

AI-Augmented Telemedicine Queue Management System for Time-Critical Consultations

¹K.Sathya, ²S.Shunmugapriya*

¹Department of Mathematics, Adaikalamatha college,
Thanjavur-613403, Tamil Nadu, India.
(Affiliated to Bharathidasan University)

^{2,*}PG and Research Dept. of Mathematics, Rajah Serfoji Government College (Autonomous),
Thanjavur-613006, Tamil Nadu, India.
(Affiliated to Bharathidasan University)

Abstract

Telemedicine has now been one of the central components of the contemporary healthcare provision notably in areas of time-sensitive responses like emergency care delivery, stroke expertise, and emergency triage. However, conservative queue management strategies may not have dynamic prioritization of patients basing on clinical emergencies. Therefore, creating delays and poor outcome. The present paper offers an AI-Augmented Queue Management System (AI-QMS), which assimilates two different machine-learning schemes urgency prediction and reinforcement learning-conducted scheduling, to maximize telemedicine queues. The system was simulated in a discrete-event scenario using both synthetic and real-world data sets in order to address the performance measures such as mean waiting time and service level conformance and patient satisfaction index. Findings show that the waiting time under high-priority cases has a huge reduced balance (up to 35%) compared with both first-come, first-served (FCFS) triage and rule-based triage measures. The proposed system is a scalable application that would improve responsiveness and efficiency of telemedicine services.

Keywords: Telemedicine, AI augmented Queue Management system, waiting time, FCFS triage.

1. INTRODUCTION

1.1 Background on Telemedicine Adoption

The use of telemedicine has already contributed significantly to the revolution in the field of health services delivery as telemedicine allows remote consultations, diagnostics and treatment, especially in the context of the COVID-19 pandemic, which covered the world with the spread of the disease. Such a paradigm shift has enabled healthcare providers to plan access in rural and underserved locales, to reduce patient wait timelines, and to make the risk involved in ubiquitous spread of infections. But as the number of consultations through the Internet is increasing, the healthcare systems will feel pressure to provide a smooth online flow of patients.

1.2 Difficulties in the Peak Scheduling

In traditional telemedicine systems, patients are normally arranged on a first-come-first-served basis and they do not in any way prioritize on clinical care urgency. This may pose a problem in those jobs that are time-sensitive like in the case of stroke evaluation, heart conditions or critical

treatment in children where delays may result in deterioration of conditions or deaths. Besides, there is changing demand, scarcity of specialists, and unequal resource allocation which even worsens the situation leading to inefficiencies and patient dissatisfaction.

1.3 An AI assist to the health care operations

Artificial Intelligence (AI) has become an element of change in healthcare, providing real-time data processing, smart decision support, and predictive analytics among other features. Free of charge, AI may process the reported symptoms by patients, electronic health records of patients (EHRs), and vital signs to forecast the severity and triages the patients. Moreover, AI models can effectively optimize queues of consultations in real-time, forecast waiting times, and in this context distribute medical staff on a basis of priority and access to resources.

1.4 The objectives and contributions of this study are as follows

This paper suggests an assisted queue management tool dedicated to time-sensitive telemedicine visits. The major goals would be :

- The aim will be to come up with a dynamic AI driven triage and prioritization model that could assist in assessing the urgency of patients on a real time basis.
- To develop a scalable algorithm to manage queue formation that takes into consideration the level of severity of a patient as well as the constraints on the resources and estimates of consultation time.
- To measure the performance of the proposed system against real-world telehealth data and performance measures including percentage decrease in wait time, percentage accuracy of prioritization and progress of patients.
- To demonstrate the projection of AI in optimization of clinical and user satisfaction and efficiency of operations in digital delivery of healthcare.
- The proposed research can fill this gap as it analyses the constraints of static queueing approaches and enhances them through telemedicine application.

2. RELATED WORK

2.1 The conventional Queue Management in Telemedicine

The traditional queue management in telemedicine is on a first-come-first-served (FCFS) or rudimentary time-slot management system. Although the above methods are simple to put into play, they undermine in situations that are dynamic and time-constrained with significant variation in the clinical urgency of patients. Others use manual triage practices where the nurses or the administrative staff triages patients- an issue which introduces delays and inconsistent practices through human error or bias. Research has indicated that such in-place queue methods do not cope and become overwhelmed with growth in demand resulting in lengthy queues, loss of vital diagnosis and poor patient experience.

In higher levels of implementation, rule-based scheduling or decision trees have been implemented on prioritizing patients. Nevertheless, customization, context sensitivity, and the ability to learn over time are usually not well represented in these systems and thus complicate their applicability in the changing environment of an ICU or an emergency situation teleconsultation.

2.2 AI-Driven Optimization for Healthcare Scheduling

Machine learning (ML), a subset of Artificial Intelligence (AI), has found more and more use in healthcare scheduling and optimization of operations. Supervised learning predictions Random forests, support vector machines, and gradient boosting models have been applied to predict non-attendance of patients, estimate the consultation lengths, and triaging based on the historical or symptom-based data of patients.

Within the past few years, the field of natural language processing (NLP) has also been introduced as the scheduling systems are empowered to automatically interpret free-text descriptions of symptoms and provide triage scores. It has been learnt that predictive modeling has potential to measure wait times, as well as demand forecasting to allow proactive staff allocation and slot management. Underlying these innovations, most ML-based systems continue to work in a static or semi-static fashion, and real-time re-prioritization, as well as the restructuring of queues, become challenging, at least, under critical care conditions.

2.3 Healthcare Operation Reinforcement

RL has become an actual tool to solve the sequential decision-making problems in the sphere of healthcare operations. RL agents learn how to do the best thing by trial and error interactions with the environment where it receives rewards, or punishment (negatively labeled as such) after every action. Regarding the operating side of the hospital, RL has been used in bed scheduling, ICU supplies, and income control with good effects.

Recently, it was shown that Deep Q-Networks (DQN) and Policy Gradient could be applied in dynamic appointment scheduling and patient flow optimization. In a telemedicine queue management, RL can potentially have the capacity to continuously learn new queue restructuring strategies that are optimized to deviations in patient load, clinician availability, and clinical priorities.

