Resource Management in Healthcare IoT Networks Using Machine Learning for Accuracy Enhancement

Archana Mongia*, Indu Bhardwaj

ECE Department, Galgotias University, Greater Noida 203201, Uttar Pradesh, India

*archanamongia287@gmail.com

* Corresponding author

doi: https://doi.org/10.21467/proceedings.7.6.31

Abstract

The objective of this paper was to improve the accuracy of resource management in IoT healthcare environments using machine learning models. A simulated dataset was created that considered client conditions, resource skills, and familiarity factors. Interactions were predicted using a Multiclass neural network and XGBoost decision tree models. The prediction model was trained and deployed using Azure ML Studio, which was integrated into the real-time framework of Azure IoT Hub. Results showed that predicted resource selection improved client satisfaction compared to random allocation. The study concluded that ML models are effective in resource optimization and can be deployed in real-time IoT environments. This instance involves an organization supplier managing IoT resources through machine learning. This might be a service provider, a business that fixes equipment, or a hospital where patients are the clients and the medical staff provides the service. The client rates the outcome of each encounter between a client and a resource on a three-point scale ('-1' for a poor interaction, '0' for a neutral interaction, and '+1' for a positive interaction). The score the following interaction with a certain resource, this collection of interactions is merged with the circumstances business keeps a note of these interactions as well as the conditions that led to them. To train a predictor for how the client would that led to the interactions. Resources are allocated to the highest-rated predictions to maximize client satisfaction.

Keywords: *Internet of Things, machine learning, resource allocation.*

1. Introduction

There will be 14.4 billion linked people and 21 billion intelligent embedded devices producing fifty trillion gigabytes of information by 2022[1]. Figure. 1 shows IoT network device projections in billions. By 2030, it is estimated that 39 billion Internet of Things (IoT) devices will be installed worldwide. For the future of smart work, the Internet of Things (IoT) offers considerable potential [13]. There are numerous difficulties associated with the widespread adoption of IoT. Among these issues are protection identification, effective transmission of information protocols, resilient networking and storage architecture for connected devices, and protection of IoT applications and devices against malicious attacks and application interfaces [2]. This paper uses a multiclass neural network to improve the data set's accuracy. For a specific set of inputs, the Azure model is used for training and for predicting the results.

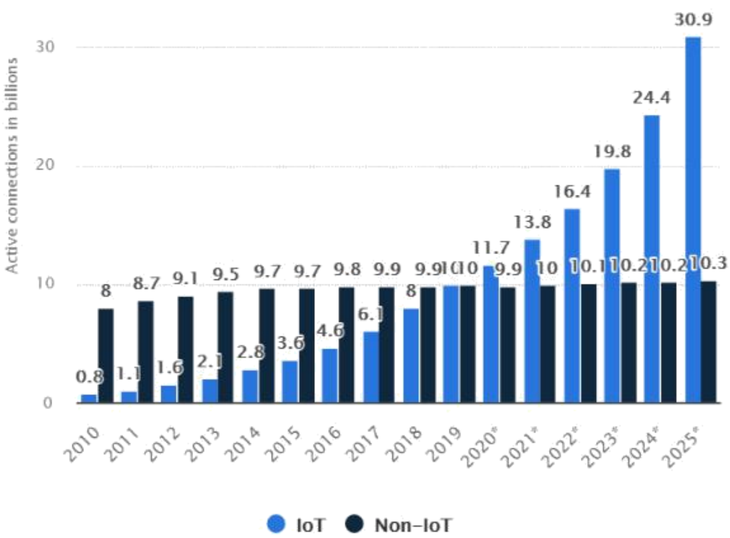

Figure1. IoT Connected Devices Projection in Billion

© 2025 Copyright held by the author(s). Published by AIJR Publisher in "Proceedings of the 3rd International Conference on Artificial Intelligence, Machine Learning and Cybersecurity". Organized by HMR Institute of Technology and Management, New Delhi, India on 1-2 May 2025.

