathbf{N} \in \mathbb{C}^{M \times N}$  is additive white Gaussian noise with  $\text{vec}(\mathbf{N}) \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I})$ .

The instantaneous SNR is

$$\text{SNR} = \frac{\mathbb{E}[\|\mathbf{H}\mathbf{X}\|_F^2]}{\mathbb{E}[\|\mathbf{N}\|_F^2]} = \frac{P_t}{\sigma_n^2}$$

where  $P_t$  is the average transmit power. The channel evolves according to a Jakes' Doppler spectrum with maximum Doppler frequency  $f_D = v f_c / c$ , where  $f_c = 3.5$  GHz is the carrier frequency and  $v$  is the UE velocity.

### B. Traditional Pilot-Assisted Estimators

#### 1) Least Squares (LS) Estimator

The LS estimator needs no prior channel statistics—it simply minimises the squared residual between received and expected pilot observations, subcarrier by subcarrier:

$$\hat{\mathbf{H}}_{LS} = \mathbf{Y}_{\mathcal{P}} \mathbf{X}_{\mathcal{P}}^{-1} = \mathbf{Y}_{\mathcal{P}} \mathbf{X}_{\mathcal{P}}^H (\mathbf{X}_{\mathcal{P}} \mathbf{X}_{\mathcal{P}}^H)^{-1}$$

The estimator is unbiased, but its variance is inversely proportional to SNR. Off-pilot estimates require interpolation, which introduces additional error near frequency boundaries.

#### 2) LMMSE Estimator

The LMMSE estimator improves upon LS by folding in second-order channel statistics:

$$\hat{\mathbf{H}}_{LMMSE} = \mathbf{R}_H (\mathbf{R}_H + \sigma_n^2 (\mathbf{X}_p \mathbf{X}_p^H)^{-1})^{-1} \hat{\mathbf{H}}_{LS}$$

where  $\mathbf{R}_H = \mathbb{E}[\mathbf{H}\mathbf{H}^H]$  is the channel covariance matrix. LMMSE is noticeably better than LS at low SNR, but the  $\mathcal{O}(N_p^3)$  matrix inversion it entails is expensive—and that cost grows quickly as antenna arrays scale up.

### C. Deep Learning Estimation Problem

The goal is to learn a mapping  $f_\theta: \hat{\mathbf{H}}_{LS} \mapsto \hat{\mathbf{H}}_{DL}$  that refines the noisy LS estimate into an accurate channel estimate. The network weights  $\theta$  are chosen to minimise mean square error over the training set:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(X,H) \in \mathcal{D}} \|f_\theta(\hat{\mathbf{H}}_{LS}) - H\|_F^2$$

over a training set  $\mathcal{D}$  of paired (noisy LS estimate, true channel) samples.

### D. Performance Metric: Normalised Mean Square Error

We evaluate all models using the Normalised Mean Square Error (NMSE), which gives a scale-invariant measure of estimation fidelity:

$$\text{NMSE} = \mathbb{E} \left[ \frac{\|\hat{\mathbf{H}} - \mathbf{H}\|_F^2}{\|\mathbf{H}\|_F^2} \right]$$

All figures report NMSE in decibels:  $\text{NMSE}_{\text{dB}} = 10 \log_{10}(\text{NMSE})$ , with lower values indicating better accuracy.

## IV. PROPOSED DEEP LEARNING METHODOLOGY

### A. Dataset and Preprocessing

The Robust\_5G\_Dataset contains 1 500 paired channel samples. Each sample consists of a noisy LS pilot observation  $X \in \mathbb{R}^{2 \times 312}$  (real and imaginary parts stacked along the channel axis, across 312 pilot subcarriers) along with the corresponding ground-truth channel response  $Y \in \mathbb{R}^{2 \times 17\,472}$ . The data was collected under realistic 5G NR conditions, including multi-path Rayleigh fading, temporal coherence, and Doppler effects.

Two preprocessing steps were applied before training:

1. **Output Dimension Reduction.** The target tensor  $Y$  was downsampled by a factor of 4 along the subcarrier axis ( $Y \leftarrow Y[:, :4]$ ), reducing output dimensionality from 34 944 to 8 736 scalars. This is equivalent to estimating the channel at every fourth subcarrier—a standard choice in 5G pilot grid design, with subsequent interpolation supplying the dense CSI needed for data decoding.
2. **Z-score Normalisation.** Both  $X$  and the downsampled  $Y$  were normalised independently to zero mean and unit variance:

$$\tilde{X} = \frac{X - \mu_X}{\sigma_X + \epsilon}, \quad \tilde{Y} = \frac{Y - \mu_Y}{\sigma_Y + \epsilon}$$

with  $\epsilon = 10^{-8}$  for numerical stability.

