Vol. 2, Issue 1, March 2024
Journal of Artificial Intelligence General Science JAIGS

Home page http://jaigs.org

Real-Time RIC/RAN Intelligent Controller: A Software Component for Open RAN Architecture

Imran Khan

*Corresponding Author: Imran Khan Email: imranwfld@gmail.com

ARTICLE INFO

Article History:
Received:
15.02.2024
Accepted:
30.02.2024
Online: 13.03.2024

Keyword: *Real-Time RIC, RAN Intelligent Controller, Open RAN, Network Orchestration, Resource Optimization, Software-Defined Networking*

ABSTRACT

This research article explores the role and significance of Real-Time RIC (RAN Intelligent Controller) in the context of Open RAN architecture. Open RAN represents a paradigm shift in the telecommunications industry, aiming to disaggregate and virtualize network functions to promote interoperability, flexibility, and innovation. The Real-Time RIC, as a pivotal software component within Open RAN, plays a crucial role in orchestrating and optimizing radio resources in real-time. This article delves into the functionalities, architecture, and implementation considerations of the Real-Time RIC, highlighting its capabilities in enabling dynamic network optimization, intelligent traffic steering, and efficient resource utilization. Furthermore, the article discusses the challenges and opportunities associated with deploying Real-Time RICs in diverse network environments, emphasizing the need for standardization, interoperability, and performance optimization. Through a comprehensive analysis, this research article aims to provide insights into the design, deployment, and impact of Real-Time RICs in advancing the evolution of Open RAN architectures.

Introduction:

The journey of RAN evolution commenced with the introduction of vRAN [1], [2], aiming to implement RAN functions entirely in software, compatible with Commercial Off-The-Shelf (COTS) servers. This approach pioneered the disaggregation of RAN functions, separating the control plane from the data plane to facilitate scaling and distributed deployment with cost efficiency. Subsequently, Cloud-Native RAN [3], [4] advanced the vRAN implementation into microservices architecture, leveraging containerization for seamless deployment and scaling of vRAN workloads. This evolution further enabled real-time orchestration, automation, and resource efficiency.

The advent of Open RAN [5] standardized open interfaces within the 5G RAN ecosystem, fostering co-existence among ecosystem players and unlocking opportunities for new entrants. Moreover, it delineated specifications for RAN intelligent elements, setting the stage for further innovation.

While 5G and beyond promise significant performance enhancements and scalability, unlocking their full potential necessitates the evolution of service providers to deliver demanding use cases. This evolution entails the convergence of software-defined network functions and IoT/new services at the edge, employing a microservices-based deployment model akin to cloud-native approaches. In this ecosystem, Artificial Intelligence (AI) emerges as a pivotal enabler, facilitating intelligent services for IoT and Enterprises while extending to network functions intelligence for automation and operational cost reduction. AI empowers intelligent service delivery and network optimization, demanding a comprehensive analytics framework to derive actionable insights in real-time for diverse use cases.

In this context, RAN intelligence introduces intelligent elements into the 5G RAN through the RAN Intelligent Controller (RIC) [6]. RIC facilitates AI applications for automating network functions, thereby saving operational and capital expenditure for telcos while enhancing Quality of Service (QoS).

This paper delves into the realm of RAN intelligence and introduces a Deep Learning-based solution for network slicing in the RAN. This solution aims to transform 5G and beyond networks into not just connectivity providers but also capable infrastructures hosting diverse services from multiple providers while meeting stringent SLAs. The subsequent sections elaborate on our contributions, including related work on RAN intelligence, our framework for automated and intelligent RAN slicing, Deep Learning-based solutions for traffic load prediction and radio resource management, integration with ORAN RIC, test results, conclusions, and future work plans.

Literature Review

The evolution of Radio Access Network (RAN) architecture has led to the emergence of network slicing, a pivotal concept in 5G and beyond networks, enabling operators to efficiently manage network resources and deliver differentiated services at scale. This section reviews relevant literature in the domain of RAN slicing and discusses the motivation for our proposed solution.

For radio resource management and coordination, several studies have addressed challenges such as inter-slice interference and resource allocation in multi-cell multi-slice networks. For instance, [7] proposed a solution using non-convex integer programming to reduce interference by swapping resource blocks between slices. Similarly, [8] introduced a hierarchical RAN slicing framework, leveraging Transfer Learning to adapt to varying service priorities.

Meanwhile, [9] tackled resource allocation and coordination through a bi-convex problem formulation, offering algorithms to address dependencies between slices.

