Predicting Patient Waiting Time and Detecting Overload in Emergency Department through Machine Learning

Zihan Qian[1,*], Xuanyi Shen[2], Rongshuo Shang[3]
{qzh@smail.nju.edu.cn[1]}

School of Intelligence Science and Technology, Nanjing University, Suzhou, China[1]
School of Electronic and Information Engineering, Tongji University, Shanghai, China[2]
College of Resources and Environmental Sciences, Nanjing Agricultural University, Nanjing, China[3]
*corresponding author
[†] These authors contributed equally to this work.

Abstract. Accurately predicting patient waiting times and detecting workflow overload in emergency departments are critical challenges that significantly impact patient care and resource management. Despite advancements in patient waiting time prediction, current methodologies often struggle with universal applicability in practical settings and fail to accurately capture extreme values. This study proposes a robust and generalized predictive model tailored to the specific challenges of ED workflows to address these gaps. A dataset containing multiple variables that record the workflow within an emergency department is utilized, and a systematic exploration and comparison of specific machine learning and deep learning models are conducted. Different machine learning models are compared, and a model is developed to enhance the accuracy of prediction. The effectiveness of the model in detecting ED overload is evaluated, and its generalization capability is improved through feature selection and feature classification. The proposed model demonstrates superior accuracy in predicting patient waiting times and exhibits high sensitivity in detecting workflow bottlenecks. The model's ability to operate effectively with fewer variables enhances its generalization ability across different ED facilities. It is found that machine learning models can effectively capture patient waiting peaks, which are critical indicators of ED overload. By detecting these overload conditions, hospitals can optimize resource allocation proactively and address overload issues promptly. In summary, this study provides a generalized model with strong predictive accuracy for patient waiting times and the ability to detect system overloads in healthcare settings, contributing to improved overall ED system performance.

Keywords: Patient Waiting Time, Emergency Department, Regression Model, Machine Learning, Deep Learning, Overload Detection

CONF-CIAP 2025, January 17-23, Eskisehir, Turkey

DOI 10.4108/eai.17-1-2025.2355323

1 Introduction

In recent years, the extraction of real-time information for the allocation of medical resources and the optimization of emergency department workflows has emerged as a prominent trend. Healthcare systems must minimize waiting times and prevent workflow overload to enhance patient satisfaction and overall quality of care. These all can be improved through the accurate patient waiting time prediction and the identification of bottlenecks in patient flow. Past research has shown that longer waiting times lead to more consumption and inefficiency [1]. The study aims to develop an accurate waiting time prediction model and to enhance resource allocation by detecting bottlenecks and dealing with them.

In this paper, an examination of both machine learning (ML) and deep learning (DL) techniques is conducted, with the intention of utilizing them for predicting patient waiting times and identifying workflow overloads. Compared to traditional basic methods, which have been extensively discussed in previous papers, deep learning models are said to offer the advantages of reducing errors and achieving higher accuracy. Therefore, the goal is to enhance the accuracy of predictions through an analysis of these advanced methods. [1, 2].

After the discussion on the paper's purpose, it is noted that there are more aspects that need to be addressed. The study is confronted with several evident challenges, including managing diverse data sources, selecting suitable machine learning models, integrating specific domain knowledge of the healthcare system, and ensuring the scalability and robustness of the proposed solutions. It is hoped that the implementation of the proposed model will enhance patient experience in the emergency department, identify workflow bottlenecks, and contribute to the comprehensive optimization of emergency room management.

2 Literature Review

In the current society, as the healthcare industry digitizing and developing really fast, the complexity and quantity of healthcare data have significantly increased. Electronic medical records (EMRs) provide healthcare organizations with extensive operational data, including processing times, scheduling records, examination types, and various resource characteristics [3]. These data are routinely recorded by hospital information systems (HIS) in a uniform format that complies with major healthcare standards such as Health Level 7 (HL7) [4], Fast Healthcare Interoperability Resources (FHIR) [5], and Digital Imaging and Communications in Medicine (DICOM) [6]. Effectively describing complex healthcare operations requires synthesizing all these data. However, due to the vast amount of data and the difficulty for humans to manually analyze its characteristics, these data are often under-utilized in operational analytics.

Modern healthcare data can be divided into several major domains based on content and application area, such as hospital information, medical imaging, and other sources. These data originate from various hospital departments, including imaging departments [7], biochemistry labs, and surgical suites. Most hospitals already have these data streams in place, generating and collecting new data on a standardized basis while ensuring the data are securely stored in compliance with legal requirements. Consequently, modern hospitals have amassed a wealth of data documenting their operations and outputs, which can be used to build operational state models.

Predicting waiting times in emergency rooms (ERs) is crucial for patient satisfaction and operational efficiency. Traditional queuing theory has been enhanced by machine learning (ML) and deep learning (DL) approaches, offering more accurate and adaptable solutions. High-dimensional Gradient Boosting Machines significantly outperformed traditional models, emphasizing the necessity of sophisticated ML models for optimal hospital operations [8]. Pianykh highlighted ML's scalability in handling complex patterns in ER operations [9].Pattnayak showed DL's superior accuracy over traditional methods, reducing human error and enhancing ER efficiency [10]. Kyritsis used a neural network whose adaptability was demonstrated across different industries [11].

While traditional machine learning (ML) methods have shown promise in healthcare applications, they often struggle with achieving high accuracy and generalization across diverse datasets. This gap in performance necessitates further refinement of prediction models. Studies have addressed this by enhancing accuracy through techniques such as outlier exclusion and the integration of system knowledge [12]. Advanced methods, including Ordinary Least Squares (OLS), ridge and LASSO regressions, Random Forest, and Quantile regression, have been shown to significantly improve predictive accuracy [13]. Additionally, in complex scenarios like multi-stage queues, transforming transactional datasets into ML-ready formats and employing grid search techniques have further optimized these models [14]. The thesis of this research is to explore and develop ML models that not only improve accuracy but also enhance generalization performance across various healthcare applications, particularly in predicting workload and optimizing resource allocation in emergency departments and beyond. This leads to the research question: How can ML models be further refined to enhance both accuracy and generalization in healthcare settings?

Furthermore, sensitive overload detection and effective load management is critical for enhancing healthcare efficiency, particularly in Emergency Department (ED) operations. Research highlights that key areas such as bottleneck detection and workload prediction can significantly improve ED efficiency. For instance, it is identified that long waiting times due to treatment delays, especially during treatment in progress and emergency room holding (ERH) procedures, using simulation models [15]. Machine learning further enhances this process by accurately predicting workload in a research with over 200,000 patient visits analyzed to predict work relative value units (wRVUs) [16]. These predictive algorithms facilitate real-time load balancing and resource optimization. Additionally, combining machine learning with optimization techniques can improve hospital scheduling systems, including operating room efficiency and appointment scheduling [17, 18]. These advancements highlight the importance of integrating machine learning and optimization to enhance resource allocation and scheduling in healthcare settings, ultimately reducing congestion and improving patient outcomes [19, 20].

