Journal of Soft Computing Paradigm (ISSN: 2582-2640)
www.irojournals.com/jscp/

Survey on Applications, Techniques and Challenges of Machine Learning for Edge Environments

Dinesh Rajassekharan

Associate Professor, School of AI Computing and Multimedia, Lincoln University College, Malaysia.

E-mail: diraja@lincoln.edu.my

Abstract

Edge computing and machine learning have changed a number of applications by extending intelligence and computation toward the data sources. A review of the present machine learning in edge applications is explained in this research focusing on areas such as IoT devices, precision agriculture, smart manufacturing, autonomous cars and healthcare monitoring. Methods like model compression and standard algorithms are used to effectively adapt and implement ML models on limited resource edge devices. The dynamic nature of edge settings, power limitations, data privacy and security, model deployment and administration and limited processing resources are some of the main challenges. This study combines detailed investigations and real-world edge machine learning implementations to address the gap between theory and practice. This study also aims to provide significant data on both the present and future advances of machine learning in edge computing by focusing on potential future applications that may benefit from expanding the fields.

Keywords: Edge Computing, ML-Machine Learning, Edge Environments, AI-Artificial Intelligence, ML Model, IoT Devices.

1. Introduction

1.1 Edge Computing

While edge computing has already pervaded several industries, it is meant to bring computation as close as possible to the "edge" of the network, or the data source. Edge

DOI: https://doi.org/10.36548/jscp-2025.4.002
Received: 12.11.2025, received in revised form: 14.11.2025, accepted: 28.11.2025, published: 07.11.2025

computing, in sharp contrast to conventional cloud computing, brings the processing of data nearer to the devices or edge servers instead of bringing the data to centralized servers for processing and analysis.

This way, a significant reduction in latency and bandwidth consumption indeed characterizes business models for real-time decision applications. Sector by sector, industries such as health care, manufacturing, and automotive are being introduced to such a model to meet their ever-increasing real-time demands. Localized data processing is increasingly becoming a necessity because of the growing number of IoT devices. If more data continues to be generated at the edge, it becomes increasingly difficult to move all information through the network into centralized cloud servers, thereby causing increased congestion and possibly delays.

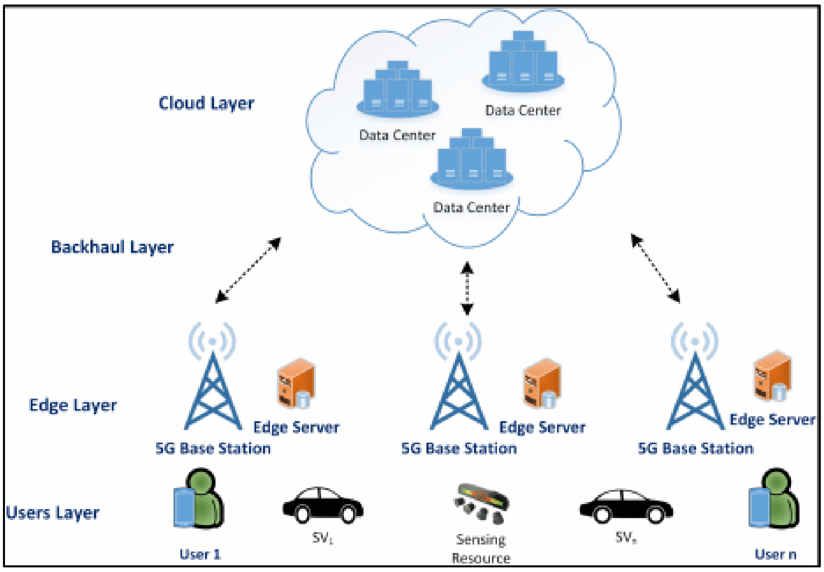

Figure 1. Cloud-Edge Interactions Architecture [16]

Figure 1 demonstrates how computation migrates from centralized cloud servers to distributed edge layers, reducing data transfer overhead. Edge computing presents a unique alternative for solving these problems as it processes data at the data-creation origin point and only sends critical or summarized data to the value-added cloud when necessary. As a result, there is reduced network traffic with faster response times in time-critical applications.

1.2 Machine Learning

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that allows computer systems to acquire knowledge from data without the need for explicit programming to do so.