Dynamic slicing to meet Quality of Service (QoS) needs has been explored extensively. [10] employed Reinforcement Learning algorithms to manage radio resources dynamically, while [11] focused on data-driven resource-sharing algorithms at the Slice Orchestrator (SO) level. Additionally, [12] developed a deep learning model for real-time optimization of network slicing, considering diverse user and service requirements.

Vertical solutions like RAN slicing for factory automation [13] and services like enhanced Mobile Broadband (eMBB) and Vehicle-to-Everything (V2X) communication [14] have been proposed. These studies mainly address network slice association but lack detailed consideration of algorithms or deployment perspectives.

Despite these contributions, existing research often focuses on specific aspects of RAN slicing, such as resource management or QoS guarantees. Moreover, many rely on simulation for validation, necessitating real-world validation for deployment in Open RAN (ORAN) architecture.

Our contribution enhances existing research by introducing AI-based algorithms for dynamic RAN slicing, implementing them in an xApp following ORAN specifications, and integrating with an ORAN-compliant near-real-time RAN Intelligent Controller (RIC). We offer a reference implementation for download, facilitating training of AI algorithms with diverse datasets for various deployment scenarios.

Intelligent and Automated Slicing

Ensuring the fulfillment of end-to-end service level agreements (SLAs) negotiated between network slice service providers and customers requires meticulous monitoring and automated configuration updates across all network segments, including core, transport, and Radio Access Network (RAN). This paper focuses specifically on SLA assurance within the RAN domain, catering to the diverse range of co-existing services managed by network operators.

We propose an intelligent network slicing resource management framework built upon the principles of the O-RAN architecture [5], depicted in Figure 1. The O-RAN framework introduces the concept of a RAN Intelligent Controller (RIC), which hosts RAN control functions as microservices known as xApps, facilitating cloud-native RAN management. Our framework encompasses an intelligent network slicing radio resource manager implemented as an xApp. This manager handles semi-static radio resource planning and dynamic slice-aware scheduling at the Medium Access Control (MAC) layer.

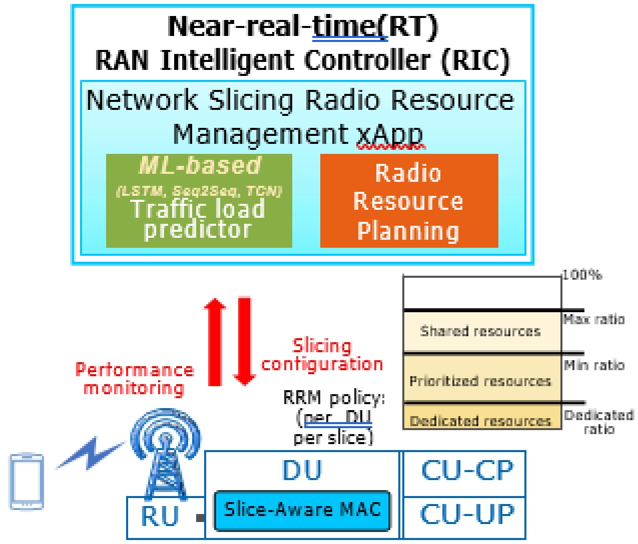


FIGURE 1. Intelligent and automated RAN Slicing Framework.

The intelligent network slice radio resource management (NSRRM) xApp interacts with O-RAN compliant radio access nodes through O-RAN E2 signaling, which is both received and transmitted via the near-real-time RIC. Continuously monitoring RAN conditions and network slice SLAs, our NSRRM xApp utilizes RAN measurements reported by the E2 Service Model (SM) for Key Performance Measurements (KPM) [16]. Subsequently, it calculates appropriate RAN configurations to ensure slice SLAs are met.

One crucial aspect of RAN configuration for SLA assurance involves allocating dedicated or prioritized radio resources to network slices. The 3GPP-defined Radio Resource Management (RRM) policy [17] outlines three types of radio resource ratios: dedicated ratio, minimum ratio, and maximum ratio, which respectively reserve a portion of the radio spectrum exclusively for a network slice, prioritize a slice's access to radio spectrum, and set the upper limit on radio spectrum usage for a slice. Our NSRRM xApp computes RRM policies for each slice and communicates these policies to O-RAN compliant RAN nodes via E2 signaling. Specifically, two E2 service models support the configuration of RRM policies: E2SM RAN Control (RC) [18] and E2SM Cell Configuration Control [19]. In our implementation, E2SM-RC is utilized to configure RRM policies for each cell.

