5-1-2020

Predictive analytics for emergency department patient flow in regards to incoming rate, admission, and leaving behaviour

Harish Kumar Manchukonda

Predictive analytics for emergency department patient flow in regards to incoming rate, admission, and leaving behaviour

By

Harish Kumar Manchukonda

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computer Science
in the Department of Computer Science and Engineering

Mississippi State, Mississippi

May 2020

Predictive analytics for emergency department patient flow in regards to

incoming rate, admission, and leaving behaviour

By

Harish Kumar Manchukonda

Approved:

Shahram Rahimi
(Major Professor)

J. Edward Swan II
(Committee Member)

Sarah B Lee
(Committee Member)

T. J. Jankun-Kelly
(Graduate Coordinator)

Jason M. Keith
Interim Dean
Bagley College of Engineering

Name: Harish Kumar Manchukonda

Date of Degree: May 1, 2020

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Shahram Rahimi

Title of Study: Predictive analytics for emergency department patient flow in regards to incoming rate, admission, and leaving behaviour

Pages of Study: 56

Candidate for Degree of Master of Science

In this work, we produce several prediction models for aspects of hospital emergency departments. Firstly, we demonstrate the use of a recurrent neural network to predict the rate of patient arrival at a hospital emergency department. The prediction is made on a per hour basis using date, time, calendar, and weather information. Then, we present our comparison of two prediction systems on the task of replicating the human decisions of patient admittance in a typical American emergency department. Again, a recurrent neural network (RNN) was trained to learn the task of selecting the next patient from the waiting-room/queue to be admitted for treatment. Lastly, we present our attempt to produce a regression model that can predict the likelihood that a given patient will leave after waiting a specific amount of time in the emergency department's waiting-room/queue. Such a model could be used to optimize the patient's waiting-room/queue of an ED to minimize the likelihood of patients leaving without receiving care.

Key words: Emergency Department, Recurrent Neural Network, Hospitals, Regression Analysis, Left Without Being Seen, Left Without Treatment, Flow Optimization

DEDICATION

To my parents for their endless love, support and encouragement.

ACKNOWLEDGEMENTS

Foremost, I would like to express my deep and sincere gratitude to the advisor and major professor, Dr.Shahram Rahimi, for allowing me to do research and for providing invaluable guidance and support throughout this research. In addition to my advisor, I would like to thank the rest of my thesis committee members: Dr. Ed Swan and Dr. Sarah Lee, for their encouragement and support.

I want to thank Potentia Analytics for supporting my research with valuable data, and this work would not have been possible without their priceless contribution. I thank my fellow lab mates at Mississippi State University: Alexander Sommers and Maya Bakhtiari Ramezani for the discussions and help in this research. Last but not least, I would like to thank my friends at Mississippi State University for all their support during my career.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

"An emergency department is a medical treatment facility specializing in emergency medicine and the acute care of patients who present without prior appointment; either by their own means or by that of an ambulance".[24]

Emergency departments (ED) are expensive and complicated systems that are critical to the mission of the hospitals they are embedded in. So much so that any process optimizations made can be pivotal for their successful operation at or beyond their previous levels. Before enforcing any changes in their processes or flows, hospitals evaluate the effect of the considered modification using simulators[3][7][10][20][26]. While most ED have resources in common (doctors, nurses, technicians, beds, medical devices, receptionists, etc...) the exact identity and quantity of each varies from hospital to hospital and even day to day within the same hospital. Efficient staffing is a perennial issue in service operations and more so in cases of medical service where the service is often vital and the volume, variety, and rapidity of patient arrival are variable. Being the department with the least prior planning involved in a person's visit to it, we identity the ED as possessing the greatest amount of uncertainty in how to staff itself across a hospital's departments.

Patient care is the primary goal of the emergency department, but to take care of the patients who come in, the availability of hospital staff is necessary. The prediction of the incoming patient rate will help the hospital management to schedule the staff to satisfy the patients. The long waiting time of the patient in the waiting-room/queue is associated with low patient satisfaction. As the patient allowed to wait longer leads the patient to leave the hospital without seeing the doctor, which is a loss for both the patient and the hospital. Prediction of the likelihood of a patient leaving the hospital will notify the hospital staff to prioritize the patient in the queue for the treatment. Analyzing the patient flow i.e., how patients are called into treatment after waiting for “t” minutes in waiting-room/queue, will explain the behavior of the hospital and improve the quality of the simulation in the emergency departments.

