AI at the Edge: Trends and Innovations in Tiny Machine Learning Models for IoT and Embedded Systems in Synergy with Neuton.AI

Aneta Trajkovska[1] and Aleksandar Markoski[1]

[1]University "St. Kliment Ohridski", Faculty of Information and Communication Technologies, Bitola, Republic of Macedonia

aneta.trajkovska@uklo.edu.mk, aleksandar.markoski@uklo.edu.mk

Abstract:
The trajectory of technological evolution is increasingly oriented towards the development of intelligent solutions that enhance both the efficiency and functionality of everyday life. As technological advancements accelerate, we are witnessing a paradigm shift in the execution of technical processes, aimed at simplifying device interactions while simultaneously enhancing control and automation. The rise of AI at the edge is revolutionizing the way we approach machine learning in the context of IoT and embedded systems. Edge AI, which brings the power of machine learning to edge devices, allows for real-time data processing and decision-making, enabling devices to operate independently of cloud-based systems. This innovation is crucial for applications requiring low-latency responses, such as autonomous vehicles, smart cities, and industrial automation. The convergence of AI, IoT, and edge computing is thus driving significant innovation in embedded systems, with trends indicating a growing emphasis on lightweight machine learning models, energy-efficient algorithms, and scalable architectures. In this paper, we will conduct an in-depth exploration of the utilization of TinyML systems, focusing particularly on practical case studies and best practices associated with neuton.ai. By examining practical use cases of neuton.ai, we will highlight its contributions to advancing the field, including innovations in model optimization, scalability, and real-world deployment strategies.

Keywords:
Internet of Things, AI, Edge intelligence, neuton.ai, tiny machine learning,

1. Introduction

The evolution of the Internet of Things (IoT) can be framed as an integral part of the successive industrial revolutions that have transformed societies and economies. Each phase of industrial development has seen the introduction of new technologies that enhance productivity, efficiency, and connectivity [1],[2]. IoT, as we know it today, can be understood as a product of the Fourth Industrial Revolution, but its foundational concepts trace back to earlier transformations in industrial history Figure 1. Each revolution brought about a new paradigm of connectivity and automation, and IoT represents the latest stage in this trajectory. By integrating data, devices, and systems, IoT is not only enhancing industrial processes but also transforming how humans interact with the world around them. The convergence of IoT and AI is driving a new wave of industrial and societal transformations, enhancing system capabilities and human-machine collaboration. [3].

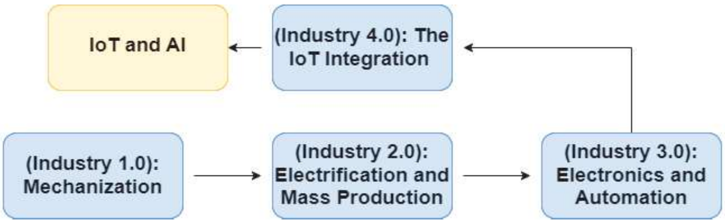

IoT and AI
(Industry 4.0): The IoT Integration
(Industry 1.0): Mechanization
(Industry 2.0): Electrification and Mass Production
(Industry 3.0): Electronics and Automation

Figure 1: The four industrial revolutions and current progress

The integration of machine learning into resource-constrained devices has led to the emergence of TinyML, a field dedicated to enabling intelligent data processing and decision-making at the edge. As the Internet of Things (IoT) continues to expand, the demand for efficient, low-latency, and energy-conscious AI solutions has become critical. TinyML addresses these demands by allowing machine learning models to operate on small, embedded devices with minimal computational power, reducing the need for continuous cloud connectivity and optimizing real-time performance [4].

One of the leading innovations in this domain is neuton.ai, which provides tools and frameworks for developing highly efficient TinyML models without the complexity of traditional machine learning pipelines [5],[6]. By automating model generation and optimizing performance for edge devices, neuton.ai has demonstrated practical applications across various industries, including healthcare, agriculture, and industrial automation. These advancements allow for real-time data analysis and decision-making directly on the device, minimizing latency, bandwidth usage, and energy consumption.

This paper explores the practical utilization of TinyML systems, with a specific focus on neuton.ai's capabilities [7]. It aims to provide a detailed analysis of current trends, challenges, and best practices for deploying TinyML in embedded systems, addressing the implications for both technological innovation and real-world applications.