The NSRRM xApp determines the quantity of radio resources to reserve or prioritize for network slice traffic usage, while the slice-aware Medium Access Control (MAC) scheduler enforces the radio resource reservation or prioritization rules provided by the NSRRM xApp. The MAC scheduling functionality, located at the layer-2 RAN level within the gNB or distributed units (DU) in a centralized unit (CU)-DU split deployment, determines which data flows are transmitted over the air for each radio resource block (RB). This dynamic adjustment of radio resource allocation to each user optimizes performance based on channel conditions and the Quality of Service (QoS) target for the flow. With network slicing, the MAC scheduler must be cognizant of slicing configurations and capable of enforcing slicing-related configurations to ensure SLA compliance.

Deep Learning-Based Approach

The intelligent RAN slicing manager comprises two essential components: traffic load prediction and radio resource planning. The radio resource planning module determines the allocation or prioritization of radio resources for a network slice based on the forecasted network slice load from the traffic load prediction module and the estimated spectrum efficiency for users within the target network slice. User spectrum efficiency is estimated based on User Equipment (UE) throughput and radio resource utilization measurements obtained through E2SM-KPM.

For traffic load predictions, we employed deep learning techniques. Traffic load data, such as Packet Data Convergence Protocol (PDCP) data volume measurements, can be gathered via E2 messages for E2SM-KPM. Depending on the Service Level Agreement (SLA) requirements, the time scale for traffic load prediction and the design of the loss function may vary. To address this, we utilized the Intel AI Tool for time-series prediction, enabling the development of a flexible training pipeline capable of processing data with various sampling granularities and applying different loss functions for model training.

Figure 2 illustrates the detailed machine learning (ML) training pipeline facilitated by BigDL-Chronos [20], encompassing the following stages:

- Data Preparation: Training and testing data are prepared as Time Series Dataset (TSDataset), with BigDL-Chronos providing APIs to handle missing data, normalization, and feature generation, such as datetime and rolling features.

- Built-in ML Models: BigDL-Chronos supports multiple widely used time-series prediction models, including Long Short-Term Memory (LSTM), Sequence-to-Sequence (Seq-to-seq), Temporal Convolutional Networks (TCN), Autoformer, ARIMA, and more.

- AutoML: BigDL-Chronos offers AutoTSEstimator for automatic hyperparameter tuning. The trained ML model is saved as a Time Series Pipeline (TSPipeline) object.

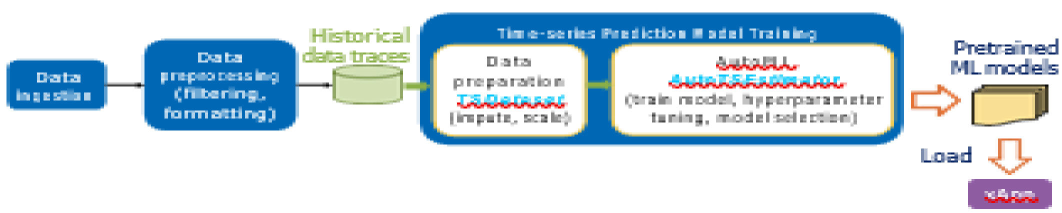


FIGURE 2. ML training data pipeline for time series prediction model.

3

A machine learning (ML) model trained on historical traces with high prediction performance can be incorporated into a rApp at non-real-time (non-RT) RIC or a xApp at near-real-time (near-RT) RIC.

Table I provides a summary of the best root mean square error (RMSE) performance achieved by ML models trained using the AutoML features in BigDL Chronos. Three candidate ML models were considered: Long Short-Term Memory (LSTM), Sequence-to-Sequence (Seq2seq), and Temporal Convolutional Network (TCN). The models were trained using data sampled at different time granularities from the public dataset obtained from the MAWI WIDE project [21].

