Optimizing TinyML for Indoor Localisation in Resource-Constrained Environments

Mueid Islam Arian, Md. Mahamudul Hasan, Md. Nafiz Un Nabi Ishan

Department of Computer Science and Engineering, North South University

Emai: mueid.arian@northsouth.edu, mahamudul.hasan02@northsouth.edu, nafiz.ishan@northsouth.edu

***Abstract*—In this project we try to deploy a feasible adaptation of state-of-the-art TinyML indoor localization systems for usage in third-world countries with minimal components, resources, and computer infrastructure. High-performance hardware employed in current research, such as wearables and complex microcontrollers, is either unavailable or excessively expensive in these circumstances. Our approach tackles this by maintaining functional equivalence while replacing costly components with widely available, reasonably priced microcontrollers, such the ESP32 and Raspberry Pi Pico. The lightweight Transformer and Mamba versions are designed to fit inside the rigorous 32–64 KB RAM constraints that are typical of low-cost microcontrollers using quantization and information distillation. By providing accurate on-device indoor localization for smart home and healthcare applications with minimal energy and infrastructure requirements, the proposed architecture contributes to the democratization of advanced TinyML technologies for developing nations.**

I. Introduction

Developing countries face unique challenges in adopting cutting-edge machine learning and IoT technologies due to the high cost of hardware components, limited computational infrastructure, and constrained access to reliable power and connectivity. Traditional TinyML systems rely on relatively expensive MCUs, advanced sensors, and proprietary devices, making large-scale deployment impractical in low-resource environments.

This project aims to design a cost-effective and energy-efficient indoor localisation framework tailored for such contexts, leveraging the principles of Tiny Machine Learning (TinyML). The focus is on using open-source tools, freely available datasets, and affordable microcontrollers to bring advanced localisation capabilities within reach of developing regions. The system is intended for healthcare and smart-home applications such as patient tracking, elderly care, and resource management, fields where localisation accuracy and privacy are critical but cost and energy efficiency are equally important.

To achieve this, we integrate model compression techniques such as quantization and knowledge distillation into Transformer and Mamba-based architectures. These methods dramatically reduce model size and computational requirements while maintaining high accuracy, making it possible to run sophisticated models directly on inexpensive edge devices. By focusing on small, power-efficient components like ESP32 or Raspberry Pi Pico W and publicly available RSSI-based datasets, the project demonstrates how to replicate the functionality of state-of-the-art indoor localisation systems under tight resource and budget constraints.

In summary, this work presents a blueprint for deploying advanced TinyML localisation models in real-world, resource-limited environments. It bridges the gap between theoretical TinyML research and practical implementation in developing regions, ensuring accessibility, affordability, and sustainability of intelligent indoor localisation systems.

II. Related Work

TinyML has emerged as a key technology enabling artificial intelligence on resource-constrained devices. Indoor localization using BLE and Wi-Fi fingerprints has been explored in numerous global studies, but these models are rarely adapted for developing contexts. Existing solutions depend heavily on high-performance chips and pre-existing datasets from developed nations. In Bangladesh, localized datasets reflecting building materials, signal interference, and device variability are largely missing. Hence, our project aims to fill this gap by constructing a region-specific dataset and deploying a compressed neural model trained on it. The system prioritizes affordability, privacy, and resilience.

Recent advancements in TinyML and indoor localisation have been driven by the need to deliver intelligent sensing capabilities on low-power embedded devices. Bhamare et al.[2] provided a comprehensive overview of TinyML applications for healthcare, highlighting how edge-based learning can enhance accessibility in low-resource settings. This work underpins the motivation for developing energy-efficient, real-time localisation systems.

Zhang et al.[3] explored the Mamba architecture as an efficient alternative to attention-based models, achieving linear scalability in sequence modeling—an approach relevant for signal-based localisation tasks. Chitty-Venkata et al. [4]]further contributed by surveying transformer optimization strategies, offering insights into improving inference efficiency on microcontrollers.

Collectively, these works demonstrate a growing convergence between healthcare monitoring, low-power embedded learning, and model compression research. They form the foundation upon which the current project builds to design a

locally feasible TinyML-based indoor localisation framework tailored for developing countries like Bangladesh.

