Journal of
Computer Technology & Applications
https://journals.stmjournals.com/jocta

ISSN: 2229-6964 (Online)
ISSN: 2347-7229 (Print)
Volume 16, Issue 2, 2025
May–August
DOI (Journal): 10.37591/JoCTA

Review JoCTA

Machine Learning Revolutionizing Server Management and Performance

Kazi Kutubuddin Sayyad Liyakat[1,*]

Abstract

The modern data center is a complex and dynamic environment, grappling with ever-increasing workloads, stringent performance demands, and the constant pressure for cost optimization. As such, applying machine learning (ML) directly to the server infrastructure offers a powerful avenue for achieving advanced automation, resource optimization, and proactive problem resolution. This article explores the transformative potential of integrating machine learning into server systems, leveraging insights gleaned from the abstract and conclusion of a broader study on the subject. The application of machine learning to server systems is rapidly evolving, moving beyond simple monitoring and analysis to sophisticated predictive and optimization strategies. By analyzing vast streams of server telemetry data, containing CPU (central processing unit) consumption, memory ingesting, network traffics, and disk I/O (input/output), machine learning algorithms can recognize patterns and irregularities which will be impossible for human operators to identify in real-time. This capability enables a range of applications, including dynamic resource allocation, predictive maintenance, and automated security threat detection. The promise lies in creating a more responsive, efficient, and resilient server infrastructure capable of adapting to fluctuating demands and unforeseen challenges. By embracing the transformative prospective of machine learning, organizations can unlock new era of efficiency and intelligence at the very core of their information technology infrastructure. The future of data centers is undoubtedly intertwined with the evolution of intelligent servers powered by machine learning.

Keywords: Machine learning, servers, management, intelligent system, cloud computing

INTRODUCTION

Machine learning (ML) is speedily transforming industries, from medical care and finance to transportation and entertainment. But behind the sophisticated algorithms and predictive power lies a crucial, often overlooked component: servers. These powerful machines are the unsung heroes, providing the computational muscle and storage capacity necessary to train, deploy, and scale modern ML models [1–4].

This article delves into the vital relationship between servers and ML, exploring how different server configurations and technologies are enabling breakthroughs in artificial intelligence (AI) and shaping the future of the field.

*Author for Correspondence
Kazi Kutubuddin Sayyad Liyakat
E-mail: drkkazi@gmail.com

[1]Professor and Head, Department of Electronics and Telecommunication Engineering, Brahmdevdada Mane Institute of Technology, Solapur, Maharashtra, India

Received Date: February 15, 2025
Accepted Date: March 19, 2025
Published Date: April 15, 2025

Citation: Kazi Kutubuddin Sayyad Liyakat. Machine Learning Revolutionizing Server Management and Performance. Journal of Computer Technology & Applications. 2025; 16(2): 36–44p.

The ML lifecycle is often broken down into three key stages:

- *Data ingestion and preparation*: This phase involves collecting, cleaning, and pre-processing vast amounts of data. Servers play a critical role in storing this data, often in distributed databases or data lakes, and providing the computational power to transform it into a usable format for training.

- *Model training*: This is the most computationally intensive stage. ML models learn patterns from the data, requiring extensive calculations and iterations. Powerful servers equipped with specialized hardware alike GPUs (graphics processing units) and TPUs (tensor processing units) are essential to accelerate the training process, reducing training times from weeks to hours or even minutes.
- *Model deployment and inference*: Once trained, model requirements to be positioned to sort predictions on innovative data in real-time. Servers provide the infrastructure for hosting the trained model and handling incoming requests. The required server configuration depends on the complexity of the model and the volume of requests, ranging from single servers to large-scale, distributed systems.