2. Technological Trends and Innovations

Technological trends and innovations in TinyML are driving significant advancements in the fields of IoT and edge computing. One major trend is the development of increasingly efficient hardware, such as low-power microcontrollers and specialized AI accelerators, which enable real-time data processing directly on edge devices. Model optimization techniques, including quantization and pruning, have also emerged as key innovations, allowing machine learning models to operate with minimal computational and energy requirements. Platforms like neuton.ai are advancing the automation of TinyML model creation, enabling the deployment of highly optimized models on resource-constrained devices without requiring deep expertise in machine learning Figure.2.

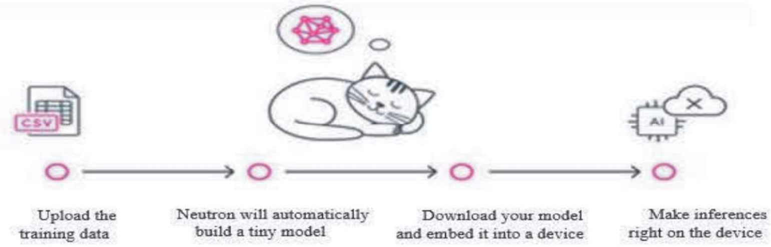

CSV
AI
Upload the training data
Neutron will automatically build a tiny model
Download your model and embed it into a device
Make inferences right on the device

Figure 2: Neuton.AI high architecture process overview [8]

Moreover, innovations in lightweight neural architectures and energy-efficient algorithms are addressing the challenges of deploying AI on embedded systems [9]. These trends are expanding the applicability of TinyML across diverse sectors, from healthcare and smart cities to industrial automation, by enabling real-time, decentralized decision-making. The convergence of these innovations is shaping the future of AI at the edge, allowing for more scalable, secure, and intelligent IoT ecosystems.

2.1. IoT and AI integration for embedded systems

A key development is the advancement of energy-efficient hardware, such as microcontrollers and edge processors, designed specifically to handle machine learning tasks directly on devices with limited computational resources. This evolution is critical for real-time, decentralized data processing, reducing reliance on cloud-based infrastructures and lowering latency. In parallel, model optimization techniques like quantization, pruning and neural architecture search have enabled the deployment of complex AI algorithms on constrained devices. Platforms such as neuton.ai are playing a pivotal role in automating the creation and deployment of highly efficient TinyML models, allowing embedded systems to execute advanced AI tasks without requiring extensive expertise or large computational overhead [10],[11].

The integration of AI and IoT in embedded systems has led to innovations in autonomous decision-making, particularly in industries such as healthcare, agriculture, and industrial automation. The ability to process data locally enables enhanced privacy, reduced bandwidth usage, and faster response times, making these systems highly adaptive for real-world applications Figure 3. As these trends continue, the convergence of IoT, AI, and TinyML is shaping the future of intelligent, low-power embedded systems, offering scalable and efficient solutions for next-generation smart environments [12].

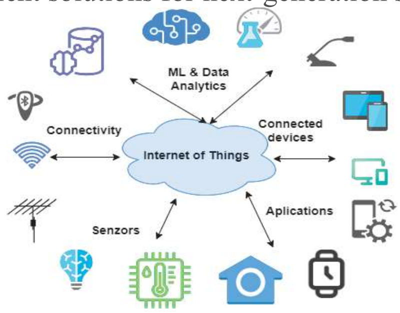


Figure 3: IoT device engineering

3. Tiny Machine Learning

TinyML is rapidly advancing field within machine learning, specifically focused on enabling AI capabilities in resource-constrained embedded systems. It represents a transformative approach to deploying machine learning models on low-power, memory-limited devices such as microcontrollers, which are integral to the Internet of Things (IoT) ecosystem. This development allows real-time, on-device data processing and decision-making, reducing the need for constant cloud connectivity, and improving both efficiency and privacy. TinyML has wide-ranging applications across industries, including healthcare, smart cities, agriculture, and industrial automation [13],[14].

The growing adoption of TinyML underscores its potential to reshape the landscape of AI and embedded systems, offering scalable, energy-efficient solutions for real-time, intelligent decision-making across a wide range of applications.

