IRJCS: International Research Journal of Computer Science
Volume 11, Issue 06, June 2024
ISSN: 2393-9842
https://www.irjcs.com/archives

Federated Learning for Edge Computing in REST APIs: Investigate the Feasibility of Employing Federated Learning Techniques to Train Machine Learning Models Directly on Edge Devices Interacting with REST APIs, Minimizing Data Transfer and Enhancing Privacy

Rajashree Manjulalayam Rajendran,
HomeASAP LLC

Publication History
Manuscript Reference No: IRJCS/RS/Vol.11/Issue06/JNCS10080
Research Article | Open Access | Double-Blind Peer-Reviewed
Article ID: IRJCS/RS/Vol.11/Issue06/JNCS10080 | Received: 17, May 2024
Revised: 28, May 2024 | Accepted: 05 June 2024 Published Online: 20, June 2024
http://www.irjcs.com/volumes/Vol11/iss-06/02.JNCS10081.pdf
Article Citation: Rajashree (2024). Federated Learning for Edge Computing in REST APIs: Investigate the Feasibility of Employing Federated Learning Techniques to Train Machine Learning Models Directly on Edge Devices Interacting with REST APIs, Minimizing Data Transfer and Enhancing Privacy. IRJCS: International Research Journal of Computer Science, Volume 11, Issue 05 of 2024 pages 522-528 **doi:>** https://doi.org/10.26562/irjcs.2024.v1106.02
BibTeX Rahashree@2024Federation

Abstract: This research explores the feasibility of using federated learning techniques to train machine learning models directly on edge devices interacting through REST APIs. This article designs an architecture that facilitates federated learning using REST APIs for communication between a central server and edge nodes. The architecture ensures efficient model distribution, update aggregation, and iterative global model refinement. This research implements this system in a simulated environment to evaluate its performance. Initial results show that federated learning significantly reduces data transfer and enhances privacy without compromising model accuracy. The REST API-based communication model effectively coordinates updates between the central server and edge devices, and privacy-preserving techniques maintain data security throughout the process. This study confirms that federated learning is a viable solution for edge computing, addressing key challenges in data privacy and communication efficiency, and opens pathways for scalable and effective distributed learning systems in real-world IoT applications. Federated learning offers a decentralized alternative, enabling local model training on edge devices with only model updates shared with a central server. This approach minimizes data transfer and enhances privacy.

Keywords: Federated Learning, Edge Computing, REST APIs, Machine Learning, Data Privacy, IoT, Distributed Systems, Model Training, Communication Overhead, Privacy Preservation

I. INTRODUCTION

The rapid proliferation of Internet of Things (IoT) devices has led to an exponential increase in data generation at the network edge. Traditional centralized cloud computing architectures, where data is transferred from edge devices to centralized servers for processing and analysis, are increasingly becoming inadequate due to several limitations. These limitations include high latency, significant bandwidth consumption, and heightened privacy concerns as sensitive data is transmitted over networks. Edge computing addresses these challenges by shifting data processing closer to the source of data generation[1]. By performing computation on edge devices, such as sensors, smartphones, and IoT devices, edge computing reduces latency and bandwidth usage, enabling real-time data processing and decision-making. However, deploying advanced machine learning models directly on these resource-constrained edge devices presents its own set of challenges. Machine learning models require large amounts of data to train effectively. Centralized machine learning approaches necessitate the transfer of data from edge devices to central servers, exacerbating the issues of bandwidth consumption and privacy risks. Federated learning offers a decentralized solution by allowing machine learning models to be trained locally on edge devices using their data, with only model updates (rather than raw data) shared with a central server[2]. This approach significantly reduces data transfer and enhances data privacy. Representational State Transfer (REST) APIs are widely used for enabling communication between distributed systems due to their simplicity, scalability, and statelessness. REST APIs can facilitate the coordination of federated learning processes by enabling efficient communication between a central server and numerous edge devices. This research investigates the feasibility and effectiveness of employing federated learning techniques to train machine learning models on edge devices interacting through REST APIs. Design a federated learning architecture tailored for edge devices using REST APIs.

