Indonesian Journal of Electrical Engineering and Computer Science
Vol. 44, No. 1, October 2026, pp. 236∼249
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v44.i1.pp236-249

ML based predictive Wi-Fi network selection method for better connection, handover and roaming

Pawan Kumar Pant, Prakhar Mishra, Sairaj Gurram
Department of Convergence (Wi-Fi), Samsung Reserch Institute Noida (SRI-N), Noida, India

Article Info

Article history:

Received Sep 21, 2025
Revised Aug 20, 2026
Accepted Sep 18, 2026

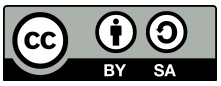
Keywords:

IEEE 802.11
Machine learning
Neural networks
Network performance
User QoE
Wi-Fi access point selection
WLAN
Wi-Fi

ABSTRACT

Wi-Fi has become an essential wireless networking technology, powering a wide range of applications in homes, enterprises, and public environments. As user expectations for quality of experience (QoE) continue to rise, intelligent access point (AP) selection is critical for ensuring consistent connectivity and optimal network performance. This paper presents a machine learning-based AP selection framework that evaluates multiple network parameters such as signal strength, link speed, round-trip time (RTT)-derived distance, channel width, congestion, latency, and interference. A large real-world dataset of 48,000 samples collected across heterogeneous environments was used to evaluate multiple deep learning models, including convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM), and artificial neural network (ANN). The lightweight CNN achieved ≈92.5% accuracy, while the heavyweight CNN reached ≈93.5%, outperforming all statistical baselines and deep learning alternatives in terms of (MSE) and R^2. Compared to the Android Google Scoring Algorithm, the proposed throughput-centric approach offers significantly improved sensitivity to congestion and interference, enabling more reliable AP association and smoother roaming. The framework is computationally efficient, supports on-device deployment, and effectively mitigates sticky-client and sticky-band issues. These results demonstrate a practical and deployable solution for next-generation AI-driven Wi-Fi network selection.

This is an open access article under the CC BY-SA license.



Corresponding Author:

Pawan Kumar Pant
Department of Convergence (Wi-Fi), Samsung Reserch Institute Noida (SRI-N)
Tower-11, Candor Techspace, Plot No. 20 & 21, Sector-135, Noida, 201304 , Uttar Pradesh, India
Email: pawan.pant@samsung.com, pantpawan@gmail.com

1. INTRODUCTION

Wi-Fi has become an essential connectivity medium across homes, offices, and public spaces; however, users frequently experience suboptimal association behavior when client devices rely solely on received signal strength for selecting access points (APs). Prior works have shown that received signal strength indicator (RSSI)-based selection is often unreliable in dense or interference-prone environments [1]-[3], where congestion and channel conditions not just signal strength determine actual throughput. As a result, clients may remain attached to overloaded or distant APs, degrading quality of experience (QoE) for latency- and throughput-sensitive applications.

To address these limitations, this work develops a machine learning (ML) based AP selection framework using parameters accessible through the Android application programming interface (API), including RSSI, frequency, round-trip time (RTT)-derived distance, channel width, link speed, interference, congestion,

and historical connection behavior. By predicting the most suitable AP based on these multi-dimensional features, the system enables faster transitions, lower latency, and more consistent connectivity [4].

Multimedia applications, which require high throughput with minimal delay variation, are particularly vulnerable to congestion and packet-level performance fluctuations [5]. Throughput prediction can support smarter handover and channel selection decisions, helping reduce inter-BSS interference. However, accurate throughput estimation in real deployments remains challenging due to user mobility, heterogeneous devices, interference patterns, and dynamically varying network density [6], [7].

In this work, the AP selection problem is formulated as a supervised regression task, where the target label is the measured throughput and the input features capture AP- and device-level characteristics. Multiple supervised learning techniques were evaluated, and convolutional neural networks (CNNs) were found to provide the best predictive performance, effectively modeling nonlinear interactions among features. The key contributions of this study include: (i) proposing a data-driven AP selection method based on Android-accessible parameters; (ii) formulating throughput-oriented AP prediction for dynamic Wi-Fi environments; and (iii) demonstrating through comparative evaluation that CNNs outperform other supervised learning techniques for this task.

Furthermore, to address the limitations of RSSI-based heuristics and the environment-specific behavior seen in earlier ML-driven AP selection methods, this study provides several additional contributions. First, we build a large real-world dataset of approximately 48,000 samples collected across heterogeneous multi-AP environments in Samsung test labs and public Noida locations, capturing realistic variations in congestion, interference, RTT-derived distance, and link quality. Second, we leverage this dataset to develop a throughput-centric regression model using an expanded feature set that includes RSSI, PHY rate, channel width, RTT-derived distance, congestion, interference, and latency. Third, we design a lightweight CNN architecture ($\approx$15.9k parameters) optimized for mobile-device deployment, achieving around 92% accuracy and outperforming traditional baselines and deep learning models such as artificial neural network (ANN), recurrent neural network (RNN), and long short-term memory (LSTM), alongside a heavier variant suitable for edge/server evaluation. Fourth, we present an Android-integrated architecture composed of a dumpstate driven data pipeline, preprocessing modules, and an AP decider unit, enabling real-world deployability. Finally, through direct comparison with the Android Google Scoring Algorithm, we show that static scoring mechanisms struggle under congestion and interference. In contrast, the proposed CNN model provides adaptive, throughput-aware AP selection that significantly enhances connection stability and roaming performance. Collectively, these contributions advance the state of the art in ML-driven WLAN optimization.

2. LITERATURE SURVEY

Dense, heterogeneous WLANs expose the limits of RSSI-only association and handover. Prior work spans supervised throughput prediction and AP selection, reinforcement/bandit control, topology-aware learning, and environment-aware planning—each improving over static rules but with trade-offs in portability, compute, and real-time deployability.

Supervised prediction and AP selection: Supervised learners (MLP, Trees, SVR, RF) predict instantaneous throughput from load, modulation, arrivals, channel/retransmissions and generally outperform RSSI heuristics on controlled datasets [8]. Practical overlays augment Google-like scoring with traffic, RSSI, bandwidth, distance/frequency, and history but remain close to fixed heuristics [9]. SNR-based/decentralized schemes account for probe behavior and interference-plus-noise [10]. Feed-forward models using SNR, load, beacon delay, failures, and station counts improve selection yet require environment-matched data [11]. Neural link-quality predictors further reduce error but raise data and on-device compute demands [12]. Insights from LTE throughput models help feature design but miss Wi-Fi MAC/interference nuances [13].

