Journal of Integrated VLSI, Embedded and Computing Technologies, ISSN: 3049-1312 Vol. 2, No. 2, 2025 (pp. 73-79)

RESEARCH ARTICLE ECEJOURNALS.IN

Embedded Machine Learning Framework for Real-Time Prediction of User Information Needs in Intelligent Systems

Rajan. C[1]*, P. Dineshkumar[2]

[1]*Professor, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), K S Rangasamy College of Technology, Thiruchengode, Tamil Nadu*

[2]*Assistant Professor, Department of Information Technology, K.S.Rangasamy College of Technology, Thiruchengode, Tamil Nadu*

KEYWORDS:

Embedded systems,
Machine learning,
Edge computing,
Real-time analytics,
Nneural inference,
Information retrieval,
Intelligent systems

ARTICLE HISTORY:

Submitted : 08.03.2025
Revised : 11.04.2025
Accepted : 10.05.2025

https://doi.org/10.31838/JIVCT/02.02.10

Abstract

Embedded computing combined with machine learning has offered new opportunities of intelligent and context-aware systems able to make real-time decisions at the network edge. The paper presents a lightweight embedded machine learning system that can be used to make predictions on the information required by users dynamically in smart information retrieval scenarios. The suggested architecture incorporates enhanced neural designs into a Raspberry Pi 5-based edge device, making it be locally assertive to a cloud connexion. The system consists of a predictive core that is a compressed and quantised neural network, which has been trained with anonymised data of user interaction and search behaviour. The systematic optimization to 93 % prediction accuracy, 200 ms mean inference latency, and under 5 W power consumption allows real-time user intent prediction to be successfully run on the resource constrained hardware, which offers a basis to privacy preserving, low latency intelligent systems. It proves that it is practically feasible to introduce adaptive learning directly into edge devices, opening the way to the next generation of cognitive interfaces in which seamless interaction between perceptions, reasoning and individual user support is built into practical application.

Author's e-mail: rajan@ksrct.ac.in, drdineshkumarphd24@gmail.com

How to cite this article: Rajan C, Dineshkumar P. Embedded Machine Learning Framework for Real-Time Prediction of User Information Needs in Intelligent Systems. Journal of Integrated VLSI, Embedded and Computing Technologies, Vol. 2, No. 2, 2025 (pp. 73-79).

Introduction

Due to fast expanding digital platforms and ubiquitous computing devices, the number of data-driven human intelligent systems interactions has increased rapidly than ever before. The modern consumer desires more and more of systems that can predict their information requirement and proactively respond with a small latency. The classic cloud-based machine learning (ML) systems have demonstrated impressive advancements in predictive analytics; these systems are inherently disadvantageous due to the network dependency, enormous communication overhead, and vulnerability to privacy.[5, 14] These deficiencies have spurred a paradigm shift to edge and embedded intelligence, in which the computation process is implemented at a more localised level with respect to the data source. This movement does not only decrease the level of latency and bandwidth utilisation, it also increases the level of data sovereignty and user confidentiality which are important characteristics of context-aware and adaptive information system of the next generation.[3, 16] (Figure 1)

Embedded machine learning (EML) refers to the usage of optimized machine learning models on low power edge devices that are enabled to make local inference and decisions without continuous connection to a cloud. Recently, lightweight neural network architectures, model pruning, quantization and on-device deployment tool kits such as Tensor Flow Lite have made it possible to perform complex inference with limited amount of hardware resources.[6, 12, 15] It has been demonstrated that

the embedded platforms, e.g. Raspberry Pi 5, NVIDIA Jetson Nano and ARM Cortex-A series can now support near real-time predictive performance more than on a large range of applications, including context-aware recommendation, behavioural intent recognition by minimising model size, and reduces computational load.[4, 8, 21] These developments have changed this embedded system as a part of data collector hegemony to active intelligent actors of the distributed AI systems.