1.1 Literature Review

Many approaches were tried to predict the patient’s admissions to the emergency department at a particular hour and day of the year. [12] says that, according to descriptive statistics arrival day, an hour of the day, week of the year, and the month of the year are the top four categories which are highly correlated with patients admittance and Gradient Boost Machines(GBM) performed more accurately than the logistic regression and decision trees. [21] says that the model which was used for prediction gave a receiver operating characteristic (ROC) curve of 0.73, and they were unable to produce models for elderly ED returns. [6] says that hospital admissions were not random and can be predicted. The model forecasted with MAPE of approx.2% on monthly data and 11% for daily data and 41% for four-hourly data and 51% for hourly data. [17] predicted the patient's admission rate using XGBoost, DNN, Logistic regression and says, machine learning can be able to predict the incoming rates of the hospital more accurately with the patient historical information and triage information. [4] had performed various analyses on the predictive models and let the user choose the model to predict the hospital admission rates.

Many papers [18][25][5][13][16] suggest that an ED can be modeled using discrete event simulation techniques. Discrete event simulation systems operate as a sequence of events across a time where each event occurs at a particular time, and when occurring, the system marks the change.

Most of the work we found in this area was conducted in the simulation of an entire hospital or one of its departments to analyze flaws and anomalies, such as long wait times and the associated patient desertion. [15] presents a hospital simulation tool *MedModel* which allows various users, hospital directors, to examine the complex operational and planning issues that emerge from the interaction of all the hospital's subsystems. [15][30] present more focused simulation models, rare in that most simulations simulate entire hospitals with less detail rather than a detailed view of one department. The simulation is of all events (an event being any time a patient has something done to them in the system: they are admitted, they are treated, they are discharged, etc.) in the ED of a specific hospital named "The Cooper Health System" to reduce of the total length of the stay of each patient in the ED. [31] performed a functional analysis of a simulated hospital and discovered that most of a patient's time in the institution is spent waiting and proposed an operational procedure to reduce the waiting times in all scenarios. [32] presents a method of improving the patient flow in emergency departments by employing a dynamic priority queue, circumventing problems with FIFO, LIFO, and static priority queues, suggesting the use of a dynamic priority “M/M/c” queue instead. [28] discussed the practice of emergency departments introducing fast-tracks-beds to improve patient safety and reduce waiting times. [11] surveyed queuing theory applications on health care systems and reported that congestion in the waiting-rooms happens when there is a “poor quality of service” in the hospital.

The count of patients leaving the hospital without seeing a doctor is very low when compared to patients treated, and it’s highly correlated with the waiting times of the patients. There are many models that tried to reduce the count of patient LWBS. [22][8][29]

1.2 Research Questions

In this work, we attempt to answer the following questions:

1. How many patients will visit the emergency department on a particular hour of the day considering predicted weather and many other calendar variables?

If this can be done, a hospital can allocate staff and resources to those times of the day and those days of the week/month where they are most needed, increasing efficiency in terms of their use and decreasing the likelihood that staff is overworked and patients unattended to.

2. How does the patient flow work in the hospitals and how the priorities are assigned to the waiting patients?

An ED waiting-room can be modeled as a queue, but how patients are drawn from that queue in a real ED be modeled accurately? If so, if how an ED prioritizes patients for admittance into the ED proper can be modeled, then this model can be used to improve simulations of EDs, and the hospitals containing them, by making the simulation more true to life.

3. How do patients behave while waiting, according to the acuity levels and waiting times?

When patients leave the waiting room without being seen, it is to everyone's detriment: the patient is untreated, and the hospital is deprived of income. If this aspect of a patient's behavior could be modeled and predicted, on a minute by minute of waiting time basis, it could then become part of a system that would edit the priority of patients in the admittance queue, and optimizing it to reduce the net likelihood of patients deserting by setting higher admittance priorities on patients more likely to desert.

1.3 Dataset

This work was performed on a combination of three datasets: Hospital dataset, Weather dataset, and holiday dataset and each dataset is explained below,

1.3.1 Hospital Data

The ED of an academic hospital in the US provides the dataset for over one year; it contains the records of approximately 65,000 patient visits to the ED. The attributes of each record are as follows:

Patient-ID: A unique identifier of a patient.