Proceedings DOI: 10.21467/proceedings.7.6; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-989164-9-5

2. Literature Review

When there are a lot of users or smart devices, resource management in real networks becomes rather difficult. The overall efficiency of all wireless network depends on how well and adaptably hyperdimensional components like radio bands, slot timings, and orthogonal codes are managed [2]. Many connections in device-to-device (D2D) are used in 5G mission-critical communication (MCC) to boost communication dependability. Mobile data from cellular users that are not orthogonal and do not require a base station (BS) can be reused by D2D users. The co-channel interference from D2D users will affect cellular subscribers' quality of service. To increase the reliability of mission-critical communication, MCC network device-to-device (D2D) communication enables mobile devices to undertake direct interaction between peers transmission using the licensed spectra assigned concerns cellular services [3]. The ability of D2D technology to offer cellular services with enormous capacity, high speeds, and quality of service (QOS) assurances drawn the attention of academia as well as industry.

2.1 An IoT network's management of resources challenge

Large-scale access to channels***:*** When multiple devices attempt to access a wireless link at the same time, it becomes overloaded. Massive deployment of IoT devices is creating issues with network congestion, channel overload, smart device networking and storage design, and effective data communication protocols [4][5]. Extension of coverage**:** In contrast to conventional devices like smartphones and tablets, IoT devices often operate at low power and have limited storage. This phenomenon results in a smaller coverage area. Power distribution and interference control: In a dense IoT network, intra and intercellular interference issues become critical, necessitating the use of power distribution and interference control solutions. A heterogeneous IoT application may have variable network topology, traffic volume, and channel characteristics. Use hand-off management in conjunction with cell selection*:* Devices in cellular IOTs must connect to a gateway node or a base station in order as a means for connect to the web. In both scenarios, uplink and downlink the relationship may not always be symmetric IoT devices are associated with several BSs at the uplink and downlink). The resource allocation is affected by this association or cell choice. Human-to-human (H2H) and IOT traffic coexist peacefully: For the old and new IoT traffic to coexist peacefully, effective resource management (such as for channel and power allocation, interference control, and utilization device association) is crucial. Both communication channels can coexist as long as they serve their respective purposes. Energy management*:* The majority of IoT sensors are powered by batteries, which have limited capacity and recharge options. To ensure adequate QoS, it is vital to allocate resources and build communication protocols that are energy efficient.

2.2 D2D Communication

In situations when there is no network infrastructure, for the transfer of information, the (crucial) MC site needs the D2D network. and offer the MCC systems access to the MC communication and data services [6]. D2D communications, which utilize WLAN resources that are not orthogonal from the mobile user through choosing a means of communication and carrying out data transfer in D2D networks with no base station (BS), can be in-band or out-band [6].

2.3 Optimal Resource Allocation:

The instantaneous decaying scenario is represented in mathematics by a random variable h and a matched instantaneous resource distribution P(h), and a performance score that occurs instantly, f(p(h), h) as a result of the resource allocation P(h) at that point in when it occurred for the cell is h. When compared to how the end user interprets the system's average over time X = E[f(P(h),h] to mean full matrices, the system's instantaneous performance tends to change too quickly. We utilized to be able to optimize the importance of the extended average X while adhering with the constraint that X = E[f(P(h),h] due to this interaction between instantaneous resource allocation and overall performance over time, which leads to unique forms [7].

2.4 Limitations of traditional resource management

The optimization method is used to address the issues with resource allocation while taking into account the users' immediate CSI and QoS needs. Globally optimal solutions cannot be achieved with the usual method. While it's possible that the answers won't be found instantly. Because of the propagation environment's attributes or consumer habits of motion, for example, a mathematical description of the issue of resource allocation may not be sufficient. The optimization problem might not be one we can formulate. The resource allocation technique based on data-driven machine learning would make sense from this perspective.

2.5 Machine Learning:

The role of extrapolating knowledge from information and using that knowledge to direct an ML agent's behavior based on that understanding can be stated in the term machine learning (ML). Classification, regression, and density estimation all involve machine-learning techniques. IoT devices produce an enormous amount of data, which may be used by data-driven ML approaches to create automated IoT service solutions. Deep learning (DL) or machine learning (ML) can be utilized for feature extraction when the data is large and multidimensional.