With the ongoing digitization of healthcare data and the development of sophisticated analytical tools, the integration of advanced machine learning and deep learning models with operational data from EMRs and HIS presents significant opportunities for enhancing healthcare system scheduling. From predicting ED waiting times to optimizing exam schedules, these technologies offer improved accuracy and efficiency.

3 Dataset

3.1 Source and Description

Specially, the dataset used in this article is provided by the Medical Analytics Group [21], placed in the core of Massachusetts General Hospital, which is ranked as the best hospital in the country by U.S. News & World Report. The data has been posted on their Nature Machine Intelligence article [9] and their official website, inviting those interested in machine learning and operations research to explore their operations dataset as a challenge [22]. The study utilizes real-world data collected by the Medical Analytics Group, with the aim of extracting more information and constructing a more precise model. This model is designed to predict patient waiting times with greater accuracy and to identify bottlenecks in the overload states of medical facilities.

Clinical workflow outcomes are influenced by a variety of factors, and no single factor can fully explain delays or patient waiting times. Current delays in healthcare organizations can be related to staffing, patient arrival patterns, time of day, complexity of tests, bottlenecks in the operating environment, holidays, weather, and many other factors [23]. Pianykh's dataset contains data on both no-appointment (F4) and appointment (F1, F2, and F3) patients, covering approximately 600 to 1,000 days of complete patient flow records [9].

3.2 Details and Features

The dataset comprises time-dependent features such as patient arrival times, examination appointment times, and examination start times. These time-stamped data are crucial for understanding patient flow and hospital operations. Additionally, dynamic features, like the number of patients matching the scheduled time after the current time, aid hospitals in managing and optimizing patient flow and waiting times.

To get a comprehensive view of daily operations, the dataset also tracks the cumulative number of exam delays, the number of exam delays in the previous hour, and the number of patients scheduled before the current patient. These metrics help hospitals adjust operational strategies in real-time to reduce delays and enhance efficiency.

Reflecting demand and resource allocation for various exam types, the dataset includes the number of patients waiting for different types of exams (e.g. chest, pediatric, neurological, abdominal, vascular, cardiac, and musculoskeletal). It also contains facility-level characteristics, such as the total performed hours for ongoing examinations to help optimize equipment use.

To assess short-term operational workload, the dataset records average waiting times for the last few customers. This information can be used to evaluate operational load and resource requirements in the short term.

In addition, the dataset is from only one hospital, so there are limitations in our model when making predictions in different hospitals.

3.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) techniques are applied to understand the data distribution, detect outliers, and identify significant patterns that could influence model development.

3.3.1 Data Cleaning

In the process of Exploratory Data Analysis (EDA), ensuring comprehensive and accurate data cleaning is crucial for the success of the research. The statistical dataset of the health care system that is being analyzed includes four worksheets, each recording statistics for different medical imaging technologies: X-ray (XR), Computed Tomography (CT), Magnetic Resonance (MR), and Ultrasound (US). Initially, these worksheets contain 89 variables, with data counts of 42,767 entries for Worksheet 1, 15,653 for Worksheet 2, 23,584 for Worksheet 3, and 48,431 for Worksheet 4.

In the face of the dataset's complexity, a series of specific data cleaning steps have been undertaken: missing values have been identified and handled, data formats have been standardized, and outliers have been identified and dealt with. Utilizing R, 3 missing values were identified in Worksheet 1, while no missing data were found in the other three worksheets. To avoid the adverse impact that directly deleting rows with missing values might have on subsequent research, a more nuanced approach was adopted: the missing values were replaced with the mean value of the respective row.

In terms of handling outliers, a detailed analysis was conducted using boxplots generated in R. All feature values were normalized to ensure a mean of zero and a variance of one. Apart from binary logical variables representing yes/no states (1 for 'yes' and 0 for 'no'), a certain number of outliers were observed in all other features within the boxplots. Considering the complexity of the healthcare system and the precision and facticity required for model predictions, these outliers were deemed to be normal occurrences within the healthcare statistical data, reflecting the actual conditions of the healthcare system. Therefore, the decision was made to retain these outliers to maintain the authenticity and integrity of the dataset.

3.3.2 Descriptive Statistics

In the dataset, the primary aim is to predict the 'waiting time'. Thus, the descriptive statistic of waiting time was done first. The dataset comprises 130,431 data points. And the mean of it is 7.295, with a standard deviation of 25.898. Upon examining its range, the minimum and maximum values are -497.000 and 360.000, respectively, while the lower quartile, median, and upper quartile are 1.000, 6.000, and 15.000, respectively. Furthermore, given the additional consideration of the relationship between 'waiting time' and the 'time series', visualization of these two variables was performed to provide an initial understanding.

From Figure A-1a, it can be found that at some time points, there are only very few waiting time that is possible, while at others, there are a lot of feasible waiting time amounts. Hence, it can be inferred that the specific time of day is likely to have a significant impact on 'waiting time' predictions.

Besides the targeted waiting time data in the collected dataset, there are 66 features that are related to the waiting time. The types of certain features of each facility differ slightly from each other due to their various functions. To describe the features, they can be divided into three kinds of variables: discrete variables, continuous variables and 0-1 two-valued variable. Examples of these features are illustrated in Figure A-1c.

3.3.3 Correlation Analysis

In pursuit of the best possible model fit, a meticulous analysis was conducted on the correlation between each independent variable and the dependent variable 'Wait', with the most influential features being carefully selected. During this process, three variables that were not quantifiable in terms of time points were eliminated to ensure the precision of the analysis.

A correlation matrix is served as a powerful means to illustrate the relationships among variables. After a correlation matrix was generated from the refined dataset, a choice was made to visualize these relationships with a heat map in R. Additionally, by applying hierarchical clustering to sort the matrix, the interpretation of the heat map was made more accessible, particularly given the extensive volume of data that was being handled. In the heat map, the darker the color of the convergence area between variables, the stronger the correlation. This approach not only enhances the visual representation of the data but also facilitates the understanding of the complex interplay between variables.

From Figure A-1a,It was found that the variable 'delayedinline' had the strongest positive correlation with waiting time in Worksheet 1, with a Spearman's rank correlation coefficient of 0.28122, while the variable 'noneinline' showed the strongest negative correlation with waiting time, with a Spearman's rank correlation coefficient of -0.15024.

4 Methodology

4.1 Previous Machine Learning Techniques

The objective of this study is to develop predictive models for patient waiting times, leveraging existing datasets to train these models for accurate forecasting. The paper commences with an examination of the comparative efficacy and efficiency of various elementary machine learning algorithms in the context of predictive modeling.

4.1.1 Dataset Split

The methodology section delineates the training strategy employed. Data from four distinct medical facilities were subjected to independent training regimens, thereby cultivating facility-specific predictive models. The dataset was partitioned into a training subset, comprising 70-80% of the data selected at random, and a test subset, encompassing the residual 20-30%. Each model underwent a series of six training iterations, with the optimal iteration, determined by performance metrics, being retained as the definitive model.

4.1.2 Experiment on Different ML models

In pursuit of a comprehensive assessment of the predictive capabilities of diverse machine learning techniques, the study encompasses a spectrum of algorithms, including linear regression, Naive Bayes classifiers, Support Vector Machines (SVMs) with Gaussian kernels, single decision trees, and ensembles of decision trees, namely random forests. Some of these methods are also models that are discussed in precious articles [9] in this field. But not every model is discussed with a concrete result.