4. Neuton.AI and its Features

Neuton AI is a cutting-edge platform that simplifies the development and deployment of machine learning models. Neuton AI leverages an automated machine learning approach, allowing users to create highly efficient models without requiring deep technical expertise in machine learning or data science. The platform is designed to address the challenges of deploying machine learning on resource-constrained devices, such as microcontrollers and embedded systems, by optimizing models for low memory usage, reduced computational power, and energy efficiency. With utilization of this platform, easily can be solved regression, classification and anomaly detection [7],[8].

It is commonly used in scientific and engineering fields for integrating machine learning into edge and embedded systems. The list of its diverse capabilities includes:

- **No-code model development** - a key feature of neuton.ai is its no-code environment, which allows users to develop machine learning models without the need for programming. This feature broadens access to AI development, making it more accessible to non-technical users and a wider audience.
- **Lightweight models** - neuton.ai specializes in creating ultra-compact models optimized for deployment on devices with minimal computational resources, such as microcontrollers and edge devices. These models are designed for efficient memory usage and energy consumption, making them well-suited for TinyML applications.
- **Low latency and real-time processing** - the platform emphasizes the development of models optimized for real-time execution, supporting low-latency decision-making on edge devices. This capability is essential for applications requiring immediate responses, including autonomous systems, healthcare monitoring, and predictive maintenance.

- **Data privacy and security** - by facilitating local data processing on devices rather than relying on cloud servers, neuton.ai enhances data privacy and security. This approach minimizes the transmission of sensitive information, addressing privacy concerns in fields such as healthcare and finance.
- **Cross-platform compatibility** - neuton.ai is designed for seamless integration across diverse hardware platforms and environments. Its models are deployable on a range of devices, from microcontrollers to more advanced embedded systems, ensuring flexible deployment options.
- **Energy efficiency** - its models are optimized for minimal energy consumption, which is crucial for battery-powered IoT devices and embedded systems. This efficiency extends the operational life of devices, reducing the need for frequent recharging or battery replacements.
- **Scalability** - supports scalable deployment of models across multiple devices and environments, making it ideal for large-scale IoT deployments where numerous devices require simultaneous data processing.

5. Utilization of Neuton.AI

The practical example of training the model we have created using data for Transport Type Detection as a source data. Next phase was training the pipeline where the model type was Multi Classification (used to predict one value of the limited number or of possible outcomes). Focus was placed on maximizing accuracy as the evaluation metric, with a training duration limited to 4 hours. The model was optimized for deployment on Intelligent Sensor Processing Units (ISPU).

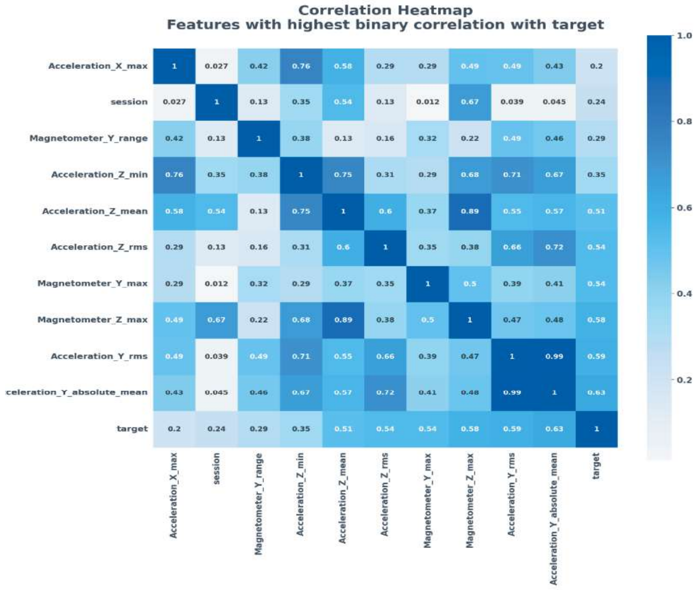


Figure 4: Heatmap of the features with highest binary correlation with target

After the phase of training the pipeline, next stage is results reviewing. Exploratory data analysis shown that for training we used data set dimensions with rows: 6210, columns:101, size memory:2.44mb. The data set based on the variables was split into 8 classes, above on the Figure 4 is shown the correlation heatmap (value 1 is indicating perfect correlation of each feature with itself, 0 indicates no correlation and -1 would indicate a perfect negative correlation, but this dataset shows only positive correlations). The top five pairs of features that exhibit high mutual correlation values exceeding 0.7, Figure 5.