While standard servers can perform some ML tasks, specialized server configurations are crucial for achieving optimal performance and scalability:

- *GPU servers*: GPUs, initially designed for graphics processing, are highly parallel processors that excel at matrix operations, which are fundamental to many ML algorithms. GPU servers can significantly speed up training of deep learning models, creating them a staple in the ML landscape.
- *TPU servers*: TPUs are custom-intended hardware accelerators established by Google precisely for ML tasks. They offer even greater performance than GPUs for certain types of models, especially those based on deep learning.
- *High memory servers*: Some ML models require massive amounts of memory to load and process large datasets. High-memory servers are equipped with terabytes of RAM (random access memory) to handle these workloads, preventing performance bottlenecks.
- *Distributed systems and cloud computing*: For very large datasets and complex models, distributing the workload across multiple servers in a cluster or leveraging cloud computing platforms is often necessary. Cloud providers offer a wide range of server configurations and services specifically tailored for ML, allowing organizations to scale their resources on demand.

The demand for real-time insights and reduced latency is driving the adoption of edge computing in ML. This involves deploying ML models on servers closer to the data source, such as in factories, autonomous vehicles, or retail stores. Edge servers enable local processing and decision-making, reducing reliance on cloud connectivity and improving responsiveness [5–12].

Despite the advancements in server technology, challenges remain in the field of ML infrastructure:

- *Cost*: Specialized servers, especially those equipped with high-end GPUs or TPUs, can be expensive. Optimizing resource utilization and leveraging cloud computing can help mitigate costs.
- *Complexity*: Managing and configuring complex server infrastructure for ML can be challenging, requiring specialized expertise.
- *Scalability*: Scaling ML infrastructure to handle growing datasets and increasing user demand can be complex and time-consuming.

Looking ahead, we can expect to see the following trends in the relationship between servers and machine learning:

- *Increased specialization*: We will likely see more specialized hardware and server configurations tailored to specific ML tasks and applications.
- *Greater adoption of cloud computing*: Cloud platforms will endure to play central role in providing scalable and cost-effective ML infrastructure.
- *Focus on energy efficiency*: As ML workloads become more demanding, there will be a greater focus on developing energy-efficient servers and algorithms.
- *Advancements in serverless computing*: Serverless computing, where cloud provider manages underlying infrastructure, will simplify the deployment and scaling of ML models.

Servers are the foundation upon which modern ML is built. By providing the necessary computational power, storage capacity, and network connectivity, they enable the training, deployment, and scaling of sophisticated ML models. As ML continues to evolve and impact more industries, the demand for powerful and specialized server infrastructure will only continue to grow, driving innovation and shaping the future of AI. Investing in the right server infrastructure is not just a technological decision, it's a strategic investment in the future of data-driven innovation.

MACHINE LEARNING IN SERVERS: POWERING THE BACKBONE OF INTELLIGENT SYSTEMS

ML is no longer confined to research labs and specialized applications. It is rapidly infiltrating the server infrastructure that powers our digital world, driving improvements in performance, efficiency, and security. From optimizing resource allocation to predicting failures, ML in servers is becoming an indispensable tool for managing and enhancing the backbone of modern computing.

The sheer volume of data generated and processed by servers presents a fertile ground for ML applications. By leveraging this data, server administrators can gain valuable insights and automate tasks that were previously manual and complex. Here are some key benefits driving the adoption of ML in servers:

- *Improved performance*: ML algorithms can analyze server workloads, identify bottlenecks, and dynamically allocate resources like CPU (central processing unit), memory, and network bandwidth. This leads to optimized performance, reduced latency, and improved user experience.
- *Enhanced efficiency*: Servers are major consumers of energy. ML can help optimize power consumption by identifying idle resources, predicting future demands, and adjusting cooling systems accordingly. This translates to significant cost savings and reduced environmental impact.
- *Proactive maintenance*: Predictive maintenance is a game-changer in server management. ML models can examine past data, recognize patterns, and forecast possible hardware disasters earlier they occur. This permits for proactive conservation, minimizing interruption and averting costly disruptions.
- *Enhanced security:* ML can be used to detect and reply to security threats in real-time. By examining network traffic and system logs, ML algorithms can identify anomalies that indicate malicious activity, such as intrusions, malware infections, and denial-of-service attacks.
- *Automated management:* Server administration can be a complex and demanding task. ML can automate many of these tasks, such as capacity planning, workload balancing, and configuration management, freeing up supervisors to emphasis on more strategic initiatives.