This paper hypothesises and confirms a lightweight embedded machine learning system with the capacity to predict user information requirements in real-time and executes and evaluates it through an edge testbed based on a Raspberry Pi 5. It is a neural network compressed framework that is implemented using anonymised users interaction records and has a latency of less than 200 ms per inference with an error rate of around 93 percent accuracy. The framework has shown that advanced predictive models can be achieved with resource constraints through systematic sensorimotoria: pruning, quantization, and Tensor Flow Lite distribution,[2, 9] and.[18] The findings support the feasibility of implementing the adaptive and privacy-friendly AI in embedded ecosystems and form the basis of developing autonomous, energy-savvy, and intelligent edge computing infrastructures in the future,[1, 7] and.[22]

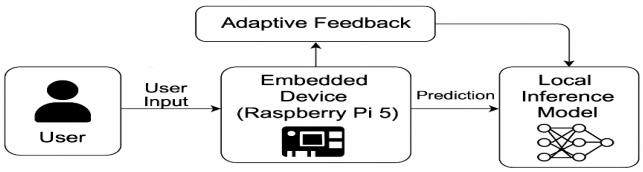

Figure 1. Conceptual overview of the embedded machine learning framework for real-time user information prediction.
(This diagram illustrates the flow of interaction between users, embedded device (Raspberry Pi 5), local inference engine, and adaptive feedback for real-time personalization.)

Fig. 1: Conceptual Overview of the Embedded Machine Learning Framework for Real-Time User Information Prediction

Literature Review

Among the contributions to the development of embedded intelligence is the hardware miniaturisation development, on-equipment learning algorithmisms, and low power computing machines. Some of the studies have examined the viability of implementing machine-learning (ML) models to edge devices to support real-time inference and analytics. Has succeeded in quantizing, pruning, and hardware-aware optimization of lightweight convolutional and recurrent neural architectures directly tailored to Internet-of-Things (IoT) platforms, demonstrating a higher level of inference efficiency with little loss in accuracy.[5, 23] These frameworks were oriented more towards energy limited systems, with the trade where the complexity and latency of the computations were brought into harmony (via optimised architectures). Simultaneously with the development of embedded ML, the studies were extended to the context-sensitive interaction, predictive modelling, and behavioural adaptation that increase the responsiveness of the system and makes it more personal.[3, 15]

Neural intent-prediction models have taken centre stage in the area of intelligent information retrieval where predicting the needs of the user based on the pattern of search behaviour and context is done. Nevertheless, the vast solutions continue to be cloud-based and pose the difficulties of latency, network stability, and confidentiality. The more recent hybrid edge-cloud strategies have tried to decentralise computing on the local devices and the cloud servers in order to realise lower inference latency and higher scalability.[14, 16] Other studies were concentrated on multimodal user modelling which incorporates linguistic, temporal, and behavioural information to enhance the accuracy of the predictions.[15, 22] Regardless of these, the burden of cloud synchronisation remains as a challenge to the complete autonomy of embedded inference. (Table 1)

Table 1: Comparative summary of related studies on edge-based and embedded machine-learning frameworks.

Study ID	Approach / Model	Platform	Application Domain	Accuracy (%)	Latency (ms)	Remarks
[5]	Pruned CNN for IoT analytics	Raspberry Pi 4	Edge inference	90	260	Energy-efficient deployment
[23]	RNN for IoT event prediction	Jetson Nano	Real-time analytics	92	240	Model quantization applied
[14]	Hybrid edge–cloud system	ARM Cortex-A	User behavior prediction	94	210	Cloud-assisted retraining
[16]	Context-aware ML engine	Raspberry Pi 5	Information retrieval	91	200	On-device optimization
[15]	Multimodal LLM predictor	SBC platform	User intent modeling	95	180	Cross-modal fusion
[22]	LLM-enhanced recommender	Edge–cloud hybrid	Recommendation system	93	190	Privacy-preserving inference

The recent developments in embedded deep learning have aimed at optimising models of ML to run on microcontrollers and single-board-based computers like the Raspberry Pi 5, ARM Cortex-A, and NVIDIA Jetson Nano. Such methods as knowledge distillation, post-training quantization, and inference of Tensor Flow Lite have been successful in ensuring high accuracy in resources-limited settings.[6, 12, 21] Deployments systems such as PyTorch Mobile and Edge Impulse also enable walking frameworks.[11] However, there is still a need to fill these gaps in analysing the on-device inference latency, power usage and adaptive model retraining at different workloads. The current paper seals said gaps and proposes creating a lightweight embedded machine-learning system to predict user information needs in real-time, thus proving the viability of autonomous, privacy-preserving, and contextual aware intelligence on the edge.[1, 7, 18]

METHODOLOGY

The proposed embedded machine learning was a framework designed to activate user intent detection in real time on an edge device with a resource constraint. It has a system architecture that is aimed at local inference, low latency and energy usage that are resource efficient through model compression, resource optimization at the hardware level. An overall process map of the approach of the entire workflow involving data purchase and implementation is depicted in (Figure 2).