RL and bandits: RL reduces unnecessary handovers and boosts throughput but often needs environment-specific retraining and careful exploration [14]. Hybrid LiFi/Wi-Fi RL improves mixed infrastructures with limited applicability to Wi-Fi-only settings [15]. For Wi-Fi 7 MLO, RL (e.g., Optimistic-Weighted VDN) enhances channel selection at higher complexity and weaker legacy compatibility [16]. Coordinated multi-AP control via deep RL aids roaming/load balancing under centralized control assumptions [17]. Bandits in mmWave reduce beam-training overhead by probing a single AP per decision but assume steady rewards/feedback [18]. Enhanced RL for WLAN AP selection reports gains yet faces convergence/state/reward-shaping issues [19].

Topology-aware and distributed learning: GNNs for cell-free massive MIMO exploit AP–AP and device–AP distances to capture spatial coupling beyond scalar RSSI/SNR [20]. Clustered AP selection (e.g., K-means) reduces coordination overhead while introducing cluster stability/reconfiguration costs [21]. Federated learning scales AP-load prediction while limiting central data movement, at the expense of communication overhead and non-uniform data/CapEx across sites [22].

Environment-aware mapping and domain cases: Deep models (CNNs/autoregressive) trained on synthetic floor plans support coverage mapping and AP placement rather than real-time client selection and are often validated in constrained buildings [23]. Smart-building usage prediction aids capacity/energy provisioning with limited value for mobile roaming [24]. An AI/ML-native 802.11 vision outlines architecture without empirical deployment guidance [25]. Domain-specific studies (AGVs in factories, smart farming, CSI sensing) show benefits yet suffer from transferability, hardware sensitivity, and maintenance burdens [26], [27]. Metaheuristics target indoor positioning/multi-objective placement more than online AP choice under mobility [28].

Gap and positioning: Across these threads, ML/DL surpass RSSI heuristics but struggle with portability (environment specificity/retraining), on-device latency/compute, heterogeneous/legacy interoperability, and reliance on proxy metrics. We address these via a throughput-oriented CNN regression using Android-accessible features (RSSI, link speed, band/width, interference/congestion, latency, RTT-based distance), with lightweight (device) and heavyweight (edge/server) realizations for real-time AP selection and roaming in multi-AP Wi-Fi.

Prior work on Wi-Fi AP selection has focused on RSSI or SNR based heuristics, supervised learners trained on limited or lab-generated datasets, or RL approaches that require retraining for each deployment scenario. While these methods show incremental gains, they struggle with real world variability, limited feature diversity, and computational overhead. Our approach differs by leveraging a large-scale, heterogeneous real-world dataset and a throughput-centric CNN model that jointly learns from RSSI, PHY rate, congestion, interference, and RTT-based distance, offering a more robust and deployable solution than existing heuristic or simulation driven methods.

3. PROPOSED MODEL

Method overview: For clarity, the proposed framework follows a five-stage processing pipeline:

a) Data collection: Wi-Fi measurements were captured from Android dumpstate logs across heterogeneous multi-AP environments in Samsung lab setups and public Noida locations. Each data sample includes RSSI, link speed, RTT-derived distance, frequency, channel width, congestion, latency, and interference.

b) Data cleaning and preprocessing: Missing values were imputed, categorical fields were encoded, and numerical features were normalized using min–max scaling. An 80/20 train–test split was applied, and feature statistics were analyzed to assess correlation and importance.

c) Model design: Two CNN architectures were developed—a lightweight model (≈15.9k parameters) optimized for on-device deployment, and a heavyweight model (≈444k parameters) for edge/server execution. Both use convolutional layers to capture nonlinear interactions among Wi-Fi metrics.

d) Training and validation: Models were trained using the MSE loss function with the Adam optimizer, employing early stopping to prevent overfitting. Hyperparameters were tuned using validation data, and performance metrics such as mean squared error (MSE) and R^2 were recorded.

e) Evaluation and comparison: CNN models were benchmarked against statistical baselines (Linear Regression, Decision Tree, Random Forest, SVR, KNN) and deep-learning alternatives (ANN, RNN, LSTM). Additionally, performance was compared with the Android Google Scoring Algorithm to assess practical deployment benefits.

We cast AP selection as a supervised learning task and repurpose a convolutional neural network (CNN) to operate on an image-like representation of per-network measurements. For each candidate AP, a fixed-size matrix is formed with channel index (frequency) along one axis and time-ordered RSSI along the other; traces are normalized and truncated/padded to standardize inputs. The CNN Figure 1 comprises stacked convolution–pooling blocks to extract local spatio–temporal patterns, followed by fully connected layers that produce a per-AP classification/score for throughput-oriented selection. Training uses annotated traces with a separate validation split for hyperparameter tuning and a holdout DUT set for final assessment. The trained

model integrates into the client stack to enable real-time decisions at initial association, proactive handover, and roaming. The architecture of the proposed model is depicted in Figure 2. Input data, gathered from the Wi-Fi settings interface and user applications, is stored in the dumpstate file of mobile devices. In the Data Preparation Unit, the Dumpstate Analyzer extracts relevant Wi-Fi metrics and contextual information, which are then refined by the Preprocessing Module. This processed data is fed into the AI/ML Unit, where convolutional neural networks (CNNs) learn patterns to predict the optimal access point. The effectiveness of CNNs depends on appropriate architecture design, hyperparameter tuning, and a representative dataset capturing diverse Wi-Fi conditions. Finally, the Access Point Decider Unit selects the best AP, enhancing connection stability, handover, and roaming performance.