Rather than being instructed precisely on how to accomplish a task, ML algorithms recognize patterns, derive insights from these patterns, and use these insights to make predictions or decisions regarding new, unseen data. This capability allows machines to continuously improve their performance as they are exposed to more data, making them invaluable for tasks where explicit rule-based programming is impractical or impossible.

The standard machine learning workflow consists of several essential stages

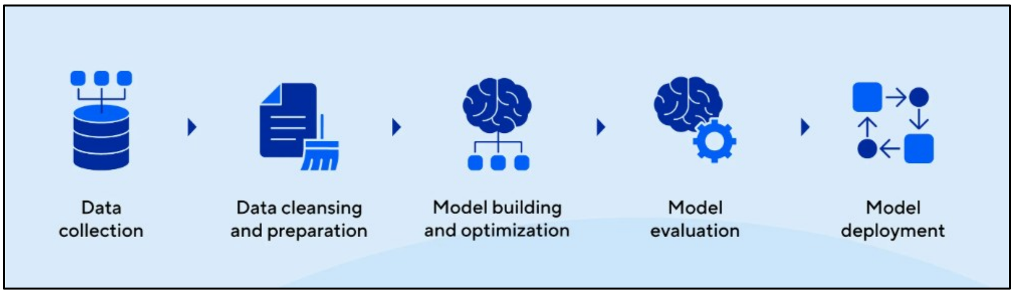

Figure 2. Machine Learning Model [17]

Figure 2 provides a schematic overview of the ML workflow, clarifying how training and inference stages differ for edge deployments. Initially, raw data is collected and subsequently cleaned, transformed, and prepared for analysis. This process may include addressing missing values, normalizing the data, and encoding various data types. The pre-processed data is then utilized to train a selected machine learning algorithm. Throughout the training phase, the algorithm identifies patterns and relationships by adjusting its parameters to reduce the discrepancy between its predictions and the actual results in the training dataset. A distinct dataset (validation or test set) is employed to evaluate the model's performance and verify its ability to generalize to new data. Different metrics are applied to assess accuracy, precision, and other pertinent factors. The model undergoes refinement through methods such as hyperparameter tuning or feature engineering to further improve its accuracy and efficiency. Once a satisfactory model is developed, it can be implemented to generate predictions or make decisions in practical applications.

1.3 Edge ML

Edge machine learning (edge ML) refers to the execution of machine learning algorithms on computing devices located at the edge of a network, enabling decisions and predictions to be made as close as possible to the data's source. This concept is also known as edge artificial intelligence or edge AI. In conventional machine learning, substantial servers typically handle vast amounts of data gathered from the Internet to deliver various advantages, such as recommending the next movie to watch or automatically tagging a cat video.

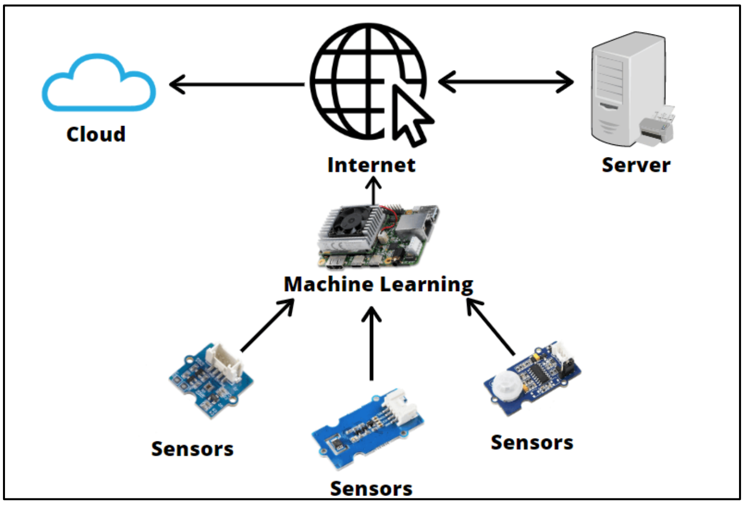

Figure 3. Edge ML – Overview [18]

Figure 3 depicts the Edge ML ecosystem, highlighting data flow between edge devices, gateways, and cloud orchestration layers. By implementing machine learning algorithms on edge devices, including laptops, smartphones, and embedded systems (like those in smartwatches, washing machines, vehicles, manufacturing robots, etc.), we can generate predictions more swiftly and eliminate the necessity of transmitting large volumes of raw data over a network.