The applications of ML in servers are diverse and continually expanding. Here are a few notable examples:

- *Resource allocation and load balancing*: ML algorithms analyze workload patterns and dynamically allocate resources to ensure optimal performance and prevent bottlenecks. This includes distributing traffic across multiple servers, adjusting CPU and memory allocations, and optimizing network routing.
- *Anomaly detection and fault prediction:* ML models analyze system logs, performance metrics, and hardware sensor data to identify anomalies that may indicate potential problems. This allows administrators to proactively address problems before they intensify into critical failures.
- *Security threat detection and response*: ML algorithms examine network traffic, system logs, and handler behavior to identify and respond to security threats, like intrusions, malware infections, and DDoS (distributed denial-of-service) attacks. This helps protect servers from malicious activity and maintain data integrity.
- *Power management and energy optimization*: ML models analyze server usage patterns and predict future energy demands. This allows for dynamic adjustment of server power settings, cooling systems, and other energy-consuming components, reducing energy consumption and lowering operational costs.
- *Capacity planning and forecasting*: ML algorithms analyze historical data and predict future resource requirements. This allows administrators to proactively plan for future growth and ensure that servers have sufficient capacity to meet growing demands.

Despite the numerous benefits, there are also challenges related with applying ML in servers:

- *Data availability and quality:* ML algorithms need large quantities of high-quality data to train effectively. Ensuring data availability and integrity can be a challenge in complex server environments.
- *Model complexity and training*: Building and training ML models may be complex and resource-intensive. Needs expertise in data science and ML.
- *Explainability and trust*: It's important to understand how ML models are making decisions, especially in critical applications. Lack of explainability can hinder trust and acceptance.

Looking ahead, we can expect to see the following trends in the application of ML in servers:

- *Edge computing*: Bringing ML closer to data source, decreasing latency and improving real-time decision-making.
- *Federated learning*: Training ML models on decentralized data sources without sharing sensitive information.
- *Auto ML*: Automating procedure of building and arranging ML models, constructing it easier for non-experts to leverage ML in server management.
- *Integration with infrastructure as code (IaC):* Integrating ML into IaC workflows for automated and intelligent infrastructure management.

Machine learning is transforming server management, offering significant improvements in performance, efficiency, and security. By leveraging power of data and intelligent algorithms, server administrators can optimize resource allocation, predict failures, and automate tasks, ultimately creating a more reliable, efficient, and secure server infrastructure. As ML technology endures to grow, we can think to see more advanced solicitations emerge, further solidifying its role as a critical component of modern server environments [13–25].

MACHINE LEARNING REVOLUTIONIZING PERFORMANCE OPTIMIZATION

In today's data-driven world, servers are the unsung heroes, diligently processing information and powering everything from e-commerce websites to complex scientific simulations. But with ever-increasing demands, maintaining optimal server performance is a constant challenge. Enter machine learning (ML), a powerful tool that is transforming how we monitor, manage, and ultimately, improve server efficiency.

For years, administrators relied on manual monitoring and reactive measures. They would identify bottlenecks after they occurred, leading to downtime, sluggish performance, and frustrated users. ML offers a paradigm shift, enabling proactive optimization and predictive problem-solving. Instead of reacting to issues, we can anticipate and prevent them.