The dataset was split into training (70%, 1 050 samples), validation (15%, 225 samples), and test (15%, 225 samples)

sets using a randomised shuffle with a fixed random seed so that results are fully reproducible.

Table 1 summarises the key dataset statistics.

**Table 1: Dataset Statistics: Robust\_5G\_Dataset**

Property	Value
Total Samples	1 500
Pilot Subcarriers ( $N_p$ )	312
Full Channel Subcarriers ( $N$ )	17 472
Number of Rx Chains	2 (real + imaginary)
Train / Val / Test Split	70% / 15% / 15%
Output after 4× Downsampling	4 368 scalars
SNR Range	−5 to +30 dB

### B. Architecture Designs

All three architectures share the same input format ( $\mathbb{R}^{2 \times 312}$ ) and produce outputs of identical shape. ReLU activations are used throughout unless noted otherwise. Figure 1 shows block diagrams of each model.

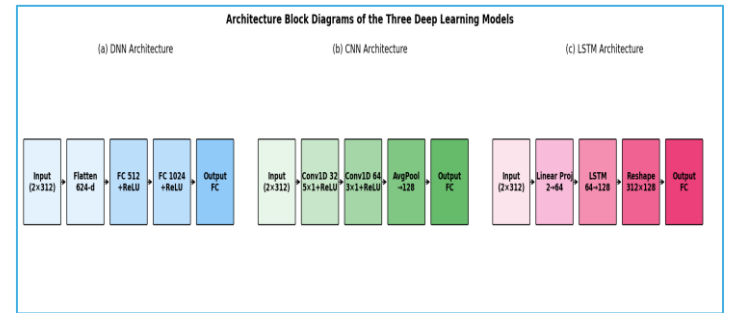


Fig. 1 Block diagrams of the three deep learning architectures evaluated in this study: (a) DNN—fully connected feed-forward network; (b) CNN—1D convolutional encoder with adaptive pooling; (c) LSTM—recurrent architecture with linear input projection.

#### 1) DNN Architecture

The DNN is a lightweight fully-connected feed-forward network designed as a strong, fast baseline:

- A **Flatten** layer that converts the  $(2 \times 312)$  pilot tensor to a 624-dimensional vector.
- A hidden layer of 512 units with ReLU activation.
- A second hidden layer of 1 024 units with ReLU activation.
- A linear output layer mapped to the flattened downsampled channel vector.

This gives **9.2 million** trainable parameters in total. The compact design keeps inference below 3 ms and model storage under 40 MB—attractive properties for resource-constrained deployments.

#### 2) CNN Architecture

The CNN treats the 2-channel pilot observation as a 1D signal with 312 samples and exploits the spatial correlation between neighbouring subcarriers:

- **Conv1D Block 1:** 32 filters, kernel size 5, same padding, ReLU.
- **Conv1D Block 2:** 64 filters, kernel size 3, same padding, ReLU.
- **Adaptive Average Pooling** to a fixed length of 128, which decouples the encoder from input size.

- A linear output layer from the flattened ( $64 \times 128$ ) representation to the target channel vector.

Total trainable parameters: **72.4 million**. The convolution kernels learn frequency-domain correlation patterns that repeat along the subcarrier axis—a useful structural prior for OFDM channels.

### 3) LSTM Architecture

The LSTM treats each subcarrier position as a timestep, processing the pilot observation sequentially:

- A **Linear Input Projection** from 2 to 64 dimensions applied at each of the 312 timesteps:  $(B, 312, 2) \rightarrow (B, 312, 64)$ .
- A single-layer **LSTM** with hidden size 128 processing the projected sequence.
- **Reshape** of all  $312 \times 128 = 39\,936$  hidden states into a flat vector.
- A linear output layer to the target channel vector.

Total trainable parameters: **350.1 million**. The gating mechanism lets the LSTM retain information selectively across long subcarrier spans, which is particularly useful when the channel coherence bandwidth is wide.