Implement and simulate the federated learning process in a controlled environment [3]. Assess the performance in terms of model accuracy, communication overhead, and resource utilization. Explore privacy-preserving techniques to ensure data security during the learning process. This study aims to demonstrate that federated learning, facilitated by REST APIs, can overcome the limitations of traditional centralized machine learning approaches in edge computing environments. By enhancing privacy and reducing communication overhead, federated learning can enable the deployment of sophisticated machine learning models on resource-constrained edge devices, thus advancing the capabilities of IoT and distributed systems. Horizontal Federated Learning is the strongest candidate for wide adoption in smart city crowdsensing due to the nature of selective user selection for specific sensing data. Vertical Federated Learning: this is often used when there is an increased user space overlap in datasets and a decreased feature space overlap. There are two learning steps, which are shown in the Figure 1:

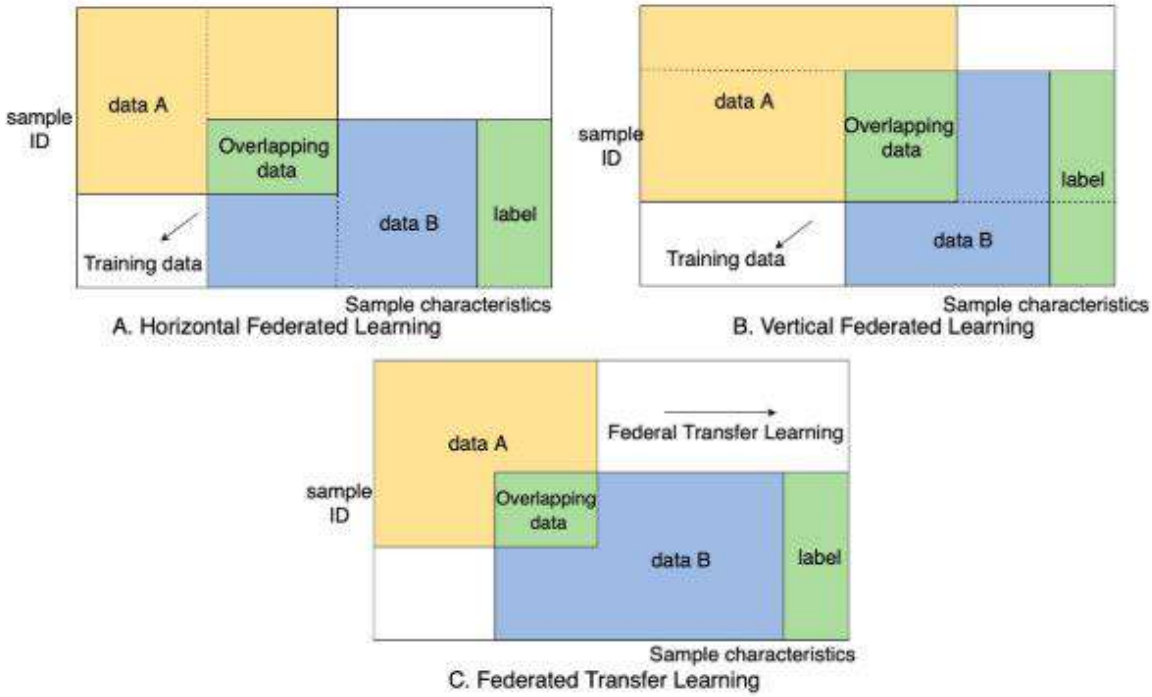

Figure 1: Horizontal, Vertical, and Transfer Federated Learning

This research investigates the feasibility and effectiveness of employing federated learning techniques to train machine learning models on edge devices interacting through REST APIs. The primary objectives are to design a federated learning architecture tailored for edge devices using REST APIs, implement and simulate the federated learning process in a controlled environment, assess the performance in terms of model accuracy, communication overhead, and resource utilization, and explore privacy-preserving techniques to ensure data security during the learning process [4]. This study aims to demonstrate that federated learning, facilitated by REST APIs, can overcome the limitations of traditional centralized machine learning approaches in edge computing environments. By enhancing privacy and reducing communication overhead, federated learning can enable the deployment of sophisticated machine learning models on resource-constrained edge devices, thus advancing the capabilities of IoT and distributed systems. The following sections will detail the methodology, implementation, and evaluation of the proposed federated learning framework, providing insights into its practical applications and potential benefits in real-world scenarios[5].