The linear regression analysis was conducted at multiple levels of complexity: a full multivariate regression incorporating all variables, a reduced multivariate regression focusing on a subset of significant predictors, and univariate regressions for individual influential predictors. The linear regression model can be mathematically described as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

where y represents the patient waiting time, β_0 is the intercept, $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for each predictor $x_1, x_2, \ldots, x_n$, and ε is the error term.

The SVM approach was standardized with a Gaussian kernel to facilitate model convergence. The SVM model with a Gaussian kernel can be expressed as:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

where $K(x_i, x_j)$ is the kernel function, x_i and x_j are data points, and σ is the bandwidth parameter.

The random forest models were parameterized with varying numbers of trees and tree depths to explore the impact of model complexity on predictive accuracy. The prediction for a random forest model is the aggregation of predictions from individual decision trees.

4.1.3 Parameter Settings

Optimal parameters for ML models are determined through cross-validation and grid search techniques to ensure the highest possible predictive performance.

The internal model parameters are dynamically optimized through the training process, contingent upon the characteristics of the training data set. Consequently, during experimentation, it is imperative to meticulously adjust the model's external tunable parameters in response to the predictive outcomes, thereby incrementally refining the model's performance. The subsequent discourse will elucidate the parameter design process for select models.

For example, within the ensemble of machine learning models, the Random Forest model necessitates a systematic refinement of both the quantity and the depth of constituent decision trees. Through iterative experimentation, it was determined that, to effectively accommodate the extensive parameter space of the medical system dataset utilized in this research, a robust ensemble of decision trees and increased tree depth are essential for achieving superior predictive accuracy. The optimal configuration identified in this study for the Random Forest model comprises 300 trees with a depth of 20 splits.

By incorporating these mathematical formulations, it can enhance the precision and clarity of the new predictive modeling approach while preserving the original narrative structure.

4.2 Neural Networks for Prediction

In addition to the foundational machine learning methodologies, the dataset for the prediction of patient waiting times is distinguished by the subtle influence of each feature on the overall waiting duration, with no single characteristic exerting a pronounced effect on the outcome. In light of this, the present study incorporates an exploration of deep learning models based on neural networks to address the prediction task, with a focus on evaluating the performance of such models in scenarios characterized by a multitude of features, each with a relatively minor impact.

4.2.1 Our Neural Network Model

The research commenced with the development of a rudimentary neural network framework to gauge its efficacy in predicting patient waiting times. The architecture was composed of either a solitary hidden layer or a dual-layer configuration, with each layer populated by either 10 or 20 neurons, which has already been used in this field to solve such patient waiting time problems [9]. Contrary to complexity, these elementary networks demonstrated an aptitude for handling the multifaceted nature of the problem, achieving a level of predictive accuracy that rivals or exceeds that of more established machine learning algorithms.

Nevertheless, while the rudimentary neural network configurations have modestly enhanced the precision of waiting time predictions, this study introduces an advanced neural network architecture specifically crafted to augment the model's predictive fidelity. This novel architecture expands the neuron count in each of the two hidden layers. Prior to each hidden layer, a Batch Normalization layer is integrated to facilitate the model's capacity to conform to the data's nonlinear dynamics. ReLU activation functions are utilized throughout to introduce nonlinearity at each layer.

The optimal configuration attained by some experiments for the neural network model entails two hidden layers, each populated with 256 neurons, preceded by a Batch Normalization layer to facilitate the fitting of data with nonlinear relationships, and activated by the ReLU function.

4.2.2 Loss Measurement

In this research, the neural network model's training was tuned with external parameters, guided by ongoing assessments of the model's predictions. The Mean Absolute Error (MAE) and Mean Squared Error (MSE) served as key performance indicators for the model's predictive accuracy. Given the volatile nature of our emergency room dataset, which includes many extreme outliers, MSE was utilized for training due to its properties that aid in gradient descent and quick convergence. Despite MSE's advantages, its susceptibility to outliers can lead to exaggerated error magnification. MAE was used for final evaluation, as it averages the absolute differences between predictions and actuals. The formulas for MSE and MAE are:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad \mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|.$$

For a more nuanced evaluation of the model's performance, U05 and U10 metrics are also considered. These metrics measure the accuracy of the model by calculating the proportion of predictions with absolute errors less than 5 or 10, respectively. The formulas are:

$$\text{U05} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(|y_i - \hat{y}_i| < 5), \quad \text{U10} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(|y_i - \hat{y}_i| < 10)$$

where $\mathbb{I}$ denotes the indicator function, and n is the total number of observations. This approach ensures a comprehensive assessment of the model's predictive capabilities, especially in the presence of variability and outliers.

4.2.3 Training

In addition, a flexible learning rate schedule has been put in place, which reduces the learning rate by a factor of ten whenever the model's performance plateaus, or in other words, when it stops improving in terms of loss reduction. This mechanism is designed to guide the model towards a more optimal solution. Moreover, a strategy for learning rate decay has been incorporated, with the training process planned to last for 50,000 epochs. The starting learning rate (η_0) is set at 0.01, and there's a mechanism in place to reduce the learning rate to one-tenth of its initial value if the model's loss does not decrease significantly over a period of 10 consecutive epochs.

The mathematical expression for the learning rate decay is as follows:

$$\eta_t = \begin{cases} \eta_0 & \text{if } t < t_0, \\ \eta_0 \cdot 0.1^{\left\lfloor \frac{t-t_0}{T} \right\rfloor} & \text{if } t \geq t_0. \end{cases}$$

Here, η_t denotes the learning rate at a specific epoch t, t_0 refers to the epoch at which the first plateau in performance is identified, and T signifies the 10-epoch interval.

The goal of the training process is to minimize the loss function $L(\theta)$, where θ symbolizes the parameters of the neural network. This optimization is carried out using the gradient descent method, and the parameters are updated according to the following rule:

$$\theta_{t+1} = \theta_t - \eta_t \nabla L(\theta_t)$$

This entire process is facilitated by the Adam optimizer in PyTorch.

4.3 Peak Capture & Overload Detection

4.3.1 Peak & Overload Definition

Predicting peaks and detecting overloads in emergency department (ED) data are crucial for optimizing resource allocation and reducing extreme waiting. In this section, the aim is to identify peaks and overload conditions using machine learning models. Various models are validated seperately to find the most sensitive overload detector, containing linear regression, decision tree or forest as well as the neural network that have been constructed.

peak is defined as a binary variable P, where

$$P = \begin{cases} 1, & \text{if the waiting time} > 10 \text{ minutes,} \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, *overload* is defined as a binary variable O, where

$$O = \begin{cases} 1, & \text{if the number of individuals waiting} > 5, \\ 0, & \text{otherwise.} \end{cases}$$

The model takes a set of variables as input, denoted as $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, where each x_i represents a different feature relevant to the prediction of waiting times and overload conditions. The model outputs the predicted waiting time $\hat{T}$, which is then used to derive the predicted peak $\hat{P}$ and the predicted overload $\hat{O}$. The aim here is to find the most suitable model offering the patients their waiting time, and another model for the hospital manager to detect overload and schedule medical resources.