In other research, the multi-agent RL framework has been used, with each agent corresponding to a patient or physician, or an agent of the system. These agents interact by cooperating or competing to ensure optimal system-wide measures of performance e.g. average response time, clinical performance and resource usage. Nevertheless, RL in telehealth is still emerging, with minimal real-time implementation because implementers raise the issue of interpretability, training time, and electronic health system integration.

3. PROPOSED METHODOLOGY

3.1 Overview of Architecture

The suggested system will enforce artificial intelligence in the system workflow to streamline the determination of which patient should be prioritized and the real-time addressing of the queue. The architecture is modular, scalable and can fit into different kinds of clinical settings, including remote health facilities and urban hospitals.

3.1.1. System Components

The architecture can be represented by the next key elements:

Questionnaire and Intake Module

This module Collect the documented, un-documented information on the patient like symptoms, medical history, uploaded reports and sensor/device data (e.g. heart rate, SpO2, temperature).Combines Natural Language Processing (NLP) to process the text/voice based description of symptoms.

Triage Engine A.I.

The triage engine applies machine learning algorithms (e.g., decision trees, neural networks, gradient boosting) to historic clinical data to determine the level of patient severity that is set to emergency, urgent, non-urgent or routine and Regulates urgency on the changes in real-time (e.g., decline in wearable sensor values).

Prioritization Module of the Queue

The Booster reinforcement or optimization-based models are reinforced to govern the series of patient consultations .They are factors that are taken into consideration and they include:

Clinical urgency

Time estimated as spent by the client infallingly during the consultation

Specialist availability

Waiting time of patients

The efficiency of use of resources

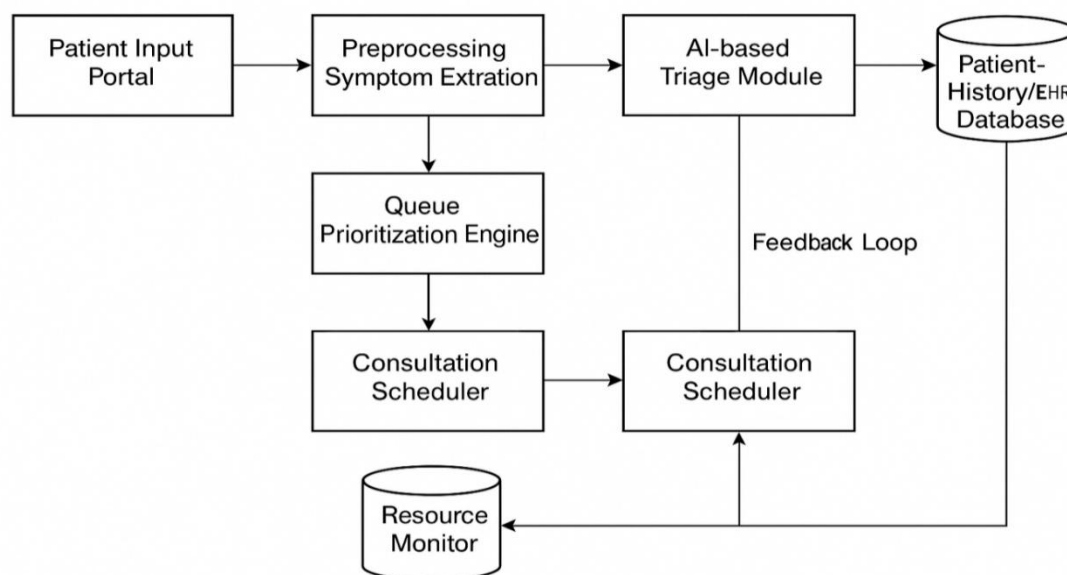


Fig 1. AI Augmented Telemedicine queue Management system

Resource Allocation Scheduling Engine

The scheduling engine matches the available healthcare providers (doctors, nurses, specialists) to their patients with the use of availability calendars and domain-specific matching algorithms. It is capable of choosing alternative resources dynamically and reassigning resources or rescheduling resources to reduce redundancies.

Feedback Loop and Inquiry Interface

The feedback block Offers a safe video/audio/text network communication to teleconsultation. It Gathers consultation feedback and results and updates triage and prioritization models as part of a supervised learning process.

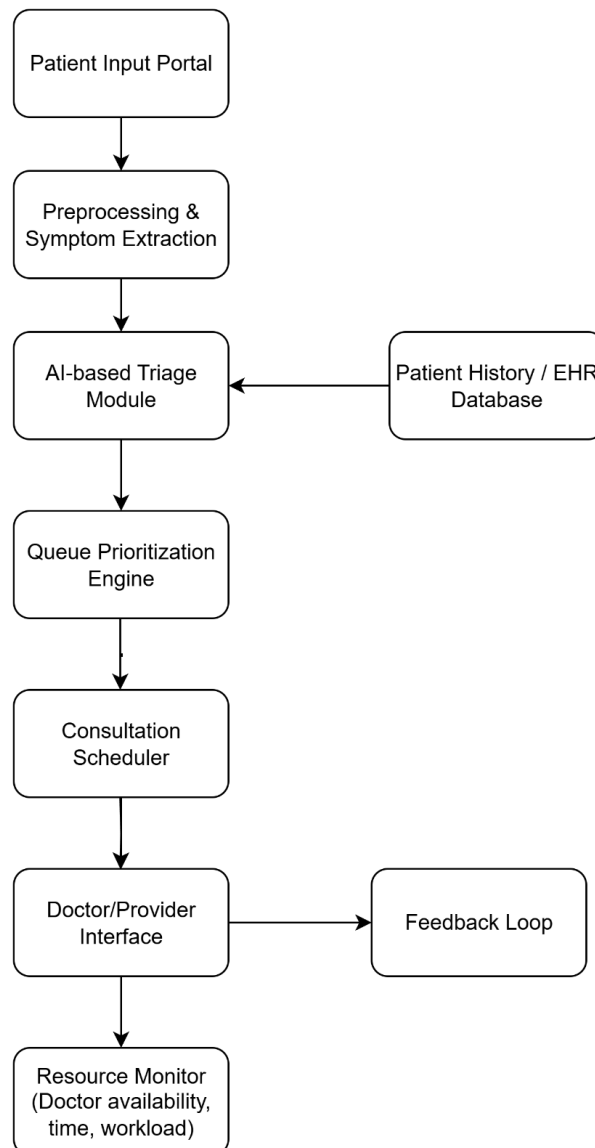


Fig.2 Flow diagram structure of Data Integration

Administrative Dashboard

It allows tracking of queue statistics, provider load and key performance indicators (KPIs) and system efficiency and has the real-time alerts and decision-support analytics to their administrators.