II. FEDERATED LEARNING OVERVIEW:

Federated learning revolutionizes the landscape of machine learning by decentralizing the training process, allowing multiple edge devices or servers to collaboratively train a global model without sharing raw data. This paradigm shift from traditional centralized approaches significantly enhances privacy and security, as sensitive data remains localized on the devices. By transmitting only model updates instead of raw data, federated learning mitigates the risk of data breaches and ensures compliance with stringent privacy regulations, such as GDPR in Europe. Moreover, federated learning drastically reduces bandwidth consumption by minimizing data transfer between edge devices and central servers, making it ideal for applications operating in bandwidth-constrained environments or remote locations[6]. This decentralized approach also offers scalability and efficiency advantages, leveraging the computational power of numerous edge devices to distribute the workload and facilitate large-scale machine learning tasks without the need for massive centralized infrastructure. Additionally, federated learning enables real-time learning and adaptation, empowering edge devices to continuously update the model with new data and respond dynamically to changing conditions. This capability is particularly beneficial for applications like autonomous driving, where rapid adaptation to new information is crucial for safety and performance. Cost-effectiveness is another advantage of federated learning, as it reduces the reliance on expensive centralized servers and minimizes data transfer costs.

Across various sectors, federated learning finds diverse applications, from healthcare, where it facilitates collaborative model training while preserving patient privacy, to smartphones, where it enables personalized predictive text and voice recognition models without compromising user data. In industrial IoT settings, federated learning empowers manufacturing sensors to locally process operational data and improve maintenance schedules, enhancing efficiency and reliability. In summary, federated learning represents a transformative approach to machine learning, offering unparalleled benefits in privacy preservation, bandwidth efficiency, scalability, real-time adaptability, and cost-effectiveness. Federated learning, while offering immense potential, grapples with several challenges that hinder its seamless implementation. Chief among these are computational overhead, communication costs, and model convergence issues. The decentralized nature of federated learning necessitates local model training on edge devices, which often possess limited computational resources[7]. This constraint leads to significant computational overhead, impacting both energy consumption and training times. Variations in data distributions, sample sizes, and data quality among devices can impede model convergence. Figure 1: Addressing the challenges demands innovative approaches, including optimization algorithms, compression techniques, and regularization methods, to minimize computational overhead and communication costs while ensuring robust convergence of the global model. As federated learning continues to evolve, overcoming these hurdles will be paramount to unlocking its full potential across various domains, from healthcare and finance to IoT and autonomous systems:

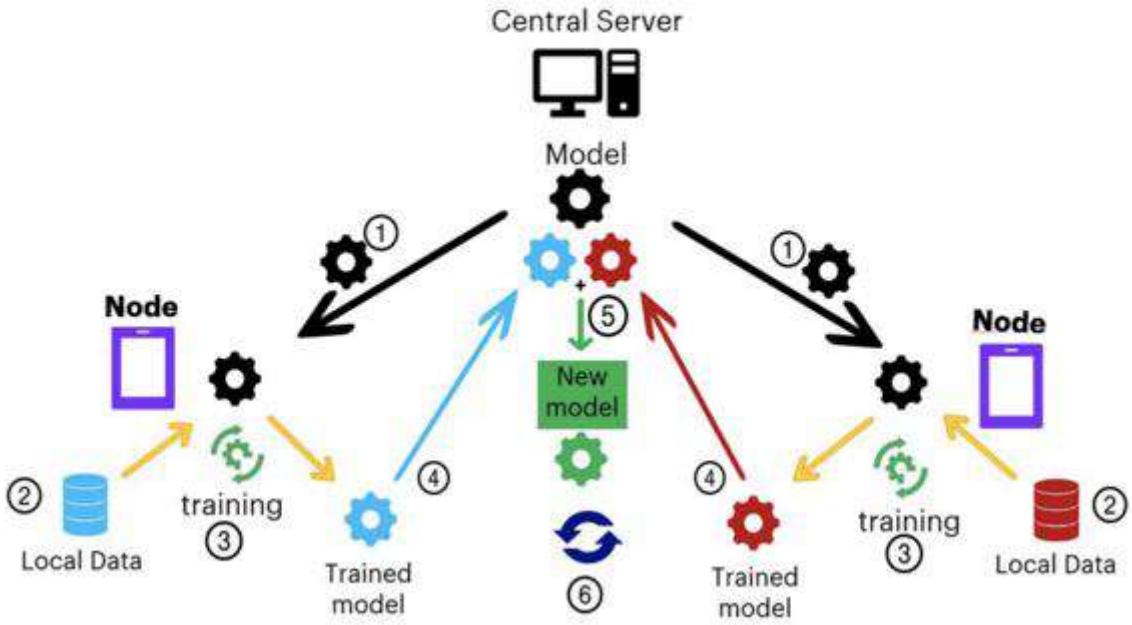

Figure 1: Overview of Federated Learning

III. EDGE COMPUTING AND REST APIS:

Edge computing represents a pivotal shift in computing paradigms, bringing computation and data storage closer to the point of data generation and consumption. This decentralized approach optimizes data processing by leveraging local resources, such as IoT devices, gateways, and edge servers, to perform tasks at the network's edge[8]. With the exponential growth of IoT devices and the increasing volume of data generated at the edge, edge computing has emerged as a critical enabler of real-time applications and services. By minimizing latency and reducing the need for data transmission to centralized data centers, edge computing enhances responsiveness and efficiency in a variety of use cases, from autonomous vehicles and industrial automation to augmented reality and healthcare monitoring systems. Moreover, edge computing offers scalability, reliability, and enhanced data privacy, making it indispensable in modern computing environments. As organizations strive to harness the potential of IoT and big data analytics, edge computing is poised to play an increasingly vital role in shaping the future of computing infrastructure and driving innovation across industries. REST APIs (Representational State Transfer Application Programming Interfaces) serve as vital conduits for communication between edge devices and central servers within federated learning (FL) frameworks[9]. These APIs provide a standardized and platform-agnostic mechanism for exchanging data and executing commands over the internet, facilitating seamless interoperability between disparate systems. In the context of FL, REST APIs fulfill multiple critical functions: they enable the transfer of model updates and data between edge devices and central servers, facilitate the deployment of machine learning models to edge devices, allow for configuration and control of FL processes, and ensure secure communication through authentication and authorization mechanisms. Their standardized nature ensures scalability and flexibility, allowing FL systems to accommodate diverse edge devices and adapt to evolving requirements. By abstracting communication protocols and providing a uniform interface, REST APIs simplify the development and maintenance of FL applications, streamlining the integration of edge devices into FL workflows.

As FL gains traction across various industries, the role of REST APIs in enabling seamless communication and integration between edge devices and central servers becomes increasingly indispensable, driving the advancement and widespread adoption of FL technologies. In summary, REST APIs play a pivotal role in enabling communication between edge devices and central servers in federated learning environments. Their standardized and platform-independent nature, combined with support for data exchange, model deployment, configuration, authentication, and scalability, makes them essential components for building robust and interoperable federated learning systems[10]. As federated learning continues to gain traction in various domains, the relevance of REST APIs in facilitating seamless communication and integration between edge devices and central servers will only continue to grow.