1.4 Edge devices

Edge devices refer to hardware components that serve as access points to a network, processing data close to its origin instead of relaying it to a central site for analysis. These devices are commonly utilized in applications such as IoT (Internet of Things), where they gather and send data from multiple sources. Examples of edge devices include sensors, gateways, routers, and specialized equipment like smartphones and medical devices.

2. Literature Survey

Hua et al. [1] define Edge Computing, a favorable computing model, and discuss its problems, traditional solutions, and limitations. They also explore the use of AI to optimize Edge Computing and apply AI to other fields under the Edge Computing architecture, guiding new research ideas and highlighting the mutually beneficial relationship between AI and Edge

Computing. Grzesik et al. [2] discuss edge computing, its benefits like reduced data transmission and latency, and its challenges such as energy consumption and security. They highlight the benefits of combining Machine Learning (ML) with Edge Computing, such as improved data privacy and reduced latency. The paper reviews edge computing platforms and frameworks, focusing on their capabilities in industrial applications, healthcare, smart cities, environmental monitoring, and autonomous vehicles. It also discusses existing use cases in these areas.

Jouini et al. [3] explore an outline of all essential techniques for ensuring the effective implementation of machine learning models, beginning with the available algorithms, frameworks, and even the selection of hardware. This paper offers an extensive analysis of machine learning architectures, algorithms, and criteria for solutions that deploy ML on IoT devices across various processing layers, with the main aim of defining the current state of the art and anticipating future requirements. Paakkonen et al. [4] present a Reference Architecture (RA) design for a big data system using machine learning techniques in edge computing environments, extending an earlier version based on 16 implementation architectures. They also discuss the deployment of architectural elements and provide a system view of software engineering.

Fangxin Wang et al. [5] survey recent studies on edge computing applications powered by deep learning, emphasizing areas such as smart industries, smart transportation, smart multimedia, and smart cities. The research highlights significant challenges and potential avenues for further investigation within these fields. Xiaofei Wang et al. [6] discuss edge computing application scenarios, methods of practical implementation, enabling technologies like DL training and inference, and constraints and upcoming developments of more pervasive and fine-grained intelligence. They aim to consolidate information across communication, networking, and DL areas.

While prior research provides a rich understanding of edge computing and ML integration, critical analysis reveals that most existing works prioritize architectural or algorithmic descriptions over quantitative performance evaluation. For instance, studies such as Hua et al. (2023) and Grzesik & Mrozek (2024) present detailed frameworks but seldom assess scalability or resource efficiency across heterogeneous devices. Moreover, inconsistencies persist in the evaluation metrics used to compare latency reduction, energy efficiency, or model accuracy in edge ML environments. Therefore, this paper identifies a lack of standardized evaluation benchmarks and proposes a structured classification for the comparative analysis of ML algorithms, optimization techniques, and deployment frameworks suitable for diverse edge conditions.

A review of existing literature reveals several research gaps and contradictory findings. For instance, while some studies (e.g., Wang et al., 2020) assert that deep learning at the edge can achieve near-cloud accuracy with optimized models, others (e.g., Chen & Ran, 2019) emphasize substantial trade-offs in latency and energy efficiency. Similarly, federated learning approaches improve privacy but often degrade convergence speed under heterogeneous edge devices. These contradictions underscore the need for a balanced framework that considers both computational and contextual trade-offs, guiding future research toward harmonized performance evaluation and adaptive learning paradigms.

3. Machine Learning Techniques and Applications

ML techniques are typically categorized based on the nature of the data and the problem they are trying to solve.

Table 1. Summary of ML Techniques and Applications

Category	Description	Algorithms	Common Applications
Supervised Learning	Training models on labelled datasets to predict outcomes. Recent progress often focuses on improving model robustness, efficiency, and generalization from limited data.	Logistic Regression, Linear Regression, Support Vector Machines (SVMs), Decision Trees, Random Forests, K-Nearest Neighbors (KNN), Naive Bayes, XGBoost, LightGBM, Neural Networks.	Image/Object Recognition, Spam Detection, Medical Diagnosis, Credit Scoring, Customer Churn Prediction, Natural Language Processing (NLP) tasks like text classification and sentiment analysis, Time Series Forecasting
Unsupervised Learning	Models discover patterns in unlabelled data. Clustering, dimensionality reduction, and generative models are key areas, with self-	Hierarchical Clustering, K-Means, DBSCAN, Apriori algorithm, Gaussian Mixture Models, PCA, ICA	Customer Segmentation, Anomaly Detection (fraud, intrusion), Recommendation Systems, Exploratory Data Analysis, Image