4.3.2 Peak Capturing Sensitivity Measurement

For the peak detection task, which is treated as a classification problem, a confusion matrix of predicted-peak and actual-peak is generated, thus calculating several classification metrics:

- Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$
- Precision $= \frac{TP}{TP+FP}$
- Recall $= \frac{TP}{TP+FN}$

where TP, TN, FP, and FN represent the true positives, true negatives, false positives, and false negatives, respectively.

A comprehensive comparison of various machine learning models, including our neural network and other baseline models, was conducted to determine the most suitable model for peak capture. The models were evaluated based on their performance in predicting both waiting times and peaks, with a particular focus on their ability to accurately detect peak conditions (i.e., when waiting time exceeds 10 minutes).

4.3.3 Overload Detection Classifier

Description The detection of overload conditions in the healthcare system is also framed as a classification task. This task involves determining whether the system is in an overload state based on patient waiting times and the current operational status of the healthcare facility. The classifier's effectiveness in this context is again evaluated using a confusion matrix, with metrics such as accuracy, precision, recall, and F1-Score being calculated.

Measurement In the context of healthcare, the primary concern is to ensure that no overload condition goes undetected. Therefore, recall, defined as the proportion of actual overload cases correctly identified by the model, is the most critical metric. Maximizing recall ensures that the system is adequately prepared for every potential overload, minimizing the risk of missing a critical situation that could compromise patient care.

$$\text{Recall} = \frac{TP}{TP + FN}$$

where TP represents true positives (correctly detected overloads), and FN represents false negatives (missed overloads). A high recall value indicates that the model is effective in capturing all instances of overload, thus providing a reliable warning system for healthcare providers.

Correlation Between Overload and Peak In addition to evaluating the individual performance of the overload and peak detection models, it is essential to assess the correlation between these two phenomena. A strong correlation between overload and peak detection would suggest that the model accurately reflects the operational status of the healthcare system, providing a holistic view of its capacity and performance. In this study, the correlation between the variables *overload* (O) and *peak* (P) is analyzed using binary classification methods. The variable *overload* (O) and *peak* (P) is defined as follows:

- $O = 1$ (True Positive): when the number of people waiting exceeds 5.
- $O = 0$ (True Negative): otherwise.
- $P = 1$ (Predicted Positive): when the waiting time exceeds 10 minutes.
- $P = 0$ (Predicted Negative): otherwise.

To evaluate the effectiveness of using the *peak* variable (P) to predict the *overload* variable (O), the confusion matrix and several evaluation metrics were computed, including accuracy, precision, recall, and F1 score.

4.4 Generalization Improvement

4.4.1 Feature Selection and Decline

Given that not all hospitals may track these features, feature selection on the dataset was conducted to enhance the model's generalization performance. The four worksheets classified by device in the original dataset were used as the starting independent variable set for feature selection. The data approximated a normal distribution, which allowed us to employ two methods for feature selection: Principal Component Analysis (PCA) and Step Regression Analysis.

Step Regression Analysis involves iteratively adding or removing predictors based on their statistical significance:

$$Y = \beta_0 + \sum_{i=1}^{n} \beta_i X_i + \varepsilon$$

where Y is the dependent variable (waiting time), X_i are the independent variables (features), β_i are the coefficients, and ε is the error term. Features are added or removed based on criteria such as the Akaike Information Criterion (AIC) or p-values of the coefficients.

Step Regression Analysis was utilized to identify features with the most significant impact on waiting times from the original set of 83 features. For instance, in the analysis, the number of the key features was narrowed down to 54 from the original 83 in the worksheet.

Subsequently, PCA was applied to rank these selected features based on their impact, from the greatest to the least. This ranking was determined by the cumulative absolute loading values of each variable across all selected principal components. Principal Component Analysis transforms the original variables into a new set of uncorrelated variables called principal components. The principal components are ordered by the amount of variance they capture from the data.

The loadings, which are the coefficients of the linear combination of the original variables, are used to rank the features. The cumulative absolute loading value for a variable X_i across k principal components is given by:

$$L_i = \sum_{j=1}^{k} |w_{ij}|$$

where w_{ij} is the loading of variable X_i on the j-th principal component.

Step Regression Analysis had already identified 54 features with substantial influence on waiting times so the first 50 original features with the highest cumulative load absolute values were focused on as revealed by the principal component analysis(in Table A-2).

4.4.2 Distinct Feature Groups with Domain Knowledge

Aiming to enhance the generalization performance of our model by leveraging domain knowledge to construct neural network architectures tailored to distinct feature groups, the strategy here involved dividing the feature set into meaningful categories, ensuring each subset of features is processed by a dedicated sub-network that captures the specific patterns and relationships inherent in each group.

The approach here starts by categorizing features into five groups: appointment status, queue status, daily efficiency, immediate efficiency, and check type. Each of these groups contains unique information that can be utilized more effectively when processed separately.

1. **Appointment Status:** This group includes features related to the timing and scheduling of appointments, such as number of patients scheduled in the 30- and 60-minute window before patient arrived.

2. **Queue Status:** This information helps assess the load and performance of the queue management system, allowing healthcare organizations to adjust resource allocation in real-time to reduce patient waiting times, such as number of patients in line measured when a patient arrives, 15, 30, 45 & 60 minutes before.

3. **Daily Efficiency:** Metrics that reflect the overall efficiency of the emergency department, such as average delay/wait for patients for that day.

4. **Immediate Efficiency:** Metrics providing a snapshot of immediate performance metrics and offering a real-time view of ongoing processes, such as the sum of the expected times to complete of the exams in progress.

5. **Check Types:** This group deals with the characteristics and waiting times for different examination types, including ‘number of chest examinations’, ‘number of neurological examinations’ and etc.

In the model implementation, each feature group is fed into a dedicated sub-network. Each sub-network consists of two fully connected layers using ReLU activation functions, designed to capture the complex non-linear relationships within each feature group. For a given feature group $\mathbf{X}_i$ (where $i = 1,2,3,4,5$), the sub-network's output can be represented as:

$$\mathbf{H}_i = \mathrm{ReLU}(\mathbf{W}_{i,2} \cdot \mathrm{ReLU}(\mathbf{W}_{i,1} \cdot \mathbf{X}_i + \mathbf{b}_{i,1}) + \mathbf{b}_{i,2})$$

where $\mathbf{W}_{i,1} \in \mathbb{R}^{h \times d_i}$ and $\mathbf{W}_{i,2} \in \mathbb{R}^{h \times h}$ are the weight matrices, $\mathbf{b}_{i,1} \in \mathbb{R}^h$ and $\mathbf{b}_{i,2} \in \mathbb{R}^h$ are the bias vectors, and $\mathrm{ReLU}(x) = \max(0,x)$ is the ReLU activation function, with h being the hidden layer dimension.