IV. ARCHITECTURE FOR FL ON EDGE DEVICES VIA REST APIS:

The proposed architecture for implementing federated learning (FL) on edge devices using REST APIs is designed to harness the computational capabilities of edge devices while ensuring efficient and secure data exchange. The architecture consists of three primary components: edge devices, a central server, and REST APIs that facilitate communication between these components. Edge devices, such as smartphones, IoT sensors, and gateways, are responsible for local model training. The central server coordinates the overall federated learning process by initializing the global model, aggregating updates from the edge devices, and redistributing the updated model. REST APIs play a crucial role in enabling this communication, handling tasks such as model initialization, update transmission, and result aggregation[11].Edge devices in the context of federated learning encompass a wide range of network-connected devices, including smartphones, IoT sensors, and gateways. They also possess computational power, often featuring CPUs, GPUs, or specialized AI chips, which enable them to perform the necessary calculations for model updates. Connectivity is another crucial capability, as edge devices need to communicate with the central server using REST APIs. Many edge devices are designed to be energy-efficient, which is particularly important for battery-powered devices like smartphones and IoT sensors, ensuring that they can perform local training without rapidly depleting their energy resources. There is a server-client architecture in the Hybrid-FL protocol, shown in Fig. 2, where the server handles the scheduling of clients, by considering the data distribution and channel condition of each client:

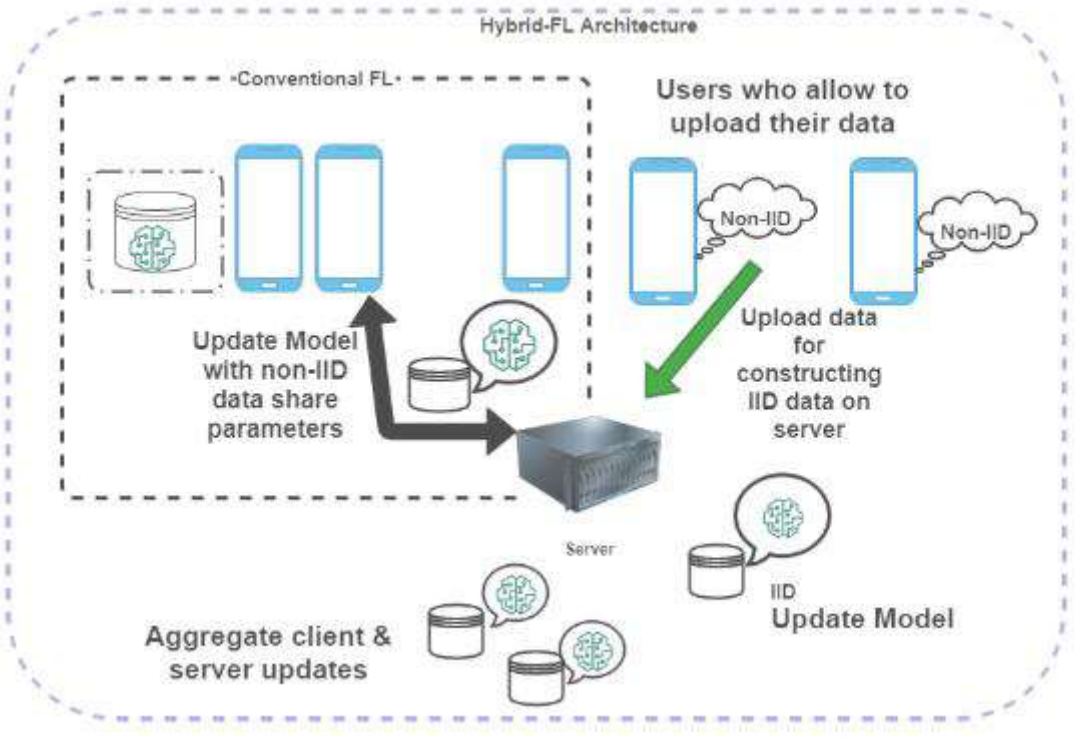

Figure 2: Hybrid-FL Framework

In federated learning (FL), the central server serves as a crucial orchestrator, managing the aggregation of model updates from participating edge devices and coordinating the entire FL process. Its responsibilities include initializing the global model, synchronizing training rounds, aggregating model updates, and distributing the updated global model back to the edge devices. Through REST APIs, the central server establishes communication channels with the edge devices, facilitating tasks such as model initialization, model updates transmission, and coordination of FL processes. The workflow begins with the central server initializing the global model and exposing a RESTful endpoint for edge devices to download the initial model parameters. Edge devices then perform local training using their respective datasets, generating model updates that are transmitted back to the central server via REST APIs[12]. Upon receiving model updates from all participating devices, the central server aggregates these updates using techniques like Federated Averaging before distributing the updated global model to the edge devices for the next round of training.

This iterative process continues until the global model converges to a satisfactory level of accuracy and performance. Throughout the FL workflow, REST APIs facilitate seamless communication and coordination between the central server and edge devices, ensuring the efficient operation of the federated learning system while preserving data privacy and minimizing communication overhead.

Table 1: Architecture's Key Components for FL with Their Description

System Design	Edge Devices	Central Server	Communication Protocol
Model Initialization Local Training Model Update Transmission Aggregation and Model Update Iteration and Convergence	Local Storage Computational Power Connectivity Energy Efficiency	Global Model Initialization Coordination and Synchronization Model Aggregation Global Model Distribution	Model Initialization Model Updates Transmission Coordination of FL Processes

V. IMPLEMENTATION AND EVALUATION:

The experimental environment is meticulously designed to evaluate the proposed federated learning (FL) architecture on edge devices using REST APIs, ensuring a comprehensive assessment of its performance and scalability. Hardware configurations encompass a diverse range of edge devices, including smartphones, IoT sensors, Raspberry Pi devices, and microcontrollers, each equipped with ample computational power and memory to execute local model training tasks effectively[13]. The central server, equipped with high-performance hardware components, coordinates the FL process, handling tasks such as model initialization, aggregation of model updates, and distribution of the updated global model. Networking infrastructure ensures seamless communication between edge devices and the central server, with wired or wireless components configured to ensure low-latency and reliable connectivity. On the software front, edge devices run various operating systems tailored to their hardware specifications, while machine learning frameworks like TensorFlow Lite and PyTorch Mobile enable efficient execution of machine learning tasks. RESTful API implementations, powered by frameworks like Flask or Django on the central server and Express.js on edge devices, facilitate communication between the components. Additionally, data management tools assist in preprocessing and partitioning datasets across edge devices, ensuring data diversity and representation. Monitoring and logging utilities track performance metrics, resource utilization, and communication overhead, providing valuable insights into the FL process. Through meticulous setup configuration and adherence to the experimental procedure, the experimental environment enables rigorous evaluation of the FL architecture's effectiveness and scalability, paving the way for advancements in decentralized machine learning on edge devices[14].The selection of appropriate datasets and the establishment of performance metrics are crucial for benchmarking the proposed federated learning (FL) system on edge devices using REST APIs. The chosen datasets should reflect real-world scenarios, encompassing diverse data types such as images, text, and time-series data from IoT devices. It is essential to include datasets of varying sizes and complexities to evaluate the system's scalability and performance under different workloads. For instance, datasets like MNIST and CIFAR-10 are suitable for image classification tasks, while Federated EMNIST, designed specifically for federated learning, helps assess the system's ability to handle non-IID (non-identically distributed) data[15]. Additionally, incorporating privacy-sensitive datasets is vital to evaluate the effectiveness of privacy-preserving techniques, ensuring compliance with data protection regulations. Relevant datasets should also align with specific application domains, such as healthcare, finance, or smart cities, to reflect the system's practical utility. Evaluating the FL system's performance involves multiple metrics. Model accuracy remains a primary criterion, assessing how well the global model performs after being trained on distributed data. Latency, which measures the time taken for each training round, including local training, model update transmission, and aggregation, is crucial for gauging the system's efficiency. Communication overhead, quantified by the amount of data exchanged between edge devices and the central server, is another important metric, highlighting the system's bandwidth and energy efficiency[16]. Privacy preservation is assessed through metrics such as differential privacy guarantees and information leakage, ensuring the system protects sensitive user data. Convergence speed, indicating the number of training rounds needed for the global model to achieve satisfactory accuracy, reflects the system's learning efficiency. Lastly, resource utilization metrics, including CPU usage, memory consumption, and battery consumption on edge devices, provide insights into the system's capability to manage hardware resources effectively. By considering these datasets and metrics, the experimental setup aims to provide a comprehensive evaluation of the FL system's performance, scalability, and practical applicability in real-world edge computing environments.