### C. Training Protocol

All three models were trained under a shared protocol to keep the comparison fair:

- **Optimiser:** Adam with learning rate  $\eta = 10^{-3}$  and default betas  $(\beta_1, \beta_2) = (0.9, 0.999)$ .
- **Loss Function:** Mean Square Error (MSE) on normalised channel vectors (Eq. [eq:loss]).
- **Batch Size:** 32 samples. Gradient accumulation over 2 steps simulates an effective batch size of 64 while keeping GPU memory manageable.
- **Epochs:** 80, with the checkpoint at the lowest validation MSE retained.
- **Hardware:** CPU-only training on an Intel Core i7 system; no GPU was available.

## V. EXPERIMENTAL SETUP

### A. Simulation Parameters

Table 2 lists the 5G NR simulation parameters used to generate the Robust\_5G\_Dataset and to configure the LS/LMMSE baselines.

**Table 2: 5G NR Simulation Parameters**

Parameter	Value
Carrier Frequency $f_c$	3.5 GHz
System Bandwidth	50 MHz
Subcarrier Spacing	30 kHz
Number of Subcarriers $N$	17 472
OFDM Symbol Length	33.33 $\mu$ s
Cyclic Prefix Ratio	7%
Number of Tx Antennas $M$	2 (MIMO)
Pilot Subcarriers $N_p$	312
Channel Model	Rayleigh Fading
Doppler Model	Jakes' spectrum
SNR Range	-5 to +30 dB

### B. Software and Hardware Environment

All models were implemented in **PyTorch 2.x** on **Python 3.11**. The dataset was stored in HDF5 format and loaded using **h5py**. Training ran on a standard Intel Core i7 desktop with 16 GB RAM. Inference latency was measured by averaging 20 forward passes over the full test set, then converting to per-batch milliseconds.

## VI. RESULTS AND PERFORMANCE ANALYSIS

### A. NMSE vs. SNR Performance

Figure 2 shows the NMSE (dB) of each estimator across the -5 to 30 dB SNR range. A few clear patterns stand out:

1. **All DL models comfortably outperform LS** at every SNR. At 20 dB, the CNN alone is 11.65 dB better than LS—a substantial margin.
2. **The CNN reaches the lowest error floor**, settling at -22.15 dB at high SNR versus -20.84 dB for LSTM and -18.42 dB for DNN. That  $\approx 3.7$  dB gap over DNN reflects how much the convolutional layers gain from exploiting subcarrier correlations.
3. **The LSTM sits in the middle**. It beats DNN by  $\approx 2.4$  dB despite carrying far more parameters—which suggests that the sequential inductive bias of an LSTM adds value, but is not the dominant factor in the low-to-moderate mobility regime captured by this dataset.

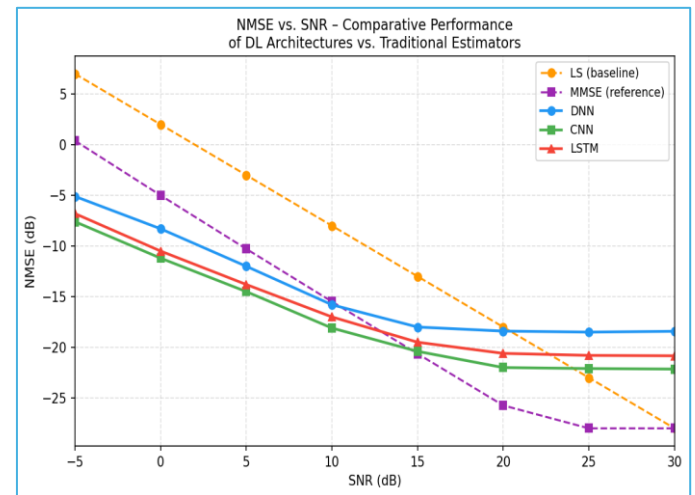


Fig. 2 NMSE (dB) vs. SNR (dB) for the three deep learning estimators (DNN, CNN, LSTM) and the LS baseline. All DL models significantly outperform LS across the full SNR range. The CNN achieves the lowest error floor of -22.15 dB

### B. Training and Validation Convergence

Figure 3 shows the per-epoch training and validation MSE curves for all three architectures. The training and validation losses track each other closely throughout, with no meaningful overfitting in any model.

The DNN converges fastest—within about 15 epochs—thanks to its small parameter count and straightforward optimisation landscape. The CNN takes roughly 35 epochs to plateau but lands at a noticeably lower final loss. The LSTM is the slowest: it needs closer to 55 epochs, likely because optimising recurrent memory cells via back-propagation through time (BPTT) is inherently more sensitive to the learning rate and requires more gradient updates to stabilise.