2.6 Resource allocation in cellular IoT networks:

IoT cellular networks (like 5G cellular and beyond) ought to support excessively high rates of data that demand constant connectivity to service heterogeneous devices and user groups together with different data QoS demands. The transmitting gadget must independently connect to either a network access node (AP) or a base station, and gain access to the channels of frequencies to achieve a suitable ratio of signal to noise. The angles of departure (AOD) and arrival (AOA) are parameters for communication in millimeter waves. The CSI and AOA values are unknown before in millimeter wave communication can be approximated using the RL approach [8] [9].

2.7 Resource management in IoT network:

The difficulties with resource management in four different IoT networks, including three types of IoT networks: cognitive, low-power, and cellular. Resource management strategies operate in many ways depending on the Cellular IoT network's communication mode, such as the Device to Device (D2D) mode, HetNet mode, NOMA mode, or dynamic spectrum access-based mode. IoT networks with low power usage, particularly those that offer long-distance communication like the low power wide area network, are becoming essential. By restricting the data speeds and energy used, LPWANS have been researched for a variety of smart applications. The cognitive networks have proposed numerous approaches for solving different resource management problems. One of an IoT network's cognitive properties is that its nodes continuously look for resources that are more advantageous for them in improving the network's performance as a whole. When the primary user is not present or is not using the resources, the cognitive networks distribute those resources to the so-called "secondary user." The secondary user must leave that channel as immediately as the primary device is engaged, which could happen at any time.

2.8 Machine learning and deep learning for resource management:

Reinforcement learning, supervised learning, and unsupervised learning are the three subcategories of machine learning. Supervised learning: Random Forest, naive Bayes classifier, and support vector machine (SVM) are typically utilized the supervised learning techniques in categorization as well as modeling the current data sets. These techniques can estimate and forecast unknown parameters using models and pre-given labels. These family methods are employed for localization, spectrum sensing, and estimating channel issues.
Unsupervised learning**:** These methods handle input data heuristically and are utilized with untagged data. Three of these algorithms' main applications in Internet of Things networks are cell clustering, user association, and

balancing the load. Reinforcement learning: RL methods don't need sets of training data a learning agent engages in interactions with its surroundings. IoT sensors and devices can learn RL approaches to make decisions and infer under uncertain and constantly changing network settings.

Deep learning: Deep learning refers to effective reinforcement learning techniques that can handle vast amounts of unstructured data. Deep learning algorithms are appropriate for managing large amounts of data and computationally demanding tasks like speech and picture synthesis, pattern identification in images, etc. As the need for CPU power rises, deep learning activities are frequently carried out using potent GPUs. Deep learning creates deep neural networks, which are similar to ANNs. The artificial neural network on which deep learning is built has numerous concealed layers in between the layers of output and input. The data set's functional relationships are estimated by a deep neural network (DNN). Predicted, categorization, distribution of resources, estimation of spectral state, object identification, translation of language, and detection of speech are just a few of the issues that DL is used to address. The advantages of DL for model training and enhancing prediction accuracy are that it can handle huge amounts of information and that its algorithms scale by combining growing data. Deep reinforcement learning: DRL acts as a hybrid of DNN and RL where a DNN acts as RL's agent. A DNN is taught by using data interaction between the surroundings and approximates the best course of action. It won't be essential to use any previously trained data.

Deep Q Networks (DQN) and Policy Gradient make up the two divisions of DRL. DQN, as well as policy gradient, are both values-driven and regulations-based methodologies, respectively. Every action within the action space has a Q value, and in a value-based technique, the selection of action is set by the Q value.

2.9 Resource Management in Cloud:

Cloud computing, which offers on-demand storage and compute capabilities via the Internet, is driving the rise of implementing the Internet of Things, also known as IoT, in industries such as healthcare, smart cities, Industry 4.0, and others. The cloud has also moved its service offering platform to its next-generation models, like fog, mist, and dew computing, to satisfy the special requirements of connected devices [11].

Resource Management in Fog Computing: An idea called fog computing provides a variety in dispersed resources along the network edge to meet QoS demands. The dynamic nature of user demands and the dispersed and diverse nature as fog computing, yet, have made effectively organizing such resources a major challenge [12].