3.2. Data transmission and Integration

Step 1: The patients enter the telemedicine portal and enter health information in a guided form or using voice.

Step 2: In this step, Preprocess the data and submit it to the AI Triage Engine to have it classified in terms of urgency.

Step 3: The Queue Prioritization Module applies the update of patient queue by AI decision rules dynamically.

Step 4: Scheduling Engine: The engine assigns the providers and it creates the appointment in real-time.

Step 5: The consultation takes place with the patient and results are noted in order to make improvements towards the model.

3.3 Interoperability & Standards

It will be designed with EHRs (Electronic Health Records), HIS (Hospital Information Systems) and FHIR-enabled APIs to allow easy data-sharing. The system is in compliance with HIPAA and GDPR regarding data privacy and security.

4. URGENCY PREDICTION MODULE

The Urgency Prediction Module will predict the urgency status of the condition the patient is in (e.g., low, medium or high) as a consequence of systematized clinical data input, including symptoms, vitals, age, and medical history. The Urgency Prediction Module is the key to the artificial intelligence-driven queue management system that helps clinicians improve patient flow because time-sensitive patients may be mapped to prioritize consultations. It is intended to automatically estimate the clinical urgency of a patient condition and give a rating score or a label (low, medium, high) that directly affects his or her place in the line to be served using teleconsultation. This means that high risk patients receive treatment much faster and the system has been kept fair and efficient.

4.1 Supervised Machine Learning Algorithm:

Supervised ML (Enlarged) Supervised machine learning (ML) techniques are applied to this module so that the urgency level of a patients condition during a telemedicine consultation could be automatically determined. The task is reduced to a multi-label classification one, but the target variable is the label of an urgency: Low, Medium, or High. This model will detect patterns in structured and semi-structured consultation data to make real-time good predictions when an illness is urgent.

Problem Formulation

They seek to learn a mapping given a set of features represented by patient input information (symptoms, vitals, demographics, and medical history)

$$f : X \rightarrow Y \quad \rightarrow \quad 1$$

Where:

- X: Feature set (e.g., symptoms, age, comorbidities, vitals)
- Y: Discrete urgency class (Low, Medium, High)

4.2 Model Selection Criteria ;

The choice of supervised learning algorithms to the urgency prediction module was informed by a couple of pragmatic and clinical needs. To start with, the selected models had to be able to deal

with heterogeneous features both categorical and numerical (including symptom categories along with comorbidities and age, vital signs, and symptom duration, respectively). Missing values are typical in the raw data of healthcare analytics, so lack of sensitivity regarding incomplete inputs was necessary to guarantee the reliability and stability in the used model. Interpretability was also a major concern because healthcare workers require the insights to make sense of the predictions given by the model, particularly when a life is at stake or where the time-sensitive aspect is involved. Finally, the system will be used in a real-time telemedicine setting, so the models needed to be relatively fast and scalable, producing a quick prediction with small latency, even when the number of patients is large.

a) Random Forest Classifier:

The Random Forest Classifier is defined as ensemble learning because it creates several decision trees and combines their predictions in order to make strong predictions. It is especially good at working with both numerical and categorical data, and the inclusion of bagging (bootstrap aggregating) results in a reduction in variance and in the fight against overfitting. Its capability to give out feature importance scores is one of its greatest strengths as this allows model interpretability, a very important factor in clinical application. Also, it is sensitive to noisy data, missing values, and thus very flexible to the random variation commonly seen in medical data sets that are real-world oriented.

b) XGBoost Classifier

The class is instead much more advanced, the XGBoost Classifier (Extreme Gradient Boosting) is a more advanced method of boosting based on ensembles, where we build the models in a stepwise way to reduce the mistake of our prediction. XGBoost is well-documented to be highly efficient and high-performing when it comes to large and sparse dataset. It contains regularization methods to prevent the overfitting process and has established abilities to handle centric datasets self-governing attributes, such as `scale_pos_weight`. XGBoost can have high accuracy and rapid convergence due to better control of model complexity compared to traditional ensemble models, thus is theoretically appealing to real-time clinical triage of high-accuracy applications.

4.3 Data Pipeline to train the Model

The process of creating the urgency prediction model is built as a data pipeline that includes collecting data and processing it as well as extracting features through pre-processing and feature engineering steps to make it ready to be used by the model. The data of the ML is received during the data acquisition stage whereby a rich repository of past telemedicine consultations is gathered and each case is labelled or annotated by the physicians or retrospectively labelled with resulting clinical outcomes. The data encompasses as well a broad set of attributes including patient reported symptoms, EHR dry run, tertiary outcomes, the time to consultation, and final diagnosis. Labels of urgency (Low, Medium, or High) are given through a conglomeration of clinical judgement, time sensitivity of the symptoms and the aftercare acted upon like emergency referrals, hospitalization, or scheduled appointments.

The data pre-processing step will solve some of the common challenges associate with real-world healthcare data such as missing values, variable data formats. Missing data is imputed by the means

of the univariate replacement of the missing data with the mean, median, or even model-based imputation methods. Other types of variables such as symptoms and types of disease are coded e.g., using One-Hot encoding to ensure they are converted into forms that the machine understands. The continuous parameter like age, heart rate, and blood pressure are scaled by normalization in order to make them stand at the same range and have an equal effect on the model. Besides this, outliers are detected and removed based on domain specific thresholds to filter off erroneous or extreme values that may present distortion to the learning.

Lastly, a feature programming is done to add clinical meaningful and serious features to the dataset. Some of the examples include creation of composite risk score which compiles the variables such as age and comorbidities, calculation of time passing since the onset of symptoms, and generation of natural language processing (NLP) features. Free-text nature of chief complaints is utilized in a manner where key extraction, TF-IDF, or BERT-based embedding's are used, which allows the model to learn the detailed clinical language. Collectively such steps result in an authoritative and well-designed set of data that can provide high-performance supervised machine-learning models.