The outputs of these sub-networks are then combined to form a consolidated representation of the input data:

$$\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4, \mathbf{H}_5]$$

Here, $\mathbf{H} \in \mathbb{R}^{5h}$ is the merged vector. This combined vector is processed through a final linear layer to generate the model predictions:

$$\hat{y} = \mathbf{W}_f \cdot \mathbf{H} + \mathbf{b}_f$$

where $\mathbf{W}_f \in \mathbb{R}^{1 \times 5h}$ is the weight matrix of the final linear layer, and $\mathbf{b}_f \in \mathbb{R}$ is the bias term.

Across all healthcare scenarios, these five dimensions make it possible to measure operating characteristics effectively for making predictions. The overall model can be succinctly represented as:

$$\hat{y} = \mathbf{W}_f \cdot \left(\sum_{i=1}^{5} \mathrm{ReLU}(\mathbf{W}_{i,2} \cdot \mathrm{ReLU}(\mathbf{W}_{i,1} \cdot \mathbf{X}_i + \mathbf{b}_{i,1}) + \mathbf{b}_{i,2}) \right) + \mathbf{b}_f$$

To summarize, features are categorized into five groups: appointment status, queue status, daily efficiency, immediate efficiency, and check types. Each category was processed by dedicated sub-networks with two fully connected ReLU layers, capturing specific patterns within the data. These outputs were merged and passed through a final linear layer for predictions. This model, combining rigorous feature selection and domain-informed subset grouping, demonstrated excellent performance on our sub-NN. By effectively capturing the nuances of various feature groups, it provides accurate waiting time predictions with better generalization performance, making it a valuable tool for emergency departments.

5 Results

5.1 Accurate Waiting Time Prediction

In this study, various prediction models are thoroughly tested, mainly in order to see how well they performed in predicting patient waiting times. The prediction accuracy of these models are evaluated by using two important metrics - mean absolute error (MAE) and mean square error (MSE). The smaller the value, the more accurate the model's prediction are. The data in several tables 1a, 1b, 1c, 1d are carefully analyzed, which show the results of some different machine learning and deep learning models, and the model is compared with these models [9]. In addition,

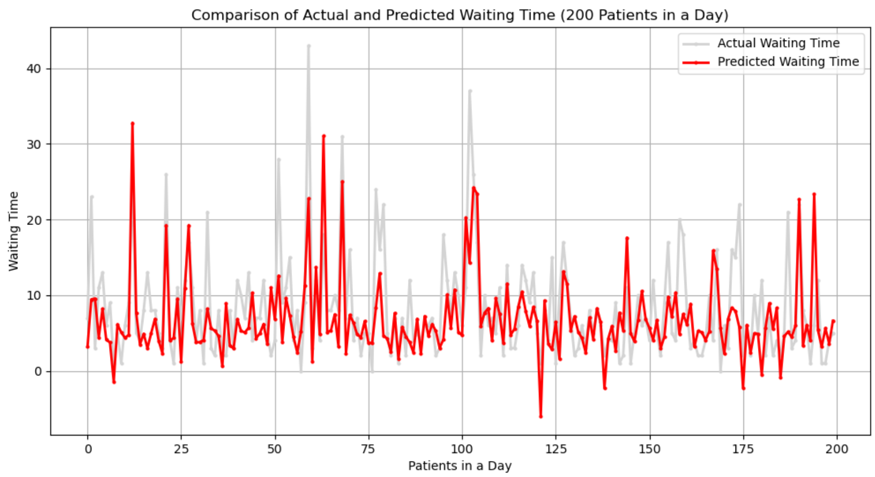


Fig. 1. Visualization of our Predictive Model

MLmodel	MAE	MSE	U05	U10	trainMAE	trainMSE
MostRecentWait	12.837	18.027	0.291	0.525	12.856	18.24
MostRecentWait-Average	12.778	17.969	0.287	0.524	12.809	18.096
LinearRegression	8.797	12.450	0.396	0.683	8.736	12.375
GaussianKernel	12.238	17.353	0.308	0.549	12.297	17.599
ForestSmall_300trees_30splits	12.659	18.132	0.334	0.602	9.012	0.270
NeuralNetwork-[10,10] Layers	9.172	12.815	0.341	0.608	9.812	13.789
NeuralNetwork-[20,20] Layers	9.847	13.629	0.312	0.559	11.033	15.350
Our_NeuralNetwork_Model	**7.617**	**10.880**	**0.196**	**0.377**	**6.479**	**9.032**

(a) Comparison of Models on f1 (US)

MLmodel	MAE	MSE	U05	U10	trainMAE	trainMSE
MostRecentWait	20.675	30.227	0.181	0.352	20.763	30.842
MostRecentWait-Average	20.706	30.887	0.181	0.351	20.621	30.519
LinearRegression	17.819	24.776	0.202	0.392	17.649	24.982
GaussianKernel	20.38	30.407	0.192	0.363	20.315	30.224
ForestSmall_300trees_30splits	25.892	36.882	0.151	0.291	0.011	0.413
NeuralNetwork-[10,10]_Layers	18.863	27.833	0.181	0.358	19.802	29.327
NeuralNetwork-[20,20]_Layers	19.083	28.541	0.19	0.365	19.166	28.479
Our_NeuralNetWork_Model	**18.594**	**27.903**	**0.145**	**0.284**	**16.499**	**22.36**

(b) Comparison of Models on f2 (MR)

MLmodel	MAE	MSE	U05	U10	trainMAE	trainMSE
MostRecentWait	32.236	48.169	0.121	0.238	32.463	48.207
MostRecentWait-Average	31.98	47.183	0.125	0.241	32.289	48.226
LinearRegression	23.329	30.086	0.139	0.275	23.144	29.864
GaussianKernel	31.568	47.288	0.124	0.244	31.819	47.746
ForestSmall_300trees_30splits	32.937	43.052	0.109	0.211	0.004	0.291
NeuralNetwork-[10,10]_Layers	23.255	30.664	0.130	0.255	25.465	33.339
NeuralNetwork-[20,20]_Layers	23.472	30.717	0.123	0.246	25.538	33.441
Our_NeuralNetWork_Model	**22.636**	**29.402**	**0.097**	**0.192**	**21.881**	**28.171**

(c) Comparison of Models on f3 (CT)

MLmodel	MAE	MSE	U05	U10	trainMAE	trainMSE
MostRecentWait	4.718	6.579	0.653	0.921	4.71	6.583
MostRecentWait-Average	4.653	6.503	0.662	0.921	4.641	6.464
LinearRegression	3.822	5.453	0.761	0.94	3.805	5.424
GaussianKernel	3.932	5.557	0.748	0.937	3.923	5.57
ForestSmall_300trees_30splits	4.906	7.201	0.669	0.876	0.015	0.344
NeuralNetwork-[10,10]_Layers	3.829	5.461	0.721	0.933	3.933	5.533
NeuralNetwork-[20,20]_Layers	3.952	5.541	0.644	0.907	4.634	6.186
Our_NeuralNetWork_Model	**3.782**	**5.407**	**0.612**	**0.851**	**3.683**	**5.249**

(d) Comparison of Models on f4 (XR)

Table 1: Comparison of Models across different datasets

the prediction results of our model is also shown in Figure 1, which lists the predicted and actual waiting times for 200 patients in one day in chronological order.