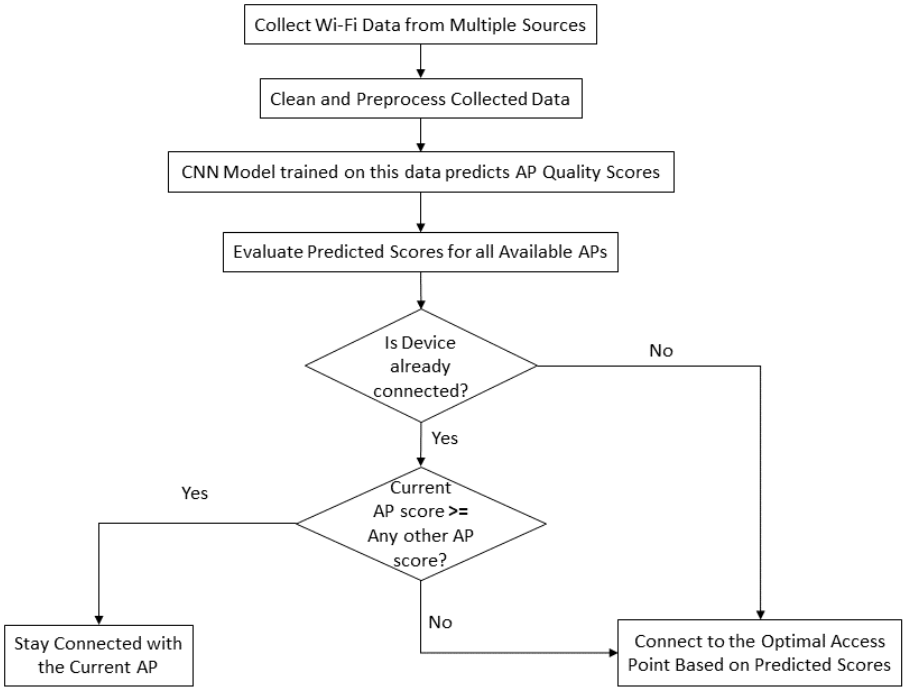


Figure 1. CNN model development flowchart

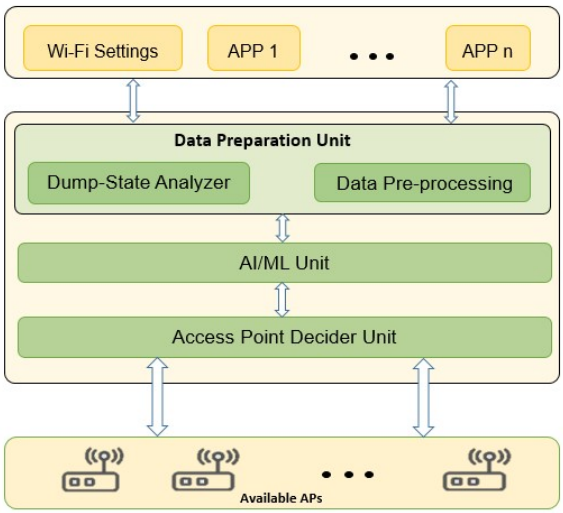


Figure 2. Device architecture of proposed method

3.1. Comparison with traditional google score algorithm

The Google score algorithm [29] evaluates candidate APs using two network nominators: the *Saved Network Nominator* (user-configured) and the *Suggested Nominator* (carrier-recommended). Each AP receives a composite score from fixed lookup mappings of factors including RSSI, estimated throughput, operating band/frequency, channel width, number of spatial streams, security type, and metering status. Additional bonuses apply for 6 GHz operation and for remaining on the current network. Category precedence is: Saved Unmetered, then Suggested Unmetered, then Saved Metered, and finally Suggested Metered. The AP with the highest total score is selected.

In contrast, the proposed CNN regression-based model dynamically learns from real-world data, where actual throughput serves as the prediction label. Input features include RSSI, latency, packet loss, SNR, and interference, allowing the CNN to capture complex feature interactions that static formulas cannot. Unlike the traditional approach, which may select APs with strong RSSI but poor throughput in congested environments, the CNN model directly optimizes throughput, thereby improving connection stability, reducing latency, and enhancing application performance. While the Google algorithm is lightweight and computationally simple, it lacks adaptability. The CNN model is more resource-intensive during training but efficient during inference, making it suitable for modern devices or cloud-assisted implementations.

4. RESULTS AND DISCUSSION

4.1. System design and data pipeline

4.1.1. Data collection

Wi-Fi data were collected from user device logs (dumpstate) in a multi-AP Samsung Noida lab with heterogeneous signal strengths and capacities Figure 3. The client associated to one AP at a time; post-association, a script retrieved logs and features Figure 4. Test measurements were repeated at varied client locations across Noida over three months, yielding ∼48,000 samples. We have placed the client device in various places and collected the data. Following a successful connection, the real-time throughput speed was checked and a complete dataset entry was saved. Using this method, we have compiled around 48000 datasets at various places in Noida city over a span of 3 months. These dataset contain several important parameters, such as:

- The RSSI measures the signal strength.
- Link temporal: network speed
- Frequency: The frequency band of the Wi-Fi network (e.g., 2.4 GHz, 5 GHz)
- Wi-Fi channel width: Wi-Fi channel bandwidth
- "Distance from AP" refers to the distance between a mobile device (station) and a Wi-Fi Access Point (AP). One common method to estimate this distance is by using RTT measurements [30], where the time it takes for a signal to travel from the device to the AP and back is recorded. By analyzing RTT values, it is possible to approximate the device's distance from the AP with reasonable accuracy.
- To give a more complete picture of the network environment, various aspects were also extracted from these parameters:
- Congestion can be modelled as:

$$\text{Congestion} = \frac{N_{\text{devices}}}{B_{\text{total}}}$$

Where:
N_{devices} is the total number of devices that are linked to the access point (AP).
B_{total} is the total available bandwidth of the AP.

- Interference level: Assessed using the SNR (signal-to-noise ratio) and the presence of overlapping channels. Interference is calculated as:

$$\text{Interference} = \sum_{i=1}^{n} \frac{P_{\text{interf}_i}}{d^2_{\text{interf}_i}}$$

Where:
P_{interf_i} is the signal strength of the ith interfering Access Point (AP).
d_{interf_i} is the distance to the ith interfering AP from client.
n represents the total amount of APs that are interfering.



Figure 3. Data collection in Samsung Lab environment

After a successful connection with AP, Script triggered.
For each connection **do**
1. Fetch parameter from the dumpstate/log file .
2. Measure real-time throughput.
3. Append parameters and throughput in the database.

Figure 4. Field data collection

– Latency: Measured as the time delay between sending a request and receiving a response.
Latency is given by:

$$\text{Latency} = 5 + \text{Congestion} + f(\text{Interference Level}) + \frac{d_{\text{router}}}{10} + \epsilon$$

Where:
5 is a base latency in milliseconds (ms).
Congestion is the congestion value from the formula above and in users/Hz.
$f(\text{Interference Level})$ is:

$$f(\text{Interference Level}) = \begin{cases} 0 & \text{if Interference Level} = \text{Low} \\ 10 & \text{if Interference Level} = \text{Medium} \\ 20 & \text{if Interference Level} = \text{High} \end{cases}$$

$\frac{d_{\text{router}}}{10}$ implies a delay that is proportional to the number of routers or hops the packet must traverse. Each router introduces processing and queuing delays, which contribute to the overall latency. In our experimental setup, we have only used one-hop or two-hop systems.
$\epsilon \sim \mathcal{N}(0, 2)$ is a noise term, has a mean of 0 and a standard deviation of 2 milliseconds, and it is regularly distributed.