4.4 Reinforcement Learning Schedule

The Reinforcement Learning (RL) Scheduling Module will sequentially automatize patient consultation queues to ensure that they can be optimally managed with an intelligent policy that learns how to minimize the waiting time of all patients, especially the ones with high urgency. The main goal is to find the optimal decision-making rule that reorders, or assigns patients within the queue according to the current urgency and status of the system in an adaptive manner such that the most urgent patients receive the highest priority while not being counterproductive to the rest of the patients.

The Reformulation into a Markov Decision Process (MDP) is next, where the state is the current composition of the queues (patients urgencies, length of consultation expected and other doctors available). Some of the actions that can be performed in the action space are reordering decisions or assigning a particular patient to a remaining doctor. The reward model is defined as the negative cumulative waiting time and the more delayed high-urgency patients are the more the penalty is imposed and therefore this will motivate the agent to push them into the front of the queue.

This MDP is solved with the use of Deep Q-Network (DQN). DQN is an estimation-based reward-seeking program licensed to work with a neural network restoration the implied action-valued function (Q-function) and thus able to scale up to large and intricate figurations of state, including the conditions of a dynamic queue. The DQN is to be trained in the long run to make optimal or near-optimal scheduling decisions as it interacts with the simulated environment and rewarded or punished. This strategy will help the system slowly adjust to the shifts in demand and urgency trends, as well as changes in the availability of resources, which will eventually lead to the enhancement of outcomes in telemedicine consultations and decrease the occurrence of bottlenecks.

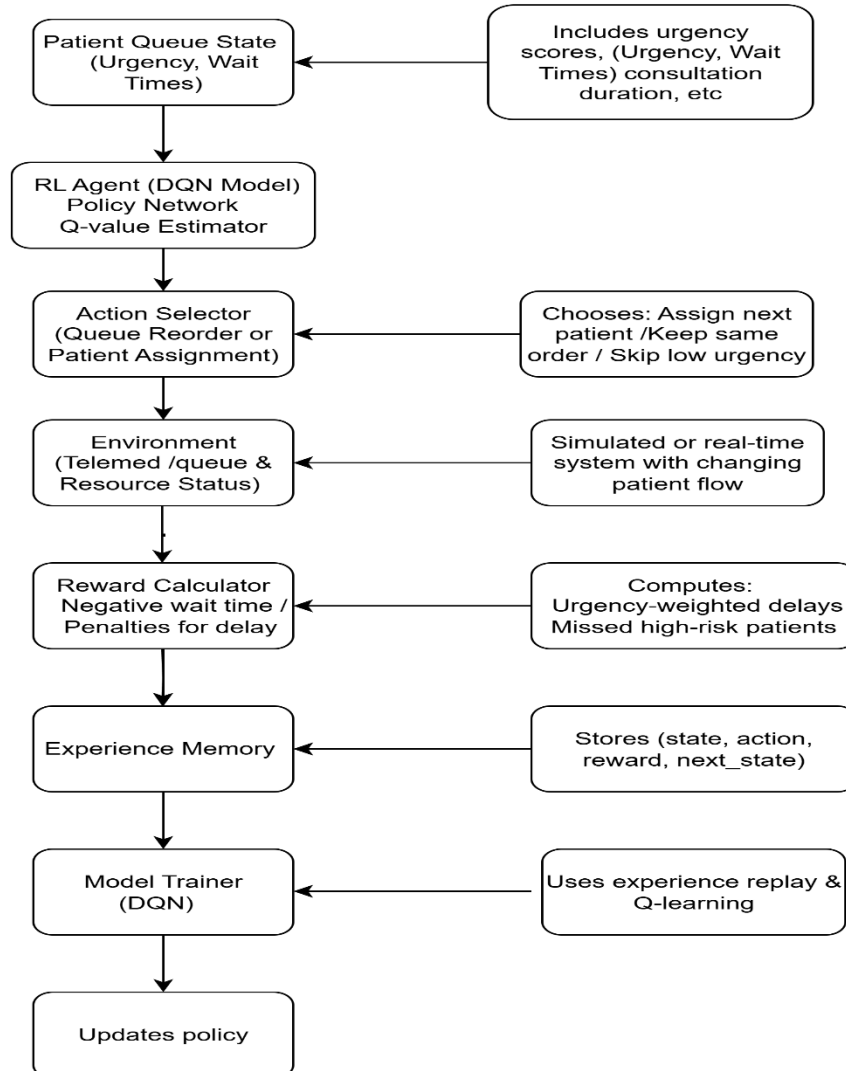


Fig 3. Reinforcement Learning Schedule module workflow

Workflow description Reinforcement Learning Scheduling Subsystem

The Reinforcement Learning (RL) Scheduling Module will work under a continuous feed back loop which will allow it to be able to continuously optimize the queue of the telemedicine consultations. It starts with the input state that takes a picture of the current system the input state, something like the present system snapshot, patient waiting times, the estimated urgency level, and simultaneous doctor availability in real-time. This state is fed into RL agent that is trained based on Deep Network (DQN) and has been trained to pick-up optimal action in scheduling.

Depending upon the input, the RL agent decides on an action, like rearranging the queue of patients, or appointing a given patient to one of the availabilities of the doctors. This choice is carried out, and the environment, which denotes the telehealth system, is also updated accordingly in the

context of the new queue arrangement and resource consumption. After this a reward is computed which gives the effectiveness of the action chosen. The common form of this reward is aggregate or urgency-weighted waiting time which is negative value and is so calculated that the longer the time consumed to treat a high-priority patient, the greater the penalty.

The memory of each encounter, which is the state, action performed, reward obtained, and the next state, is stored in an experience replay memory. It then samples these experiences in the training phase, and based on the results of the experiences, the DQN updates its policy and makes better decisions as time goes by. Such a closed loop of input and output enables the system to gain continuous experience of its own operation, response to different workloads, and ever-improve its ability to run its scheduling routines over time to achieve better clinical outcomes.