	supervised learning gaining significant traction.		Compression, Topic Modelling
Reinforcement Learning	Agents learn by interacting with an environment, optimizing actions to maximize cumulative rewards. RL has seen remarkable breakthroughs in complex decision-making scenarios.	Q-learning, SARSA, policy gradient methods (like REINFORCE, PPO, TRPO), and actor-critic methods (like A2C, A3C, DDPG)	Robotics Control, Game Playing (e.g., Go, Chess), Autonomous Navigation, Resource Management, Recommender Systems, Algorithmic Trading
Semi-Supervised Learning	Algorithms learn from a combination of a limited quantity of labeled data and a more substantial volume of unlabeled data. This approach can be useful when labelling data is expensive or time-consuming.	Self-training Co-training Label Propagation Generative Models Graph-based methods	Text and Image Classification with limited labelled data, Speech Recognition, Medical Image Analysis
Deep Learning	Uses artificial neural networks with many layers to learn complex patterns and representations from vast amounts of data.	LSTM (Long Short-Term Memory), MLPs (Multilayer Perceptrons), FNNs (Feedforward Neural Networks), GRUs (Gated Recurrent Units), RNNs (Recurrent Neural Networks), CNNs (Convolutional Neural Networks), NLP (Natural Language Processing)	Facial recognition systems, self-driving cars, chatbots, social media monitoring, converting spoken language into text, call center automation.

4. Machine Learning in Edge Computing

ML in edge computing [8] refers to deploying ML models directly on edge devices instead of depending on centralized cloud servers, such as smartphones, IoT sensors, drones, or embedded systems. By storing sensitive data locally, this method improves privacy, lowers latency, saves bandwidth, and permits real-time data processing. ML at the edge is particularly valuable in time-critical applications like autonomous vehicles, industrial automation, remote healthcare monitoring, and smart surveillance [10]. Due to the limited computational resources on edge devices, lightweight models, model quantization, pruning, and hardware acceleration (Example: Using TPUs or NPUs) are commonly employed to optimize performance. As the number of connected devices grows, combining machine learning with edge computing is becoming increasingly essential for building intelligent, responsive, and scalable systems.

Table 2 addresses key advantages of Integrating Machine Learning with Edge Computing and their practical applications [7]. The increasing number of smart devices spread throughout the world and the expanding usages for machine learning and edge computing make it even more important.

Table 2. Advantages of Combining ML and Edge Computing

Advantage	Description	Example
Reduced Latency	ML inference at the edge allows real-time decisions without sending data to the cloud.	Industrial automation, healthcare monitoring, autonomous vehicles
Improved Privacy & Security	Local data processing keeps sensitive information on-device, minimizing transmission risks.	Personal health tracking, security camera footage analysis
Lower Bandwidth Usage	Less need to transmit raw data continuously, conserving bandwidth and reducing network congestion.	Remote environmental sensors, video surveillance
Offline Capability	Edge devices can run ML models without needing internet connectivity.	Rural medical devices, drones in remote areas
Energy and Cost Efficiency	Reduces cloud communication/storage needs, lowering both energy usage and operational costs.	Smart meters, home automation systems
Scalability	Distributed processing across many edge devices eases load on central systems, allowing wider deployment.	Smart city infrastructure, IoT deployments
Context-Aware Intelligence	Edge ML can adapt to local context like user behavior or environment for more relevant decision-making.	Smart home assistants, adaptive traffic systems

4.1 Criteria for Selecting ML Algorithms for Edge Domains

In edge computing, algorithm selection requires several parameters. The model's memory and computational footprint are one of the important factors for algorithm selection. For microcontrollers, IoT devices, or embedded systems, lightweight architectures like MobileNet, SqueezeNet, and ShuffleNet are suggested for their fewer parameters and faster inference times. Complex models using GPUs or NPUs, such as ResNet or Transformer-based models, are suitable for high-capacity edge servers. Another important factor is the application's latency requirement. Algorithms that can provide predictions in milliseconds are needed for real-time applications such as augmented reality, robotic control, and autonomous cars. Ensemble techniques such as Random Forest or Gradient Boosted Trees can be applied for increased robustness and interpretability.