It is observed that, under identical modeling conditions, the predictive accuracy for various facilities exhibits notable divergence, with potential for substantial discrepancies. Despite these variations, the models demonstrate a commendable level of precision, thereby validating their utility. However, this observation necessitates a deeper inquiry, particularly given the dataset's inherent imbalance. The volume of data associated with different facilities is uneven, with those facilities exhibiting suboptimal performance also being underrepresented in the data, suggesting that the scarcity of training instances may be a contributing factor to their diminished performance.

Furthermore, a comparative analysis in single facility, taking Table 1a for example, reveals that the neural network predictive model, as orchestrated in this study, surpasses both traditional machine learning approaches and rudimentary neural network configurations in terms of predictive efficacy. This means that the model successfully transcend those basic learning models [9] clearly. This model adeptly fulfills the objective of patient waiting time prediction, corroborating the initial hypothesis posited at the outset of the paper. The meticulously calibrated neural network architecture is adept at tackling scenarios characterized by a multitude of features with attenuated individual impacts.

Additionally, this study incorporates a feature selection endeavor to bolster the models' versatility and applicability. In this section, the findings post-feature selection are delineated and interpreted. It emerges that employing a curated subset of features, as per the methodologies previously outlined, for model training results in a quantifiable diminution of predictive efficacy in correlation with the reduced feature count. The fidelity of predictions is intricately linked to the cardinality of the features engaged in the training regimen. Although a robust feature set can maintain an elevated level of predictive performance even with a modest decline, the predictive outcomes become increasingly stochastic with a diminished feature set. This unpredictability is heavily dependent on the particular features selected, thereby not ensuring the precision of predictions in a universally applicable context.

5.2 Overload Detection

Peak Capture with Linear Regression Experiments are conducted by using a Linear Regression model across four different modalities. The results of these experiments are summarized in Table 2a. The table displays the Accuracy, Precision, and Recall for each modality, which reflects the model's capability in detecting peaks.

Facility	Accuracy	Precision	Recall
1	**0.821**	**0.774**	0.679
2	0.699	0.711	**0.748**
3	0.714	0.730	0.730
4	0.788	0.660	0.473

(a) Peak Capture Measurement (LR)

ML Model	Accuracy	Precision	Recall
Linear Regression	**0.821**	0.774	**0.679**
LR (best features)	0.779	0.696	0.602
Decision Tree	0.711	0.572	0.582
Random Forest	0.721	**0.891**	0.211
Neural Network	0.785	0.798	0.486

(b) Comparison of Different Models

Table 2: Performance Metrics for Facilities and ML Models

The results indicate that Modality 1 achieved the highest accuracy (0.820) and precision (0.755), making it the most reliable model for predicting peak conditions in this context. However, Modality 2 shows the highest recall (0.748), suggesting that it is better at identifying all peak occurrences, albeit with a slightly lower precision. These findings provide valuable insights for selecting the appropriate model based on the specific requirements of peak detection and overload management in hospital settings.

Meanwhile, other methods such as decision tree, random forest or Neural Network has an accuracy of 70%-80%, lower than the LR model, taking Facility-1 as an example. The result shows that although NN can predict patient waiting time quite well, the Linear Regression model is more suitable for Peak Capturing Task. To summarize, can use the NN can be used to make waiting time prediction for the patients, while LR model is used as the Overload Detector presented to the hospital manager.

Time Peak Indicating Overload When the TN, FP, FN, TP are defined in Section 4.3.3. Peak of the waiting time representing Predicted-Overload, while True-Overload is defined as large amount of patients delayed in line. The confusion matrices for tables F1 and F3 are as follows:

$$\mathbf{CM}_{F1} = \begin{pmatrix} \text{TN} & \text{FP} \\ \text{FN} & \text{TP} \end{pmatrix} = \begin{pmatrix} 29258 & 13087 \\ 88 & 333 \end{pmatrix}, \quad \mathbf{CM}_{F2} = \begin{pmatrix} \text{TN} & \text{FP} \\ \text{FN} & \text{TP} \end{pmatrix} = \begin{pmatrix} 11326 & 11359 \\ 168 & 730 \end{pmatrix}$$

The corresponding evaluation metrics of F1 and F3 are summarized in Table 3. (There's no necessary for F2's or F4's overload to be measured because they are running well.)

Metric	**F1**	**F3**
Accuracy	0.6919	0.5112
Precision	0.0248	0.0604
Recall	**0.7910**	**0.8129**

Table 3: Evaluation Metrics for Tables F1 and F3

From the confusion matrices and evaluation metrics, it is evident that the recall rates for both F1 and F3 are relatively high, at 0.7910 and 0.8129, respectively. This indicates that the *peak* variable (P) is effective in identifying most of the cases where the *overload* variable (O) is true. In other words, when the waiting time exceeds 10 minutes (P=1), it successfully identifies the majority of situations where the number of people waiting exceeds 5 (O=1). While the *peak* variable can be used as a reliable indicator for *overload* with a high recall rate, the model's low precision indicates that additional factors should be considered to reduce the false positive rate and improve overall prediction accuracy. To summarize, long patient waiting time can be used to detect the department's overload, and the high recall rate shows that the detection method is of high sensitivity.

By focusing on the high recall for overload detection and analyzing the correlation between overload and peak predictions, this approach not only aims to detect critical conditions within the healthcare system but also strives to provide a model that offers a robust and accurate reflection of the system's current state. This ensures that the healthcare system can respond proactively to potential overloads, ultimately improving patient outcomes and operational efficiency.

5.3 High Generalization Performance

After feature selection, the filtered features are then integrated into our model, beginning with a subset of 5 features and incrementally increasing the count by 10 with each subsequent test. The findings indicated that as the number of features expanded, the Mean Absolute Error (MAE) of the model's predictions decreased, with minimal changes in the mutation value, the model's performance exhibits a gradual improvement (in Figure 2). Notably, this model can significantly reduce the required number of features without a marked decline in accuracy. This attribute is particularly advantageous when dealing with incomplete datasets or different scenarios, as the architecture can still train an effective patient waiting time prediction model using a subset of the original features.

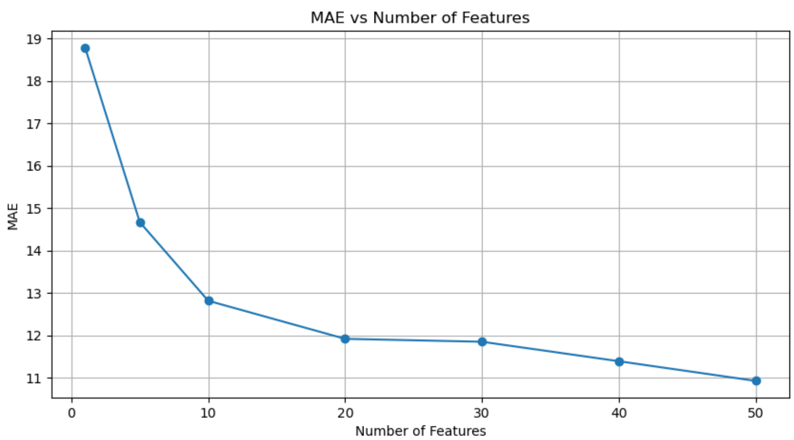


Fig. 2. MAE vs the number of features

By selecting five dimensions representing the current state of the healthcare system, the model maintains a high prediction accuracy. A neural network architecture was constructed comprising five integrated sub-networks, which achieved an accuracy level with a Mean Absolute Error (MAE) of 8.778, compared to the most accurate MAE of 7.617 (seen in Tables 1a, tested for Facility-1 as an example). This performance level remains notably high.