Age: (number) Represents the age of the patient.

Sex: (Categorical variable) Represents the Sex of the patient, and the values are Male, Female.

ESI Level: (Categorical variable) Represents the Severity of the patient, and the values are 1,2,3,4,5.

Time of patient arrival: (Date and Time) Represents the time stamp of the patient's arrival to the hospital(YYYY-MM-DD HH: MM: SS).

Length of time between arrival in the waiting-room (AIWR): (Time in minutes) Represents the number of minutes patient taken after arrival and been in the waiting room.

Length of time between AIWR and termination of treatment: (Time in minutes) Represents the number of minutes patient taken from the waiting room to terminate from treatment.

Length of time between AIWR and checkout from hospital: (Time in minutes) Represents the number of minutes the patient taken from the waiting room to check out from the hospital.

Time of patient departure: (Date and Time) Represents the time stamp of the patient departure from the hospital(YYYY-MM-DD HH: MM: SS).

Total length of stay of the patient in the hospital: (Time in minutes) Represents the number of minutes a patient's length of the stay in the hospital.

1.3.2 Weather Data

The weather data was taken for a specific location and specific time interval at every hour, and the attributes are: ***Temperature***, ***Dew Point***, ***Relative Humidity***, ***Head Index***, ***Wind Speed***, ***Wind Direction***, ***Visibility***, ***Precipitation***, and ***Sea Level Pressure***

1.3.3 Holiday Data

The calendar attributes are added to capture seasonal patterns and are as follows:

Year, Month, Day, Day of the week, Hour of the day

Is it holiday: (Categorical variable) A holiday indicator (A boolean which is set to 1 for a day that is a holiday, 0 otherwise. All U.S. public federal holidays are used and all weekends are considered holidays)

Weight of holiday: A holiday weight attribute (A real that ticks down, from 1 to 0, in increments of 1/(24*n), from the first to the last hour of the holiday. n is the number of contiguous holidays, so, for example, if a public holiday occurs on a Friday, then n = 3 and the weight reaches 1/72 on the last hour of Sunday).

1.4 Thesis Organization

This thesis is organized as follows: Chapter 2 explains about the background of emergency department process flow and the technical background used in this work in brief, chapter 3 works on predicting the incoming rates of patients to emergency departments, chapter 4 works on patients admittance predictors, chapter 5 works on the prediction of patients likelihood of desertion(leaving the hospital without seen by doctor) and follows by conclusion and future work.

CHAPTER 2

BACKGROUND

Our aim in this chapter is to describe the patient flow in the emergency department. This patient flow is common, where it has exceptional cases when a patient arrives with a very severe cardiac attack. This chapter also aim's to describe the research procedure followed on dataset and machine learning algorithms in detail, which are used in this work.

2.1 Flow of Emergency Department

A patient entering the ED will pass through 4 stages as described below and as illustrated in the Figure 2.1.

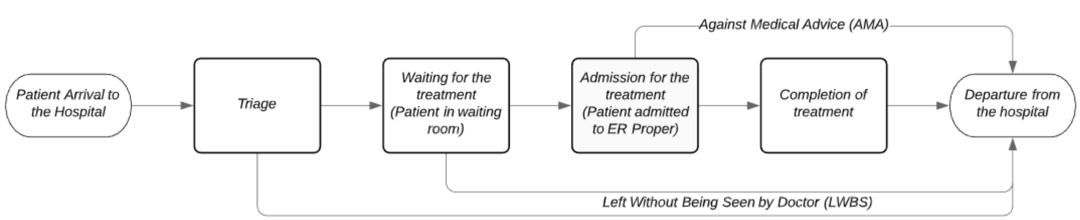

Figure 2.1

Flowchart of ED Visit

Stage 1, Triage: Triage assigns an emergency severity index (ESI) to each patient. This value ranges from 1-5 and indicates the severity of their condition with 1 being the most severe. Table 2.1 presents the meaning of each level.

Table 2.1

ESI Levels and Description [33]

Level	Description
ESI 1	Patient requires immediate intervention to avoid death.
ESI 2	Emergency, patient is in a high risk condition, vitals are dangerously abnormal.
ESI 3	Urgent, multiple medical personnel are required to stabilize the patient but vitals are not dangerously abnormal.
ESI 4	Semi-Urgent, one staff member is required to stabilize the patient.
ESI 5	Non-Urgent, The patient is already in a stable condition.