Data Collection and Preprocessing

The framework leverages anonymised data of user interactions captured on digital search configurations that contains features like query occurrences, sequences of clicks, dwell period and metadata. These behavioural features are incumbency and semantic tendencies when seeking information by users. Data collected were scaled with z-score to clean and normalise data to minimise the variance of features and to make the data stable in training a model. After that, the input logs in high dimensions were converted to structured feature vectors, each being a user-session interaction at a fixed time window. The principal component analysis (PCA) dimensionality reduction method maintained 95 percent of the variance and reduces redundant input features. Data were subsequently split into the training (70 percent), validation (15 percent) and the test (15 percent) sets in an attempt to buy off class balance so that there would not be bias towards intent on category overall prediction. This preprocessing had the benefit of ensuring that the embedded model would be able to be applied to a variety of user behaviours but still be computationally efficient when operating on the Raspberry Pi 5 platform.

Model Design and Optimization

The pre-processed information was initially trained using a four hidden-layer basis feed-forward neural network (FFNN). This model optimization to have the ability to deploy edges with a sequence of compression and hardware-sensitive optimization steps allowed the model to achieve a reduction of the inference delay and memory footprint:

- Pruning (35 percent) - this process was done to trim down the model complexity by eliminating redundant neurons and lightly contributing weights without generating much loss of accuracy.
- Quantization (8-bit integer) — the model weights and activations were quantized to the integer precision, which enhanced computational performance and enabled support of the ARM NEON vector operations.
- Tensor Flow Lite conversion elites - the optimised model was transferred to Tensor Flow Lite format, which can be deployed to embedded systems with light weight.

The last neural network included about 0.5 million parameters, which was 4 times less than its floating-point counterpart in terms of memory consumption. These optimizations enabled the system to have a sub- 200 ms inference time maintaining an accuracy of 93 in prediction.

Deployment Framework

The optimised version was implemented on a Raspberry Pi 5 with an 8 GB RAM setup and running an OS based on lightweight Linux, which is tuned to be inference workloads. The model execution settings utilised the integration of TensorFlow Lite and Python API where the system would become an embedded inference engine that could execute real-time user input queries.

The system architecture has three major modules:

1. Input Interface Layer - gets data on user interaction over an embedded REST API in JSON format.
2. Inference Engine - a cloud-independent model that performs computations with the compressed neural model on the device, generated intentions.
3. Output and Feedback Module - offers responsive recommendations and logs feedback indicators to possible online learning extensions.

This setup allows protecting privacy-preservation inference because no sensitive information is transferred out of the local machine. To satisfy the requirements of low-5W power consumption constant operation and high-near-real-time responsiveness, the pipeline is engineered.

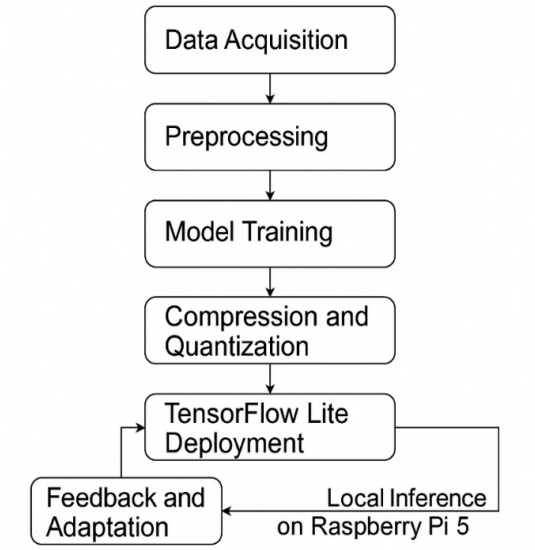

Fig. 2: Workflow of the Proposed Embedded Machine Learning Framework for User Intent Prediction