The Table1 displays an overview of research challenges and strategies for problem-solving a choice made about action using machine learning [1]. Resource Allocation: IoT network resource allocation issues can be resolved with machine learning (ML), as large data sets can be gathered to train algorithms that generate incredibly reliable solutions for a range of resource allocation issues.

Power Allocation and Interference Management: For interference and power control in D2D communication networks, distributed Q-learning and CART Decision Tree algorithms have been discussed in [10]. In addition to increasing system capacity and energy efficiency, the researchers were able to simplify the time complexities. In [10], a technique for measuring interference levels and modifying power in wireless networks based on interference intensity was presented. The scientists employed a linear regression technique that leverages CSI to forecast transmit power levels, thereby reducing interference and power loss. Energy-efficient Collaborative Multiband Spectrum Detection: Cognitive radios (CR) and dynamic spectrum access (DSA) aim to use the underutilized radio spectrum by allowing secondary users to opportunistically use the permissible frequencies for access. Due to the rising demand for wireless services and regulatory agencies' strict spectrum distribution policies, the freely accessible radio spectrum is an important asset. Nevertheless, spectral measurements have found that the majority of the time, significant portions of the spectrum are underutilized [9]. The goals of cognitive radio and dynamic spectrum access (DSA), which seek to identify unlicensed radio spectrum and grant access to secondary users (SU) while guaranteeing that is the licensed main user won't be impacted, were spurred by this discovery.

Table1: *Reviews On Research Problem*

Reviews on Research Problems	**Machine Learning Techniques**
Resource Allocation (Scheduling, Random Access)	Q-Learning
	Reinforcement Learning
	Deep Neural Network
	Simulated Annealing
	Bayes Classification
Power Allocation and Interference Management	Deep Reinforcement Learning
	Linear Regression
	K-Means Clustering

But after a detailed literature study, some important research gaps have emerged:

Benefits of Real-world Human-Centric Feedback: Most studies are limited to theoretical or simulation-based models. There is a lack of ML implementation based on real-world interactions between clients and IoT resources (such as feedback from hospital patients and medical staff) — which is very critical in actual healthcare scenarios.

Limited Use of Multiclass Predictive Models: Much of the research so far is limited to supervised learning models such as Random Forest, SVM or regression. But using multiclass neural networks to predict client interaction outcome has not yet been extensively explored, especially in the healthcare domain.

Lack of Synthetic Data & IoT Integration Pipeline: Many studies either stop at synthetic data creation or just ML model training. But a complete end-to-end system where synthetic data is generated, the model is trained, and then deployed via real IoT devices (such as Azure IoT Hub) — such a pipeline approach is missing in the literature.

Lack of Healthcare Domain-Specific Resource Allocation: While resource allocation has been extensively studied in industrial and smart city applications, there is very little research available on the healthcare environment, where patient condition, resource familiarity, and client satisfaction matter.

Explainability and lack of Prescriptive Analytics: Prediction models are available in many places, but prescriptive analytics – i.e., suggestions for which resource would be best and why – this approach has been largely ignored so far. Especially in a critical field like healthcare, explainable ML is a must.

3. Proposed Multi-Layer Neural Network Model

The script for "Generate Outcome History" produces a set of fictitious interaction data. Three main groups of underlying factors are produced by it: The chronic conditions of each client. 1. The Skills for each resource, 2. The Familiarity between resources and clients. After that, a set of 100,000 interactions is created, with the (client ID, resource ID), client conditions (Cond_0) through (Cond_19), and interaction `outcome (rating) all recorded. The training data is kept in a flat CSV file. From the same hidden factors, a second set of testing data is produced. For every 10,000 possible client conditions, five sets of outcomes are created (for five different resources), for a total of 50,000 points. With the same (client ID) and set of conditions, each of the five outcomes is provided for a distinct resource ID. In an arbitrary scenario, the initial result for every interaction ID would be selected.

Prescriptive analytics for resource allocation is demonstrated in Figure. 2. The Azure training model is displayed in Figure. 3[14].