Stage 2, Waiting to be treated: After being assigned an ESI the patient is admitted into the waiting room. Mathematically the waiting-room acts as a queue out of which patients are admitted into the ED proper. Ideally, this queue is a priority queue where a patient's priority is mathematically determined from their ESI and time already spent waiting. For patients of identical ESI and wait time the queue resolves these ties randomly. Deviations may occur from expected queue behavior due to human decision on the part of the patients (they might leave), staff, or based on resource (medical personnel, beds, equipment, etc...) availability (a patient might not be admitted, even if they would be otherwise, due to the unavailability of a necessary resource, like a doctor specialized in their condition).

Stage 3, Admittance and treatment: Once a patient is admitted they are treated as necessary including them being assigned a bed, administration/prescription of drugs, and the performing of any needed medical imaging.

Stage 4, Departure: Once patients can return home they are discharged along with medical advice, equipment, and prescriptions to any needed medications. An alternative exit to the process comes when triaged patients leave the ED without being seen, desertion most often due to waiting times the patient deems unacceptable. The third transition to departure occurs when a fully admitted patient leaves before treatment is complete These last two methods of departure are anomalous and unwanted events that, when they do occur, occur most often with low severity patients. Higher severity patients are understandably less likely/able to desert.

2.2 Research Procedure

Data Science is extracting the knowledge and meaningful insights by applying many scientific methods , statistical process and algorithms on the data, can be structured and unstructured in format.

Data science workflow is the recurrent procedure of asking a question on the data by exploring the data and modeling the data with an algorithm and communicating with it. The stages that we need to follow to get useful information from the dataset are explained in Figure 2.2:

Data Preprocessing: Conversion of raw data into useful and efficient format using data mining techniques. These are the steps involved in data preprocessing:

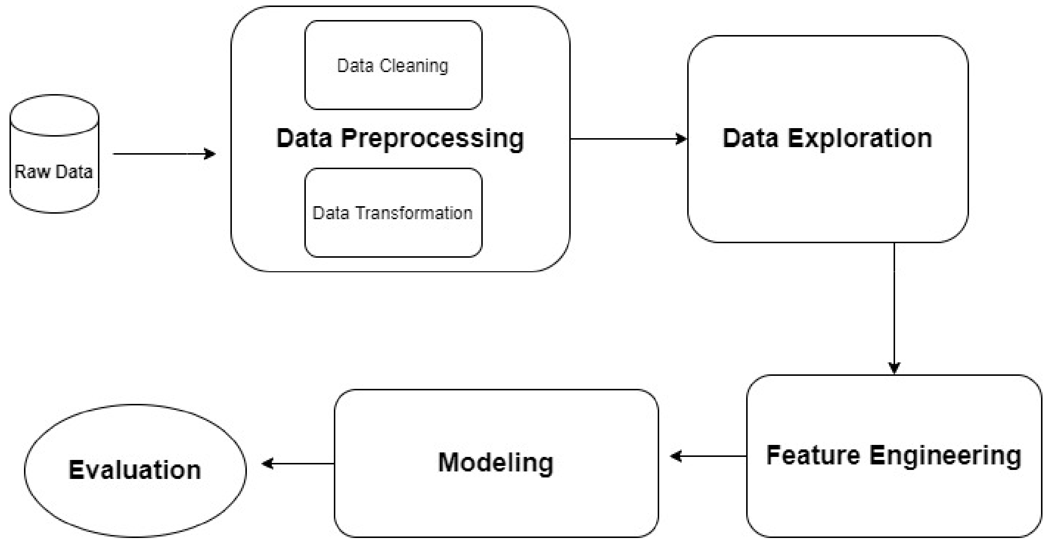

Figure 2.2

Flow of Data Science

1. Data Cleaning: The data may have lot of irrelevant information and missing values. By removing them we can reduce the noise from the knowledge.

2. Data Transformation: Transformation of the raw data into required forms by applying statistical methods, algorithms. Transformation involves many ways like Normalization, Discretization , Data Reduction and etc.

Data Exploration: Data Exploration is the first step of data analysis where exploring the patterns in the dataset , characteristics of the variables, correlation and causation of the variables. This process helps in recognizing the important trends and that lead to create a broad picture of the dataset.