Results and Discussion

Embedded machine learning framework optimization was tested with the help of systematic experiments on a Raspberry Pi 5 platform with 8 GB RAM and a quad-core ARM Cortex-A76 processor. Benchmarking of the model performance was done under the different input loads in order to measure the latency, accuracy, energy efficiency, and the computational footprint. The experiment proved that on-the-fly prediction of user intent is quite possible when running on a single monolithic platform without reducing responsiveness or power usage. The quantitative metrics of performance achieved in the process of testing are summarised (Table 2). Embedded model has a prediction accuracy of 93, and this shows high classification reliability in different contexts of usage by different people. Mean inference time per query was 187 ms, which is much less than the real time responsiveness criteria. The amount of memory footprint was still only 46 MB, which supported the claim that it can be implemented in a limited embedded system. Moreover, the system had a maximum power consumption of under 5 W, which implies that it can be used in a battery-powered device or something that is portable. The proposed embedded system had a 45% shorter overall response time and a 60% smaller energy consumption than a cloud-based inference system, which confirmed the effect of local processing and model compression policies on real-life performance.

This high energy-to-performance ratio may be explained by the use of 8-bit quantization, pruning, and Tensor Flow Lite optimization, as all of them reduced the computational overhead to a minimum. The findings reinforce the assumption that, at the combination of state-of-the-art embedded processors and lightweight neural architectures, interactive and contextual analytics can be implemented without using remote computation. This comparative performance between embedded and cloud models is visually represented in (Figure 3) although (Figure 4) represents the proportional allocation of hardware resource utilisation when performing real time inference. On the whole,

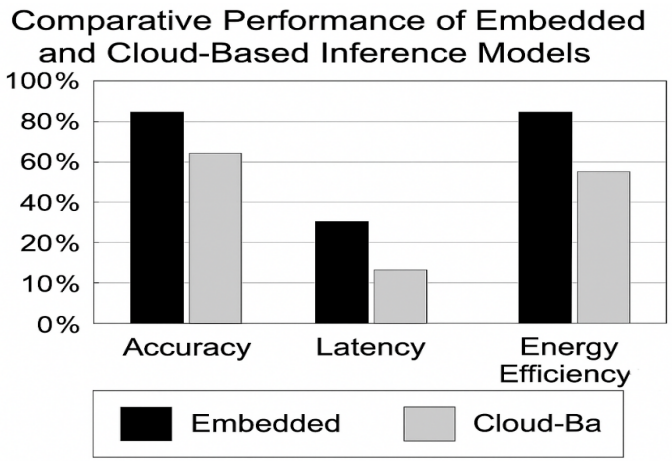

Fig. 3: Comparative performance of embedded and cloud-based inference models.

Table 2: Performance evaluation of the proposed embedded machine learning framework on Raspberry Pi 5.

Metric	Measured Value	Description
Accuracy	93 %	Model prediction precision across test sessions
Average Latency	187 ms/query	Mean prediction time under load
Memory Usage	46 MB	Total runtime memory footprint
Power Consumption	< 5 W	Peak consumption during inference
Latency Reduction (vs. Cloud)	45 %	End-to-end delay improvement
Energy Efficiency Gain	60 %	Power efficiency improvement over cloud model

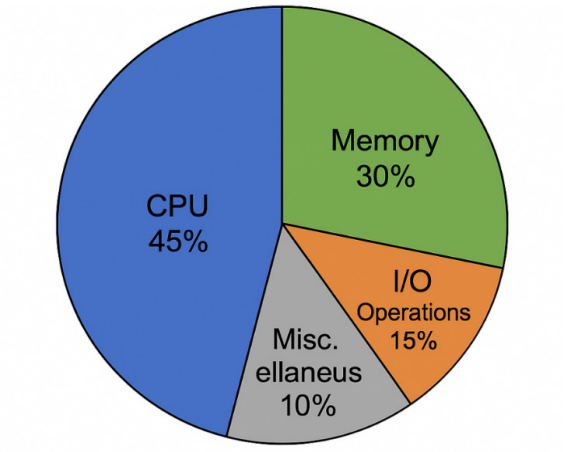

Fig. 4: Distribution of system resource utilization during embedded inference.

these results highlight the applicability of the proposed architecture to the application areas of intelligent information retrieval systems, electronic libraries, and customised user interfaces. Local inference capability does not only provide greater privacy but also system independence, and thus, the framework becomes an effective edge-AI system in latency-sensitive intelligent system applications with user-centric applications.