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

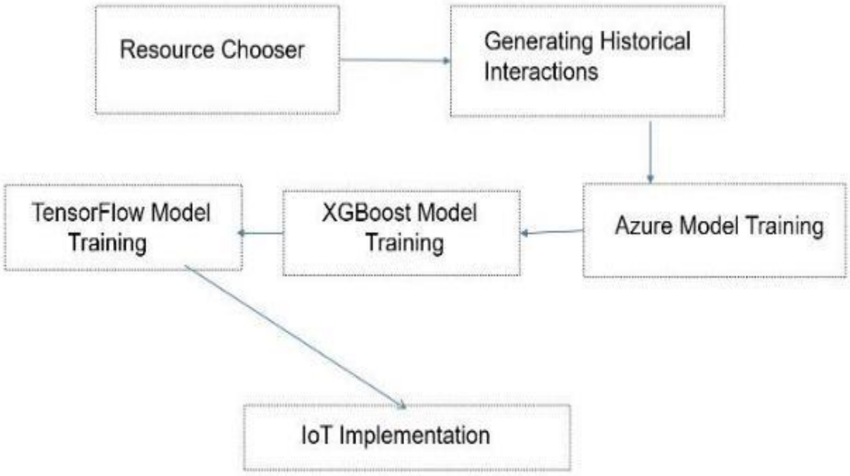

Figure 2. Demonstration of prescriptive analytics for resource allocation.

Training using Azure Model and XG Boost: The [Azure Machine Learning Studio] was used to train a predictive model using the 'train_outcomes.csv' data. To make predictions about the results of a set of inputs, the model employs a Multiclass Neural Network. Creating a predictive web service from the trained model. Locally, the author also developed an XG Boost decision tree model to examine how using the best prediction as opposed to a random resource could affect the outcome of client/resource interactions. The best forecast and the actual result for each "interaction ID" are then determined by looking at each interaction. Tensor Flow Model Training**:** A comparable implementation is carried out using Tensor Flow. This approach saves two separate models, Model 1 and Model 2, each with an accuracy of roughly 85%. Model 1 is trained on a subset of the training data, whereas Model 2 was trained on the entire training set. IoT Implementation**:** An "Azure IoT Hub," several client and resource devices (built using "node-red" on a local server), and an "Azure Web App" are used to implement the scenario. The latter allows for network monitoring and prediction retrieval via an "Azure ML Studio" web app endpoint.

A simulated client device sends randomized data to the user interface dashboard, waits for a button hit to send an alert, and updates the device twin into alert status after reading a random client/resource interaction from the test outcomes data file. A collection of device administration flows. These processes configure the client and resource device tags, enable manual scenario resetting, and enable monitoring of the IoT Hub. In a pattern that resembles the server web app, IoT Hub alerts are monitored, the current resource status and client condition are compiled from the alerts, and those configurations are sent to the Azure ML Studio API for forecasting. Choosing the most accurate prediction prints the most useful source.

4. Algorithmic

This algorithmic outline provides an organized method for implementing the functionality described by the code. It includes data creation, data visualization, result computation, model training, and evaluation. It also highlights how crucial testing, documentation, and iterative improvement are to creating reliable and efficient machine-learning pipelines.

Data Generation and Visualization: Generate synthetic data for resource skills, client familiarity, and client conditions. Save the generated data to CSV files. Visualize the resource skills using a heatmap. Visualize the distribution of outcomes. Outcome Calculation: Define a function to calculate outcomes based on resource skills, client familiarity, and client conditions. Iterate through a large number of samples to generate outcomes for each combination of resource and client.