Data sensitivity and privacy requirements are important for algorithm selection. The algorithms should enable privacy-preserving learning techniques in industries such as healthcare, banking, and military fields. Energy efficiency is important for battery operation or energy-constrained systems. Algorithms are designed to achieve low power consumption to ensure long-term operational uptime. The application domain and data characteristics, such as structured vs. semi-structured and static vs. streaming, also play a role. CNN-based models use images or spatial data, while recurrent algorithms work better with time-series or sensor data. In industrial systems, network intrusion detection or anomaly detection is suggested for unsupervised or semi-supervised algorithms. This decision-making process affects system support and hardware compatibility. Portability and deployment effectiveness are two advantages of algorithms that can be executed successfully using edge inference systems.

5. Challenges of Machine Learning for Edge Environments

Edge machine learning is also subject to various challenges [9] owing to inherent hardware limitations, including low battery life, memory capacity and processing power. These limitations restrict the ability to run sophisticated machine learning models, and data handling in edge settings poses specific challenges. Edge data can be sparse, disjointed, and of varying quality, posing privacy and security issues.

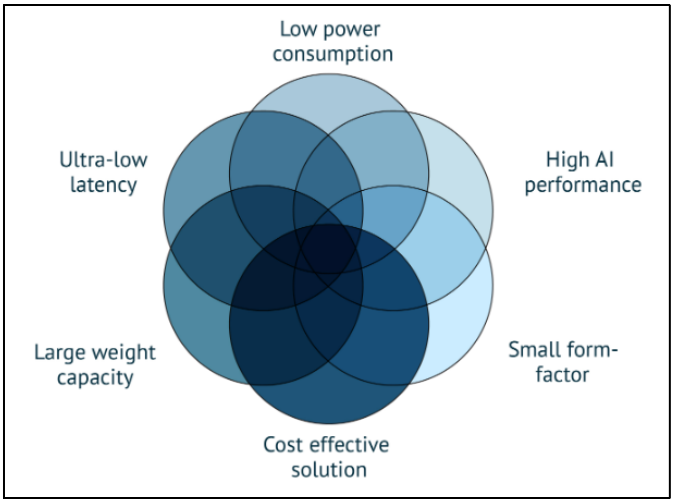

Figure 4. Challenges of ML in Edge [19]

Figure 4 visualizes the interrelation of key challenges, limited computation, power constraints, data privacy, and security vulnerabilities, thereby establishing a base for the optimization frameworks discussed later. A crucial challenge in deploying ML at the edge lies in data availability and quality. Edge environments often generate fragmented, low-quality, or non-IID (non-independent and identically distributed) data streams. Such heterogeneity impacts model accuracy and generalization. Data captured by IoT sensors or mobile devices may suffer from missing values, transmission errors, or inconsistent sampling rates. Moreover, privacy regulations such as GDPR and HIPAA limit centralized data aggregation, necessitating decentralized data cleaning and federated learning solutions. Therefore, developing adaptive preprocessing and on-device validation pipelines is vital for maintaining data reliability in edge ML ecosystems.

To close the gap between resource availability and computational requirements, advanced model optimization and adaptation methods are necessary. Methods such as model compression, quantization, and knowledge distillation are essential for shrinking model size and computational overhead without incurring considerable loss of accuracy [11]. Such compressed models must be adapted and optimized for targeted hardware platforms on edge devices, usually through the use of special processing units or optimized software packages.

Rolling out and managing machine learning models to a large and geographically distributed fleet of edge devices adds a new level of complexity. Orchestration mechanisms are required to deploy a new model or software update smoothly and reliably to all applicable devices [12]. The absence of standardization of hardware and software platforms throughout

ISSN: 2582-2640

the edge ecosystem can make integration and deployment even more complicated. Making the ML applications executed on edge devices reliable and fault tolerant is also important since these devices can operate in harsh environments and have occasional network connectivity issues.

Security and building trust in edge AI are crucial for their mass adoption and acceptance. Edge devices, usually installed in physically accessible sites, are vulnerable to physical compromise and cyber-attacks that seek to undermine the integrity of the ML models or the personal information they handle. Debugging errors, maintaining security, and building user confidence depend on ML models executing on edge devices, making their decision-making process visible. These security problems must be addressed to ensure transparency and accessibility for edge AI systems that are widely utilized and accepted.

6. Case Studies and Practical Implementations in Edge Machine Learning

Case studies discussing the way domain-specific requirements such as ultra-low latency, data privacy and energy efficiency affect the design, optimization and selection of edge ML algorithms show the practical importance of edge machine learning (ML) deployment in real industrial applications [1-15].