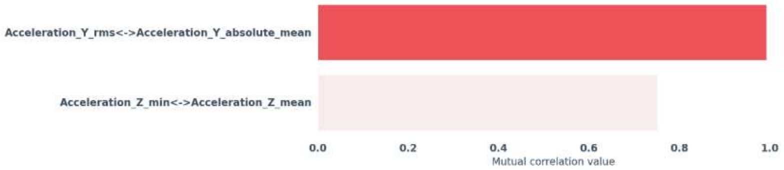


Figure 5: Columns with high mutual correlation (>0.7) TOP 5

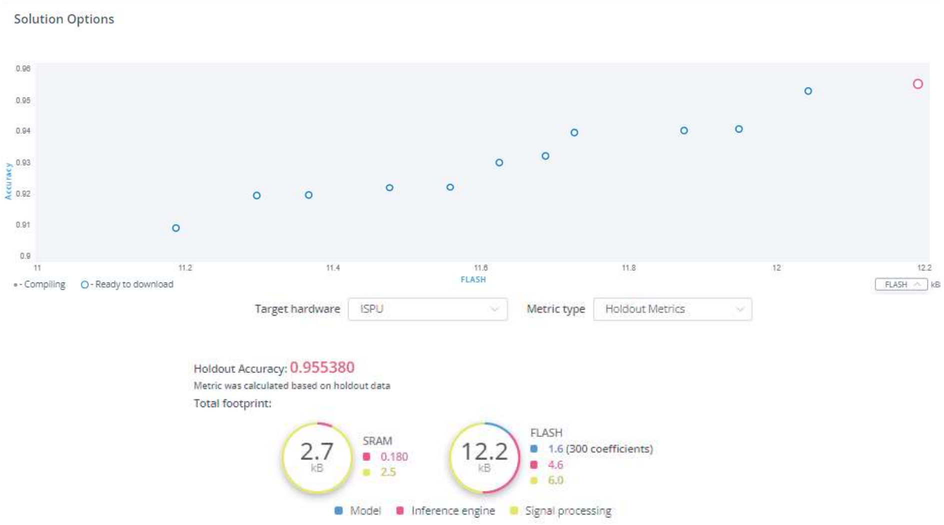
Based on the metric type for the target hardware ISPU, there is an option to select if we want to use the holdout metrics or training metrics (more to read about the difference in Table1).

Table 1:
Key differences between Training and Holdout Metrics

Feature	Training Metrics	Holdout Metrics
Data Used	Training dataset	Holdout(validation/test)dataset
Purpose	Evaluate model learning	Access generalization ability
Implications	Can indicate overfitting	Reflect real-world performance
Common metrics	Accuracy, loss, precision, recall	Accuracy, loss, precision, recall

Both training and holdout metrics are essential for building and validating machine learning models. Training metrics assess how well the model has learned from the training data, while holdout metrics evaluate the model's performance on unseen data, providing a clearer picture of its potential effectiveness in real-world applications.

In our case, with utilization of neuton.ai platform the monitoring of the holdout metrics performance for the model show 0.95% of accuracy and good performance in the FLASH and SRAM (Static Random Access Memory) memory, Figure 6.



Target hardware ISPU

Metric type Holdout Metrics

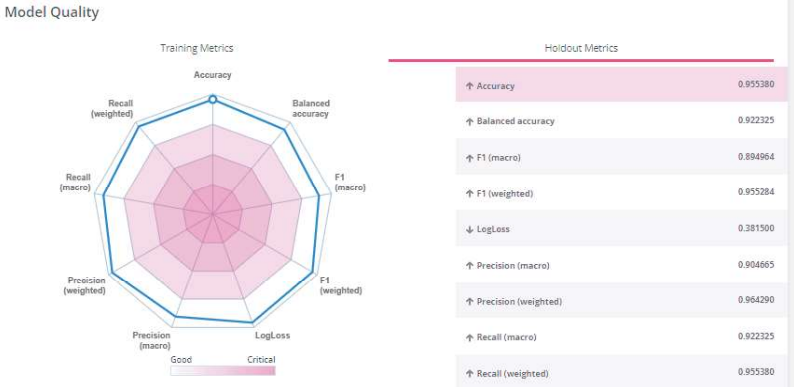
Holdout Accuracy: 0.955380

Metric was calculated based on holdout data

Total footprint:

Figure 6: Performance holdout metrics accuracy

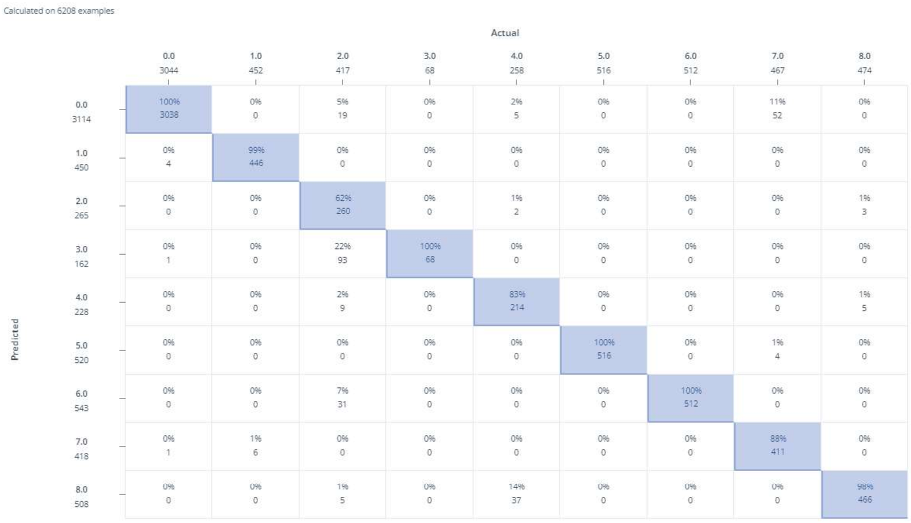
If we focused even more on the model quality, based on the Figure 7 easy we can detect the stable version of the model.



Holdout Metrics	
↑ Accuracy	0.955380
↑ Balanced accuracy	0.922325
↑ F1 (macro)	0.894964
↑ F1 (weighted)	0.955284
↓ LogLoss	0.381500
↑ Precision (macro)	0.904665
↑ Precision (weighted)	0.964290
↑ Recall (macro)	0.922325
↑ Recall (weighted)	0.955380

Figure 7: Analytics of the model

Classification tasks only can be represented in confusion matrix Figure 8, that shows the number of correct and incorrect predictions based on the validation data.



Calculated on 6208 examples

Actual

Predicted

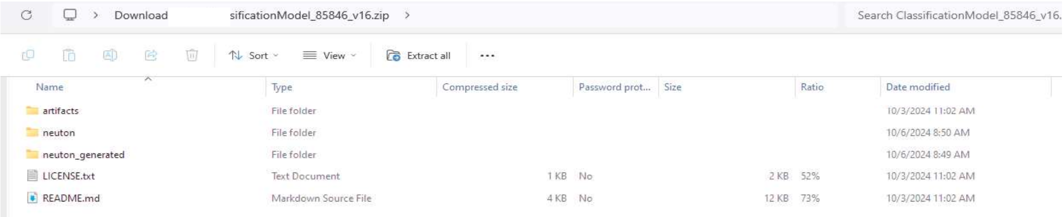
	0.0 3044	1.0 452	2.0 417	3.0 68	4.0 258	5.0 516	6.0 512	7.0 467	8.0 474
0.0 3114	100% 3038	0% 0	5% 19	0% 0	2% 5	0% 0	0% 0	11% 52	0% 0
1.0 450	0% 4	99% 446	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0
2.0 265	0% 0	0% 0	62% 260	0% 0	1% 2	0% 0	0% 0	0% 0	1% 3
3.0 162	0% 1	0% 0	22% 93	100% 68	0% 0	0% 0	0% 0	0% 0	0% 0
4.0 228	0% 0	0% 0	2% 9	0% 0	83% 214	0% 0	0% 0	0% 0	1% 5
5.0 520	0% 0	0% 0	0% 0	0% 0	0% 0	100% 516	0% 0	1% 4	0% 0
6.0 543	0% 0	0% 0	7% 31	0% 0	0% 0	0% 0	100% 512	0% 0	0% 0
7.0 418	0% 1	1% 6	0% 0	0% 0	0% 0	0% 0	0% 0	88% 411	0% 0
8.0 508	0% 0	0% 0	1% 5	0% 0	14% 37	0% 0	0% 0	0% 0	98% 466

Figure 8: Confusion Matrix

After completing the training, C libraries will be prepared for a wider range of hardware types, and source code will be available for enterprise plans. With downloading the "C Library" of the selected model that can integrated it on the device Figure 9.

