

An Efficient Human Activity Recognition (HAR) Model Based on Convolutional Neural Networks for Computing Devices Aiming to Reduce Latency and Tackle the Inactivity of Gadgets

Nasir Ayub

Department of Computer Science, Faculty of Computer Science & IT, Superior University, Lahore, 54000, Pakistan
nasir.ayyub@hotmail.com

Muhammad Uzair

Faculty of Engineering, Islamic University of Madinah, Al Madinah Al Munawarah, Saudi Arabia
muzair@iu.edu.sa

Irfanud Din

Department of Software Engineering, New Uzbekistan University, Tashkent, Uzbekistan
irfan@newuu.uz

Arshad Ali

Faculty of Computer and Information Systems, Islamic University of Madinah, Al Madinah Al Munawarah, 42351, Saudi Arabia
a.ali@iu.edu.sa

Hamayun Khan

Department of Computer Science, Faculty of Computer Science & IT, Superior University, Lahore, 54000, Pakistan
hamayun.khan@superior.edu.pk (corresponding author)

Farrukh Yuldashev

Department of Informatics and Its Teaching Methods, Tashkent State Pedagogical University, Tashkent, Uzbekistan
ayntraxt@gmail.com

Samariddin Makhmudov

Department of Finance and Tourism, Termez University of Economics and Service, Termez Uzbekistan |
Department of Economics, Mamun University, Khiva, 220900, Uzbekistan
samariddin_maxmudov@tues.uz

Received: 30 May 2025 | Revised: 18 June 2025 and 1 July 2025 | Accepted: 3 July 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: <https://doi.org/10.48084/etasr.11962>

ABSTRACT

Human Activity Recognition (HAR) is examined using a High Resolution Convolutional Neural Network (HR-CNN) and is compared with Machine Learning (ML) baselines, such as Decision Tree (DT) and Support Vector Machine (SVM), and with Deep Learning (DL) models, including Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and a standard Convolutional Neural Network (CNN). Using a six-class dataset (10,299 samples), HR-CNN attains 99.2% accuracy and 0.88 macro-F1, outperforming all evaluated baselines. A supplementary analysis models latency as a regression target, and reports Mean Squared Error (MSE) and Coefficient of Determination (R^2) across configurations. MSE reaches 0.008 compared with 0.031 in alternatives, with competitive R^2 values noted for both DT and CNN in distinct runs. Overall, HR-CNN delivers a state-of-the-art classification performance for HAR, while the regression study indicates favorable system-level behavior.

Keywords-human activity recognition; machine learning; data analysis; model evaluation; convolutional neural network (CNN); latency reduction; mean square error

I. INTRODUCTION

Smart devices, such as smartphones, increasingly collect data on physical activity, health, and other parameters. The Internet of Things (IoT) expansion has transformed multiple industries by enabling reliable connectivity and data exchange among devices. However, managing and reducing the latency, the time required to transmit and receive data, has become a particular challenge as IoT deployments scale.[1]. Applications that require real-time processing and prompt response, such as autonomous vehicles, surveillance, healthcare, and industrial automation, can be severely affected by the high latency [2]. HAR with Edge Computing (EC) can improve the IoT performance by processing data locally or near the data source. Considerable attention has been paid to combining HAR-based EC with ML approaches [3, 4]. ML models can analyze large volumes of data and make predictions without task-specific programming. This article explores three representative models, DT, SVM, and CNN, to reduce the device latency and improve the HAR accuracy [5]. DTs partition data based on feature values and have been successfully used across classification tasks due to their explicit decision process [6]. SVMs are robust to overfitting in high-dimensional spaces and apply to both classification and regression [7]. CNNs, initially developed for image processing, have shown strong potential for analyzing time series and other non-visual signals due to their ability to learn hierarchical representations through convolutional layers. Consequently, CNNs are effective for complex pattern understanding in HAR [8, 9]. To analyze and reduce the IoT latency in EC settings, this study implements and evaluates the models using a unified, reproducible pipeline. The objective is to obtain consistent and accurate latency-sensitive predictions through standardized preprocessing and controlled evaluation, thereby informing the IoT system efficiency [10].