ML algorithms are adept at analyzing vast quantities of server data, identifying patterns invisible to the human eye. This allows for a variety of performance improvements, including:

- *Predictive maintenance*: ML models can learn to predict server failures based on historical data, like CPU use, memory ingesting, and disk I/O (input/output). By identifying anomalies and subtle warning signs, administrators can schedule maintenance before a critical fault occurs, minimizing downtime and preventing catastrophic failures.
- *Resource allocation optimization*: Server resources like CPU, RAM, and network bandwidth often get allocated statically. ML can dynamically adjust these allocations based on real-time workload demands. By understanding the specific needs of different applications and processes, the system can optimize resource distribution, ensuring that resources are available where and when they're needed most.
- *Anomaly detection*: ML excels at identifying unusual patterns in server behavior. Sudden spikes in CPU usage, unexpected network traffic, or unusual log entries can all be flagged as potential anomalies. This allows for early detection of security threats, performance bottlenecks, and other issues that could negatively impact server stability.

- *Workload forecasting and scheduling*: By analyzing historical workload patterns, ML models can predict future demand. This allows for proactive scaling of server resources, ensuring that the system is always prepared to handle peak loads. It also enables intelligent scheduling of tasks, optimizing resource utilization and minimizing contention.
- *Automated configuration optimization*: Fine-tuning server configurations for optimal performance can be a complex and time-consuming task. ML can automate this process by analyzing performance metrics and suggesting optimal parameter settings for various server components.

Integrating ML into server management offers numerous benefits:

- *Reduced downtime:* Predictive maintenance and anomaly detection can identify and resolve issues before they lead to outages.
- *Improved performance*: Optimized resource allocation and workload scheduling result in faster response times and a smoother user experience.
- *Increased efficiency*: Automated configuration and resource management free up proprietors to emphasis on more tactical tasks.
- *Enhanced security*: Anomaly detection can identify and respond to security threats more quickly and effectively.
- *Cost savings*: Reduced downtime, improved efficiency, and optimized resource utilization all contribute to significant cost savings.

While the potential of ML in server optimization is immense, there are also challenges to consider:

- *Data availability and quality*: ML models need large quantities of high-quality data to be effective. Safeguarding data accuracy and wholeness is crucial.
- *Model complexity:* Building and deploying ML models may be complex, needing specialized expertise.
- *Computational resources*: Training and running ML models can be computationally intensive, requiring significant processing power.
- *Explainability and trust*: Sympathetic why an ML model makes a particular prediction is important for building trust and confidence in its decisions.

ML is poised to revolutionize server management, ushering in an era of intelligent, self-optimizing systems. As ML algorithms become more sophisticated and readily available, we can expect to see even greater improvements in server performance, reliability, and efficiency. By embracing this technology, organizations can unlock significant benefits and ensure that their servers are ready to meet the demands of the future. The transition from reactive to proactive server management is underway, and machine learning is leading the charge [26–29].

DESIGNING A MACHINE LEARNING PIPELINE ON SERVERS

ML is rapidly transforming businesses, offering powerful solutions for prediction, automation, and personalization. While developing ML models is crucial, deploying them effectively on servers is where the real value lies. This article outlines the key design steps for building a robust and scalable ML pipeline on server infrastructure, ensuring your models deliver tangible results.

Define the Problem and Business Goal

Before diving into the technicalities, clearly define the problem you aim to solve with ML and how it aligns with your business objectives. This includes identifying:

- *The specific business problem*: What are you trying to predict, classify, or optimize?
- *Desired outcome*: What impact will the ML model have on your business key performance indicators (KPIs)? (e.g., increased sales, reduced costs, improved customer satisfaction).
- *Key performance indicators:* How will you measure the success of your ML deployment?
- *Data availability*: Do you have the necessary data to train and validate the model?

A well-defined problem statement will guide subsequent design decisions and ensure the project stays focused on delivering value.