Feature Engineering: Applying data mining techniques, process of extracting the features from the raw data with the help of domain knowledge. Performance of machine learning algorithms will increase with the essential features when we compare with all features.

Data Modeling: Data Modeling is a process of defining and analyzing the requirements which are essential to thrive the business within the scope of information systems of the organizations. Process of producing the diagrams and the relationships among various attributes to store in the database.

Applying machine learning algorithms based on the features retrieved and the pattern recognized. Modeling is applying the dataset on the model and where the model grabs almost the standard deviation of the dataset.

Model Evaluation: Model Evaluation is an critical part in development process of the model. Selecting the best model that actually represents the data and how well the model that predicts the future.

2.3 Regression

Regression is a statistical approach or data mining technique used to predict the numeric value(continuous value) of the target variable with the relation of other variables in the dataset. For example, Prediction of the distance from one city to another city, weather of the hour in a day considering many variables in the dataset provided.

There are many types of regression models used in this work and they are:

2.3.1 Linear Regression

Linear regression prediction models assume a linear relationship between the predicted (dependent) variable(s) and the predicted from (independent) variable(s). Simple linear regression has only one independent variable while "multiple" linear regression has several. The general form of linear regression can be expressed with the following function:

$$y = h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{2.1}$$

Where y, the predicted value, is predicted by calculating the weighted sum of x_1, x_2, ..., x_n, each of which is multiplied by some weight theta[2].

2.3.2 Decision Trees

This method can be used to produce regression predictors though it is more commonly known for producing classifiers. On the basis of its attributes, a record will traverse down the tree, being sorted at each junction of branches on the basis of a splitting rule. A record is classified or a real-valued prediction is made for it on the basis of which terminal, "leaf", node it is finally sorted into. We varied the "maximum depth" hyperparameter in this work to examine the effects this had on performance. This max depth value determines the maximum number of sorting rules that can exist in the longest such path down a tree. If this number is too high for a given dataset then over-fitting is more likely and the tree may generalize to new data poorly. We used max depths of 1, 3, and 5 for our experiments

2.3.3 Random forests

Random forests are several different decision trees generated using the same data-set. This ensemble of trees produces predictions by having the predictions of the individual trees aggregated using a variety of methodologies[19].

2.3.4 Support Vector Machine

The version of the support-vector machine (SVM) used for regression is called a support-vector regression (SVR) model. There is a hyperparameter for this model called its " margin". The margin is a distance about the line which will come to be the regression line formed via training. Points within this distance are those used to calculate the error of the model, and it is on the basis of this error that the line of best fit is adjusted. Because of the margin, only a subset of the dataset is used to form the final regression model in SVR[2].

2.3.5 K-Nearest Neighbors

In the K-Nearest Neighbor (KNN) method a data-point for which some value is to be predicted is projected into the already possessed dataset, the points of which have all their values known. The k nearest points to this projected point, its k nearest neighbors, are used to calculate the new point's predicted variable value. This is most often done by averaging the salient variable value of these k neighboring points. The hyperparameter k is obviously of importance and selection of a good k is a problem on its own. We used a k value of 50 neighbours for this problem.

2.3.6 Isotonic regression

In this approach, using at least a 2D attribute set, a line is fitted to the data with the following constraints: (1) The line must be as close to all points as possible, which amounts to minimizing the distance between the line and all data-points. (2) The line's trend must be monotonic, meaning that it must exclusively have a positive or negative trend, though segments are permitted to be flat, with a slope of zero.Following the above two constraints over multiple iterations, the system will converge to a solution.

2.3.7 AdaBoost

Adaptive boosting, or AdaBoost, is a kind of meta learning model. An ensemble of learning methods (such as those discussed above) is produced and the outputs of each of them are then the inputs to the meta model, which similar to standard regression, learns a series of weights. These weights are adjusted in a way to minimize error across the training data-set.

2.4 Clustering

Clustering is a unsupervised machine learning technique that groups the data points into clusters. Clusters represents the group of similar data points. Grouping the data points will be based on similar properties or features using statistical methods.

There are a couple of clustering techniques which were used in this work, they are:

2.4.1 K-Means

K means clustering is an unsupervised learning algorithm and popular clustering algorithm which represents by name with K clusters in a set of data and means represent the finding the centroid of the cluster. Clustering/Grouping the data is defined by centroid, which is the center of the cluster and every point in that cluster is closure to the centroid of the cluster.