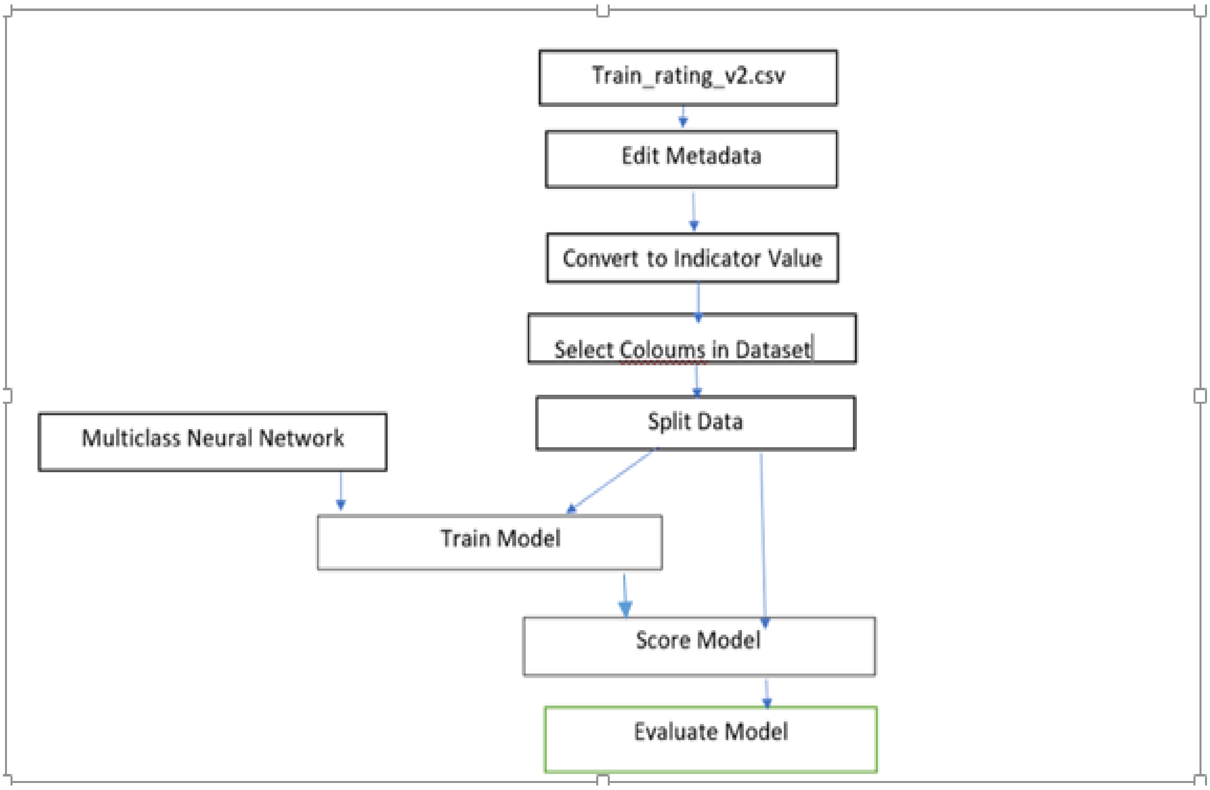

Figure 3. Azure Model Training

Data Preparation: Combine the generated outcomes with client and resource identifiers. Save the combined outcomes data to a CSV file. Model Training: Load the outcomes data. Split the data into train, validation, and training. Preprocess the data (e.g., one-hot encoding categorical variables). Train machine learning models such as neural networks and XGBoost on the training data. Evaluate the models on the validation set. Fine-tune the models based on validation results. Evaluate the final models on the test set.
Documentation and Code Organization**:** Add comments to the code explaining each step and any complex logic. Ensure code readability and organization for easier maintenance and collaboration.
Testing and Validation: Test each component of the code independently. Validate the entire pipeline with synthetic data to ensure correctness and reliability. Iterative Improvement**:** Analyze model performance and identify areas for improvement. Experiment with different model architectures, hyperparameters, and data generation strategies. Iterate on the process to improve model accuracy and generalization.

5. Results

Figure 4. shows samples of the resources used and client interactions. The figure illustrates the sample interactions that we obtained after the simulation. Instead of using a random resource, the author employed the best prediction. Trains a model by loading the training data, and then makes predictions for every potential event. Next, for each interaction ID, the author looks up the best estimate and the actual result for that interaction. The average result of the sample interaction and the sub-rolling average are shown in Figures 4 and 5 respectively. The train test best rolling and the train test best average are shown in Figures 6 and 7 respectively.