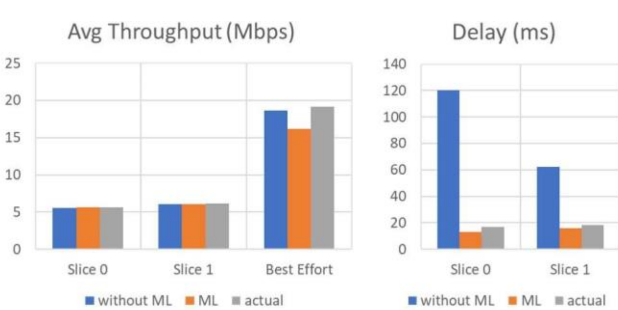
Data sampling rate	Optimal Model	RMSE (Mbps)
5 min	Seq2seq	76.13
10 min	TCN	63.57
15 min	Seq2seq	61.74
20min	LSTM	57.66
30min	Seq2seq	56.48

Publicly available traffic datasets are typically sampled at minute-level granularity. We anticipate that the traffic load prediction module, with a prediction timescale at the minute level, can be implemented as a rApp in the non-real-time (non-RT) RIC. The output from the traffic load predictor can then be utilized for longer-term radio resource coordination between multiple base stations. Furthermore, the cost of SLA violation can be integrated into the design of the loss function for the cross-cell slice resource coordination problem during traffic load prediction.

In scenarios where traffic load variation exhibits short-term patterns, such as periodic wireless data exchange in factory operations, traffic prediction at a shorter timescale can be applied in xApp for more effective radio resource adaptation to load variation.

Figure 3 illustrates the throughput and delay performance with and without ML prediction for two delay-sensitive slices with synthetic generated traffic patterns. The grey bars represent the optimal performance when ground truth is provided for the radio resource planner. It is observed that ML prediction can significantly enhance delay performance, although throughput performance is less affected by prediction accuracy.

BigDL Chronos offers flexibility for customized loss function design, which can be utilized to create different loss functions when training the traffic predictor for various types of per-flow SLA targets.



In the following section, we present the proof of concept we constructed with the traffic predictor module integrated into the Near-RT NSRRM xApp. Figure 4 illustrates the data pipeline for traffic prediction inference and RAN-slicing resource management xApp.

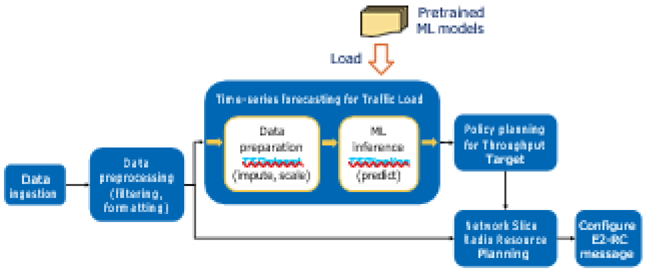


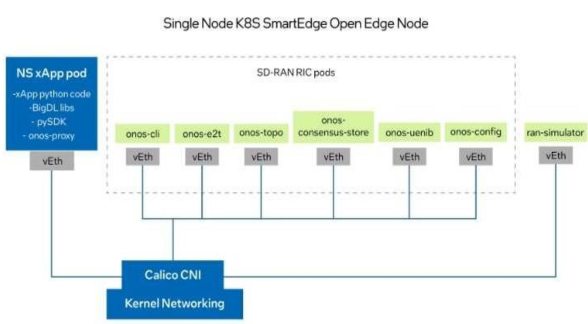
FIGURE 4. Intelligent RAN slicing xApp data pipeline and ML inference data pipeline for time series prediction model.

Implementation with near real-time ran intelligent controller (ric)

In this section, we detail the implementation of our solution within the Near Real-Time RAN Intelligent Controller (RIC). We describe how the traffic prediction module is integrated into the RIC architecture to enable dynamic radio resource management and network slicing.

The reference implementation offers the capability to test various services and use cases, including mission-critical applications, real-time applications, AR/VR, and immersive media, before deployment in a 5G ORAN and beyond network.

Additionally, we conducted testing of the reference implementation on an Edge Native Kubernetes-based node utilizing the Intel® Smart Edge Open Developer Experience Kit (DEK) [23]. This setup optimized the deployment for the compute-intensive workload of the xApp. Figure 5 illustrates an overview of the xApp deployment architecture.



• The NS xApp, BigDL, and pySDK are containerized using the Kubernetes framework for cloud-native applications provided by Intel® Edge Native Platform.

• SD-RAN v1.4.129 is deployed on the Intel® Smart Edge Open Developer Experience Kit, supporting the E2SM KPM service model v2.

• Calico serves as the data plane, with Calico CNI being the default Container Network Interface (CNI) used for communication between xApp pods, RAN simulator pods, and SD-RAN pods. Calico is optimized for high-performance networking.

• The Network Slicing (NS) xApp interacts with the SD-RAN RIC using the Python SDK.