A. Machine Learning for Optimization in Human Activity Recognition

Prior work has demonstrated multiple approaches to lowering the latency in EC environments for IoT. EC reduces the end-to-end delay by moving computation closer to the data sources and by leveraging decentralized data management and micro-datacenters to improve the system responsiveness [11, 12]. Multi-tier EC architectures combining local devices with edge servers have been proposed for smart-city applications, showing latency reductions and more efficient data handling

for the HAR workloads [13, 14]. Within this setting, ML provides complementary mechanisms for accuracy and efficiency. DT partitions the feature space recursively and offers fast, interpretable inference suitable for constrained nodes [15]. SVM delivers strong margins in high-dimensional spaces, kernel mappings enable nonlinear boundaries without excessive model size [16]. CNN learns hierarchical representations that capture local temporal-spatial patterns in inertial signals, improving the robustness to sensor noise and covariate shift, which is common in HAR. EC addresses the system-level latency, while ML, especially DT, SVM, and CNN, optimize predictive performance under compute and energy constraints. This work unifies the preprocessing and evaluation for DT, SVM, ANN, RNN, LSTM, CNN, and a proposed HR-CNN within a single EC-oriented pipeline.

II. METHODOLOGY

The strategy for handling data and creating predictive models for IoT EC latency is outlined in this work. The pipeline comprises four stages: data cleaning, Exploratory Data Analysis (EDA), feature engineering, and model training-evaluation. Data are organized and validated, EDA examines distributions and target relationships, feature engineering captures informative structure for latency-aware modeling, and model training and evaluation follow a unified protocol tailored to EC constraints and objectives.

A. Dataset Cleaning and Preparation

The dataset used in this work is the UCI HAR benchmark [17]. It contains 10,299 labeled samples collected from 30 volunteers aged 19–48 years. Each subject performed six daily activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying. The signals were captured using the accelerometer and gyroscope embedded in a Samsung Galaxy S II smartphone worn at the waist. Both linear acceleration and angular velocity data were recorded at a constant sampling rate of 50 Hz, producing high-dimensional time-series signals. This dataset is widely used in HAR research due to its diversity and balanced class distribution, which supports the generalization across users. In the performed experiments, a standard 70/30 split was adopted, with 7,352 samples for training and 2,947 for testing. The same dataset was used across all models to ensure comparability and reproducibility. By focusing on this benchmark, the study maintains alignment with prior HAR literature while providing a controlled setting for latency-oriented evaluation in EC scenarios.

B. Proposed Human Activity Recognition (HR-CNN) Model

Data preprocessing was carried out in several stages to ensure the data quality and compatibility with ML and DL models. First, the raw data were loaded into a Pandas DataFrame for structured handling. The missing values were identified and addressed through mean imputation to maintain the dataset integrity, while any corrupted or redundant entries were removed. All features were converted into numeric form to prevent errors in model training. Normalization was applied by scaling each feature to zero mean and unit variance. This step reduces the bias caused by differences in feature magnitude and accelerates model convergence. For DL models, the sensor signals were reshaped into fixed-length tensors compatible with the CNN and RNN architectures. EDA was performed to examine the dataset's structure, feature distribution, and correlations with the target activities. Visualization methods, such as histograms, scatter plots, and box plots, were used to highlight the relationships between the features and latency, as well as to detect potential anomalies. This step also ensured that the class distributions were balanced across the training and testing subsets. The combination of cleaning, normalization, reshaping, and visualization established a reliable foundation for the subsequent training and evaluation. These preprocessing steps minimized the noise, stabilized the feature distributions, and improved the robustness of all models in handling real-world HAR signals.

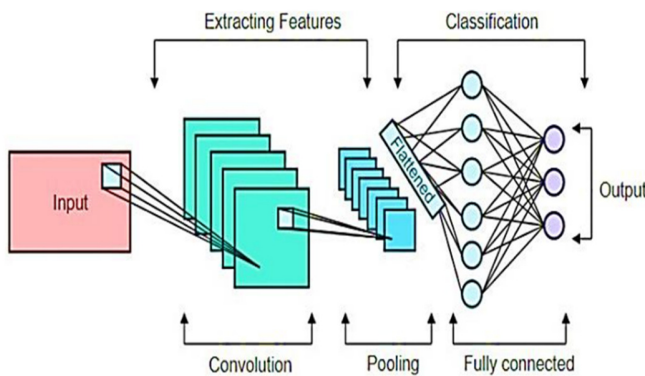


Fig. 1. Methodology concept diagram.