Data Acquisition and Preparation: The Foundation of Success

The quality and availability of data are paramount. This stage involves:

- *Data collection*: Identify all relevant data sources, including databases, APIs (application programming interfaces), log files, and external datasets. Automate data ingestion processes to ensure a continuous and reliable data stream.
- *Data cleaning*: Address missing values, outliers, and inconsistencies. Implement data validation checks to maintain data integrity.
- *Data transformation*: Convert data into a suitable format for ML algorithms. This may involve scaling, normalization, feature engineering, and encoding categorical variables.
- *Data storage*: Choose a robust storage solution that may handle volume and velocity of your data. Options include cloud-based data lakes (Azure Data Lake Storage, Amazon S3), relational databases, and NoSQL databases.

Model Selection and Training: Choosing the Right Tool for the Job

This is where the core ML model is developed:

- *Algorithm selection:* Select appropriate ML algorithms based on problem type (regression, classification, clustering), data characteristics, and desired performance metrics. Experiment with multiple algorithms to identify the best performer.
- *Feature engineering*: Generate new topographies from present data to advance model accuracy and interpretability. Domain expertise can be invaluable in this process.
- *Model training*: Train the selected model using your prepared data. Employ techniques like cross-validation to prevent overfitting and ensure generalization performance.
- *Hyperparameter tuning*: Enhance model's hyperparameters to attain the best possible performance. Consider using automated hyperparameter optimization techniques like grid search or Bayesian optimization.
- *Model evaluation:* Assess the trained model using suitable metrics (e.g., accuracy, precision, recall, F1-score, Area under the curve (AUC)) on a held-out validation dataset.

Model Deployment: Putting Your Model to Work

This stage focuses on making the trained model accessible for real-time predictions:

- *Deployment architecture*: Choose a deployment architecture that meets your performance, scalability, and availability requirements. Options include:
  - *REST API*: Expose the model as a REST API endpoint, allowing applications to send requests and receive predictions.
  - *Batch prediction*: Process large batches of data offline and store the predictions in a database.
  - *Stream processing*: Integrate the model into a stream processing pipeline to generate predictions in real-time.
- *Model serialization*: Save the trained model in a portable format (e.g., Pickle, ONNX, PMML) for deployment on the server.
- *Server infrastructure*: Select appropriate server infrastructure based on the model's computational requirements. Consider cloud-based solutions (AWS EC2, Azure Virtual Machines, Google Compute Engine) or on-premise servers.
- *Containerization*: Package the model and its dependencies into a docker container for easy deployment and portability.
- *Load balancing*: Distribute incoming requests across multiple server instances to ensure high availability and scalability.
- *Monitoring and logging:* Implement monitoring and logging to pathway model performance, detect errors, and identify potential issues.

Monitoring and Maintenance: Ensuring Long-Term Performance

ML models are not static; their performance can degrade over time due to data drift and concept drift. Continuous monitoring and maintenance are crucial:

- *Performance monitoring*: Track key performance metrics (e.g., accuracy, latency, throughput) in real time.
- *Data drift detection*: Monitor the input data distribution for changes that may impact model performance.
- *Concept drift detection*: Monitor the relationship between input features and the target variable for changes that may indicate model degradation.
- *Model retraining:* Retrain the model periodically with new data to maintain accuracy and adapt to evolving patterns.
- *Versioning*: Implement model versioning to track changes and facilitate rollback to previous versions if necessary.
- *Automated retraining pipelines*: Automate the model retraining process to ensure consistent performance and reduce manual effort.

The specific technologies used will depend on your organization's needs and infrastructure. However, some common components include:

- *Programming languages*: Python (with libraries like Scikit-learn, TensorFlow, PyTorch) or R.
- *Data processing tools*: Spark, Hadoop, Dask.
- *Database technologies*: PostgreSQL, MySQL, MongoDB, Cassandra.
- *Cloud platforms*: AWS, Azure, Google Cloud Platform.
- *Monitoring tools*: Prometheus, Grafana, ELK stack.
- *Containerization*: Docker, Kubernetes.
- *MLOps platforms*: MLflow, Kubeflow, AWS SageMaker, Azure Machine Learning.