A systematic approach was used to improve the predictive power and understandability of the model by carefully differentiating the influence of each group of features. Incorporate expertise into models by grouping features and designing specialized subnetworks. The feature selection method effectively predicts patient waiting times even with a subset of features, especially when the available data is less complete than the features in the dataset, and detects systematic trends across different healthcare settings.

The model is powerful because the designed sub-neural networks demonstrate good ability to adapt to new situations based on different feature sets and medical knowledge. Through feature selection and clustering processing, an algorithm that predicts stable latency even when the amount of data is not large is obtained. The model optimizes emergency department resource allocation and maintains high accuracy by using key feature sets and separating them. This improves the understanding of the model and makes it a useful tool for healthcare professionals.

6 Conclusions

In this study, the challenge of predicting patient waiting times was addressed by developing a neural network. The method significantly improves the prediction accuracy, which is better than previous models. Based on this, the predictions can be presented to patients in the emergency room to reduce their waiting anxiety. An overload detection model based on linear regression was developed and examined its sensitivity. In addition, A feature selection process was conducted to identify key attributes of different diagnostic devices in the hospital environment. The model remains the similar accuracy when a subset of most important feathers are reserved. This process not only highlights the importance of these features, but also validates the predictive ability of our model at different scales, thus demonstrating its generalisation performance and usefulness in patient waiting time prediction.

This paper, evidently, carries profound practical significance and real-world applicability. The study presents a predictive model based on neural networks that offers accurate waiting time predictions. The predictive capabilities of our model can be seamlessly integrated into the emergency departments of hospitals. Furthermore, the use of the overload detector can help hospital managers allocate medical resources more efficiently. By learning from the relevant data of these medical institutions, the model can adeptly fulfill predictive tasks, thereby enhancing the allocation and coordination of medical systems across various hospitals, offering substantial assistance.

The findings contribute to the body of knowledge in the field of health informatics, offering insights into improving patient flow and resource allocation within healthcare facilities. The robustness of the model, as evidenced by its performance across different scenarios, underscores its potential for real-world application. As looking to the future, this work lays the groundwork for further exploration into optimizing patient experience and operational efficiency in healthcare settings.

However, it is important to acknowledge that this study has certain limitations. For instance, this approach primarily focused on refining simple neural network architectures, without incorporating a wide variety of complex neural network models. Additionally, the dataset that was utilized may not be extensive, as it was specific to one medical system, which could potentially lead to inaccuracies in prediction under certain special circumstances.

The insights gleaned from this research open avenues for future work. For instance, there is potential in experimenting with diverse feature combinations and model architectures to uncover more accurate and effective predictive methodologies. Moreover, the application of predictive outcomes presents numerous opportunities for refinement. While the current predictions are confined to the present moment, facilitating short-term adjustments and management within the hospital's medical system, there is scope to incorporate broader temporal references. This could enable more long-term, strategic forecasting and regulation of the healthcare system.

References

[1] Hijry H, Olawoyin R. Predicting patient waiting time in the queue system using deep learning algorithms in the emergency room. International Journal of Industrial Engineering. 2021;3(1):33-45.

[2] Shafaf N, Malek H. Applications of machine learning approaches in emergency medicine; a review article. Archives of academic emergency medicine. 2019;7(1).

[3] Heart T, Ben-Assuli O, Shabtai I. A review of PHR, EMR and EHR integration: A more personalized healthcare and public health policy. Health Policy and Technology. 2017;6(1):20-5.

[4] Dolin RH, Alschuler L, Boyer S, Beebe C, Behlen FM, Biron PV, et al. HL7 clinical document architecture, release 2. Journal of the American Medical Informatics Association. 2006;13(1):30-9.

[5] Ayaz M, Pasha MF, Alzahrani MY, Budiarto R, Stiawan D. The Fast Health Interoperability Resources (FHIR) standard: systematic literature review of implementations, applications, challenges and opportunities. JMIR medical informatics. 2021;9(7):e21929.

[6] Bidgood Jr WD, Horii SC, Prior FW, Van Syckle DE. Understanding and using DICOM, the data interchange standard for biomedical imaging. Journal of the American Medical Informatics Association. 1997;4(3):199-212.

[7] Curtis C, Liu C, Bollerman TJ, Pianykh OS. Machine learning for predicting patient wait times and appointment delays. Journal of the American College of Radiology. 2018;15(9):1310-6.

[8] Nelson A, Herron D, Rees G, Nachev P. Predicting scheduled hospital attendance with artificial intelligence. NPJ digital medicine. 2019;2(1):26.

[9] Pianykh OS, Guitron S, Parke D, Zhang C, Pandharipande P, Brink J, et al. Improving healthcare operations management with machine learning. Nature Machine Intelligence. 2020;2(5):266-73.

[10] Pattnayak P, Mohanty A, Das T, Patnaik S. Deep Learning Based Patient Queue Time Forecasting in the Emergency Room. In: 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS). IEEE; 2023. p. 541-5.

[11] Kyritsis AI, Deriaz M. A machine learning approach to waiting time prediction in queueing scenarios. In: 2019 Second International Conference on Artificial Intelligence for Industries (AI4I). IEEE; 2019. p. 17-21.

[12] Kuo YH, Chan NB, Leung JM, Meng H, So AMC, Tsoi KK, et al. An integrated approach of machine learning and systems thinking for waiting time prediction in an emergency department. International journal of medical informatics. 2020;139:104143.

[13] Pak A, Gannon B, Staib A. Predicting waiting time to treatment for emergency department patients. International Journal of Medical Informatics. 2021;145:104303.

[14] Al-Mousa A, Al-Zubaidi H, Al-Dweik M. A machine learning-based approach for wait-time estimation in healthcare facilities with multi-stage queues. IET Smart Cities. 2024.

[15] Zhao Y, Peng Q, Strome T, Weldon E, Zhang M, Chochinov A. Bottleneck detection for improvement of emergency department efficiency. Business Process Management Journal. 2015;21(3):564-85.

[16] Joseph JW, Leventhal EL, Grossestreuer AV, Chen PC, White BA, Nathanson LA, et al. Machine learning methods for predicting patient-level emergency department workload. The Journal of Emergency Medicine. 2023;64(1):83-92.

[17] Eshghali M, Kannan D, Salmanzadeh-Meydani N, Esmaieeli Sikaroudi AM. Machine learning based integrated scheduling and rescheduling for elective and emergency patients in the operating theatre. Annals of Operations Research. 2024;332(1):989-1012.