6.1 Case Study 1 – Autonomous Vehicles (Automotive Sector)

Autonomous vehicles (AVs) need advanced and highly computational edge machine technologies for safety. On-device deep learning models are used by Tesla, NVIDIA Drive PX and Waymo for analyzing high-resolution inputs from multi-camera, radar and LiDAR sensors. CNN-based architectures with inference latency of less than 20ms per frame such as YOLOv5 and EfficientNet are used for lane tracking and object detection. Model optimization receives the highest priority by these systems, that minimize model size while maintaining 97% detection of accuracy.

6.2 Case Study 2 – Predictive Maintenance in Smart Manufacturing

Siemens and General Electric used edge-devices federated learning systems in Indutry 4.0 environments for real-time anomaly detect and failure prediction in devices. Temperature and vibration data are transmitted to local machine learning nodes via sensors in motor, compressors and turbines. Lightweight models that analyze data directly on edge gateways such as Random Forest classifiers and 1D Convolutional Neural Networks can detect vibration spectrum abnormalities and predict mechanical problems upto 72 hours in advance.

6.3 Case Study 3 – Healthcare Monitoring and Biomedical Edge AI

In the healthcare sector, edge machine learning (ML) has been utilized continuously for anomaly detection, diagnostics and real-time patient monitoring. Wearable technology and bedside monitoring systems from companies like Fitbit, Philips and Medtronic uses embedded machine learning algorithms to measure mobility, blood oxygen, glucose and ECG data. When a cloud connection is not available in the system, the model can identify arrhythmia patterns with an accuracy of 94% and provide on-device notifications. Low-latency medical imaging processes are also accessible by edge-based machine learning.

6.4 Case Study 4 – Smart Energy Grids and IoT-Based Infrastructure

Smart energy grids are distributed devices that perform localized decision-making for identifying defects, load balancing and demand forecasting, serving as ideal examples of edge machine learning applications. Real-time consumption patterns are analyzed by Edge ML models on smart meters and substations that can predict increasing loads with more than 90% accuracy. Startups like AutoGrid and companies like Schneider Electric are using edge inference networks to reduce the requirement for central communication and optimize dynamic energy routing.

6.5 Case Study 5 – Edge Intelligence in Retail and Smart Cities

Edge ML improves resource use, safety and customer experience in urban infrastructure and retail. Real-time tracking, stock prediction and theft prevention are achieved by computer vision models in cameras. Deep reinforcement learning is used by Amazon Go businesses to automate checkout processes. In smart cities, edge sensors with AI capabilities control the collection of waste, monitor pollution, and manage traffic flow. Edge nodes transform traditional networks into responsive, data-driven environments by detecting cars and pedestrians with a latency of less than 30ms.

7. Future Directions

The development of machine learning (ML) for edge environments has the potential to revolutionize a wide range of applications. As edge computing becomes more popular, several future directions are expected to influence the subsequent stages of development and implementation.

Federated and collaborative learning [13] in machine learning improves convergence speed, handles heterogeneous data, and ensure robustness in non-IID scenarios. Research into model compression and efficiency techniques drives advancements in quantization, pruning, knowledge distillation, and neural architecture search, which are automated, adaptive, and fit for specific edge hardware constraints. Future machine learning models for edge devices will require real-time and adaptive inference, refined techniques like early-exit strategies and attention-based mechanisms, enhanced security and privacy [14] through secure enclaves, homomorphic encryption, and differential privacy, and robustness against adversarial attacks. Emerging hardware, like custom edge AI accelerators and neuromorphic computing, will revolutionize energy-efficient and high-speed machine learning processing. Emphasizing self-supervised and unsupervised learning paradigms [15], context-aware AI is crucial for edge environments, as models adapt to changes in the environment, user behavior, and task requirements.

8. Conclusion

Machine learning for edge environments is an innovative change that enables intelligence and autonomy at the edge of the network, which was not previously discovered. Data governance and security issues are major challenges when resources are limited. Continuous research and development in projects are examples of innovative solutions. As a result of the widespread application of Edge ML across multiple sectors, more flexible, secure, and reliable intelligent systems will be produced due to the combination of effective machine learning algorithms, specialized hardware, and distributed learning approaches. It is expected that future research will focus on developing Edge ML systems that are more flexible, secure, and energy-efficient, allowing them to easily adapt to changes in the edge computing environment.