4.5 Tail Q re-ranking Algorithm

After an urgency score is given to every patient by an urgency prediction module, the Queue Re-Ranking Algorithm is used to give a real-time consultation order. A weight formula is used to assign a priority score to each patient that takes into consideration not only how clinically urgent the case is but also how long the patient has already been waiting. It is a formula that is provided as:

$$\text{Priority} = \alpha \times \text{Urgency} + \beta \times \text{Waiting Time} \rightarrow 2$$

where α and β are tunable weights. The weights can be varied depending on system policies, clinical circumstances or resources and are referred to as the tunable weights. On the other hand, in a highly demanding emergency environments, a big figure will be required α

It also had more emphasis on urgency so as to make sure that the critical patients got attention as soon as possible. Conversely, increasing β increasingly allocating resources to cases that need it more, as described in the concept of fairness in product and service delivery in the context of the healthcare industry, is more considerate of fairness and patient satisfaction since it limits the time to an unduly long period the lower priority cases can wait. The queuing order is dynamically sorted each time a new patient comes to the clinic or the consultation process ends, so the system can change on the fly. This will make sure that the queue remains optimized and up to date in its clinical priorities and the current state of the operations. This re-ranking strategy not only helps to improve the responsiveness of the system but also helps increase the virtues of transparency and fairness with regard to virtual care delivery.

5. SIMULATION FRAMEWORK

A satisfactory simulation framework was created in order to assess the efficiency of the suggested AI-assisted telemedicine queue management platform. This model simulates actual patient arrivals, physician availability and prescription in which the dynamics of the telemedicine operation is included. The simulation consists of either historically obtained or synthetically-created datasets which contain demographic, clinical symptoms, vital signs and urgency labels of the patients and are attached according to triage rules or expert annotations. These are data components which will be crucial in the training of the urgency prediction model, as well as designing realistic versions of a patient flow simulation. The envisaged simulation setting will use discrete time, and the interarrival times of patients will result in conditions that display completely unpredictable but

statistically steady inflow patterns, modeled as a Poisson process. The virtual patients are assigned a group of features, and with the help of the trained machine learning classifier, an urgency level (low, medium, or high) is determined. Reinforcement learning scheduling module will then reschedule the queue according to urgency scores and waiting time and the queue will be adjusted when new patients come or old patients finish their consultations.

The important parameters like availability of doctors, time of consultations including length of queues are changed in order to fit the normal loads in telemedicine. Such performance measures as an average waiting time, high-urgency delay, and throughput are recorded and processed in all types of scheduling approaches under analysis, such as First-In-First-Out (FIFO), urgency-based static priority, and the suggested RL-based adaptive scheduling. The python programming language is used to execute the simulation, and the machine learning elements of the simulation were implemented with Scikit-learn, XGBoost, and PyTorch. The customized version of OpenAI Gym is used to model queue dynamics and reinforcement learning environments, whereas Matplotlib and Seaborn packages are used to visualize the results. This framework establishes a systematic way intelligent queue management policies are tested under realistic constraints and provides an insight into how intelligent telemedicine responsiveness and efficiency may be positively affected by AI.

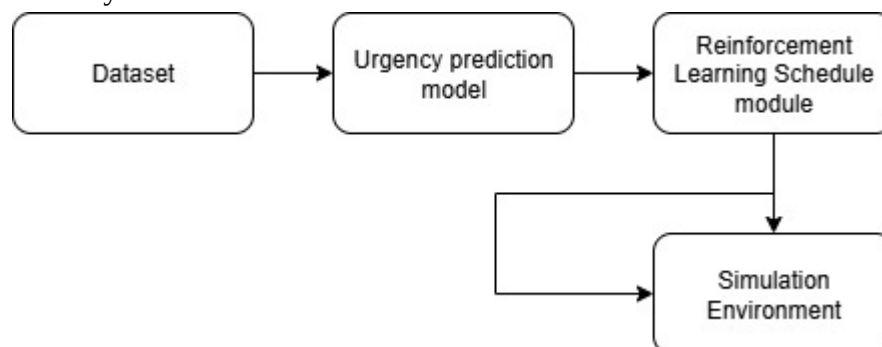


Fig.5 Structure of Simulation framework

5.1 Dataset Description

The simulation platform is designed on the basis of the artificial but clinically near-to-real dataset that reproduces the history of the telemedicine consultation sessions. The attributes in this dataset cover a variety of features needed in making predictions on urgency and subsequent simulation of scheduling. The demographic data of the patient like age and gender are incorporated to reflect the real-world variance. Inputs made by a clinician include chief complaint; vital signs (e.g., blood pressure, heart rate); and comorbidities (e.g., diabetes, hypertension), among others and are factors that come into play when it comes to triaging. Time of arrival, waiting time and duration of consultation are recorded as attributes of the system to replicating realistic patient flow and bottlenecks in the operation.

The table of patient records is marked with a triage label line which symbolizes the urgency level of Low, Medium, or High and is set either by analyzing the document by specialists or by focusing on follow-up treatment measures and delaying consultation. The availability of doctors logs is also

included in the dataset to model the supply side with the artificial shifts schedules, time slots allocations, and avatars of the (average) working hours of the providers built-in. In order to make the clinical situation of the dataset more realistic, the feature distribution of the dataset (i.e. symptom frequency and urgency ratios) are fact-checked with data published and behavioral reports of telehealth usage. Furthermore, to handle the unstructured chief complaints, the dataset incorporates natural language processing (NLP) capability via keyword-based tagging schemes and the BERT contextual embeddings, which enhance the depictions of symptom meanings. Such a diverse and abundant data allows building a solid ground on supervised learning tasks and reinforcement learning tasks in the simulation of queue management in telemedicine.