The objective variable (*score*) is the real-time throughput measured immediately after the device associates to an AP. This scalar target reflects end-user performance and is modeled as a function of Android-accessible features that capture both link and environment conditions, including RSSI, link (PHY) speed, frequency band, channel width, RTT-based distance to AP, a congestion index, an interference/SNR indicator

(including overlapping channels), and measured latency. Descriptive statistics for all features and the target (means, standard deviations, and ranges) are summarized in Figure 5, while Figure 6 presents the score/class distribution, highlighting the balance of samples across throughput levels.

	Signal Strength (dBm)	Network Speed (Mbps)	Congestion (ms)	Latency (ms)	Distance From Router (mtr)	Temperature (°C)	Humidity (%)	Score
count	48390.000000	48390.000000	48390.000000	48390.000000	48390.000000	48390.000000	48390.000000	48390.000000
mean	-54.492374	395.465600	51.688755	102.465279	92.861502	24.974623	47.034656	117.031777
std	13.467859	236.013552	28.283130	85.321359	55.629539	7.784488	16.470218	75.358726
min	-94.995537	0.001725	3.890000	8.242571	0.500000	12.000000	19.000000	0.070555
25%	-59.546487	235.955942	26.811021	60.477717	44.016542	18.000000	33.000000	60.980468
50%	-55.655756	322.588365	52.995210	86.807185	92.355990	25.000000	47.000000	93.554346
75%	-50.499943	502.630886	79.159765	113.120379	141.048850	32.000000	61.000000	153.493975
max	-13.560000	1105.256700	99.000000	499.965020	152.318108	38.000000	75.000000	486.237529

Figure 5. Statistical report

```
Class Distribution (Compact Format):
Security Type: WPA2: 22029, WPA3: 11967, WPA: 9629, Open: 4765
Frequency Band: 5 GHz: 24232, 2.4 GHz: 24158
Channel Width: 20 MHz: 18133, 40 MHz: 18061, 160 MHz: 6101, 80 MHz: 6095
Device Type: Phone: 24203, Tablet: 9671, Laptop: 9577, Watch: 4939
User Density: Medium: 24101, Low: 12167, High: 12122
Interference Level: Medium: 19464, High: 19290, Low: 9636
```

Figure 6. Class distribution report

4.1.2. Data cleaning and preprocessing

Missing values were imputed (mean/median for numeric, mode for categorical). Categorical features (SSID, security type, standard) used one-hot encoding. Numeric features were min–max scaled to $[0, 1]$. An 80/20 train–test split was used.

4.1.3. Generalization considerations

Although the dataset spans multiple locations with varying AP densities across Noida, the current study has not yet evaluated cross-city or cross-country generalization. All experiments were performed within a single metropolitan region using heterogeneous Samsung lab environments and public hotspots. As a result, the reported performance reflects in-scope deployment conditions. Future work will extend the evaluation to different building layouts, enterprise networks, and geographical regions to assess model portability and robustness under broader environmental variability.

4.2. Models and training

Weather variables (e.g., temperature, humidity) were excluded due to negligible effect. Baseline statistical models—Linear Regression, Decision Tree, Random Forest, SVR, and KNN—were tuned via grid search with cross-validation and evaluated using MSE and R^2. In parallel, deep learning candidates (ANN, CNN, RNN, LSTM) were assessed under comparable tuning protocols (depth/width, activation functions, learning rate), and results were logged for comparative analysis.

4.3. CNN architecture and training

Two CNNs were designed to trade off deployability and accuracy.

– Lightweight CNN model: It is designed with a focus on reducing the number of parameters to ensure deployability on resource-constrained devices such as mobile phones Figure 7. The architecture included

fewer convolutional layers and smaller dense layers. This model achieved an accuracy of approximately 92%, which, while slightly lower than the more complex models, is sufficient for many practical applications. Figure 8: ∼15,893 params; input 6×1 (RSSI, link speed, congestion, latency, distance, score); 1D Conv(16, k=2) → Flatten → Dense(64) → Output(1). Accuracy ∼ 92%; mobile-friendly footprint.