[18] Ala A, Alsaadi FE, Ahmadi M, Mirjalili S. Optimization of an appointment scheduling problem for healthcare systems based on the quality of fairness service using whale optimization algorithm and NSGA-II. Scientific Reports. 2021;11(1):19816.

[19] Shi Y, Mahdian S, Blanchet J, Glynn P, Shin AY, Scheinker D. Surgical scheduling via optimization and machine learning with long-tailed data. arXiv preprint arXiv:220206383. 2022.

[20] Wang Y, Zhang Y, Zhou M, Tang J. Feature-driven robust surgery scheduling. Production and Operations Management. 2023;32(6):1921-38.

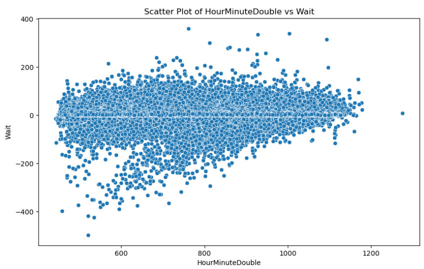
[21] Medical Analytics Group of Massachusetts General Hospital. Bringing Data-Driven Improvements to Healthcare; 2024. `https://medicalanalytics.group/`.

[22] Medical Analytics Group of Massachusetts General Hospital. Challenging Problems and Dataset; 2024. `https://medicalanalytics.group/operational-data-challenge/`.

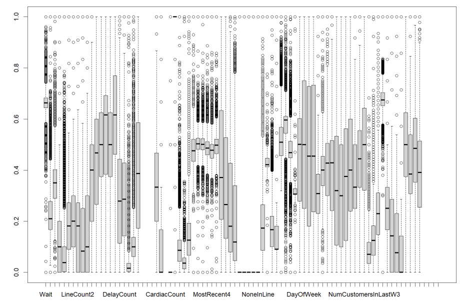
[23] Tang KJW, Ang CKE, Constantinides T, Rajinikanth V, Acharya UR, Cheong KH. Artificial Intelligence and Machine Learning in Emergency Medicine. Biocybernetics and Biomedical Engineering. 2021;41(1):156-72.

Appendix

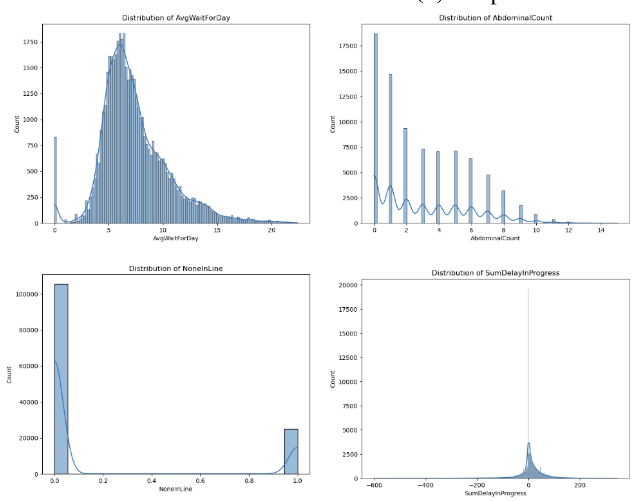
A-1. EDA Visualization



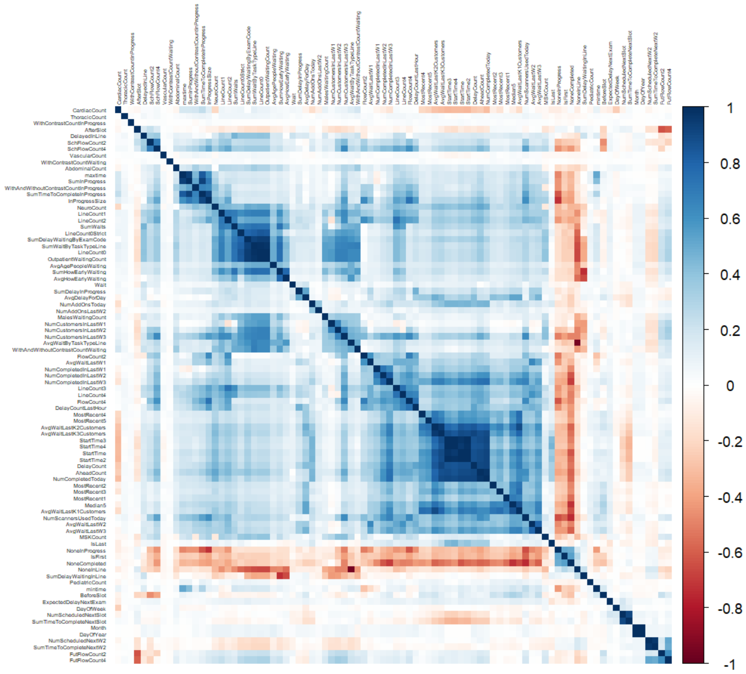
(a) Distribution of waiting time



(b) Boxplot of the dataset in Worksheet1



(c) Distribution of features



(a) Correlation between each variable in Worksheet1

Fig. A-1. EDA Visualization figures

A-2. PCA Results

Index	Variable Name	Absolute Load Value	Index	Variable Name	Absolute Load Value
1	NumCustomersInLastwl	6.382846	26	DelayCount	5.285572
2	F1owCount2	6.094908	27	Linecount0strict	5.274416
3	LineCount4	6.055510	28	IsFirst	5.272488
4	NumCompletedInLastwl	5.938251	29	SchF1owCount2	5.271732
5	AvgWaitLastw3	5.932923	30	NoneCompleted	5.255597
6	maxtime	5.873289	31	mintime	5.236939
7	LineCount2	5.856698	32	AvgWaitLastK3customers	5.236619
8	AvgDelayForDay	5.830229	33	AbdominalCount	5.222655
9	SumDelayInProgress	5.815193	34	IsLast	5.191561
10	NumCompletedInLastw3	5.778408	35	MostRecent1	5.157611
11	NumCustomersInLastw2	5.771374	36	SumTimeToCompleteNextW2	5.152805
12	NoneInProgress	5.714588	37	NumCustomersInLastw3	5.150708
13	Vascularcount	5.686995	38	AvgwaitLastk2customers	5.121899
14	LineCount1	5.648676	39	DelayedInLine	5.103083
15	Sumwaits	5.624718	40	Afterslot	5.067978
16	SumDelayWaitingInLine	5.605646	41	Aheadcount	5.061976
17	InProgresssize	5.524258	42	AvgwaitLastw2	5.046041
18	Median5	5.503731	43	FutFlowCount2	5.044763
19	DelayCountLastHour	5.500591	44	StartTime	5.037080
20	NumAddonsToday	5.493103	45	AvgHowEarlyWaiting	5.015407
21	AvgwaitLastwl	5.433343	46	FlowCount4	4.993000
22	sumInProgress	5.406005	47	SumTimeToCompleteNextslot	4.974212
23	Beforeslot	5.377051	48	MostRecent	4.915264
24	NumAddonsLastw2	5.310590	49	NumScheduledNextw2	4.842634
25	NumCompletedInLastw2	5.290821	50	NumScheduledNextslot	4.818803

Table A-2: Principal Component Analysis (PCA) Result of Worksheet1 Table