Grenze International Journal of Engineering and Technology, June Issue

Performance evaluation of TinyML and CNN Models on Edge Devices for Real Time Medical Diagnosis

Sharon Joy[1], Somnath Sinha[2] and Binayak Dutta[3]
[1-3]Christ University, Bangalore, India
Email: sharon.joy@bsccsh.christuniversity.in, Somnath.sinha@christuniversity.in, binayak.dutta@christuniversity.in

***Abstract*— As the use of wearable health devices like smartwatches and fitness bands increases, the need for quick analysis also increases. In these situations, researchers are introducing some new TinyML systems that help to process data directly without the cloud. This study is primarily about the analysis that compares convolutional neural networks (CNNs) and TinyML systems for two medical datasets: diabetes prediction and electrocardiogram (ECG) analysis. In this study, we can clearly see how a model's complexity is affected to make accurate predictions, the training time, and the quantity of resources. This study shows that TinyML models perform better than other models with less computing power. The TinyML model achieves an accuracy of 85.71% on the diabetes dataset, whereas the CNN model achieves only 76.60%. The TinyML model is more effective when compared with its training time. For the ECG datasets, both models achieve a similar accuracy level of around 71%. The TinyML model trains 1900 times faster than the CNN. This shows that the TinyML model can deliver output much faster than CNN, maintaining the same level of accuracy. From these findings, we can conclude that the lightweight machines give more efficient results. This model enables data analysis without cloud systems. This helps the health monitoring systems to be more practical, affordable, and accessible—especially in remote areas.**

***Index Terms*— TinyML, Medical Diagnostics, CNN, Edge Computing, Diabetes Prediction, ECG Classification, Healthcare IoT, Machine Learning.**

I. Introduction

The major health challenges faced by the population are cardiovascular diseases and diabetes [1]. Managing these medical conditions has shifted from traditional models to proactive, continuous, and personalized approaches [2]. The vast amount of medical data generated by wearable devices and easy-to-use medical tools helped improve the healthcare system [3]. For early detection and effective management of these medical conditions, an effective analysis is required [4]. Machine learning (ML) has developed as an efficient tool in medical diagnostics [5]. Using some deep learning models, such as Convolutional Neural Networks (CNNs), the medical applications have achieved excellent results [2]. These advanced models require high computational power and cloud-based processing. This makes them less practical for real time application [6]. TinyML models allow machine learning (ML) to run on very small devices such as smartwatches with limited power, memory, and computing capacity [3]. This method allows instant analysis of health data with limited latency and with no internet connectivity. This is very crucial in health care, as many real-time errors occur during patient monitoring.

Grenze ID: 01.GIJET.12.2.292_20

This paper describes two widely used techniques – CNN and TinyML for diabetes prediction and electrocardiogram (ECG) signal analysis. This study also analyzes the differences in training time, computational efficiency, memory requirements, and real-world applicability.

II. Related Works

The aim of machine learning in medical diagnostics has been researched deeply. Many researchers have studied various algorithms using the Pima Indians Diabetes Database [7]. Traditional machine learning methods, such as decision trees, logistic regression, and support vector machines, have given good results [9][10]. Model comparison techniques include pruning and knowledge distillation. These methods help to improve hardware effectiveness [12]. Deep learning methods for medical diagnosis have many attentions. Hannun et al. showed that deep neural networks can work with cardiologists in finding arrhythmias using ECG data [5]. Similarly, many studies have shown that CNNs and other deep learning models can predict diabetes, with different results [4][9]. The introduction of TinyML has given new possibilities for medical diagnostics on the brink. Recent studies that emphasize the potential of TinyML's in the health care domain [6] have shown the importance of keeping a balance between model performance and computational efficiency. Studies have discussed that TinyML – optimized neural networks actually decrease memory footprint and inference latency in comparison with traditional CNN implementations, which can be efficiently used in devices of the microcontroller class [13]. There are not many comparisons between TinyML and traditional deep learning methods in the medical domain, though. While comparing the conventional CNN deployments and the present CNN deployments, the present CNN deployments are more developed in the fields of energy consumption and execution time [14]. Even though the full-scale CNN models have a high predictive performance in clinical applications, they are weak in edge hardware [15]. Deep convolutional neural networks need essential computational resources and GPU acceleration to train and inference, which limits their direct deployment on resource – constrained implanted and edge devices [16]. By contrasting CNN and TinyML models on standard medical datasets, our study removes this gap. We give importance to the rule for edge device execution in the real world. We concentrate on the practical needs to deploy edge devices.