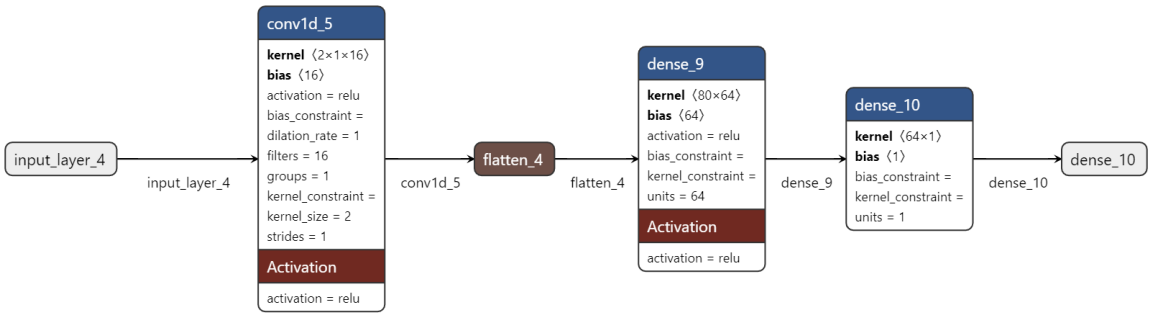


Figure 7. Lightweight CNN model architecture

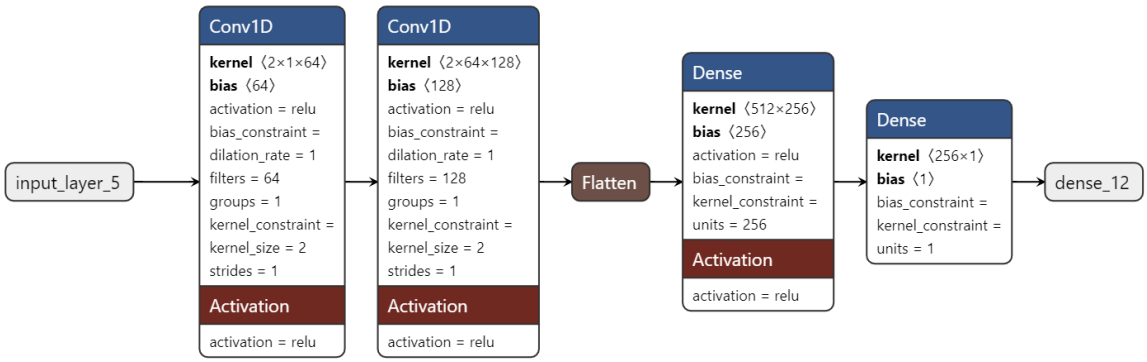


Figure 8. Heavyweight CNN model architecture

– Heavyweight CNN model: To maximize performance, a deeper CNN with more convolutional layers, larger filter sizes, and more units in the dense layers was developed Figure 8. This model achieved a higher accuracy of 93.3%, but its larger size and memory requirements made it unsuitable for deployment on devices with limited resources, such as mobile phones. Figure 8: ∼444,869 params; 1D Conv(64, k=2) → 1D Conv(128, k=2) → Flatten → Dense(256) → Output(1). Accuracy ∼ 93.3%; higher resource demand.

All layers used ReLU with Adam optimization. Models were trained for 100 epochs (batch size 32) using MSE loss with early stopping on validation loss; post-training, test performance was evaluated via MSE and R^2, with the heavyweight model slightly outperforming the lightweight variant.

4.4. Results, analysis, and performance comparison

4.4.1. Exploratory analysis

A correlation matrix Figure 9 and scatter plots Figure 10 informed feature relevance. RSSI correlated negatively with distance (-0.8); throughput correlated negatively with congestion (-0.51) and latency (-0.23), and positively with link speed (0.36). Distance and throughput were negatively correlated (-0.69). These trends motivated throughput-centric modeling.

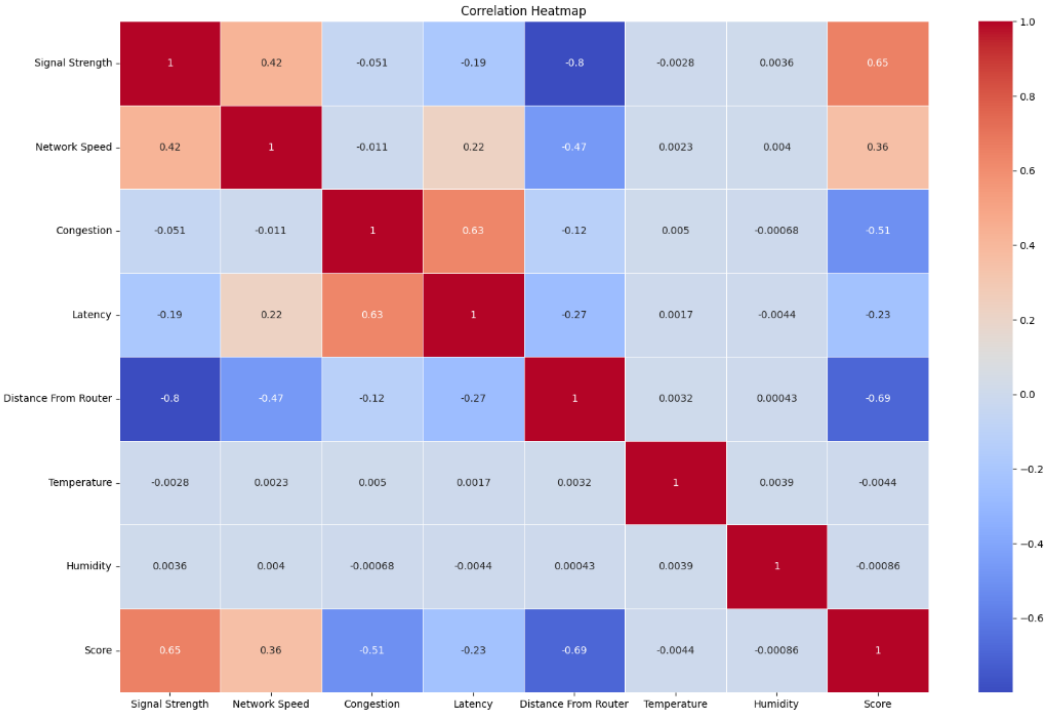


Figure 9. Correlation Heat-map

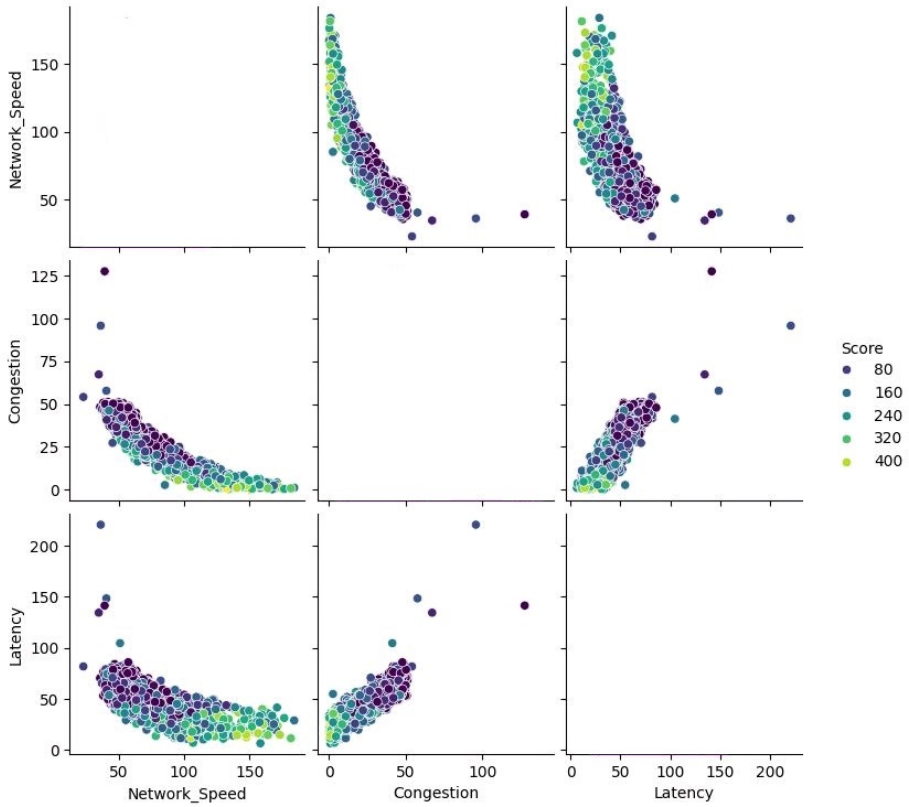


Figure 10. Scatter-plots between highly correlated features

4.4.2. Baselines vs. deep models

Statistical and deep learning results are summarized in Figures 11 and 12. Random Forest and KNN were the strongest statistical baselines Figure 11; CNNs outperformed all baselines Figure 12. Lower MSE and higher R^2 indicate better fit.