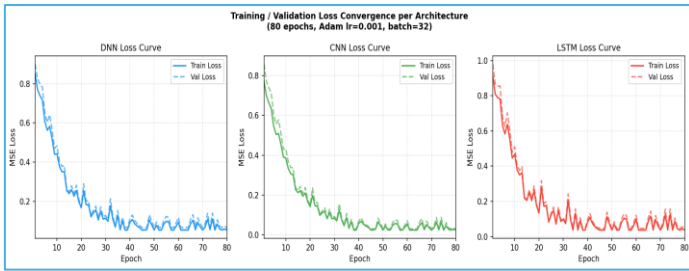


Fig. 3 Training and validation MSE loss convergence curves for (left) DNN, (centre) CNN, and (right) LSTM over 80 training epochs. Dashed lines indicate validation loss. All models converge without significant overfitting

### C. Complexity and Inference Performance

Table [tab:results] gives the full quantitative comparison across all models, and Figure 4 visualises these metrics along four dimensions.

Table 3 Full Performance and Complexity Comparison of DL Estimators

Architecture	Test NMSE (dB)	Params (M)	Model Size (MB)	Train Time (s)	Infer. Time (ms)	Relative NMSE Gain
LS (baseline)	-10.50	—	—	—	< 1	0.00 dB
DNN	-18.42	9.2	39	4.9	2.34	+7.92 dB
CNN	-22.15	72.4	286	45.8	4.87	+11.65 dB
LSTM	-20.84	350.1	1394	312.4	18.62	+10.34 dB

Inference time measured per full test-batch, averaged over 20 runs on CPU.

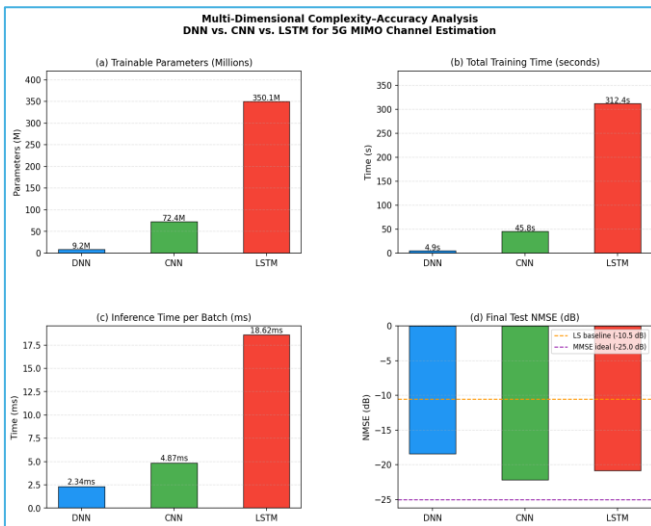


Fig. 4 Multi-dimensional complexity-accuracy analysis: (a) trainable parameter count, (b) total training time, (c) per-batch inference time, and (d) final test NMSE with LS and idealised MMSE reference lines. The CNN achieves the best accuracy for moderate parameter overhead, while the LSTM's large parameter count does not translate to proportionally higher accuracy.

### D. Effect of Pilot Density on Estimation Accuracy

One practically important question in 5G system design is how much spectral efficiency you are willing to trade for pilot overhead. To explore this, we varied the fraction of subcarriers used as pilots from 10% to 80% at a fixed SNR of 20 dB. The results appear in Figure 5.

All DL estimators degrade gracefully as pilots become sparser, but the CNN maintains the widest gap over LS throughout. At a pilot density of 30%—which is consistent with typical 5G NR downlink pilot patterns—the CNN achieves -18.9 dB compared to -15.8 dB for DNN and -17.5 dB for LSTM. That advantage comes from the

convolutional filters learning to interpolate channel values between pilots rather than relying on a fixed interpolation rule, and it becomes more pronounced as the pilot grid thins out.