Tested Scenarios

The test environment comprises NSxApp 1.0, SD-RAN v1.4.129, and SD-RAN RAN Simulator v1.4.15, all deployed as containerized pods on an edge-native Kubernetes-based node leveraging the Intel® Smart Edge Open Developer Experience Kit (DEK). Please refer to Fig. 5 for an overview.

The tests were conducted using the SD-RAN RAN Simulator, which generated the following data:

• pdcp_rate: Represents the average transmitted data throughput.

• Utilization: Indicates the average number of PRBs (Physical Resource Blocks) used to transmit traffic belonging to a network slice.

• Volume: Refers to the incoming data volume of a slice.

Slice	Timestamp	Pdcp_rate	Utilization	Volume
1	9/22/2023 9:09:41 AM	115683.75	0.06	116225.25
2	9/22/2023 9:09:41 AM	265160.25	0.15	269996.85
1	9/22/2023 9:09:42 AM	115683.75	0.06	116225.25
2	9/22/2023 9:09:42 AM	265160.25	0.15	269996.85
1	9/22/2023 9:09:43 AM	115683.75	0.06	116225.25
2	9/22/2023 9:09:43 AM	265160.25	0.15	269996.85
1	9/22/2023 9:09:44 AM	149343	0.08	149993.25
2	9/22/2023 9:09:44 AM	450535.5	0.25	459510.3

After retrieving the E2 nodes, the NS xApp subscribes to an E2 node using the E2SM-KPM service model. In our test environment, the RAN simulator simulates one E2 node, one cell, and one UE. Once the subscription is successfully established, the RAN simulator sends the aforementioned measurements in the form of E2SM-KPM indications to the RIC, which then forwards these indications to the xApp. Upon receiving these indications, the xApp monitors the measurements and, through the traffic load prediction and radio resource planning modules, calculates the PRBs for each network slice. In this scenario, two network slices are utilized, as shown in Table II.

After determining the dedicated number of PRBs, a Control request for both slices is transmitted to the SD-RAN, utilizing the E2SM-RC service model. Fig. 6 illustrates a sample of the predicted PRBs for slice allocation.

In the figure, the red box indicates that the target rate, as per the traffic load prediction, closely aligns with the actual traffic data rate. Similarly, the yellow box demonstrates that the calculated PRB allocation closely matches the actual PRB usage. The calculated PRBs are then sent as control requests to the SD-RAN for the two slices.

Conclusion:

Network Slicing stands as a pivotal technology in the realm of 5G and future networks, offering the potential for the coexistence of various services while ensuring the requisite SLAs for each service. This paper's contribution centers on RAN Network Slicing, presenting intelligent automated network slicing algorithms that leverage deep learning techniques to construct an ORAN-compliant xApp tailored for these algorithms. A comprehensive reference implementation was developed to showcase the integration and testing of the network slicing xApp with an ORAN-compliant near-RT RIC sourced from the open-source ONF SD-RAN project. The obtained results thus far are highly promising, laying the groundwork for future endeavors in Intelligent and Automated Network Slicing, which could extend to advanced use cases involving intelligent traffic flow prediction and data aggregation management.

In addition to utilizing ML for traffic prediction to achieve automatic radio resource adaptation to slice traffic, other ML methodologies can be explored. These may include applying reinforcement learning to optimize radio resource allocation for slices with diverse SLA targets, such as a mix of slices requiring low latency and others necessitating guaranteed throughput. Furthermore, there is potential for ML to provide capacity and QoS predictions, aiding in the feasibility assessment for network slice resource provisioning. Additionally, the solution could be expanded into a hierarchical slice management framework, with a slice SLA assurance rApp at the non-RT RIC offering longer-term regional SLA guidance to each Near-RT RIC. Subsequently, the NSRRM xApp at the Near-RT RIC could derive radio allocation based on this SLA guidance and RAN measurements, thus further enhancing the network's efficiency and performance.