	Train MSE	Test MSE	Train R2	Test R2
Linear Regression	27.6942	27.1753	0.902268	0.905226
Decision Tree	10.6299	33.807	0.962487	0.882098
Random Forest	15.8639	26.7029	0.944017	0.906874
Support Vector Regressor	30.8318	30.2741	0.891195	0.894419
K-Nearest Neighbors	17.8117	25.9596	0.937143	0.909466
CNN	19.6882	19.4599	0.930521	0.932134

Figure 11. Performance comparison (statistical models)

	Train MSE	Test MSE	Train R2	Test R2
CNN	19.6882	19.4599	0.930521	0.932134
Lightweight CNN	21.6344	21.6237	0.923653	0.924588
Heavyweight CNN	17.6336	18.437	0.937771	0.935701
RNN	20.6616	23.9243	0.927086	0.916564
LSTM	32.9552	31.5824	0.883702	0.889857
ANN	14.1191	21.7193	0.950174	0.924254

Figure 12. Performance comparison (deep learning models)

4.4.3. Learning dynamics and efficiency

Across CNN, lightweight/heavyweight CNN, RNN, LSTM, and ANN Figure 13, CNNs converged rapidly and stably, offering favorable accuracy–latency trade-offs and efficient inference relative to sequential models [31]-[34]. LSTMs showed higher variance; ANNs converged quickly but were more prone to overfitting; RNNs converged slower.

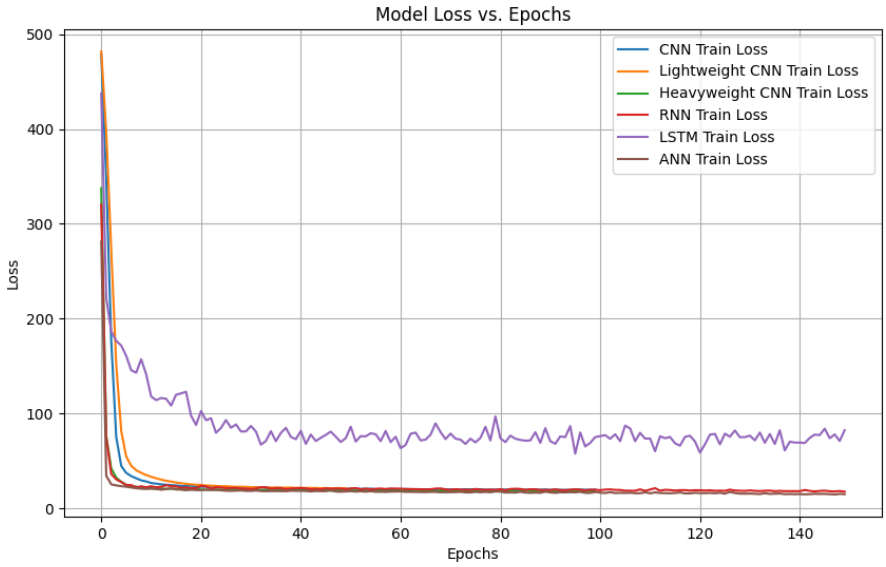


Figure 13. Model Loss vs. Epochs

4.4.4. Final recommendation and deployment

For resource-constrained devices, the lightweight CNN is recommended due to its small footprint and near-parity accuracy with the heavyweight model. The model improves association, anticipates handover by ranking neighboring APs before degradation, and enables smoother intra-SSID roaming by jointly considering interference, congestion, and current link quality reducing latency and stalls.

Unlike existing works that predominantly evaluate AP selection models in simulated settings or small, controlled environments, our study demonstrates that both the lightweight and heavyweight CNN architectures achieve superior performance across a large real-world dataset of 48,000 samples. The proposed models consistently outperform statistical baselines as well as deep learning alternatives such as ANN, RNN, and LSTM in terms of MSE and accuracy. Furthermore, by directly comparing our approach with the Android Google Scoring Algorithm, we show that learned throughput prediction enables adaptive and congestion-aware AP selection, whereas heuristic scoring fails to capture interference, load, and real-time link dynamics. These findings position our framework as a practical and deployable enhancement over prior ML-based and heuristic-driven solutions and highlight its unique contributions to the state of the art in ML-driven WLAN optimization.

4.4.5. Practical impact

In operational terms, the 92–93% prediction accuracy translates into noticeably fewer misassociations, reduced attachment to congested or interference-heavy APs, and more stable throughput during real usage. This results in faster initial connection, smoother intra-SSID roaming, and improved QoE for applications such as video streaming, online gaming, and real-time communication. The lightweight CNN's low inference latency further enables on-device decision making without impacting battery or performance.

5. CONCLUSIONS AND FUTURE WORK

This work formulated Wi-Fi AP selection as a throughput-oriented prediction task and evaluated statistical and deep models on a large, real-world dataset. Among all candidates, convolutional neural networks (CNNs) consistently achieved the best fit (lower MSE, higher R^2), capturing feature interactions beyond static heuristics. Two CNN variants were examined: a lightweight model ($\sim$15.9k parameters) delivering $\approx$92.5% accuracy with mobile-class deployability, and a heavyweight model ($\sim$444.9k parameters) reaching $\approx$93.5% accuracy at higher computational cost. These results indicate that learned, data-driven scoring improves AP selection under diverse conditions.

Recommendation: For on-device use, the lightweight CNN offers the best accuracy–efficiency trade-off; the heavyweight model suits server/edge deployments where resources are ample. However, a key limitation of the present study is that all evaluations were conducted within a single metropolitan region and under controlled mobility patterns, meaning the reported accuracy reflects in-scope deployment conditions rather than full variability across cities, enterprise networks, or diverse building geometries.