III. Proposed Method

A. Datasets

This comparative study was basically based on two publicly available medical datasets

Diabetes Database: For the diabetics prediction purpose, s cursed dataset was used. It contains patient records with factors such as glucose level, blood pressure, BMI, insulin level, age, and other clinical indicators relevant to diabetes diagnosis. The dataset was preprocessed to remove duplicate entries and missing values, ensuring data consistency. And the focus variable is binary, with values 1 and 0 for a diabetics and non-diabetics patient.

ECG Database: The PTB-XL electrocardiography (ECG) dataset from PhysioNet was taken and it includes ECG recordings and the patient metadata. In this paper, we highlight on patient details such as age, sex, height, and weight to classify ECG patterns as normal or abnormal, after converting multi-class diagnostic labels into a binary format.

B. Data Preprocessing

For both datasets, standard preprocessing techniques were applied:

- Feature scaling using a standard normalization approach was used to bring all numerical features to the same range.
- Missing values in continuous variables were inserted using the mean value for each variable.
- Binary representation for categorical variables where suitable.
- Train-test split with 80% training and 20% testing data.

For the PTB-XL dataset, additional preprocessing involved parsing medical codes and converting multi-class problems into binary classification tasks (Normal vs. Abnormal).

C. Model Architectures

CNN Model: A sequential neural network implemented using TensorFlow/Keras with the following architecture:

- Input layer with dimensions matching dataset features.
- Dense layer with 64 units and ReLU activation.
- Dropout layer (rate = 0.3) for regularization.

- Dense layer with 32 units and ReLU activation.
- Dropout layer (rate = 0.3).
- Output layer with 1 unit and sigmoid activation for binary classification.

TinyML Model: Decision Tree classifier optimized for edge deployment with:

- Maximum depth constraint to prevent overfitting.
- Minimum samples split and leaf parameters tuned for generalization.
- Scikit-learn implementation for compatibility and efficiency.

D. Training Protocol

All models were developed using identical data splits to ensure fair comparison. The CNN was trained for 20 epochs using the Adam optimizer with binary cross entropy loss. The Decision Tree was developed using default parameters with depth limitations to prevent overfitting.

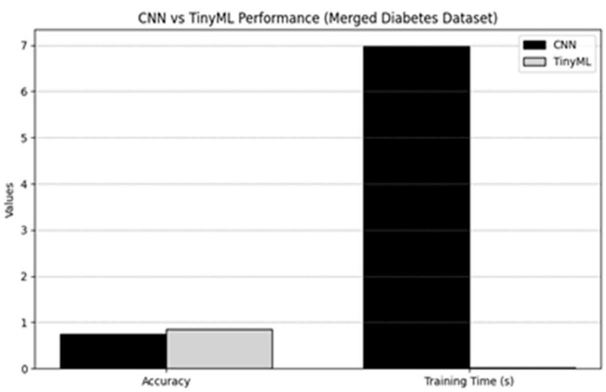

Figure 1: Performance comparison on Diabetes dataset showing accuracy and training time metrics for CNN vs. TinyML models.

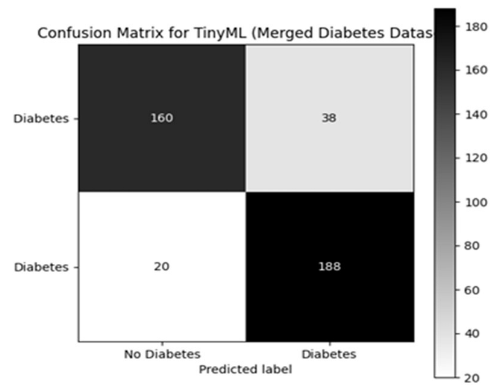

Figure 2: Confusion matrix for TinyML model on the Diabetes dataset.

IV. RESULTS AND DISCUSSION

A. Overall Performance Comparison

Table (I) outlines the in-depth performance analysis across both datasets.

B. Diabetes Classification Analysis:

Figure (1) highlights the notable performance difference between CNN and TinyML models on the diabetes dataset.

The TinyML model obtained an accuracy of 85.71%, while the CNN reached 76.60%.