The "C Library" contains the following files (which is not recommended to be modified, since unsupervised changing of files can cause errors in model inference):

- **Artifacts** – contains models converted to various formats, as well as an executable file for predictions on the desktop.
- **Neuton –** supported libraries for embedding, Cortex M0, Cortex M4, Cortex M33 and STMicro ISPU.
- **Neuton-generated –** contains information necessary for the correct operation of libraries.
- **LICENSE –** contains the possibilities and restrictions on the use of its intellectual property.
- **README –** contains instructions for leveraging libraries.


Download
sificationModel_85846_v16.zip
Search ClassificationModel_85846_v16.
Sort
View
Extract all
...

Name	Type	Compressed size	Password prot...	Size	Ratio	Date modified
artifacts	File folder					10/3/2024 11:02 AM
neuton	File folder					10/6/2024 8:50 AM
neuton_generated	File folder					10/6/2024 8:49 AM
LICENSE.txt	Text Document	1 KB	No	2 KB	52%	10/3/2024 11:02 AM
README.md	Markdown Source File	4 KB	No	12 KB	73%	10/3/2024 11:02 AM

Figure 9: C Library folder structure

6. Future Directions and Limitation

Neuton.AI is increasingly recognized as a leading platform in TinyML and edge computing, prompting the exploration of several emerging future directions. Several future directions are emerging:

- The platform is poised to significantly enhance **AI integration at the edge** with the **expanding IoT landscape**. By supporting a wider range of edge hardware, such as microcontrollers and low-power devices, its application in areas like smart cities, autonomous vehicles and healthcare wearables can deepen human-machine collaboration and advance edge computing.
- **Enhancing real-time data processing** capabilities is crucial for meeting the growing demands of low-latency applications, such as autonomous drones and smart medical devices. Optimizing inference speed and reducing latency will be essential for advancing autonomous systems and supporting mission-critical industries.
- **On-device adaptive learning**, where models evolve from real-time data without cloud retraining, represents a key area of advancement in edge AI.

- It could expand its utility by developing **domain-specific models** tailored to fields like genomics, personalized medicine, and environmental monitoring. This would involve creating customized solutions for handling specialized datasets and optimizing algorithms for specific industry applications [15].

Despite its considerable potential, Neuton.AI encounters several limitations that require attention and resolution [16]:

- **Limited customization for advanced users -** it effectively automates model development for users with limited technical expertise. However, this focus on simplicity may restrict flexibility for experienced machine learning practitioners. Users seeking greater customization or the ability to fine-tune complex models for specific applications may find Neuton.AI automated approach insufficient for their advanced needs.
- **Challenges with ultra-low-power devices** - even highly optimized models may underperform, limiting their applicability in certain edge computing scenarios.
- **Non-Sensor-based data -** its versatility may be constrained when applied to computationally intensive tasks, such as natural language processing (NLP) or computer vision, which often necessitate larger models and greater processing power than typical edge devices can support.
- **Scalability in large-scale edge networks -** scaling across numerous edge devices introduces significant operational challenges, including model update management, device synchronization, and addressing specific hardware limitations. Although Neuton.AI performs effectively in smaller deployments, larger-scale edge AI networks may necessitate more robust infrastructure for efficient model deployment and maintenance.

7. Real-World Application

Today this platform is used wide world for different applications such as: predictive maintenance in industrial IoT (to monitor equipment sensors to detect unusual patterns or vibrations that indicate potential failures to reduce downtime and operational costs in industries), health monitoring devices (process physiological data like heart rate, oxygen levels or ECG patterns directly on the device without needing a cloud connection), smart agriculture sensors (detect soil moisture, temperature and crop health status), environmental monitoring (air quality), smart home and building automation (adjust lighting) and etc. Neuton.AI is adaptable across different sectors, where the platform’s lightweight, cloud-independent approach makes it ideal for real-time, cost-effective AI solutions.

8. Conclusions

Neuton.AI presents significant advantages in enabling AI at the edge, especially through its automation and TinyML capabilities, its future success will depend on overcoming current limitations related to customization, hardware constraints, data security and scalability. As it continues to evolve, addressing these challenges will be essential to broadening its applicability and pushing the boundaries of edge AI. In our paper, we have provided a comprehensive overview of Neuton.AI's capabilities, demonstrating its platform features and evaluating its current strengths. Additionally, we identified future directions and potential innovations, particularly in embedding AI within IoT devices in more efficient and intelligent ways. These advancements promise to unlock new possibilities for real-time data processing and autonomous decision-making at the edge, further enhancing the integration of AI in smart technologies and paving the way for its expanded adoption across diverse industries.