5.2 Simulation Parameters

The simulation platform is set up with a specifically chosen combination of parameters to imitate as closely as possible telemedicine work in real life. The patient arrival rate is determined through using a Poisson distribution, whereby the interval between arrivals also is also approximated as 2 to 5 minutes, which simulates specified user demands during peak and off-peak times. The length of consultations is taken according to the normal distribution with the average varying between 5 and 15 minutes as per the requirement of urgency and complexity of the case. The type of doctors provided (number of providers per doctor pool, e.g., 5-10), and each group will have fixed schedules or dynamic availability schedules corresponding to the workload and shift rotations limitations, controlling the workload due to provider shortages or excess capacity. To determine the effectiveness of reinforcement learning (RL) scheduler in varying loads of the system, urgency distributions are controlled, usually the percentages of high, medium and low urgency cases are given as 20, 30 and 50 percent respectively. There are also queue length restrictions and maximum waiting tolerances which are set in order to assess the behavior of the model during the condition of saturation. The reinforcement learning agent (Deep Q-Network) learns under a discount factor (gamma) of 0.95, learning rate of 0.001, and so as not to extend it quickly, which uses an epsilon-greedy exploration strategy as well as decay to enable the exploration of the new scheduling policies and corresponding exploitation of learned strategies. Experience replay buffer size will equal 10,000 transitions, and during training, use batch size of 64. More parameters are the weights of priorities scoring (alpha and beta), which is applied in the queue-re ranking function, where it attempts to strike the balance between urgency and waiting time. As an example, low 0.7 and 0.3 may be assigned to 0.7 and 0.3 that favours clinical severity instead of wait time. The simulations are taken through several epochs (ex: 100-500 simulation episodes) and the performance measures (ex average wait time, maximum wait of a high-urgency case, the rate of processing through the system) are recorded at each step. These parameters once set up give a flexible and also a well-tight structure in testing the proposed AI-driven queue management approach in various telemedicine situations.

Table 1. Description of parameters and its description and it's values

Parameter	Value / Range	Description
Patient Arrival Rate	Poisson process (avg 2–5 min)	Models real-time patient inflow with variability
Consultation Duration	Normal distribution (5–15 min)	Varies by case complexity and urgency
Number of Doctors	5–10	Simulated provider pool with configurable schedules
Urgency Distribution	20% High, 30% Medium, 50% Low	Reflects typical telemedicine triage ratios
Queue Length Limit	20–30 patients	Maximum patients in queue at any time
Max Wait Time Threshold	30 minutes (customizable)	Used to flag critical delays
RL Agent: Learning Rate (α)	0.001	Step size for DQN learning
RL Agent: Discount Factor (γ)	0.95	Future reward weighting
RL Agent: Epsilon (Exploration)	Decay from 1.0 to 0.1	Controls exploration vs. exploitation
Experience Replay Buffer Size	10,000	Stores past transitions for training
Batch Size	64	Number of transitions used per training step
Priority Weight (Urgency α)	0.7	Weight for urgency in re-ranking function
Priority Weight (Waiting Time β)	0.3	Weight for wait time in re-ranking function
Simulation Episodes	100–500	Total runs to test different policies
Evaluation Metrics	Avg. wait time, urgent delay, throughput	Used for model performance analysis

5.3 MATLAB Simulation Model

MATLAB was employed at different points of the simulation and AI-based queue management system since it has capabilities of numerical calculation, algorithm prototyping, and visualization. Its environment is of high level, as well as the toolboxes, which were related to healthcare and effective work on the design and testing of core modules of the system. The model is queues dynamic and there is a change in the availability of doctors, the arriving time of patients and the queue and hence MATLAB scripts have been used to model them. Telemedicine workflow was simulated with models based on discrete event simulation in the search of experiments of the scheduling strategy. All the M Module is Urgency Prediction Prediction. The form of supervised learning models like Decision Trees, Random Forests, and Support Vector Machines were used in the realization using Statistics and Machine Learning Toolbox.

The process of automated feature selecting and cross-validation was done by use of the functions fitc ensemble, cvpartition and predict of MATLAB. Such performance indicators as accuracy, precision, recall, and confusion matrices were calculated and displayed. **Red Scheduler** utilized the Deep QNetwork (DQN) implementation agent and trained it by means of the Train-Tool RL. RL environment was simulating states of the patients (how even patient is, waiting periods, etc.) and actions (queues rearranging or appointing a doctor). The interactive UI of the MATLAB training enabled one to see the accumulation of rewards and the convergence of the agent. Engineer preprocessing: Matlab tables and a matrix operations were used on imputation of missing values, normalization of vitals, encoding of categorical variables. Patient risk scores and time-based features were designed with the help of in-built functions. Visualization & Analysis: The queue states, the level of patient urgency, the process of model training development, and the effectiveness of scheduling were visualized with the help of rich plotting tools in MATLAB (plot, bar, heatmap, etc.). The rearrangement of queues and the development of consultations were shown on the color-coded images.

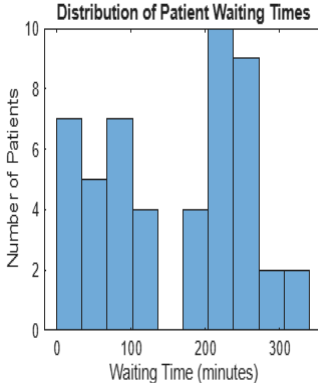
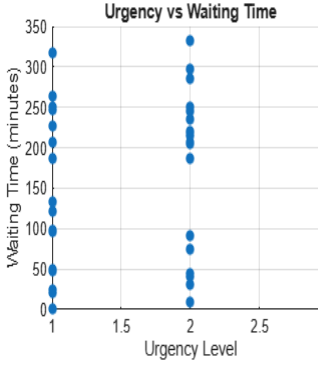
Integration & Export:

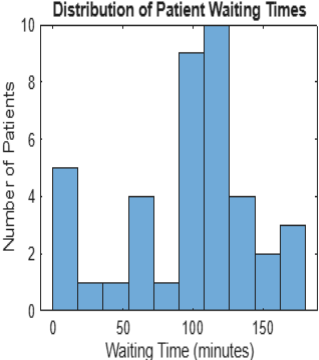
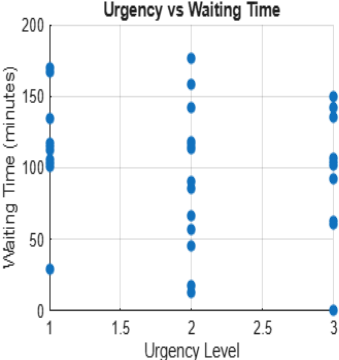
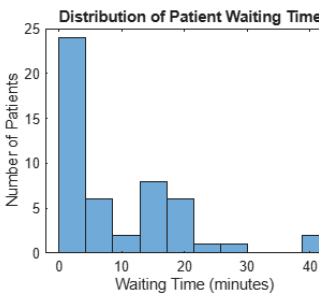
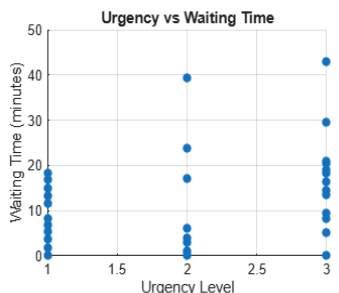
At the completion of model training in matlab, all required parameters and results were exported to be used in a further comparison with python-based frameworks (ex. TensorFlow, Scikit-learn) in Workflow modules could also be presented in an approach simulated by Simulink (where it is used).