Various reasons explain this big difference:

- Overfitting in CNNs: Although CNN performed very well, but its deep structure still risks mild overfitting as the dataset size is limited, as it has only 768 instances, finally lead to non-generalization to the test set.
- Feature Relevance: As the diabetic's dataset has a tabular format with carefully chosen medical features, it helped in working better with conventional machine learning strategies than the other deep learning strategies.
- Training Efficiency: Here TinyML proved its efficiency by just taking the training time of 0.010 seconds when compared to 2.612 seconds of the CNN model, thus making it about 256 times faster.

Deeper insights based on the TinyML model's performance is clear from the confusion matrix as shown in the figure 2, showing balanced precision and recall across both the classes.

TABLE I: PERFORMANCE COMPARISON: CNN VS TINYML MODELS

Dataset	Model	Accuracy (%)	Training Time (s)
Diabetes	CNN	76.60	2.8671
	TinyML	85.71	0.0120
ECG	CNN	71.04	12.93
	TinyML	71.08	0.007

TABLE II: PERFORMANCE OF NEURAL NETWORKS ON DIABETES DATASET

Model	Accuracy (%)	ROC_AUC	Training Time (s)
Full Neural Network	0.7734	0.7724	7.11
TinyML Neural Network	0.7512	0.7503	5.05

C. ECG Classification Analysis:

The ECG categorization method revealed more comparable performance between the two models, as seen in Figures 4 and 3. As both have obtained almost the same accuracy levels (CNN: 71.04%, TinyML: 71.08%), but with dramatically different computational requirements.

The time taken to train the TinyML model is just 0.007 seconds, when compared to the CNN model with 12.93 seconds. This represents 1900 times faster training.

This great achievement has made a vital finding in the case of edge deployment environments. This gain has made an important conclusion for edge deployment scenarios.

D. Computational Efficiency Analysis

From the outputs of the analysis, it is clear that TinyML has major importance in the case of computational speed. Key findings include:

- Training Speed: TinyML models have faster training orders than CNN models across both datasets.
- Model Size: Compared to CNN models, decision tree models need less memory under 100 nodes.
- Even though inference speed is not explicitly measured, from this study, we can see that decision trees offer faster inference.

E. Clinical Implications

The higher performance of TinyML models, mainly on the diabetes dataset, has important clinical suggestions:

- Real-time Monitoring: Continuous model updates and real-time health monitoring are enabled using fast training and inferences.
- Privacy Preservation: The need to transmit some sensitive data to cloud servers is eliminated.
- Accessibility: Advanced diagnostics are accessible in resource-constrained environments using lower computational requirements.

V. COMPARATIVE ANALYSIS OF FULL AND TINYML NEURAL NETWORKS

Along with the CNN and Decision Tree–based TinyML models, two neural network architectures were implemented and evaluated on both datasets: a full-scale neural network and a TinyML-compatible lightweight neural model. Both were trained using standardized input features, the Adam optimizer, and binary cross-entropy loss.

A. Diabetes Dataset– Neural Network Models

The curated diabetes dataset was used to assess the performance of these neural networks. The full neural network achieved an accuracy of 77.34% and an ROC–AUC of 0.7724, with a training time of 7.11 seconds, while the TinyML neural model obtained a comparable accuracy of 75.12% and ROC AUC of 0.7503, training slightly faster in 5.05 seconds, as shown in Table II.

The confusion matrices (Tables III and IV) show that Both models effectively distinguish between diabetic and non-diabetic cases.

The TinyML neural model maintained strong recall for diabetic patients while achieving faster training, highlighting its suitability for edge deployments.

TABLE III: CONFUSION MATRIX– FULL NEURAL NETWORK (DIABETES)

	Predicted: 0	Predicted: 1
Actual: 0	43	7
Actual: 1	10	19

TABLE IV: CONFUSION MATRIX– TINYML NEURAL NETWORK (DIABETES)

	Predicted: 0	Predicted: 1
Actual: 0	141	57
Actual: 1	44	164

B. ECG Dataset– Neural Network Models

For the PTB-XL ECG dataset, both neural network variants demonstrated consistent performance. The full neural network achieved an accuracy of 71.76% and ROC–AUC of 0.703, while the TinyML version achieved 71.44% and ROC–AUC of 0.701, with a shorter training duration, as presented in Table V.