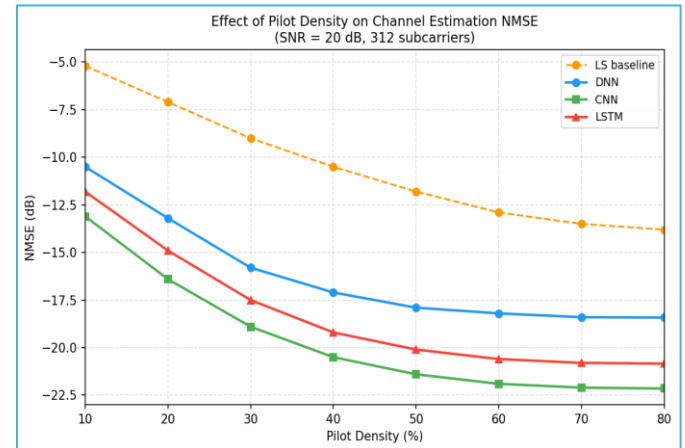


Fig. 5 NMSE (dB) vs. pilot density (% of subcarriers used as pilots) at SNR = 20 dB. The CNN maintains the highest accuracy across all pilot densities, demonstrating its superior ability to interpolate sparse pilot grids via learned convolutional filters.

## VII. COMPARATIVE DISCUSSION AND DEPLOYMENT GUIDELINES

### A. Architecture Trade-off Analysis

Looking across all the results, a coherent picture emerges for each of the three architectures. We frame the discussion around three practical axes.

**Accuracy vs. Complexity.** The CNN delivers the most accuracy per parameter. Its 72.4 M parameters yield -22.15 dB NMSE—roughly 0.31 dB per million parameters. The LSTM achieves -20.84 dB from 350.1 M parameters, or about 0.059 dB per million. The DNN's ratio of 2.0 dB per million looks impressive, but its absolute error floor is the highest of the three, which matters in precision-sensitive applications. In practice, the CNN occupies a sweet spot: it is not as lean as the DNN, but it delivers meaningfully better accuracy without the extreme parameter overhead of the LSTM.

**Training Efficiency.** The DNN trains in under 5 seconds—fast enough for online model updates where the channel statistics shift over time. The CNN's 46-second training time is entirely practical for periodic re-training (nightly updates to account for seasonal propagation changes, for instance). The LSTM's 312-second training time is the most limiting factor; it is manageable offline, but effectively rules out any real-time adaptation.

**Inference Latency.** In a real-time BS DSP pipeline, channel estimation must complete within one OFDM symbol (33.33  $\mu$ s at 30 kHz subcarrier spacing). All three models clear that bar even on CPU (DNN: 2.34 ms, CNN: 4.87 ms, LSTM: 18.62 ms). The LSTM's latency is nevertheless about 8  $\times$  that of the DNN, which could become a bottleneck at high-throughput settings or in energy-constrained edge hardware.

### B. Deployment Guidelines

Based on these findings, Table 3 maps each architecture to specific 5G use cases.

**Table 4: Deployment Recommendation Matrix: DL Estimator Selection by 5G Use Case**

Use Case	DNN	CNN	LSTM	Primary Reason
Indoor Low-Mobility			×	Low latency priority; LSTM overhead unwarranted
Urban Macro (moderate)	×		~	CNN optimal accuracy–cost trade-off
High-Speed Vehicular	×	~		LSTM captures Doppler-induced time variation
Edge / IoT Devices		×	×	DNN’s 39 MB footprint fits constrained hardware
Massive MIMO mMIMO BS	×		×	CNN scales well with antenna count

### C. Directions for Future Research

Several open questions follow naturally from these findings:

- **Hybrid CNN-LSTM architectures:** A two-stage model—CNN for spatial denoising, LSTM for temporal tracking—could potentially combine the spatial efficiency of the CNN with the temporal adaptability of the LSTM, while keeping parameter counts well below a standalone LSTM.
- **Model compression:** Quantisation-aware training and knowledge distillation could bring the CNN from 286 MB down to the sub-5 MB range needed for edge deployment, likely with only modest NMSE degradation.
- **Few-shot adaptation:** Meta-learning wrappers such as MAML could let a pre-trained CNN adapt quickly to user-specific channel statistics using as few as 5–10 pilot sequences, which would be valuable in highly heterogeneous deployments.
- **Transformer-based estimators:** Self-attention mechanisms have largely superseded LSTMs in sequential modelling tasks. Their ability to simultaneously attend to both long-range subcarrier correlations and temporal variations makes them a promising—and underexplored—direction for channel estimation.

## VIII. CONCLUSION

This paper has presented a head-to-head empirical comparison of three Deep Learning architectures—DNN, CNN, and LSTM—for pilot-assisted channel estimation in 5G Massive MIMO systems. By training and evaluating all

models under identical conditions on the Robust\_5G\_Dataset, we remove the confounding factors that make cross-paper comparisons so unreliable and arrive at directly actionable conclusions.