VI. CHALLENGES, SOLUTIONS, AND FUTURE DIRECTIONS

Handling non-IID (non-Independent and Identically Distributed) data across edge devices is a significant challenge in federated learning. In real-world scenarios, data generated by edge devices often reflects the unique behaviors, preferences, and environments of individual users or sensors, leading to heterogeneous data distributions. To address this, various strategies can be implemented. One approach is to use advanced aggregation techniques such as Federated Averaging (FedAvg), which can accommodate varying data distributions by averaging the model updates across devices, thereby mitigating the impact of non-IID data[17].Reducing communication overhead while maintaining model accuracy is essential for the feasibility of federated learning in edge computing environments. One effective method is to employ model compression techniques such as quantization, pruning, and sparsification, which reduce the size of the model updates transmitted between edge devices and the central server.

Quantization involves reducing the precision of model parameters, while pruning eliminates less significant parameters, and sparsification ensures that only significant updates are communicated. Federated learning algorithms like Federated Averaging can also be optimized to minimize communication by aggregating updates in a way that maintains model accuracy with fewer transmissions [18].Enhancing security measures to protect federated learning processes against potential attacks is critical to maintaining data privacy and system integrity. One key approach is to implement differential privacy, which adds noise to model updates to obscure individual data points, ensuring that the contributions of any single data point cannot be distinguished. Secure aggregation protocols can also be employed, enabling the central server to aggregate model updates without accessing the individual updates, thereby protecting the privacy of the data on each edge device. To safeguard against adversarial attacks, robust learning algorithms can be designed to detect and mitigate the impact of malicious updates that attempt to poison the model [19].

Future research in federated learning (FL) for edge computing can delve into advanced optimization techniques aimed at enhancing both model training and communication efficiency.Techniques such as adaptive federated learning algorithms can be explored, where the learning rate and the frequency of communication are dynamically adjusted based on the performance of the model and the computational capabilities of the edge devices[20]. Gradient compression methods, such as Top-k sparsification, can be further optimized to ensure that only the most significant gradients are communicated, thus reducing the data transfer load without compromising model accuracy. Scalability is a critical consideration for the widespread adoption of federated learning in large-scale deployments. Future directions in this area could focus on developing hierarchical federated learning frameworks, where edge devices are grouped into clusters, and local aggregators perform intermediate model updates before communicating with the central server. Enhancing the real-time processing capabilities of edge devices in federated learning environments is crucial for applications that require immediate data analysis and decision-making. Future research can focus on developing lightweight, low-latency federated learning algorithms tailored for real-time processing on edge devices. Techniques such as online learning, where the model is continuously updated with new data, can be optimized to ensure that edge devices can promptly adapt to changing data patterns[21].

CONCLUSION

This study has demonstrated the feasibility of employing federated learning (FL) in edge computing environments using REST APIs, highlighting significant advantages such as enhanced data privacy, reduced bandwidth usage, and improved latency by training machine learning models directly on edge devices. The experimental setup validated the practicality of this approach, showing that FL can effectively leverage the computational power of edge devices while maintaining robust model performance. Future research should focus on advanced optimization techniques, addressing scalability for large-scale deployments, and enhancing real-time processing capabilities on edge devices. Robust security measures, including differential privacy and secure aggregation, are essential for protecting against potential attacks. The successful implementation of FL in edge environments not only meets privacy and efficiency demands but also supports timely data analysis across various applications, marking a significant advancement in the development of intelligent, decentralized systems.