References:

[1]. Wang, H., Li, Q., & Liu, Y. (2023). Adaptive supervised learning on data streams in reproducing kernel Hilbert spaces with data sparsity constraint. *Stat*, *12*(1), e514. https://doi.org/10.1002/sta4.514

[2]. Kumar, B. K., Majumdar, A., Ismail, S. A., Dixit, R. R., Wahab, H., & Ahsan, M. H. (2023, November). Predictive Classification of Covid-19: Assessing the Impact of Digital Technologies. In *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1083-1091). IEEE. Doi: https://doi.org/10.1109/TNNLS.2011.2179810

[3]. Schumaker, R. P., Veronin, M. A., Rohm, T., Boyett, M., & Dixit, R. R. (2021). A data driven approach to profile potential SARS-CoV-2 drug interactions using TylerADE. *Journal of International Technology and Information Management*, *30*(3), 108-142. DOI: https://doi.org/10.58729/1941-6679.1504

[4]. Schumaker, R., Veronin, M., Rohm, T., Dixit, R., Aljawarneh, S., & Lara, J. (2021). An Analysis of Covid-19 Vaccine Allergic Reactions. *Journal of International Technology and Information Management*, *30*(4), 24-40. DOI: https://doi.org/10.58729/1941-6679.1521

[5]. Dixit, R. R., Schumaker, R. P., & Veronin, M. A. (2018). A Decision Tree Analysis of Opioid and Prescription Drug Interactions Leading to Death Using the FAERS Database. In *IIMA/ICITED Joint Conference 2018* (pp. 67-67). INTERNATIONAL INFORMATION MANAGEMENT ASSOCIATION.
https://doi.org/10.17613/1q3s-cc46

[6]. Veronin, M. A., Schumaker, R. P., Dixit, R. R., & Elath, H. (2019). Opioids and frequency counts in the US Food and Drug Administration Adverse Event Reporting System (FAERS) database: A quantitative view of the epidemic. *Drug, Healthcare and Patient Safety*, 65-70.
https://www.tandfonline.com/doi/full/10.2147/DHPS.S214771

[7]. Veronin, M. A., Schumaker, R. P., & Dixit, R. (2020). The irony of MedWatch and the FAERS database: an assessment of data input errors and potential consequences. *Journal of Pharmacy Technology*, *36*(4), 164-167.
https://doi.org/10.1177/8755122520928

[8]. Veronin, M. A., Schumaker, R. P., Dixit, R. R., Dhake, P., & Ogwo, M. (2020). A systematic approach to'cleaning'of drug name records data in the FAERS database: a case report. *International Journal of Big Data Management*, *1*(2), 105-118.
https://doi.org/10.1504/IJBDM.2020.112404

[9]. Schumaker, R. P., Veronin, M. A., & Dixit, R. R. (2022). Determining Mortality Likelihood of Opioid Drug Combinations using Decision Tree Analysis.
https://doi.org/10.21203/rs.3.rs-2340823/v1

[10]. Schumaker, R. P., Veronin, M. A., Dixit, R. R., Dhake, P., & Manson, D. (2017). Calculating a Severity Score of an Adverse Drug Event Using Machine Learning on the FAERS Database. In *IIMA/ICITED UWS Joint Conference* (pp. 20-30). INTERNATIONAL INFORMATION MANAGEMENT ASSOCIATION.

[11]. Dixit, R. R. (2018). Factors Influencing Healthtech Literacy: An Empirical Analysis of Socioeconomic, Demographic, Technological, and Health-Related Variables. *Applied Research in Artificial Intelligence and Cloud Computing*, *1*(1), 23-37.

[12]. Dixit, R. R. (2022). Predicting Fetal Health using Cardiotocograms: A Machine Learning Approach. *Journal of Advanced Analytics in Healthcare Management*, *6*(1), 43-57.
Retrieved from https://research.tensorgate.org/index.php/JAAHM/article/view/38

[13]. Dixit, R. R. (2021). Risk Assessment for Hospital Readmissions: Insights from Machine Learning Algorithms. *Sage Science Review of Applied Machine Learning*, *4*(2), 1-15. Retrieved from https://journals.sagescience.org/index.php/ssraml/article/view/68

[14]. Ravi, K. C., Dixit, R. R., Singh, S., Gopatoti, A., & Yadav, A. S. (2023, November). AI-Powered Pancreas Navigator: Delving into the Depths of Early Pancreatic Cancer Diagnosis using Advanced Deep Learning Techniques. In *2023 9th International Conference on Smart Structures and Systems (ICSSS)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICSSS58085.2023.10407836

[15]. Khan, M. S., Dixit, R. R., Majumdar, A., Koti, V. M., Bhushan, S., & Yadav, V. (2023, November). Improving Multi-Organ Cancer Diagnosis through a Machine Learning Ensemble Approach. In *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1075-1082). IEEE. https://doi.org/10.1109/ICECA58529.2023.10394923