Building an ML pipeline on servers requires careful planning and execution across multiple stages. By following these design steps, you can ensure that your ML models are deployed effectively, deliver tangible business value, and are continuously monitored and maintained for optimal performance. Remember that this is an iterative process, and you should continuously refine your pipeline based on feedback and performance monitoring. Choosing the right technologies and embracing MLOps best practices will pave the way for a successful ML journey.

CONCLUSION

The integration of ML into server systems is poised to revolutionize how data centers are managed and operated. While challenges remain in terms of data quality, model complexity, and the need for specialized expertise, the potential benefits are undeniable. By embracing ML, organizations can unlock a new level of efficiency, resilience, and security within their server infrastructure, enabling them to respond more effectively to the demands of the modern digital landscape. Furthermore, the continued development of specialized hardware, such as accelerated processors and dedicated AI chips, will further enhance the performance and scalability of ML -powered server solutions. As these technologies mature and become more accessible, we can expect to see even wider adoption of machine learning in servers, leading to a future where intelligent infrastructure plays a crucial role in driving innovation and business success.

REFERENCES

1. Khadake S, Kawade S, Moholkar S, Pawar M. A review of 6G technologies and its advantages over 5G technology. In: Pawar PM, et al, editors. Techno-Societal 2016, International Conference on Advanced Technologies for Societal Applications. Cham, Switzerland: Springer International Publishing; 2022. pp. 1043–1051.
2. Patil VJ, Khadake SB, Tamboli DA, Mallad HM, Takpere SM, Sawant VA. Review of AI in power electronics and drive systems. In: 2024 3rd International conference on Power Electronics and IoT Applications in Renewable Energy and Its Control (PARC), Mathura, India, February 23–24, 2024. pp. 94–99.