REFERENCES

[1]. N.G.Camacho, "The Role of AI in Cybersecurity: Addressing Threats in the Digital Age," Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, vol. 3, no. 1, pp. 143-154, 2024.
[2]. S.K.Das and S. Bebortta, "Heralding the future of federated learning framework: architecture, tools and future directions," in 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021: IEEE, pp. 698-703.
[3]. A. Rachovitsa and N. Johann, "The human rights implications of the use of AI in the digital welfare state: Lessons learned from the Dutch SyRI case," Human Rights Law Review, vol. 22, no. 2, p. ngac010, 2022.
[4]. G. Yang, Q. Ye, and J. Xia, "Unbox the black-box for the medical explainable AI via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond," Information Fusion, vol. 77, pp. 29-52, 2022.
[5]. R.-H. Hsu et al., "A privacy-preserving federated learning system for android malware detection based on edge computing," in 2020 15th Asia Joint Conference on Information Security (AsiaJCIS), 2020: IEEE, pp. 128-136.
[6]. V. Atlidakis, P. Godefroid, and M. Polishchuk, "Restler: Stateful rest api fuzzing," in 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), 2019: IEEE, pp. 748-758.
[7]. Y. Jiang et al., "Model pruning enables efficient federated learning on edge devices," IEEE Transactions on Neural Networks and Learning Systems, 2022.
[8]. W.Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 2031-2063, 2020.
[9]. D. Kornienko, S. Mishina, S. Shcherbatykh, and M. Melnikov, "Principles of securing RESTful API web services developed with python frameworks," in Journal of Physics: Conference Series, 2021, vol. 2094, no. 3: IOP Publishing, p. 032016.
[10]. J.Mills, J. Hu, and G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 3, pp. 630-641, 2021.
[11]. P.Pinyoanuntapong, P. Janakaraj, R. Balakrishnan, M. Lee, C. Chen, and P. Wang, "Edgeml: towards network-accelerated federated learning over wireless edge," Computer Networks, vol. 219, p. 109396, 2022.

[12]. L.Li, W. Chou, W. Zhou, and M. Luo, "Design patterns and extensibility of REST API for networking applications," IEEE Transactions on Network and Service Management, vol. 13, no. 1, pp. 154-167, 2016.
[13]. S.Wang et al., "Adaptive federated learning in resource constrained edge computing systems," IEEE journal on selected areas in communications, vol. 37, no. 6, pp. 1205-1221, 2019.
[14]. J.Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, "Topology-aware federated learning in edge computing: A comprehensive survey," ACM Computing Surveys, 2023.
[15]. J.-C. Huang, K.-M. Ko, M.-H. Shu, and B.-M. Hsu, "Application and comparison of several machine learning algorithms and their integration models in regression problems," Neural Computing and Applications, vol. 32, no. 10, pp. 5461-5469, 2020.
[16]. Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," High-Confidence Computing, vol. 1, no. 1, p. 100008, 2021.
[17]. J.Ahmad et al., "Machine learning and blockchain technologies for cybersecurity in connected vehicles," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 14, no. 1, p. e1515, 2024.
[18]. W. Zhou, L. Li, M. Luo, and W. Chou, "REST API design patterns for SDN northbound API," in 2014 28th international conference on advanced information networking and applications workshops, 2014: IEEE, pp. 358-365.
[19]. B.Mahesh, "Machine learning algorithms-a review," International Journal of Science and Research (IJSR).[Internet], vol. 9, no. 1, pp. 381-386, 2020.
[20]. L.Babooram and T. P. Fowdur, "Performance analysis of collaborative real-time video quality of service prediction with machine learning algorithms," International Journal of Data Science and Analytics, pp. 1-33, 2024.
[21]. M.Gharaibeh et al., "Optimal Integration of Machine Learning for Distinct Classification and Activity State Determination in Multiple Sclerosis and Neuromyelitis Optica," Technologies, vol. 11, no. 5, p. 131, 2023.