References

[1] Hua, Haochen, Yutong Li, Tonghe Wang, Nanqing Dong, Wei Li, and Junwei Cao. "Edge Computing with Artificial Intelligence: A Machine Learning Perspective." ACM Computing Surveys 55, no. 9 (2023): 1-35.

[2] Grzesik, Piotr, and Dariusz Mrozek. "Combining Machine Learning and Edge Computing: Opportunities, Challenges, Platforms, Frameworks, and Use Cases." Electronics 13, no. 3 (2024): 640.

[3] Jouini, Oumayma, Kaouthar Sethom, Abdallah Namoun, Nasser Aljohani, Meshari Huwaytim Alanazi, and Mohammad N. Alanazi. "A Survey of Machine Learning in Edge Computing: Techniques, Frameworks, Applications, Issues, and Research Directions." Technologies 12, no. 6 (2024): 81.

[4] Pääkkönen, Pekka, and Daniel Pakkala. "Extending Reference Architecture of Big Data Systems Towards Machine Learning in Edge computing Environments." Journal of Big Data 7, no. 1 (2020): 25.

[5] Wang, Fangxin, Miao Zhang, Xiangxiang Wang, Xiaoqiang Ma, and Jiangchuan Liu. "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey." IEEE Access 8 (2020): 58322-58336.

[6] Wang, Xiaofei, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey." IEEE communications surveys & tutorials 22, no. 2 (2020): 869-904.

[7] Li, Wenbin, Hakim Hacid, Ebtesam Almazrouei, and Merouane Debbah. "A Comprehensive Review and a Taxonomy of Edge Machine Learning: Requirements, Paradigms, and Techniques." Ai 4, no. 3 (2023): 729-786.

[8] Murshed, MG Sarwar, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. "Machine Learning at the Network Edge: A Survey." ACM Computing Surveys (CSUR) 54, no. 8 (2021): 1-37.

[9] Truong, Hong-Linh, Tram Truong-Huu, and Tien-Dung Cao. "Making Distributed Edge Machine Learning for Resource-Constrained Communities and Environments Smarter: Contexts and Challenges." Journal of Reliable Intelligent Environments 9, no. 2 (2023): 119-134.

[10] Guo, Yinghao, Rui Zhao, Shiwei Lai, Lisheng Fan, Xianfu Lei, and George K. Karagiannidis. "Distributed Machine Learning for Multiuser Mobile Edge Computing Systems." IEEE Journal of Selected Topics in Signal Processing 16, no. 3 (2022): 460-473.

[11] Duc, Thang Le, Rafael García Leiva, Paolo Casari, and Per-Olov Östberg. "Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing: A Survey." ACM Computing Surveys (CSUR) 52, no. 5 (2019): 1-39.

[12] Chen, Jiasi, and Xukan Ran. "Deep Learning with Edge Computing: A Review." Proceedings of the IEEE 107, no. 8 (2019): 1655-1674.

[13] Merenda, Massimo, Carlo Porcaro, and Demetrio Iero. "Edge Machine Learning for AI-Enabled IoT Devices: A Review." Sensors 20, no. 9 (2020): 2533.

[14] Voghoei, S., Tonekaboni, N. H., Wallace, J. G., & Arabnia, H. R. (2018, December). Deep Learning at the Edge. In 2018 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE. 895-901.

[15] Shuja, Junaid, Kashif Bilal, Waleed Alasmary, Hassan Sinky, and Eisa Alanazi. "Applying Machine Learning Techniques for Caching in Next-Generation Edge Networks: A Comprehensive Survey." Journal of Network and Computer Applications 181 (2021): 103005.

[16] Hoffpauir, Kyle, Jacob Simmons, Nikolas Schmidt, Rachitha Pittala, Isaac Briggs, Shanmukha Makani, and Yaser Jararweh. "A Survey on Edge Intelligence and Lightweight Machine Learning Support for Future Applications and Services." ACM Journal of Data and Information Quality 15, no. 2 (2023): 1-30.

[17] https://www.adjust.com/glossary/augmented-reality/

[18] https://embeddedcomputing.com/technology/processing/compute-modules/thermal-challenges-of-high-performance-embedded-ai-modules

https://www.wevolver.com/article/2024-state-of-edge-ai-report/edge-ai-challenges-and-real-world-mitigations