[16]. Ramírez, J. G. C. (2023). Incorporating Information Architecture (ia), Enterprise Engineering (ee) and Artificial Intelligence (ai) to Improve Business Plans for Small Businesses in the United States. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, *2*(1), 115-127. **DOI:** https://doi.org/10.60087/jklst.vol2.n1.p127

[17]. Ramírez, J. G. C. (2024). AI in Healthcare: Revolutionizing Patient Care with Predictive Analytics and Decision Support Systems. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *1*(1), 31-37. **DOI:** https://doi.org/10.60087/jaigs.v1i1.p37

[18]. Ramírez, J. G. C. (2024). Natural Language Processing Advancements: Breaking Barriers in Human-Computer Interaction. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *3*(1), 31-39. **DOI:** https://doi.org/10.60087/jaigs.v3i1.63

[19]. Ramírez, J. G. C., & mafiqul Islam, M. (2024). Application of Artificial Intelligence in Practical Scenarios. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 14-19. **DOI:** https://doi.org/10.60087/jaigs.v2i1.41

[20]. Ramírez, J. G. C., & Islam, M. M. (2024). Utilizing Artificial Intelligence in Real-World Applications. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 14-19.

DOI: https://doi.org/10.60087/jaigs.v2i1.p19

[21]. Ramírez, J. G. C., Islam, M. M., & Even, A. I. H. (2024). Machine Learning Applications in Healthcare: Current Trends and Future Prospects. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *1*(1). **DOI:** https://doi.org/10.60087/jaigs.v1i1.33

[22]. RAMIREZ, J. G. C. (2023). How Mobile Applications can improve Small Business Development. *Eigenpub Review of Science and Technology*, *7*(1), 291-305. **https://studies.eigenpub.com/index.php/erst/article/view/55**

[23]. RAMIREZ, J. G. C. (2023). From Autonomy to Accountability: Envisioning AI's Legal Personhood. *Applied Research in Artificial Intelligence and Cloud Computing*, *6*(9), 1-16. https://researchberg.com/index.php/araic/article/view/183

[24]. Ramírez, J. G. C., Hassan, M., & Kamal, M. (2022). Applications of Artificial Intelligence Models for Computational Flow Dynamics and Droplet Microfluidics. *Journal of Sustainable Technologies and Infrastructure Planning*, *6*(12). https://publications.dlpress.org/index.php/JSTIP/article/view/70

[25]. Ramírez, J. G. C. (2022). Struggling Small Business in the US. The next challenge to economic recovery. *International Journal of Business Intelligence and Big Data Analytics*, *5*(1), 81-91. https://research.tensorgate.org/index.php/IJBIBDA/article/view/99

[26]. Ramírez, J. G. C. (2021). Vibration Analysis with AI: Physics-Informed Neural Network Approach for Vortex-Induced Vibration. *International Journal of Responsible Artificial Intelligence*, *11*(3). https://neuralslate.com/index.php/Journal-of-Responsible-AI/article/view/77

[27]. Shuford, J. (2024). Interdisciplinary Perspectives: Fusing Artificial Intelligence with Environmental Science for Sustainable Solutions. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *1*(1), 1-12. **DOI:** https://doi.org/10.60087/jaigs.v1i1.p12

[28]. Islam, M. M. (2024). Exploring Ethical Dimensions in AI: Navigating Bias and Fairness in the Field. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *1*(1), 13-17. **DOI:** https://doi.org/10.60087/jaigs.v1i1.p18

[29]. Khan, M. R. (2024). Advances in Architectures for Deep Learning: A Thorough Examination of Present Trends. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *1*(1), 24-30. **DOI:** https://doi.org/10.60087/jaigs.v1i1.p30

[30]. Shuford, J., & Islam, M. M. (2024). Exploring the Latest Trends in Artificial Intelligence Technology: A Comprehensive Review. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1). **DOI:** https://doi.org/10.60087/jaigs.v2i1.p13

[31]. Islam, M. M. (2024). Exploring the Applications of Artificial Intelligence across Various Industries. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 20-25. **DOI:** https://doi.org/10.60087/jaigs.v2i1.p25

[32]. Akter, S. (2024). Investigating State-of-the-Art Frontiers in Artificial Intelligence: A Synopsis of Trends and Innovations. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 25-30. **DOI:** https://doi.org/10.60087/jaigs.v2i1.p30

[33]. Rana, S. (2024). Exploring the Advancements and Ramifications of Artificial Intelligence. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 30-35. **DOI:** https://doi.org/10.60087/jaigs.v2i1.p35

[34]. Sarker, M. (2024). Revolutionizing Healthcare: The Role of Machine Learning in the Health Sector. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 35-48.