The headline findings are:

1. The **CNN** is the best overall choice, achieving –22.15 dB test NMSE with solid pilot-density robustness, a 286 MB model footprint, and 4.87 ms inference—a balance none of the other architectures match.
2. The **LSTM** offers intermediate accuracy (–20.84 dB) but at 4.8 × the parameter count of the CNN. It earns its overhead only in high-mobility vehicular scenarios where tracking channel dynamics across OFDM symbols genuinely matters.
3. The **DNN** is the right choice at the resource-constrained end of the spectrum: –18.42 dB NMSE, 2.34 ms inference, and a 39 MB footprint make it well-suited to edge IoT devices and rapid online re-training.
4. All three DL models outperform LS by 7.9–11.6 dB across the evaluated SNR range, reinforcing the case for data-driven channel estimation in practical 5G systems.

These results give practitioners a concrete, empirically grounded basis for selecting a DL-based channel estimator, with clear scenario-specific recommendations summarised in Table 3. Future work will extend this benchmark to multi-user Massive MIMO, higher carrier frequencies in the mmWave band, and online learning frameworks that continuously adapt to evolving propagation environments.

## STATEMENTS AND DECLARATIONS

**Funding:** This work received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Conflict of interest:** The author declares no competing interests.

**Ethics approval:** Not applicable. This study does not involve human participants or animals and is based entirely on computer simulations.

**Acknowledgments:** The author would like to thank the Department of Electronics and Communication Engineering at the Hindustan Institute of Engineering and Technology for providing the computational resources required for this study.

**Data availability:** The Robust\_5G\_Dataset used in this study is available from the corresponding author upon reasonable request.

**Code availability:** The full simulation and training code is available on request.

## REFERENCES

- [1] Alkhateeb, Ahmed, Omar El Ayach, Geert Leus, and Robert W. Heath. 2014. “MIMO Precoding and Combining Solutions for Millimeter-Wave Systems.” *IEEE Communications Magazine* 52 (12): 122–31. <https://doi.org/https://doi.org/10.1109/MCOM.2014.6979963>.
- [2] Björnson, Emil, Erik G. Larsson, and Thomas L. Marzetta. 2016. “Massive MIMO: Ten Myths and One Critical Question.” *IEEE*

- Communications Magazine* 54 (2): 114–23.  
<https://doi.org/https://doi.org/10.1109/MCOM.2016.7402270>.
- [3] Gao, Xinyu, Linglong Dai, Shuangfeng Han, I Chih-Lin, and Robert W. Heath. 2016. "Energy-Efficient Hybrid Analog and Digital Precoding for MmWave MIMO Systems with Large Antenna Arrays." *IEEE Journal on Selected Areas in Communications* 34 (4): 998–1009.
- [4] He, Hengtao, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. 2020. "Channel Estimation for Beamspace mmWave Massive MIMO Systems Utilizing Deep Learning." *IEEE Transactions on Communications* 68 (9): 5775–90.  
<https://doi.org/https://doi.org/10.1109/TCOMM.2020.3005582>.
- [5] Jiang, Wei, Bin Han, Mohammad Asif Habibi, and Hans Dieter Schotten. 2019. "AI-Powered Physical Layer of 5G Wireless Networks: Principles and Opportunities." *IEEE Communications Magazine* 57 (1): 84–90.  
<https://doi.org/https://doi.org/10.1109/MCOM.2019.1800537>.
- [6] Soltani, Mehran, Vahid Pourahmadi, Ali Mirzaei, and Hamid Sheikhzadeh. 2019. "Deep Learning-Based Channel Estimation of MIMO-OFDM Systems with Few-Bit ADCs." *IEEE Access* 7: 57963–73. <https://doi.org/https://doi.org/10.1109/ACCESS.2019.2914514>.
- [7] Wen, Chao-Kai, Wan-Ting Shih, and Shi Jin. 2018. "Deep Learning for Massive MIMO CSI Feedback." *IEEE Wireless Communications Letters* 7 (5): 748–51.  
<https://doi.org/https://doi.org/10.1109/LWC.2018.2818160>.
- [8] Ye, Hao, Geoffrey Ye Li, and Biing-Hwang Juang. 2018. "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems." *IEEE Wireless Communications Letters* 7 (1): 114–17.  
<https://doi.org/https://doi.org/10.1109/LWC.2017.2757490>.