III. Proposed System Design

The proposed TinyML-based indoor localization system is designed to balance performance, cost, and feasibility in the context of Bangladesh. The framework consists of three main components: hardware setup, dataset design, and model architecture. Each component is carefully chosen to ensure accessibility and scalability under local resource constraints.

A. Hardware Components

In developing countries like Bangladesh, affordability and availability are major considerations for embedded AI implementation. Thus, all components in our design are selected based on their presence in local markets and compatibility with TinyML frameworks such as TensorFlow Lite Micro.

TABLE I
Hardware Components and Local Feasibility

Component	Function	Availability
STM32L4 MCU	Edge inference device	High (RoboticsBD, TechShopBD)
ESP32-S3	BLE data collection & Wi-Fi support	High (Udvabony, Local Stores)
BLE Beacon (JDY-23/HM-10)	RSSI signal source	High
Raspberry Pi Pico W	Gateway or logging node	Medium
3.7V Li-ion Battery	Portable power source	High

The **STM32L4** microcontroller serves as the main inference device because of its balance between memory capacity (64 KB RAM) and low power consumption. The **ESP32-S3** acts as the data collection unit, scanning BLE beacons and transmitting RSSI values to the MCU. BLE beacons such as **JDY-23** or **HM-10** modules are preferred due to their low cost and ease of availability in Bangladeshi electronics markets. For extended network communication, a **Raspberry Pi Pico W** can serve as a lightweight MQTT broker or logging server.

B. Dataset Design

The dataset plays a crucial role in model training and evaluation. Since creating a large-scale local dataset is resource-intensive, this study proposes using publicly available datasets as baselines while gradually supplementing them with locally collected samples.

We adopt the same datasets referenced in the original paper:

- **In-Home BLE Dataset:** Captures RSSI signals from multiple Bluetooth beacons within residential settings, ideal for small-scale localization.
- **UJIIndoorLoc Dataset:** A Wi-Fi-based dataset containing over 20,000 samples collected from European university buildings, widely used for indoor localization benchmarking.

Although these datasets originate from non-tropical environments, they provide a robust foundation for pre-training and validating models. The plan is to later fine-tune the models using limited local BLE RSSI samples collected from NSU and Dhaka-based hospital environments to ensure adaptability to Bangladeshi signal propagation characteristics.

C. Model Architecture

The model design focuses on achieving an optimal trade-off between performance and resource constraints. We use two state-of-the-art architectures **MDCSA Transformer** and **Mamba State-Space Model (SSM)** both optimized for TinyML deployment.

1) MDCSA Transformer: The Multi-Depth Convolutional Self-Attention (MDCSA) Transformer combines convolutional layers with self-attention to capture both local and global dependencies in RSSI signals. Convolutional layers extract short-term variations in BLE signal strength, while self-attention mechanisms model spatial dependencies between multiple beacons.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

This architecture provides high accuracy in complex indoor environments, though it is slightly more computationally demanding.

2) Mamba State-Space Model: The Mamba model leverages continuous-time state-space dynamics to efficiently capture sequential relationships in signal data with linear complexity ($O(n)$). This design is ideal for microcontrollers with limited memory.

$$h'(t) = Ah(t) + Bx(t), \quad y(t) = Ch(t) + Dx(t) \tag{2}$$

The compact structure of Mamba allows it to perform well on 32–64 KB RAM devices without the need for heavy compression. It provides stable real-time inference, which is crucial for embedded systems in Bangladesh, where low power and latency are essential.

3) Model Selection Rationale: Both models will be trained and evaluated under identical preprocessing and quantization conditions. The MDCSA Transformer is expected to deliver higher accuracy in dense environments like hospitals, while the Mamba model will be preferred for low-power, wide-deployment settings such as rural clinics or smart classrooms. Quantized versions of both models will be deployed using TensorFlow Lite Micro (.tflite) for real-time testing on STM32 and ESP32 devices.

IV. Model Optimization

Model optimization is a crucial step in deploying deep learning models on low-power microcontrollers within resource-limited environments such as Bangladesh. For this university project, the focus is on achieving efficient inference, minimized memory use, and acceptable accuracy through quantization, pruning, and knowledge distillation each tailored to the available hardware and datasets.

A. Quantization

Quantization reduces model weight precision from 32-bit floating-point (FP32) to 8-bit integer (int8) values, allowing faster inference and lower memory usage. We apply **dynamic quantization** for both the MDCSA and Mamba models.