TABLE V: PERFORMANCE OF NEURAL NETWORKS ON ECG DATASET

Model	Accuracy (%)	ROC_AUC	Training Time (s)
Full Neural Network	0.7176	0.7030	62.18
TinyML Neural Network	0.7144	0.7010	54.23

Even through the normal and abnormal ECG classification, both the models maintained a stable accuracy and stability, ensuring that TinyML can provide resource-efficient inference for integrated healthcare systems.

TABLE VI: CONFUSION MATRIX– FULL NEURAL NETWORK (ECG)

	Predicted: 0	Predicted: 1
Actual: 0	1086	817
Actual: 1	391	1984

TABLE VII: CONFUSION MATRIX– TINYML NEURAL NETWORK (ECG)

	Predicted: 0	Predicted: 1
Actual: 0	1103	800
Actual: 1	422	1953

VI. PROPOSED METHOD

The proposed methodology involves data handling, model framework, optimizing, and Performance across CNN, TinyML, and Neural Network models. The complete workflow of the entire study is shown in Fig. 5.

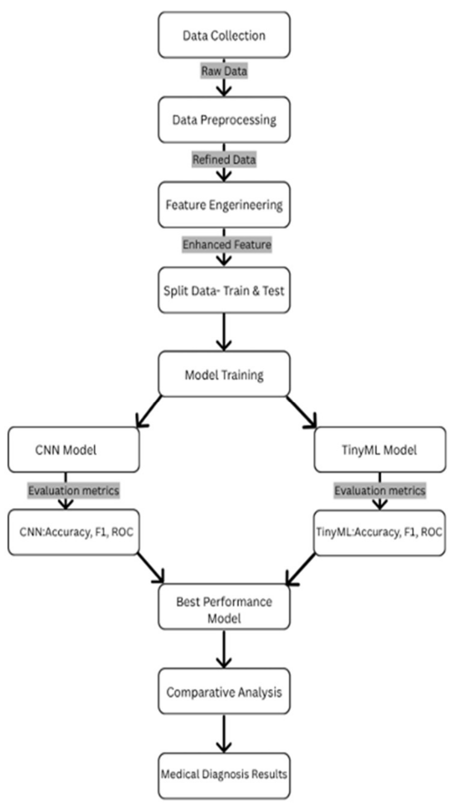

Figure 5: Workflow of the suggested comparative analysis between CNN, TinyML, and neural network models.

VII. Result and Discussion

The experimental results highlight a clear difference in computational efficiency and predictive capability between CNN and TinyML models across both datasets. For the Diabetes dataset, the TinyML Decision Tree model attained the highest accuracy of 79.75%, with the Full Neural Network (77.34%) and the TinyML Neural Network (75.12%) coming next. The CNN model showed good performance, reaching 76.60%. Even though the differences in accuracy are comparatively less, the training times differed greatly—TinyML models trained in only 0.002-5.05 seconds compared to CNN's 7.11 seconds. This effectiveness advantage shows TinyML's capability for fast, resource-constrained surroundings. And the slight change in the performance also confirms that simpler or lightweight neural networks can perform comparably to deeper CNNs in the case of small tabular datasets. And in the case of ECG dataset, both models show comparable outputs, where CNN shows 71.04% accuracy and TinyML shows 71.08%. Though there were a substantial difference—TinyML needs only 0.007 seconds, while CNN needed 12.93 seconds, proving TinyML approximately 1900 times faster. This proved the strong potential of TinyML for real-time analysis and deployment in low power integrated medical systems. From the confusion matrix it was evident that both models were efficient and effective in normal and abnormal ECG signals with stable accuracy. In general, regarding the reduction in operation time and resource usage TinyML showed comparable predictive accuracy rather than CNN. And from the results it is very evident that TinyML is a trustable and effective method for edge-based healthcare applications, which permits instance tracking and real time diagnostics on low-power devices without relying on cloud-based computing.

VIII. Limitations and Future Scope

Even though TinyML models demonstrated strong performance, this study was limited to structured tabular datasets. Future work should explore:

- Implementation of TinyML Neural Networks on real integrated devices (e.g., Arduino, ESP32).
- Testing on more complex multimodal datasets merging ECG signals and vital sensor data.
- Incorporation of federated learning to ensure data privacy in healthcare IoT networks.