Future Work: We will broaden real-world data coverage, incorporate richer environmental and contextual features, optimize the lightweight model for edge deployment via quantization/pruning, and integrate explainability methods to make AP-selection decisions transparent. To further validate and generalize the proposed framework, several key experiments should be conducted: (i) cross-environment testing across different buildings, cities, and deployment densities to assess model portability; (ii) mobility-driven roaming experiments to measure handover delay, packet loss, and application-level QoE during walking and vehicular movement; (iii) multi-vendor interoperability testing to ensure consistent performance across heterogeneous AP hardware; (iv) on-device benchmarking of inference latency, CPU/memory footprint, and energy consumption to confirm deployability on mobile clients; and (v) hybrid edge–device evaluations to study trade-offs between lightweight on-device inference and heavyweight cloud-assisted prediction. These experiments will strengthen scalability, robustness, and practical readiness for real-world deployments.

ACKNOWLEDGMENTS

This work was conducted using institutional resources at Samsung Research Institute Noida (SRI-N). No individuals outside the author list contributed sufficiently to merit acknowledgment.

FUNDING INFORMATION

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Pawan Kumar Pant	✓	✓			✓	✓		✓		✓		✓	✓	
Prakhar Mishra	✓		✓	✓	✓	✓		✓	✓		✓			
Sairaj Gurram			✓	✓		✓	✓	✓		✓	✓			

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject Administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding Acquisition
Fo	: **Fo**rmal Analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The dataset was collected at Samsung Research Institute Noida (SRI-N) test labs and selected public locations in Noida and contains internal AP identifiers/configurations as well as third-party public AP details. Owing to corporate confidentiality and third-party privacy constraints, the full dataset cannot be made publicly available. An anonymized illustrative snippet and the feature/schema description used in the study are provided within the article and/or its supplementary materials. Additional de-identified subsets or aggregated statistics may be available from the corresponding author upon reasonable request, subject to institutional review and a data-sharing agreement. No personally identifiable information was collected.

REFERENCES

[1] S. Szott, K. Kosek-Szott, P. Gawłowicz, J. T. Gómez, B. Bellalta, A. Zubow, and F. Dressler, "Wi-Fi meets ML: A survey on improving IEEE 802.11 performance with machine learning," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 3, pp. 1843–1893, 2022, doi: 10.1109/COMST.2022.3179242.

[2] J.-C. Chen, T.-C. Chen, T. Zhang, and E. van den Berg, "WLC19-4: Effective AP selection and load balancing in IEEE 802.11 Wireless LANs," in *in Proceedings of the IEEE Global Telecommunications Conference*, 2006, pp. 1–6, doi: 10.1109/GLOCOM.2006.724.

[3] L. Du, M. R. Jeong, A. Yamada, Y. Bai, and L. Chen, "QoS aware access point selection for pre-Load-balancing in multi-BSSs WLAN," in *in Proceedings of the IEEE Wireless Communications and Networking Conference*, 2008, pp. 1634–1638, doi: 10.1109/WCNC.2008.292.

[4] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives," *IEEE Communications Magazine*, vol. 44, no. 12, pp. 51–57, 2006, doi: 10.1109/MCOM.2006.273099.

[5] K. Xu, Y. Tian, and N. Ansari, "Improving TCP performance in integrated wireless communications networks," *Computer Networks*, vol. 47, no. 2, pp. 219–237, 2005, doi: 10.1016/j.comnet.2004.07.006.

[6] M. Abusubaih, "A new approach for interference measurement in 802.11 WLANs," in *in Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2010, pp. 2336–2341, doi: 10.1109/PIMRC.2010.5671707.

[7] F. Xu, X. Zhu, C. C. Tan, Q. Li, G. Yan, and J. Wu, "SmartAssoc: Decentralized access point selection algorithm to improve throughput," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2482–2491, 2013, doi: 10.1109/TPDS.2013.10.

[8] M. A. Khan, R. Hamila, N. A. Al-Emadi, S. Kiranyaz, and M. Gabbouj, "Real-time throughput prediction for cognitive Wi-Fi networks," *Journal of Network and Computer Applications*, vol. 150, Art. no. 102499, 2020, doi: 10.1016/j.jnca.2019.102499.

[9] V. Akash, S. M. Howale, A. J. Gupta, N. Kayarvizhy, and R. K. Saranappa, "Predictive Wi-Fi network selection: Enhancing user experience through a machine learning-based approach," in *in Proceedings of the IEEE Global Conference on Artificial Intelligence and Computing Technologies*, 2023, pp. 1–6, doi: 10.1109/GCAT59970.2023.10353492.

[10] P. B. Oni and S. D. Blostein, "Decentralized AP selection in large-scale wireless LANs considering multi-AP interference," in *in Proceedings of the International Conference on Computing, Networking and Communications*, 2017, pp. 13–18, doi: 10.1109/ICCNC.2017.7876094.

[11] B. Bojovic, N. Baldo, J. Nin-Guerrero, and P. Dini, "A supervised learning approach to cognitive access point selection," in *in Proceedings of the IEEE Global Communications Conference Workshops*, 2011, pp. 1100–1105, doi: 10.1109/GLOCOMW.2011.6162348.

[12] M. Biagi, G. Bartolini, F. Graziosi, and S. Tennina, "Predicting Wi-Fi link quality through artificial neural networks," *Internet Technology Letters*, vol. 4, no. 3, Art. no. e326, 2021, doi: 10.1002/itl2.326.

[13] G. A. Fernandez, "Machine Learning for Wireless Network Throughput Prediction," *Advances in machine learning and artificial intelligence*, vol. 5, no. 1, pp. 01–06, Jan. 2024. [Online]. Available: https://scholarworks.utrgv.edu/mss_fac/447/

[14] M. A. Khan, R. Hamila, A. Gastli, S. Kiranyaz, and N. A. Al-Emadi, "ML-based handover prediction and AP selection in cognitive Wi-Fi networks," *Journal of Network and Systems Management*, vol. 30, no. 4, p. 72, 2022, doi: 10.1007/s10922-022-09684-2.

[15] D. F. Fonseca, B. G. Guzman, G. L. Martena, R. Bian, H. Haas, and D. Giustiniano, "Prediction-model-assisted reinforcement learning algorithm for handover decision-making in hybrid LiFi and WiFi networks," *Journal of Optical Communications and Networking*, vol. 16, no. 2, pp. 159–170, Feb. 2024, doi: 10.1364/JOCN.495234.