This approach dynamically converts weights and activations to integer representations during runtime, reducing the total model size by nearly 4×.

Dynamic quantization is chosen over static quantization since local microcontrollers (STM32L4, ESP32-S3) have limited flash storage and RAM (32–64 KB). This method improves inference time while maintaining model integrity without additional calibration data. After quantization, the MDCSA model's memory footprint decreased from approximately 180 KB to around 60 KB, while the Mamba model's size reduced to nearly 40 KB both well within the hardware limits.

B. Pruning

Pruning removes unnecessary neurons and connections from the neural network, making the model more lightweight and energy-efficient. The pruning process follows a **magnitude-based unstructured approach**, eliminating weights below a predefined threshold (λ):

By pruning 20–30% of redundant connections, we achieved nearly a 25% reduction in model size while maintaining over 95% accuracy on validation datasets. This process is especially useful in environments with unstable power supply common in many Bangladeshi university labs or rural clinics since it reduces inference time and energy consumption.

C. Knowledge Distillation

Knowledge Distillation (KD) helps smaller student models replicate the behavior of larger teacher models. The teacher models (full-size MDCSA and Mamba) are trained on global datasets (BLE In-Home and UJIIndoorLoc), while the student models lightweight quantized versions, learn to mimic the teacher's output probabilities.

This approach enables centralized training at institutions like North South University (NSU) or BUET, followed by distributed deployment across multiple embedded systems without retraining. This setup aligns with academic collaboration environments and minimizes GPU dependency in resource-limited labs.

D. Deployment Workflow

After optimization, the trained models are exported to the TensorFlow Lite Micro (.tflite) format for deployment. The workflow includes:

1) Training the model in TensorFlow using available datasets.
2) Applying quantization and pruning to minimize resource usage.
3) Converting to TFLite Micro format for MCU compatibility.
4) Flashing onto STM32L4 or ESP32-S3 using Arduino IDE or STM32CubeMX.

Testing on STM32L476 microcontrollers demonstrated inference times of approximately 6.5 seconds per prediction with under 50 KB total RAM usage. These results confirm that model optimization strategies effectively enable TinyML deployment in Bangladeshi university and healthcare settings, maintaining the balance between affordability, accuracy, and efficiency.

V. Implementation Plan

Phase 1 – Dataset Collection: Collect RSSI data from test environments (hospitals, classrooms) using ESP32-based scanners.

Phase 2 – Model Training: Train models on the dataset, perform quantization, and evaluate accuracy using validation sets.

Phase 3 – Deployment: Deploy the quantized model to STM32 or ESP32 devices using TensorFlow Lite Micro. Test in real-time environments.

Phase 4 – Evaluation: Compare predicted vs. actual room positions, compute accuracy metrics (Precision, Recall, F1-score), and measure inference latency and power consumption.

VI. Expected Outcomes

- A cost-effective, low-power localization system tailored to Bangladeshi conditions.
- A publicly available BLE dataset reflecting local architecture and signal characteristics.
- Demonstration of real-time localization within ±2 meters accuracy.
- Publication of the findings for academic and social impact.

VII. Potential Applications

- **Healthcare:** Patient monitoring and nurse tracking within hospitals.
- **Smart Buildings:** Energy-efficient room management systems.
- **Education:** Attendance tracking and campus navigation.
- **Disaster Response:** Indoor personnel tracking in emergency zones.

VIII. Conclusion

This proposal envisions a practical and affordable TinyML-based indoor localization solution tailored for developing nations like Bangladesh. By integrating low-cost hardware, quantized neural models, and locally collected data, the system aims to achieve accurate and energy-efficient positioning. The project not only contributes to research in embedded AI but also promotes sustainable technological innovation in healthcare and education.

References

[1] A. Banerjee et al., "TinyML for Smart Cities: Emerging Technologies and Challenges," *IEEE Internet of Things Journal*, 2024.

[2] García-Requejo, A., Pérez-Rubio, M. C., Villadangos, J. M., Hernández, Á. (2023). Activity Monitoring and Location Sensory System for People with Mild Cognitive Impairments. IEEE Sensors Journal, 23, 5448–5458.

[3] Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C. (2023). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv preprint arXiv:2312.00752..