3. Dudgikar AB, Ingalgi AA, Jamadar AG, Swami OR, Khadake SB, Moholkar SV. Intelligent battery swapping system for electric vehicles with charging stations locator on IoT and cloud platform. Int J Adv Res Sci Commun Technol. 2023; 3 (1): 204–208.
4. Javagar P, Surendar V, Jayakumar K, Riyas KA, Dhanush K. Solar charging power station for electric vehicle. In: 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, April 7–9, 2022. pp. 824–827.
5. Zohuri B, Mossavar-Rahmani F. The symbiotic relationship unraveling the interplay between technology and artificial intelligence (an intelligent dynamic relationship). J Energy Power Eng. 2023; 17 (2): 63–68.
6. Zhong F, Wang J, Cheng X. Application of intelligent traffic control based on PLC. In: Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), March 2013. Dordrecht, Netherlands: Atlantis Press; 2013. pp. 1044–1048.
7. Magar SS, Sugandhi AS, Pawar SH, Khadake SB, Mallad HM. Harnessing wind vibration, a novel approach towards electric energy generation – review. Int J Adv Res Sci Commun Technol. 2024; 4 (2): 73–82.
8. Abass N, Vaidya R, Satav R, Jeyavel J. Automatic sanitizer dispenser with temperature screening. Int J Adv Res Ideas Innov Technol. 2021; 7 (3): 283–285.
9. Dhawale A, Kamble A, Ghule S, Gawande M. Solar powered indoor air purifier with air quality monitor. In: 2023 9th International Conference on Electrical Energy Systems (ICEES), Chennai, Insia, March 23–25, 2023. pp. 206–209.
10. Shabnam S, Latha HN. Design and implementation of saliency detection model in H.264 standard. Int J Sci Res. 2014; 3 (6): 2014–2020.
11. Muddamsetty SM, Sidibé D, Trémeau A, Mériaudeau F. Salient objects detection in dynamic scenes using color and texture features. Multimedia Tools Appl. 2018; 77: 5461–5474.
12. Guleria M, Rathi A, Sajwan DS, Negi JS, Chauhan N. DTMF based irrigation water pump control system. Int J Smart Device Appl. 2017; 5 (1): 19–24.
13. Arulious JA, Earlina D, Harish D, Priya PS, Rexy AI, Mary JN. Design of solar powered electric vehicle. J Phys Conf Ser. 2021; 2070 (1): 012105.
14. Kumar TP, Haidari M, Vimala R, Subramanya R. Low-cost battery management system for e-vehicles. In: 2022 International Conference on Edge Computing and Applications (ICECAA), Tamil Nadu, India, October 13–15, 2022. pp. 723–727.
15. Chounde A, Gopnarayan BB, Patil KB, Kamble SS. Human health care system: a new approach towards life. Grenze Int J Eng Technol. 2024; 10: 5487–5494.
16. Patil VJ, Mallad HM, Gopnarayan BB, Pati KB. Maximize farming productivity through agriculture 4.0 based intelligence, with use of agri tech sense advanced crop monitoring system. Grenze Int J Eng Technol. 2024; 10: 5127–5134.
17. Atabani AE. Biodiesel: a promising alternative energy resource. In: Alternative Fuels Research Progress. Najaf, Iraq: International Energy and Environmental Foundation; 2013. 93 pp.
18. Veena C, Sridevi M, Liyakat KK, Saha B, Reddy SR, Shirisha N. HEECCNB: an efficient IoT-cloud architecture for secure patient data transmission and accurate disease prediction in healthcare systems. In: 2023 Seventh International Conference on Image Information Processing (ICIIP), Solan, India, November 22–24, 2023. pp. 407–410.
19. Kazi KS. Computer-aided diagnosis in ophthalmology: a technical review of deep learning applications. In: Garcia MB, de Almeida RPP, editors. Transformative Approaches to Patient Literacy and Healthcare Innovation. Hershey, PA, USA: IGI Global; 2024. pp. 112–135.
20. Kazi KS. AI-driven-IoT (AIIoT)-based decision making in kidney diseases patient healthcare monitoring: KSK approach for kidney monitoring. In: Polat LO, Polat O, editors. AI-Driven Innovation in Healthcare Data Analytics. Hershey, PA, USA: IGI Global; 2025. pp. 277–306.
21. Pradeepa M, Jamberi K, Sajith S, Bai MR, Prakash A. Student health detection using a machine learning approach and IoT. In: 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, October 16–17, 2022. pp. 1–5.

22. Aldahiri A, Alrashed B, Hussain W. Trends in using IoT with machine learning in health prediction system. Forecasting. 2021; 3 (1): 181–206.
23. Mulani AO, Sardey MP, Kinage K, Salunkhe SS, Fegade T, Fegade PG. ML-powered internet of medical things (MLIOMT) structure for heart disease prediction. J Pharmacol Pharmacother. 2025; 16 (1): 38–45.
24. Payalan YF, Guvensan MA. Towards next-generation vehicles featuring the vehicle intelligence. IEEE Trans Intell Transport Syst. 2019; 21 (1): 30–47.
25. Patel PP, Pavesha CL, Sabat SS, More SS. Deep learning based driver drowsiness detection. In: 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, May 9–11, 2022. pp. 380–386.
26. Yahaya L, Oye ND, Garba EJ. A comprehensive review on heart disease prediction using data mining and machine learning techniques. Am J Artif Intell. 2020; 4 (1): 20–29.
27. Kutubuddin KA, Nerkar PR, Sultanabanu KA. IoT-based skin health monitoring system. Int J Biol, Pharm Allied Sci. 2024; 13 (11): 5937–5950.
28. Sreelakshmy R, Sruthy R, Rajeshwari R, Thyla B. Patient health monitoring system using smart IoT devices for medical emergency services. In: 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, July 15–16, 2022. pp. 1–10.
29. Malathi S, Razool Begum S. Architecture for reliable cyber attack detection in IOT networks to increase the trust worthiness between nodes. Int J Model SimulSci Comput. 2024. Epub. doi: 10.1142/S179396232441023X.