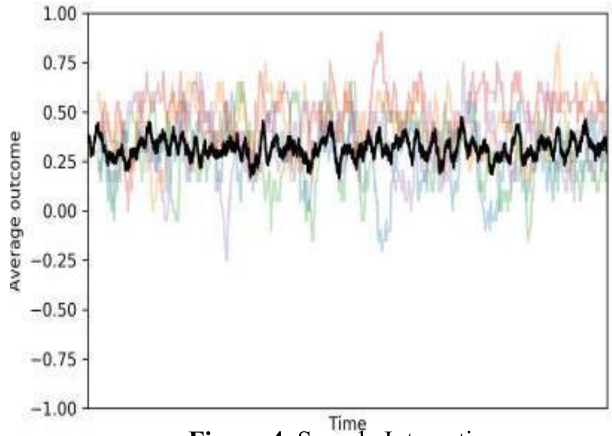

Figure 4. Sample Interaction

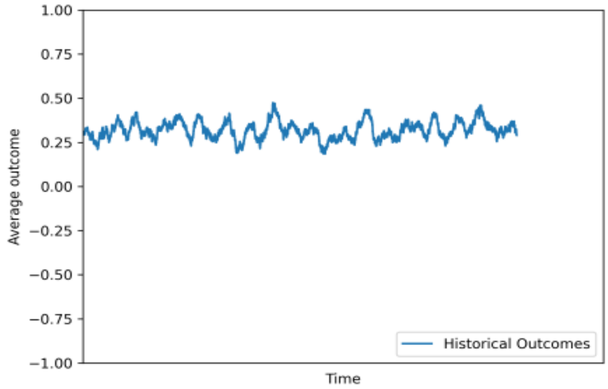

Figure 5. Train sub-rolling average

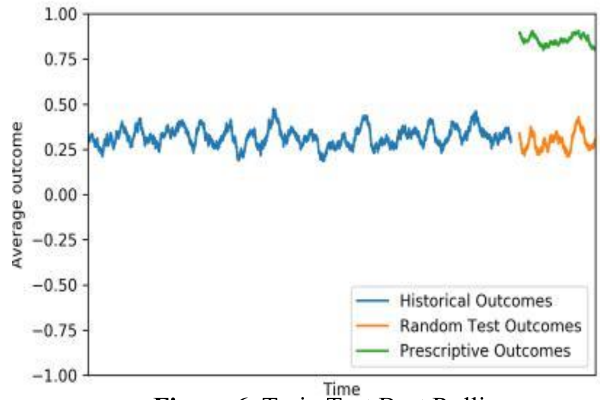

Figure 6. Train Test Best Rolling

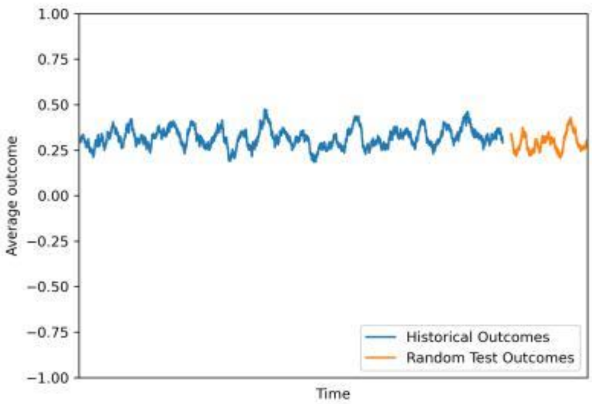

Figure 7. Train Test Best Average

Positive outcomes are displayed on these graphs. According to the graph, the result value in this case is 0.3, which is the prerequisite for the research. The best interaction between the client and the resource occurs with positive output. The results of our discussion should show a positive trend for the graph. On the time base axis, seconds, minutes, and hours will be displayed.