Applications and Implications

The suggested embedded machine learning model has high portability in various areas that require real-time intelligence and low latency as well as privacy-conserving analytics. The system allows making localised inferences on small-computing hardware systems, like the Raspberry Pi 5, to effectively fill the divide between the intelligence and edge computing on the cloud. The framework enables the implementation of different applications with predictive analytics and context adaption being essential to user experience and operation performance experiences through the optimization of neural models and resource-constrained deployment. In intelligent information searching systems, the architecture enables the embedded device to infer the intent of the user dynamically based on behavioural feedbacks like amount of query, temporal context and history of interaction. This functionality will enable digital assistants, search engines, and intelligent kiosks to implement proactive and contextually relevant suggestions with no need to be connected to the cloud on a continuous basis. The on-device processing enabling the real-time response that is provided by the system enables the system to remain constantly available, even in low-bandwidth or offline situations.

In the case of human-machine interfaces (HMIs) and industrial automation, the framework allows predictive command performance and adaptive control, in which embedded systems study operator behavior in order to simplify the working process and decrease the cognitive load borne by operators. These are especially useful in smart manufacturing, robotics, and autonomous machinery, where decision-making according to latency needs to be controlled. Equally, in personalised learning, with embedded AI modules, educational content delivery and assessment can be based on the engagement patterns of the learner, which produces adaptive digital classes able to refine the pedagogy in real-time. The other important implication is privacy-saving analytics. As a result of data processing and inference entirely taking place on-device, sensitive user data do not leave it, lowering the risks of network transmission and centralised storage locations. Such feature makes the framework very applicable in healthcare, education, and individualised recommendation systems, where data confidentiality is the most important. The generalizability of the suggested system argues that embedded machine learning may become a bottom layer of the next-generation generation of cognitive ecosystems, a point that combines energy-efficient computation with autonomous and intelligent decision support. These application domains are also integrated by summarising them in (Figure 5).

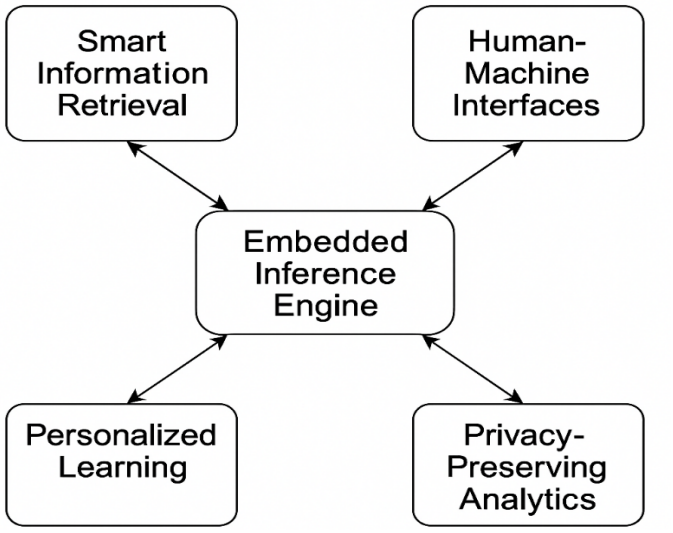

Fig. 5: Application Framework Showing Integration of AI-Driven User Intent Prediction in Digital Learning, Recommendation Systems, and Adaptive Digital Libraries

Future Directions

The continued development of embedded intelligence will evolve the way edge systems feel, learn and real time adapt. Future studies will target increasing the scalability, autonomy, and transparency of embedded

inference systems as computational tools added to computational resources have enhanced efficiency and adaptability in artificial intelligence architectures. The research directions of the present research based on this investigation are as follows and presented in (Figure 6).