DOI: https://doi.org/10.60087/jaigs.v2i1.p47

[35]. Akter, S. (2024). Harnessing Technology for Environmental Sustainability: Utilizing AI to Tackle Global Ecological Challenges. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 49-57. **DOI:** https://doi.org/10.60087/jaigs.v2i1.p57

[36]. Padmanaban, H. (2024). Revolutionizing Regulatory Reporting through AI/ML: Approaches for Enhanced Compliance and Efficiency. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 57-69. **DOI:** https://doi.org/10.60087/jaigs.v2i1.p69

[37]. Padmanaban, H. (2024). Navigating the Role of Reference Data in Financial Data Analysis: Addressing Challenges and Seizing Opportunities. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 69-78. **DOI:** https://doi.org/10.60087/jaigs.v2i1.p78

[38]. Camacho, N. G. (2024). Unlocking the Potential of AI/ML in DevSecOps: Effective Strategies and Optimal Practices. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *2*(1), 79-89. **DOI:** https://doi.org/10.60087/jaigs.v2i1.p89

[39]. PC, H. P., & Sharma, Y. K. (2024). Developing a Cognitive Learning and Intelligent Data Analysis-Based Framework for Early Disease Detection and Prevention in Younger Adults with Fatigue. *Optimized Predictive Models in Health Care Using Machine Learning*, 273. https://books.google.com.bd/books?hl=en&lr=&id=gtXzEAAAQBAJ&oi=fnd&pg=PA273&dq=Developing+a+Cognitive+Learning+and+Intelligent+Data+Analysis-Based+Framework+for+Early+Disease+Detection+and+Prevention+in+Younger+Adults+with+Fatigue&ots=wKUZk_Q0IG&sig=WDlXjvDmc77Q7lvXW9MxIh9Iz-Q&redir_esc=y#v=onepage&q=Developing%20a%20Cognitive%20Learning%20and%20Intelligent%20Data%20Analysis-Based%20Framework%20for%20Early%20Disease%20Detection%20and%20Prevention%20in%20Younger%20Adults%20with%20Fatigue&f=false

[40]. Padmanaban, H. (2024). Quantum Computing and AI in the Cloud. *Journal of Computational Intelligence and Robotics*, *4*(1), 14–32. Retrieved from https://thesciencebrigade.com/jcir/article/view/116

[41]. Sharma, Y. K., & Harish, P. (2018). Critical study of software models used cloud application development. *International Journal of Engineering & Technology, E-ISSN*, 514-518. https://www.researchgate.net/profile/Harish-Padmanaban-2/publication/377572317_Critical_study_of_software_models_used_cloud_application_development/links/65ad55d7ee1e1951fbd79df6/Critical-study-of-software-models-used-cloud-application-development.pdf

[42]. Padmanaban, P. H., & Sharma, Y. K. (2019). Implication of Artificial Intelligence in Software Development Life Cycle: A state of the art review. *vol*, *6*, 93-98. https://www.researchgate.net/profile/Harish-Padmanaban-2/publication/377572222_Implication_of_Artificial_Intelligence_in_Software_Development_Life_Cycle_A_state_of_the_art_review/links/65ad54e5bf5b00662e333553/Implication-of-Artificial-Intelligence-in-Software-Development-Life-Cycle-A-state-of-the-art-review.pdf

[43]. Harish Padmanaban, P. C., & Sharma, Y. K. (2024). Optimizing the Identification and Utilization of Open Parking Spaces Through Advanced Machine Learning. *Advances in Aerial Sensing and Imaging*, 267-294. **https://doi.org/10.1002/9781394175512.ch12**

[44]. PC, H. P., Mohammed, A., & RAHIM, N. A. (2023). *U.S. Patent No. 11,762,755*. Washington, DC: U.S. Patent and Trademark Office. https://patents.google.com/patent/US20230385176A1/en

[45]. Padmanaban, H. (2023). Navigating the intricacies of regulations: Leveraging AI/ML for Accurate Reporting. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, *2*(3), 401-412. DOI: https://doi.org/10.60087/jklst.vol2.n3.p412

[46]. PC, H. P. Compare and analysis of existing software development lifecycle models to develop a new model using computational intelligence. https://shodhganga.inflibnet.ac.in/handle/10603/487443