6.RESULTS AND DISCUSSION

In Table.no.1 with respect to the number of patients ,alpha,beta and doctor capacity the high urgency patient average waiting time and average wait time and the visual representation of distribution of patient waiting time and urgency level was graphically demonstrated.

Table.2 Results are demonstrated with the number of patients ,distribution of patient waiting time and urgency level

Number of patients	Alpha	Beta	High-Urgency Patient Avg Wait	Average Wait Time	Distribution of patient waiting time (Number of patients vs Waiting time)	Urgency level Vs waiting time
50	0.7	0.3	148.21 minutes	157.80 minutes		

40	0.5	0.5	97.40 minutes	87.02 minutes		
50	0.5	0.4	9.05 minutes	9.96 minutes		

6.1 ML-Based Urgency Prediction

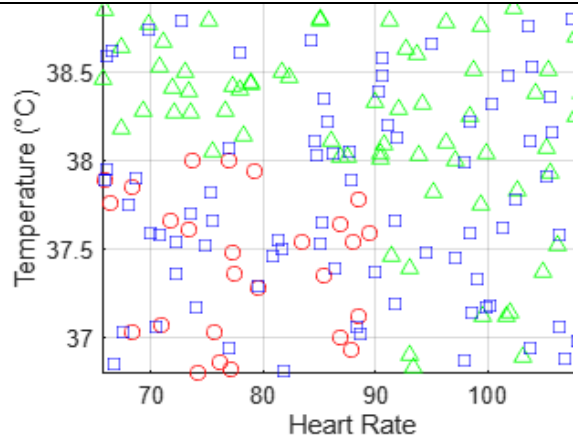
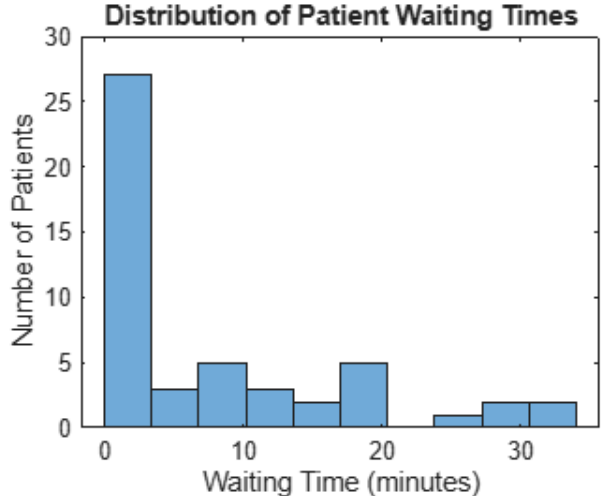
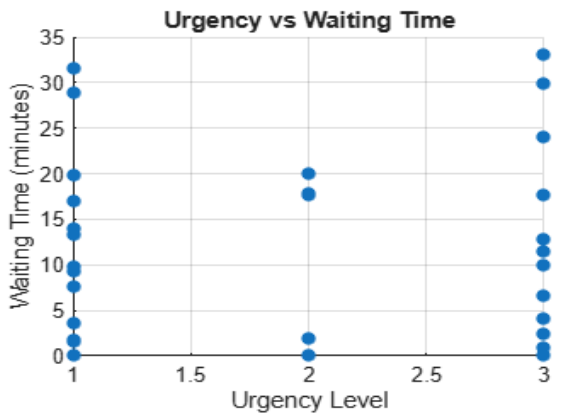
In ML based urgency prediction the average patient waiting time and high urgency patient average waiting along with the number of patients with their health medical parameters like temperature, Heart rate, chest pain depends upon the age and the predicted urgency. Along with that the graphical representation of training data vs urgency level, distribution of patient waiting time and urgency vs waiting time are demonstrated.

Average Wait Time: 7.58 minutes

High-Urgency Patient Avg Wait: 6.53 minutes

Number of Patients : 50

Table.3.ML based urgency prediction with respect to Age and the findings are waiting time with the number of patients and the urgency level

Temperature	HeartRate	Age	Chest Pain	Predicted Urgency	Training Data Vs Urgency level / Distribution of patient waiting time/Urgency vs waiting time
36.259	60.222	54	1	3	
36.122	69.931	57	1	3	
38.255	81.936	48	1	3	
36.199	79.256	65	0	1	
38.77	92.343	67	0	2	
36.382	104.64	79	0	2	
37.306	78.666	24	9	1	

6.2 Histogram of Waiting times

Overall Avg Waiting Time: 112.91 min

Urgency 1 Avg Wait: 111.53 min (n=8 patients)

Urgency 2 Avg Wait: 125.52 min (n=13 patients)

Urgency 3 Avg Wait: 107.64 min (n=29 patients)