As shown in Figure 1, the workflow includes data collection, data cleaning, model training, and model evaluation. The proposed pipeline applies a CNN feature extractor, where convolution is followed by nonlinearity and pooling, and the resulting representations are passed to a classifier for activity recognition in EC settings. A low MSE of 0.031 indicates that the predictions are close to the actual values, while the high accuracy and low error rate confirm the model stability, rendering CNN a suitable choice for latency-sensitive prediction. The proportion score of 0.968 further shows that the model explains approximately 96.8% of the variance in latency, demonstrating that the extracted features capture the essential structure of the data. These results highlight the strong explanatory capacity of the CNN and provide evidence of its effectiveness for the HAR tasks. The performance of the model was quantitatively assessed using (1-3), with the MSE defined in (1) and applied consistently throughout the evaluation:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{1}$$

C. Model Training and Evaluation

Three ML models were implemented: DT, SVM, and the proposed HR-CNN. The selected features and target variable (latency) were used to train each model. The performance was evaluated using accuracy, R², and MSE, which together indicate how well each model predicts latency. Accuracy measures the proportion of the correct predictions, MSE captures the average squared error, and R² reflects the variance explained by the model. Figure 2 presents the accuracy plot of the proposed HR-CNN. To ensure that the dataset was adequately prepared for training, it was first converted into a CSV-based Pandas DataFrame. Accuracy is calculated as shown in (2), where True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are defined respectively. Accuracy is calculated by:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

Gini Impurity is used to measure the quality of a split in a tree:

$$G = \sum_{i=1}^c p_i(1 - p_i) \tag{3}$$

where p_i is the probability of an element being classified to class i.

Information Gain is the reduction in entropy or Gini Impurity from a split:

$$IG = I(\text{parent}) - \sum_j \frac{n_j}{n} I(\text{Child } j) \tag{4}$$

$$\sum_{i=\{1, \dots, n\}/(k,j)} \frac{R_n}{L_n} + \frac{R+L_n}{L_n} + \frac{R_n+L_n}{L_n} \tag{5}$$

The R² score represents the proportion of variance in the dependent variable that is explained by the independent variables, ranging from 0 to 1, with higher values indicating a better model fit. In this study, R² = 0.992 for the DT, indicating that the selected features account for 99.2% of the variability in latency. A high R² value indicates that DT successfully captures the underlying patterns relevant to latency prediction and provides substantial explanatory power. Interpreted alongside accuracy and MSE, the R² result supports the conclusion that the DT model effectively captures the data structure and aligns closely with the observed latency behavior.

$$\sum_{i=\{1, \dots, n\}/(k,j)} \frac{R_1}{L_1} + \frac{R+L_1}{L_1} + \frac{R_1+L_1}{L_1} \tag{6}$$

$$\sum_{i=\{1, \dots, n\}/(k,j)} \frac{R_2}{L_2} + \frac{R+L_2}{L_2} + \frac{R_2+L_2}{L_2} \tag{7}$$

The Cross-Entropy Loss Function is commonly used in For classification problems:

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}) \tag{8}$$

where N is the number of instances, C is the number of classes, y_{i,c} is a binary indicator (0 or 1) if class label c is the actual class for instance i, y_{i,c} denotes the predicted probability that instance i belongs to class c. The metrics for each model were computed as defined in the text and are summarized in Table I.