References:

[1] David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry. "IoT Fundamentals Networking Technologies, Protocols and Use Cases for the Internet of things. " Cisco Press (2017): 2017937632.

[2] Radouan A. Mouha. "Internet of Things (IoT)", Journal of Data Analysis and Information Processing (2021): 10.4236/jdaip.2021.92006

[3] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions", Elsevier - Future Generation Computer Systems (2013): https://doi.org/10.1016/j.future.2013.01.010

[4] In Lee and Kyoochun Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises", Elsevier, Business Horizons Volume 58, Issue 4 pp.43-440 (2015): https://doi.org/10.1016/j.bushor.2015.03.008

[5] Norah N. Alajlan and Dina M. Ibrahim. "TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications", Micromachines (2022): https://doi.org/10.3390/mi13060851

[6] M. Giordano, N. Baumann, M. Crabolu, R. Fischer, G. Bellusci, M. Mango. "Design and Performance Evaluation of an Ultralow-Power Smart IoT Device With Embedded TinyML for Asset Activity Monitoring", IEEE Transactions on Instrumentation and Measurement, Volume 71, https://ieeexplore.ieee.org/document/9758676

[7] C. Banbury, V. J. Reddi, A. Elium, S. Hymel, D. Tischler, D. Situnayake, C. Ward, L. Moreau, J. Plunkett, M. Kelcey, M. Baaijens, A. Grande, D. Maslov, A. Beavis, J. Jongboom and J. Quaye. "Edge Impulse: An MLOps Platform for Tiny Machine Learning", Proceedings of Machine Learning and Systems 5 (MLSys 2023): https://proceedings.mlsys.org/paper_files/paper/2023/file/49fe55f5e9574714dda575bfb2177662-Paper-mlsys2023.pdf

[8] Neuton.AI official documentation, https://neuton.ai/, last accessed 2024/09/15

[9] Satyanarayan Kanungo. "Edge-to-Cloud Intelligence: Enhancing IoT Devices with Machine Learning and Cloud Computing ", IRE Journals, Volume 2 Issue 12 (2019): ISSN: 2456-8880

[10] Cosmina M. Rosca. "Convergence Catalysts: Exploring the Fusion of Embedded Systems, IoT, and Artificial Intelligence", Springer Singapore (2024): https://doi.org/10.1007/978-981-97-5979-8_4

[11] Franklin Olivera, D.G. Costa, Flavio Assis, I. Silva. "Internet of Intelligent Things: A convergence of embedded systems, edge computing and machine learning", Internet of Things Elsevier, Volume 26 (2024): https://doi.org/10.1016/j.iot.2024.101153

[12] Z. Zhang, J. Li. "A Review of Artificial Intelligence in Embedded Systems", Micromachines (2023): https://doi.org/10.3390/mi14050897

[13] Syed Ali R. Zaidi, Ali M. Hayajneh, M. Hadeez, Q.Z Ahmed. "Unlocking Edge Intelligence Through Tiny Machine Learning (TinyML)", IEEE Access, vol. 10, pp. 100867-100877, (2022), doi: 10.1109/ACCESS.2022.3207200

[14] Ji Lin, L.Zhu, Wei-Ming Chen, Wei-Chen Wang, Song Han. "Tiny Machine Learning: Progress and Futures [Feature]", IEEE Circuits and Systems Magazine, vol. 23, no. 3, pp. 8-34, (2023), doi: 10.1109/MCAS.2023.3302182.

[15] H. B. Pasandi, F. B. Pasandi, F. Parastar, A. Moradbeikie, T. Nadeem. "Echoing the Future: On-Device Machine Learning in Next-Generation Networks – A Comprehensive Survey", ResearchGate(2023),https://www.researchgate.net/profile/Hannaneh_Barahouei_Pasandi/publication/371139760.

[16] A. Elhanashi, P. Dini, S. Saponara, Q. Zheng." Advancements in TinyML: Applications, Limitations, and Impact on IoT Devices", Electronics (2024): https://doi.org/10.3390/electronics13173562 .