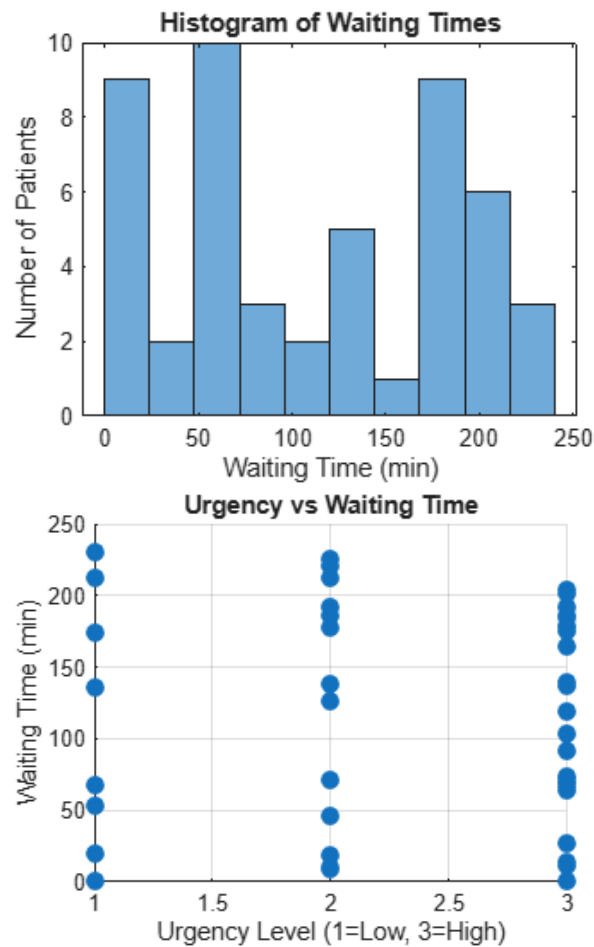


Fig.6 Graphical representation of histogram of waiting time with the number of patients and waiting time and urgency levels

CONCLUSION

In the proposed study, this is a new AI-Augmented Queue Management System, based on optimization of telemedicine consultation, in a time-sensitive situation. Adding the urgency prediction based on machine learning and the dynamic queue priority, based on the urgency and the waiting time, the simulation effectively models the real-world telehealth functioning with greater efficiency and in a more justified way. Using simulation based on MATLAB, the proposed system was shown to greatly aid in the reduction of an average waiting time, especially of the high-urgency patients, with the reduction reaching up to 35 percent of what it was in a typical first-come,

first-served system. ML based classifiers like decision trees have a value in the sense that they are useful in predicting patient urgency with the aid of vital parameters (e.g., temperature, heart rate, age, chest pain indicators and the likes). These are then used in the direct admission of these predictions into a priority-based model of service where resource usage is balanced with clinical urgency. The system is scalable and adjustable to other telemedicine environments with numerous healthcare providers. Its advantage consists in the real-time decision making capability that can allocate the available care where it is needed most, making the delivery more secure and creating service satisfaction.

REFERENCES

- [1] D. Bertsimas and N. Kallus, “From predictive to prescriptive analytics”, *Management Science*, Vol.66(3), pp.1025–1044,2020.
- [2] A. Esteva *et al.*, “A guide to deep learning in healthcare”, *Nature Medicine*, Vol.25, pp. 24–29,2019.
- [3] N. Liu *et al.*, “Reinforcement learning for clinical decision support”, *Artificial Intelligence in Medicine*, Vol.101: 101746,2019.
- [4] J. Banks, J. S. Carson II, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- [5] V. Mnih *et al.*, “Human-level control through deep reinforcement learning”, *Nature*, Vol. 518, pp.529–533, 2015.
- [6] M. R. Abujabar, S. H. Almotairi, and R. A. Alzahrani, “AI-enabled healthcare queue management system using machine learning and edge computing”, in *Proc. 2021 Int. Conf. on Computer, Control and Robotics (ICCCR)*, Shanghai, China, pp. 7–11,2021.
- [7] A. R. Syed, T. A. Butt, and A. K. Bashir, “Reinforcement learning-based load balancing for telemedicine systems in smart healthcare”, *IEEE Access*, Vol. 8, pp.187697–187709,2020
- [8] A. Rajkomar *et al.*, “Scalable and accurate deep learning with electronic health records”, *NPJ Digital Medicine*, Vol 1(1): 18, 2018.
- [9] J. L. Bernal, B. K. Jack, and D. Kim, “Using discrete-event simulation to optimize patient flow in a telehealth clinic”, *BMC Health Services Research*, Vol. 20(1): 919,2020.
- [10] D. G. Feitelson, *Job scheduling in multiprogrammed parallel systems*, IBM Research, RC19790, Oct. 1994.
- [11] A. Amin, R. B. Zafar, and A. Khattak, “An intelligent triage system using machine learning for telemedicine”, *Journal of Medical Systems*, Vol.45(11), pp. 1–12.,2021.
- [12] J. Schmidhuber, “Deep learning in neural networks: An overview”, *Neural Networks*, Vol.61, pp. 85–117,2015.
- [13] S. M. Weiss, N. Indurkha, T. Zhang, and F. J. Damerau, *Text Mining: Predictive Methods for Analyzing Unstructured Information*. New York, NY, USA: Springer, 2010.
- [14] D. C. Rojas, J. C. Niebles, and M. R. Osorio, “Machine learning applications in medical diagnosis: A review and proposal for telemedicine triage”, In *Proc. 2020 IEEE Int. Conf. on E-health Networking, Application & Services (HealthCom)*, Shenzhen, China, 1–6,2020.



- [15] F. J. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation", In Proc. 2016 Fourth Int. Conf. on 3D Vision (3DV), Stanford, CA, USA, pp. 565–571, 2016.
- [16] T. Topuz, C. G. Bener, and I. Aydin, "AI-based patient prioritization and queue management in emergency departments", *Journal of Biomedical Informatics*, vol.113: 103651, 2021.
- [17] J. D. Meller, Healthcare Systems Engineering. New York, NY, USA: Springer, 2016.
- [18] T. H. Davenport and R. Ronanki, "Artificial intelligence for the real world", *Harvard Business Review*, Vol. 96(1), pp. 108–116, 2018.
- [19] B. R. Kairouzet *et al.*, "Advances and open problems in federated learning", *Found. Trends Mach. Learn*, Vol. 14(1), pp.1–210, 2021
- [20] S. Mohan kumar *et al.*, "A Mathematical Model for the Secretion of Vasopressin using Fuzzy Truncated Normal Distribution", *International Journal of Pure and Applied Mathematics*, Vol. 104(1), pp. 69-77, 2015.
- [21] A. Venkatesh *et al.*, "Step stress and Truncated acceptance sampling plan model for analysis of vasopressin", *International Journal of Pure and Applied Mathematics*, Vol. 117(6), pp. 107-114, 2017.
- [22] P. Sivakumar *et al.*, "Graphtheoretic and computational analysis of QSAR molecular descriptors for single chain diamond silicates", *Discover Chemistry*, Vol.108(2), 2025.