[16] P. E. Iturria-Rivera, M. Chenier, B. Herscovici, B. Kantarci, and M. Erol-Kantarci, "Channel selection for Wi-Fi 7 multi-link operation via optimistic-weighted VDN and parallel transfer reinforcement learning," *arXiv preprint* arXiv:2307.05419, 2023. [Online]. Available: https://arxiv.org/abs/2307.05419

[17] L. Zhang, H. Yin, S. Roy, and L. Cao, "Multiaccess point coordination for next-gen Wi-Fi networks aided by deep reinforcement learning," *IEEE Systems Journal*, vol. 17, no. 1, pp. 904–915, Mar. 2023, doi: 10.1109/JSYST.2022.3183199.

[18] S. Hashima, K. Hatano, H. Kasban, and E. M. M. Mahmoud, "Wi-Fi assisted contextual multi-armed bandit for neighbor discovery and selection in millimeter wave device to device communications," *Sensors*, vol. 21, no. 8, Art. no. 2835, 2021, doi: 10.3390/s21082835.

[19] G. P. Dubey, S. Stalin, O. Alqahtani, A. Alasiry, M. Sharma, A. Aleryani, P. K. Shukla, and M. T.-H. Alouane, "Optimal path selection using reinforcement learning based ant colony optimization algorithm in IoT-Based wireless sensor networks with 5G technology," *Computer Communications*, vol. 212, pp. 377–389, 2023, doi: 10.1016/j.comcom.2023.09.015.

[20] V. Ranasinghe, N. Rajatheva, and M. Latva-aho, "Graph neural network based access point selection for cell-free massive MIMO systems," in *in Proceedings of the IEEE Global Communications Conference*, 2021, pp. 1–6, doi: 10.1109/GLOBECOM46510.2021.9685221.

[21] S. Biswas and P. Vijayakumar, "AP selection in cell-free massive MIMO system using machine learning algorithm," in *in Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking*, 2021, pp. 158–161, doi: 10.1109/WiSPNET51692.2021.9419450.

[22] S. Shriwastav *et al.*, "Wireless signal prediction using deep learning models for WiFi coverage mapping," in *in Proceedings of the Hawaii International Conference on System Sciences*, 2024, pp. 5691–5700. [Online]. Available: https://hdl.handle.net/10125/107070

[23] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: recent progress and future challenges," *arXiv preprint* arXiv:2104.02151, 2021. [Online]. Available: https://arxiv.org/abs/2104.02151

[24] G. H. Apostolo, F. Bernardini, L. C. S. Magalhães, and D. C. Muchaluat-Saade, "A unified methodology to predict Wi-Fi network usage in smart buildings," *IEEE Access*, vol. 9, pp. 11455–11469, 2021, doi: 10.1109/ACCESS.2020.3048891.

[25] F. Wilhelmi *et al.*, "Machine learning and Wi-Fi: toward AI/ML-Native IEEE 802.11 networks," *arXiv preprint* arXiv:2405.11504, 2024. [Online]. Available: https://arxiv.org/abs/2405.11504

[26] S. Samour, "Performance enhancement in Wi-Fi networks using machine learning," Ph.D. dissertation, Univ. Clermont Auvergne, 2023. [Online]. Available: https://theses.hal.science/tel-04393773v1/file/2023UCFA0041_SAMMOUR.pdf

[27] M. Ali *et al.*, "A comparison of machine learning algorithms for Wi-Fi sensing using CSI data," *Electronics*, vol. 12, no. 18, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/18/3935

[28] W. Zhang, K. Yu, W. Wang, and X. Li, "A self-adaptive AP selection algorithm based on multiobjective optimization for indoor WiFi positioning," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1406–1416, 2021, doi: 10.1109/JIOT.2020.3011402.

[29] Android Open Source Project, "Wi-Fi network selection," 2024. [Online]. Available: https://source.android.com/docs/core/connect/wifi-network-selection (Accessed: Apr. 24, 2025).

[30] F. Xu, K. A. Nguyen, and Z. Luo, "WiFi round-trip time (RTT) fingerprinting: an analysis of the properties and the performance in non-line-of-sight environments," *Journal of Location Based Services*, vol. 17, no. 4, pp. 307–339, 2023, doi: 10.1080/17489725.2023.2239748.

[31] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021, doi: 10.1186/s40537-021-00444-8.

[32] A. Ghosh, N. D. Jana, S. Mallik, and Z. Zhao, "Designing optimal convolutional neural network architecture using differential evolution algorithm," *Patterns*, vol. 3, no. 9, Art. no. 100567, 2022, doi: 10.1016/j.patter.2022.100567.

[33] M. A. K. Raiaan, S. Sakib, N. M. Fahad, A. Al Mamun, M. A. Rahman, S. Shatabda, and M. S. H. Mukta, "A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks," *Decision Analytics Journal*, vol. 11, Art. no. 100470, 2024, doi: 10.1016/j.dajour.2024.100470.

[34] A. Sehgal and N. Kehtarnavaz, "Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time apps," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 450–465, 2019, doi: 10.3390/make1010027.

BIOGRAPHIES OF AUTHORS



Pawan Kumar Pant is Senior Chief Engineer (Convergence) at Samsung India Electronics Pvt. Ltd., Noida. He holds an M.Tech. in Information Technology from IIIT Allahabad. His research focuses on WLAN security, IEEE 802.11 performance, and AI/ML in wireless networks. He has authored some Scopus-indexed publications and is a co-inventor on three Wi-Fi–related patent applications filed in 2024–2025; publications pending. He can be contacted at email: pawan.pant@samsung.com.



Prakhar Mishra is an Engineer at Samsung India Electronics Pvt. Ltd., Noida, working in the Convergence team with a focus on WLAN and AI/ML in wireless networks. He holds a B.Tech. in Electronics and Communication Engineering from NIT Silchar and is a co-inventor on Wi-Fi–related patent applications filed in 2024–2025; publications pending. He can be contacted at email: prakhar.m1@samsung.com.



Sairaj Gurram is a Software Engineer at Samsung India Electronics Pvt. Ltd., Noida, focusing on Android development and Wi-Fi frameworks. He holds a B.E. in Computer Science from CBIT Hyderabad and is a co-inventor on AI-driven Wi-Fi patent applications filed in 2024–2025; publications pending. He can be contacted at email: s2.gurram@samsung.com.