IX. Conclusion

The work showed a comparative study of CNN and TinyML models for medical diagnostics using Diabetes and ECG datasets. The results showed that TinyML models can get comparable or even superior accuracy to CNNs. It requires significantly less computational resources and time. The TinyML Decision Tree model got the highest accuracy of 79.75% while the Full Neural and CNN models got 77.34% and 76.60%, respectively, for the Diabetes dataset. The TinyML Neural Network performed well at 75.12% training in just 5.05 seconds in comparison to CNN's 7.11 seconds. These results show that TinyML models can give comparable accuracy with necessary lower computational cost. For the ECG dataset, TinyML got similar accuracy (71.08%) to CNN (71.04%) but trained nearly 1900 times faster. From these interpretations, we can make sure that TinyML models are strongly effective, light in weight, and have the ability for maintaining strong predictive functioning across medical categorization activities. These results show TinyML is a practical and dependable solution for edge-based healthcare applications or systems, particularly for wearable devices and IoT medical systems where computational resources are limited. By allowing on-device processing, TinyML assures speedier decision-making, data security, data privacy, and ongoing real-time health tracking without depending on cloud infrastructure. Later, these surveys will be based on implementing these models on integrated hardware platforms like Arduino or ESP32, which extends the study to multimodal medical datasets while exploring other combined learning possibilities to emphasize privacy and overall performance. Finally, this research paper highlights TinyML's ability to transform medical diagnostics by creating smart, real-time healthcare that is more available, efficient, and durable.

References

[1] World Health Organization, "Cardiovascular Disease (CVD)," 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ cardiovascular-diseases-(cvds).

[2] C. L. Listers, T. K. B., E. Bejnordi, et al., "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60–88, 2017.

[3] P. Warden and D. Situnayake, TinyML: Machine Learning with Tensor Flow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media, 2019.

[4] M. A. Alayarajah and T. Meya, “Efficient data mining method to predict the risk of diabetes mellitus using deep learning,” International Journal of Cognitive Informatics and Natural Intelligence, vol. 10, no. 4, pp. 11–18, 2020.
[5] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, et al., “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network,” Nature Medicine, vol. 25, no. 1, pp. 65 69, 2019.
[6] S. K., J. H., D. V. Tirado, and A. K. Jha, “TinyML for healthcare: A review of current trends and applications,” Journal of Healthcare Engineering, vol. 2022, Article ID 9951185, 2022.
[7] J. W. Smith, J. E. Everhart, W. C. Dickson, C. W. Knowler, and R. S. Johannes, “Using the ADAP learning algorithm to forecast the onset of diabetes mellitus,” in Proceedings of the Symposium on Computer Applications and Medical Care, 1988, pp. 26–35.
[8] P. Wagner, N. Strothoff, R. D. Bosselut, W. Samek, and T. Schaeffer, “PTB-XL: A large publicly available electrocardiography dataset,” Scientific Data, vol. 7, 2018.
[9] S. I. Khandaker and M. M. Islam, “Diabetes prediction: A deep learning approach,” International Journal of Information Engineering and Electronic Business, vol. 11, no. 2, pp. 21–27, 2019.
[10] D. Sioda and D. S. Sisodia, “Predicting diabetes using classification algorithms,” Procedia Computer Science, vol. 132, pp. 1578–1585, 2018.
[11] B. Jacob, S. Kilgys, B. Chen, M. Tan, A. Howard, H. Adam, and D. Kalenichenko, “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2204–2213.
[12] S. Han, H. Mao, and W.J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” in Proceedings of the International Conference on Learning Representations (ICLR), 2016.
[13] M C. Banbury, C. Zhou, I. Fedorov, R. Navarro, U. Thakker, D. Gope, V. J. Reddi, et al., “MLPerf Tiny benchmark: Measuring performance of tiny machine learning systems,” in Proceedings of the Neural formation Processing Systems (NeurIPS) Datasets and Benchmarks Track, 2021.
[14] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, and F. Kawsar, “DeepX: A software accelerator for low-power deep learning inference on mobile devices,” in Proceedings of the 14th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2015. pp. 23–34.
[15] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, M. Sun, P. Sundberg, H. Yee, K.Zhang, G. E. Duggan, G. Flores, and J. Dean, “Scalable and accurate deep learning with electronic health records,” npj Digital Medicine, vol. 1, no. 1, pp. 1–10, 2018
[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1097–1105.