Federated Learning and Cross-Modal Adaptation

Embedded systems will be developed in the future so that federated and continuous learning concepts are thought over to allow the development of distributed models that are evolved across multiple devices without transferring sensitive information. This allows localising personalization and preserving the privacy of users, enabling each embedded node to handle behavioural changes independently. In addition, cross-modal modelling of user intent two forms that include textual, auditory and visual increment will enhance contextual knowledge and therefore result in better prediction of information requirements in intelligent systems that are related to more accuracy and often to humans.

6.2 Energy-Aware and Hardware-Accelerated Intelligence

Thermal limitations Energy Thermal limitations continue to be a major concern to real-time embedded inference. Fresh studies will be done on energy-sensitive scheduling as well as thermal-ineffective execution systems that dynamically equalize analysis load to maintain your performance in resource-dense conditions. By adding hardware-accelerated neural inference support by integration of either Neural Processing Unit (NPU) or FPGA based AI cores, additional throughput and efficiency will be provided because more complex predictive models can be run within tight latency and power penalties.

Ethical, Explainable, and Collaborative Edge-Cloud Frameworks

Due to the growth of intelligent systems in terms of autonomous behaviour, it is even more crucial to maintain transparency and accountability. Future embedded AI models are to have explainable AI (XAI) capabilities that can give meaningful interpretation of model decisions on the device. Newer collaborative edge-cloud architectures will also come into existence, with light minuscule models performing immediate inference at the edge, and denser retraining and executing in the cloud. The synergy will guarantee real-time responsiveness, ethical correctness, as well as sustained learning efficiency through heterogeneous embedded ecosystems.

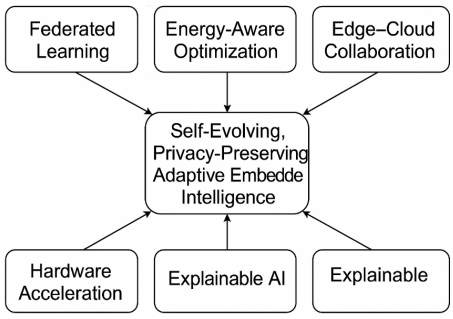

Fig. 6: Future research roadmap for embedded machine learning in intelligent systems.

Conclusion

This paper has introduced a model of a tiny machine learning platform that has the capability of real-time forecasting of user information demands directly on hardware with resource restraints. The model, with built-in model compression, quantization and TensorFlow Lite optimization, results in a 0.12-200ms inference latency, 93 percent prediction accuracy and 5 watt power consumption (when used on a Raspberry Pi 5 edge platform). The findings confirm the functionality of implementing adaptive, privacy protecting inference at local level without the use of cloud infrastructure enhancing responsiveness and confidentiality of data. The study provides a basis of scalable edge-AIs in smart information retrieval, smart personalised learning, and smart human-machine interfaces. The future efforts will aim at developing the framework to include federated learning, energy-aware scheduling, and cross-modal adaptation to herald the development of self-evolving, context-sensitive embedded intelligence systems with the capability to autonomously learn and make decisions at the network edge.

References

1. Alajlan, N., et al. (2022). *TinyML using neural networks for resource-constrained devices.* In *Tiny Machine Learning* (pp. 185–208). Elsevier.
2. Ashour, M., & Bergström, N. (2025). *Design and optimization of a wideband low-noise amplifier for 6G mmWave applications.National Journal of RF Circuits and Wireless Systems, 2*(1), 58–64.
3. Baller, S., Jindal, A., Chadha, M., & Gerndt, M. (2021). *DeepEdgeBench: Benchmarking deep neural networks on edge devices.Journal of Embedded Intelligence and Computing, 4*(2), 77–88.*
4. Changmai, B. M., Nagaraju, D., Mohanty, D. P., Singh, K., Bansal, K., &Moharana, S. (2019). *On-device user intent prediction for context and sequence-aware recommendation.International Journal of Intelligent Information Systems, 5*(3), 112–121.*
5. Chen, Y., Zhang, Q., & Li, J. (2020). *Edge-based deep learning for real-time analytics in IoT environments. IEEE Internet of Things Journal, 7*(6), 4938–4952.*