6. Discussion

The focus of this research was on how machine learning can be used to optimize resource allocation in the IoT healthcare environment. Using a simulation-based dataset, client and resource interactions were analyzed, considering client conditions, resource skills, and familiarity between them. Results suggested that supervised models such as multiclass neural networks and XGBoost are effective in providing accurate predictions of client satisfaction. The main observation was that ML-based predictions yield more positive outcomes than random resource allocation. This strategy can be impactful in critical environments such as hospitals, where patient experience and timely response are critical. A real-time implementation of this model was demonstrated via Azure IoT Hub, wherein predictions were made from ML models by collecting data from edge devices. Research also showed that compared to traditional optimization techniques, ML models provide adaptive and scalable solutions, especially when data is continuously changing. However, for real-world deployment, other parameters such as latency, security, and explainability need to be addressed.

7. Conclusion

This research concluded that the use of machine learning is highly useful for optimizing resource allocation in IoT healthcare environments. ML and DL models, such as neural networks and XGBoost, accurately predict client-resource interactions, thereby improving both resource selection and client satisfaction. Using Azure IoT Hub and ML Studio, an integrated real-time system was developed that selects the best resource through predictive analytics. The research pipeline includes data generation, model training, and evaluation, making the system adaptive and scalable. Future work requires model explainability, latency optimization, and validation on real-world datasets.

8. References

[1] F. Hussain, Internet of Things; Building Blocks and Business Models. Springer, 2017[2] Fatima Hussain, Syed Ali Hassan, Rasheed Hussain, Ekram Hossain "Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges," *arXiv:1907.*08965v1 [cs.NI] 21 Jul 2019.

[2] D. B. J. Sen, "Internet of Things - Applications and Challenges in Technology and Standardization," IEEE Transactions in Wireless Personal Communication, May 2011.

[3] D. Wang, D. Chen, B. Song, N. Guizani, X. Yu, X. Du "From IoT to 5G I-IoT: The next generation IoT-based intelligent algorithms and 5G technologies", IEEE Commun. Mag. 56 (10) (2018) 114–120.

[4] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A Brief Survey of Machine Learning Methods and their Sensor and IoT Applications," IEEE Conference on Information, Intelligence, Systems and Applications, March 2018.

[5] Q. et al., "Real-time multi-application network traffic identification based on machine learning," in Springer, vol. 92, 2015, pp. 473–480.

[6} Dan Wanga, Hao Qin, Bin Song, Ke Xu, Xiaojiang Du, Mohsen Guizani "Joint resource allocation and power control for D2D communication with deep reinforcement learning in MCC" Elsevier.

[7] Mark Eisen_ Clark Zhang_ Luiz F. O.Chamon_ Daniel D. Leey Alejandro Ribeiro_ "Learning Optimal Resource Allocations in Wireless Systems" 1053-587X (c) 2018 IEEE.

[8] T. et al., "Machine learning-based classification of encrypted internet traffic," in Springer, 2017, p. 578592.

[9] Xiaolin Ma, Shiwen Ning, Xinhua Liu, Hailan Kuang, Youying Hong "Cooperative Spectrum Sensing using Extreme Learning Machine for Cognitive Radio Networks with Multiple Primary Users" IEEE 978-1-5386-4509-3/18/$31.00 ©2018

[10] P. Lynggaard, "Using machine learning for adaptive interference suppression in wireless sensor networks," IEEE Sensors Journal, vol. 18, no. 21, pp. 8820–8826, Nov 2018

[11] Rathinaraja Jeyaraj, Anandkumar Balasubramaniam, Ajay Kumar M.A., Nadra Guizani, Anand Paul Authors Info & Claims "Resource Management in Cloud and Cloud-influenced Technologies for Internet of Things Applications" ACM Computing Surveys, Volume 55, Issue 12 Article No.: 242, Pages 1 – 37 March 2023

[12] Muhammad Fahimullah , Shohreh Ahvar, Mihir Agarwal & Maria Trocan "Machine learning-based solutions for resource management in fog computing" arXiv:2209.03066 volume 83, pages 23019-23045,(2024) https://doi.org/10.1145/3571729 Aug 2023

[13] Alhusein Almahjoub , Dongyu Qiu "A Grant-Free Random-Access Process for Low-End Distribution System Using Deep Neural Network" Appl. Sci. 2022, 12(14), 7070; https://doi.org/10.3390/app12147070.

[14] https://github.com/madsenmj/ml-iot-resource-allocation/blob/master/README.md