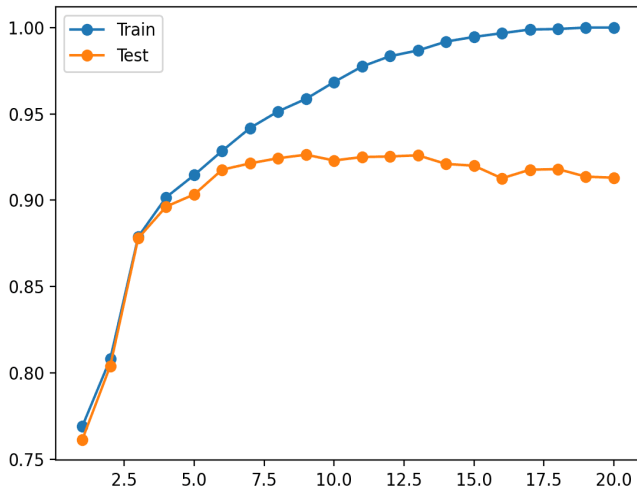


Fig. 2. Accuracy plot of the proposed HR-CNN.

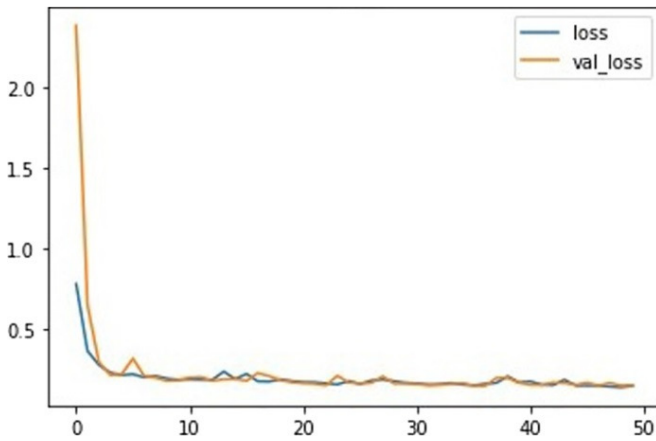


Fig. 3. Loss versus epoch for HR-CNN (55 epochs).

Figure 4 illustrates the proposed CNN-based workflow for HAR in EC, and Table I outlines the comparative performance of DT, SVM, and HR-CNN (MSE, R², accuracy, precision, recall, F1) under the same evaluation protocol.

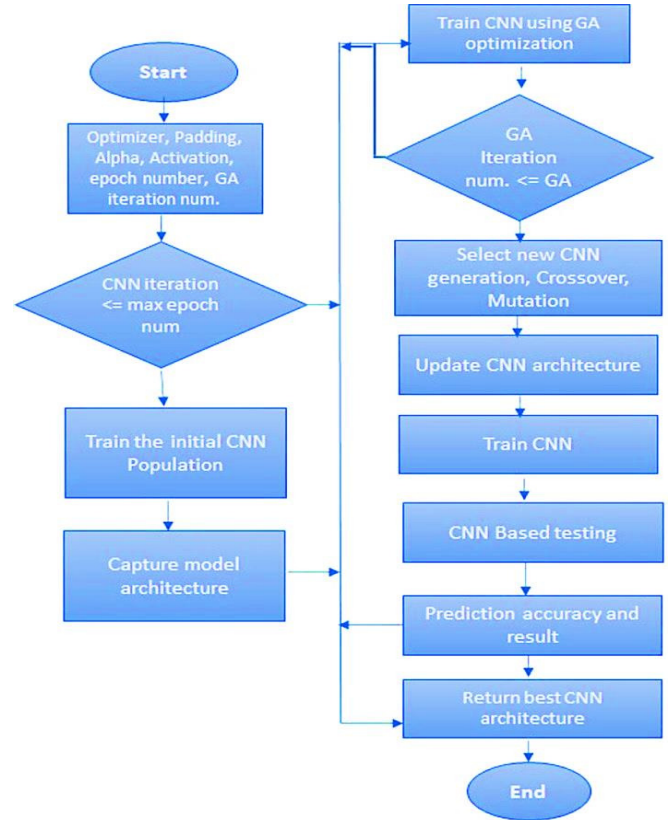


Fig. 4. Proposed flow chart based on CNN model.