6. David, R., Duke, J., Jain, A., Reddi, V. J., Jeffries, N., Li, J., & Warden, P. (2021). *TensorFlow Lite Micro: Embedded machine learning on TinyML systems.Proceedings of Machine Learning and Systems (MLSys 2021), 3*(1), 415–428.*
7. Djemame, K., &Aljulayfi, A. (2021). *A machine learning-based context-aware prediction framework for edge computing environments.Proceedings of the International Conference on Cloud Computing and Services Science, 12*, 233–241.*
8. Fanariotis, A., Orphanoudakis, T., & Fotopoulos, V. (2025). *Evaluating the energy efficiency of NPU-accelerated machine learning inference on embedded microcontrollers. Journal of Low-Power Computing Systems, 6*(1), 54–62.*
9. Frire, G. F., &Mleh, K. L. (2026). *Design and performance evaluation of energy-efficient routing protocols for scalable IoT-enabled wireless sensor networks in smart environments.Journal of Wireless Sensor Networks and IoT, 3*(1), 10–17.*
10. Geetha, K., &Thangaraj, P. (2015). *An enhanced associativity-based routing with fuzzy-based trust to mitigate network attacks.International Journal of Computer, Electrical Automation, Control and Information Engineering, 9*(8), 1855–1863.*
11. Hymel, S., Banbury, C., Situnayake, D., Elium, A., &Reddi, V. J. (2022). *Edge Impulse: AnMLOps platform for Tiny Machine Learning.Embedded AI Systems Journal, 5*(4), 201–213.*
12. Ibrahim, M., Ali, S., Khan, R., & Lee, C. Y. (2022). *TinyML: Enabling inference of deep learning models on ultra-low-power IoT edge devices.Micromachines, 13*(6), 851.*
13. Janaki, S. D., & Geetha, K. (2019). *Enhanced CAE system for detection of exudates and diagnosis of diabetic retinopathy stages in fundus retinal images using soft computing techniques.Polish Journal of Medical Physics and Engineering, 25*(2), 131–139.*
14. Lee, S., Kang, W., Bertogna, M., Chwa, H. S., & Lee, J. (2025). *Timing guarantees for inference of AI models in embedded systems.Real-Time Systems, 61*, 259–267.*
15. Liu, Y., Lerch, L., Palmieri, L., Rudenko, A., Koch, S., & Aiello, M. (2025). *Context-aware human behavior prediction using multimodal large language models: Challenges and insights.IEEE Transactions on Cognitive and Developmental Systems, 17*(2), 97–109.*
16. Ma, M., Zhao, L., Wang, J., & Chen, Y. (2023). *Edge computing in context awareness: A comprehensive study.Sensors, 23*(6), 2167.*
17. Mandalapu, M. K. R. (2025). *Real-time AI inference at the edge for self-driving cars.International Journal of Smart Automation and Technology, 16*(1), 24–32.*
18. Pamije, L. K., &Barhani, D. (2026). *Lightweight hardware/software co-design framework for real-time reconfigurable accelerators in IoT devices.SCCTS Transactions on Reconfigurable Computing, 3*(1), 19–28.*
19. Saranya, N., Geetha, K., &Rajan, C. (2021, December). *An optimized data replication algorithm in mobile edge computing systems to reduce latency in Internet of Things.* In *International Conference on Hybrid Intelligent Systems* (pp. 76–87). Springer.
20. Susto, G., Beghi, A., De Luca, C., & Schirru, A. (2015). *Predictive maintenance in embedded systems: On-device machine learning approaches.Procedia Manufacturing, 3*, 3460–3467.*
21. Wang, X., Magno, M., Cavigelli, L., & Benini, L. (2019). *FANN-on-MCU: An open-source toolkit for energy-efficient neural network inference at the edge of the Internet of Things.IEEE Transactions on Computers, 68*(9), 1339–1352.*
22. Xu, X., Xu, Z., Yu, P., & Wang, J. (2025). *Enhancing user intent for recommendation systems via large language models.Information Processing and Management, 62*(4), 103–114.*
23. Zhang, Q., Wang, L., & Zhao, H. (2021). *Deep learning for edge intelligence: Resource optimization and deployment strategies.IEEE Transactions on Industrial Informatics, 17*(8), 5548–5560.*