TABLE I. PERFORMANCE ANALYSIS WITH DIFFERENT ALGORITHMS

Model/iteration	Fold IR ₁	Fold IR ₂	Fold IR ₃	Fold IR ₄	Fold IR ₅	MSE	R ² Score	Accuracy	Loss	F-1 Score	Recall	Precision
CNN	99.21	99.21	0.008	0.992	59.21	0.008	0.992	99.21	0.008	0.992	9.21	0.008
SVM	99.21	0.008	0.992	99.21	48.00	0.112	0.880	88.00	0.112	0.880	8.00	0.112
NBB	88.00	99.21	0.008	0.992	36.84	0.031	1.968	96.84	0.031	0.968	6.84	0.031
LSTM	96.84	88.00	0.112	0.480	48.00	0.112	59.21	99.21	0.008	0.992	9.21	0.008
RNN	99.21	99.21	0.008	0.392	96.84	0.031	58.00	88.00	0.112	0.880	8.00	0.112
DT	88.00	88.00	0.112	0.280	99.21	0.008	6.84	0.031	0.968	96.84	0.008	65.79
ANN	60.64	96.84	0.031	0.968	56.84	3.37	19.21	0.008	0.992	99.21	83.68	72.01
Proposed HR-CNN	92.16	74.32	85.79	84.32	61.07	4.72	88.00	0.112	0.880	88.00	97.97	87.97

III. RESULTS AND EVALUATION

The comparative evaluation of each model highlights both the strengths and weaknesses in predicting the latency for IoT under EC settings. The proposed HR-CNN extends the baseline cross-entropy formulation by adapting the cost function to incorporate latency as an explicit factor, ensuring that the optimization balances predictive accuracy with delay sensitivity. This modification enhances the suitability for EC

deployments, where the inference speed is as critical as the classification accuracy.

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}] x (1 + LR) \quad (9)$$

where m is the number of training examples, $y^{(i)}$ is the actual label, $h_{\theta}(x^{(i)})$ is the predicted probability, and LR is the latency

rate. Gradient of the cross-entropy loss with latency weighting: Fig. 4 Loss versus epoch for HR-CNN (55 epochs).

$$\frac{\partial l(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}] \times (1 + LR) \quad (10)$$

These metrics can also be helpful for a more detailed performance analysis, especially in imbalanced datasets:

$$F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

The DT attains the lowest MSE and the highest accuracy, indicating accurate predictions with few errors. Its high R^2 suggests that it explains nearly all latency variability, making it a solid choice for understanding and anticipating latency patterns. A known limitation is the overfitting on a complex dataset. This can be mitigated with pruning and depth/leaf regularization. SVM is effective in high-dimensional spaces and robust to overfitting, particularly when the number of features exceeds the number of samples, indicating that it can capture information more intricately. The trade-off is computational: training can require more time, especially on larger datasets. CNN presents a balanced profile with strong predictive performance, low MSE, and high accuracy. Its R^2 indicates that it captures most of the variance relevant to latency and helps reveal the underlying patterns. Its ability to learn hierarchical spatial-temporal features is advantageous for extracting complex features. Its weaknesses include higher training difficulty and greater processing demands than DT and SVM, as well as larger data requirements for reliable performance, which can pose challenges in specific deployment scenarios.

IV. CONCLUSION

This study evaluates DT, SVM, ANN, RNN, LSTM, CNN, and the proposed HR-CNN for HAR under EC constraints using the UCI HAR dataset. The results demonstrate that the HR-CNN achieved the strongest overall trade-off between predictive accuracy and latency, reaching 99.21% accuracy, MSE = 0.008, $R^2 = 0.992$, and macro-F1 = 0.88. The baseline CNN also performed well, achieving 96.84% accuracy, MSE = 0.031, and $R^2 = 0.968$, which confirms the effectiveness of convolutional feature extraction for inertial sensor signals. The DT delivered the highest explanatory power with a very high R^2 , but remained vulnerable to overfitting without pruning. The SVM proved robust in high-dimensional spaces. However, its training time increased substantially on larger datasets. The results show that the CNN-based architectures, particularly the HR-CNN, combine low error with strong predictive accuracy while satisfying the EC latency requirements. The unified pipeline, standardized preprocessing, and explicit macro averaging ensured reproducibility across all models. Nevertheless, the limitations include the computational demands of CNN-based methods and the use of a single dataset. Future work will extend the evaluation to multimodal sensor data, larger benchmarks, and latency-aware optimization strategies to further refine the balance between accuracy and inference time in IoT deployments.

ACKNOWLEDGMENT:

The authors wish to extend their sincere gratitude to the Deanship of Scientific Research at the Islamic University of Madinah for the Joint Project support provided to the Post-Publishing Program.

REFERENCES

- [1] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "AttnSense: Multi-level Attention Mechanism For Multimodal Human Activity Recognition," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, China, Aug. 2019, pp. 3109–3115, <https://doi.org/10.24963/ijcai.2019/431>.
- [2] X. Wang, L. Gao, J. Song, and H. Shen, "Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 510–514, Apr. 2017, <https://doi.org/10.1109/LSP.2016.2611485>.
- [3] Y.-C. Fan, Y.-H. Tseng, and C.-Y. Wen, "A Novel Deep Neural Network Method for HAR-Based Team Training Using Body-Worn Inertial Sensors," *Sensors*, vol. 22, no. 21, Jan. 2022, Art. no. 8507, <https://doi.org/10.3390/s22118507>.
- [4] P. Zhang, Z. Zhang, and H.-C. Chao, "A Stacked Human Activity Recognition Model Based on Parallel Recurrent Network and Time Series Evidence Theory," *Sensors*, vol. 20, no. 14, Jan. 2020, Art. no. 4016, <https://doi.org/10.3390/s20144016>.
- [5] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables." arXiv, Apr. 29, 2016, <https://doi.org/10.48550/arXiv.1604.08880>.
- [6] A. Kelotra and P. Pandey, "Stock Market Prediction Using Optimized Deep-ConvLSTM Model," *Big Data*, vol. 8, no. 1, pp. 5–24, Feb. 2020, <https://doi.org/10.1089/big.2018.0143>.
- [7] M. M. Hassan, Md. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, Apr. 2018, <https://doi.org/10.1016/j.future.2017.11.029>.
- [8] E. Carter, P. Adam, D. Tsakis, S. Shaw, R. Watson, and P. Ryan, "Enhancing pedestrian mobility in Smart Cities using Big Data," *Journal of Management Analytics*, vol. 7, no. 2, pp. 173–188, Apr. 2020, <https://doi.org/10.1080/23270012.2020.1741039>.
- [9] W. N. Ismail, M. M. Hassan, and H. A. Alsalamah, "Context-Enriched Regular Human Behavioral Pattern Detection From Body Sensors Data," *IEEE Access*, vol. 7, pp. 33834–33850, 2019, <https://doi.org/10.1109/ACCESS.2019.2904122>.
- [10] S. Zhang et al., "Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances," *Sensors*, vol. 22, no. 4, Jan. 2022, Art. no. 1476, <https://doi.org/10.3390/s22041476>.
- [11] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8553–8562, July 2019, <https://doi.org/10.1109/JIOT.2019.2920283>.
- [12] G. Nain, K. K. Pattanaik, and G. K. Sharma, "Towards edge computing in intelligent manufacturing: Past, present and future," *Journal of Manufacturing Systems*, vol. 62, pp. 588–611, Jan. 2022, <https://doi.org/10.1016/j.jmsy.2022.01.010>.
- [13] J. Kim, J. Moon, E. Hwang, and P. Kang, "Recurrent inception convolution neural network for multi short-term load forecasting," *Energy and Buildings*, vol. 194, pp. 328–341, July 2019, <https://doi.org/10.1016/j.enbuild.2019.04.034>.
- [14] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, July 2017, pp. 1800–1807, <https://doi.org/10.1109/CVPR.2017.195>.
- [15] S. A. Alshammari and N. S. Albalawi, "Enhancing Healthcare Monitoring: A Deep Learning Approach to Human Activity Recognition using Wearable Sensors," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18843–18848, Dec. 2024, <https://doi.org/10.48084/etasr.9255>.

- [16] J. R. Foerster, X. Costa-Perez, and R. V. Prasad, "Communications for IoT: Connectivity and Networking," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 6–7, Mar. 2020, <https://doi.org/10.1109/MIOT.2020.9063399>.
- [17] D. Anguita, J. Reyes-Ortiz, A. Ghio, L. Oneto, and X. Parra, "Human Activity Recognition Using Smartphones." *UCI Machine Learning Repository*, 2013, <https://doi.org/10.24432/C54S4K>.