2.4.2 Toeplitz Inverse Covariance-Based Clustering

The Toeplitz inverse co-variance based clustering method is used for multivariate time series clustering. This method defines each cluster as a dependency network showing the relationships between the different features in short sequences. Here the clusters correspond to a network called Markov Random Field(MRF) where it can show how the transition from time t to time t+1 [14].

2.5 Deep Learning

Deep learning is subset of a broader family of artificial intelligence and machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. Deep learning architectures such as deep neural networks, artifical neural networks like perceptron and multilayer perceptron ,recurrent neural networks, and convolutional neural networks.

2.5.1 Artificial Neural Network

“An artificial neural network(ANN) is a computational model based on the structure and functions of biological neural networks. ANNs are considered nonlinear statistical data modeling tools where the complex relationships beetween inputs and outputs are modeled or patterns are found” [1]. The Figure 2.3 shows the basic feed forward neural network.

2.5.2 Recurrent Neural Network

A recurrent neural network (RNN) is type of artificial neural network where the links between nodes form a network graph along a temporal sequence. This network graph

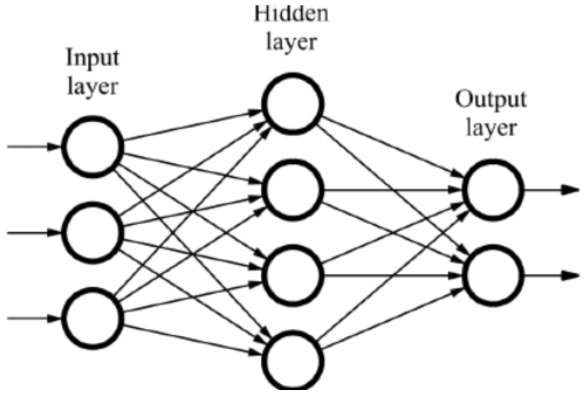

Figure 2.3

Feed Forward Neural Network

learns the temporal sequences while training and unlike perceptron and multilevel perceptron, RNN has their internal memory to process the sequence of inputs through temporal patterns.

The recurrent neural network reads the data from left to the right. The prediction at time "t" using RNN not only uses the input at the current state but also the information from the previous states. Inputs with various lengths can be processed with RNN architecture and model size doesn't increase with the size of the data because the data is considered as the sequence in training. The prediction of RNN will be based on the previous inputs but not on future inputs.

2.5.3 Long Short Term Memory (LSTM)

Long short-term memory (LSTM) is an type of recurrent neural network architecture which is used in deep learning. LSTM networks are mostly used for making predictions based on time series data because it has memory to store the required information which may have significant effect in the future.

RNN's architecture can able to predict the next state based on the current and previous state, but if there is a context where we need more information from the past to connect with the future state, then RNN cannot be helpful. LSTM's can solve this problem because it can handle long term dependencies. It uses tanh as activation function, and it has forget gate, input gate, output gate, and cell state, which represents the current state of the cell. LSTM's require more memory than the RNN because to store long term dependencies, and it takes more computation time to train because of the complexity in the architecture. LSTM has a lot of parameters to tune, and it can perform better on large datasets.

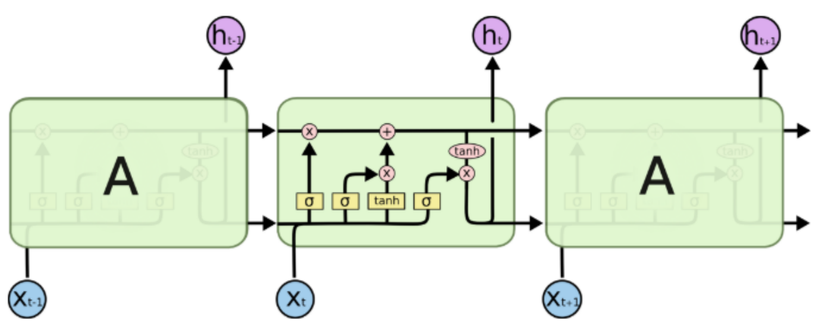

Figure 2.4

Long short-term memory: Variant of RNN

It has forget gate which leaves the unnecessary information and input gate which takes the current information and output gate gives the output considering the past information and cell state represents the current state of the cell.

2.6 Tools used in research

Tools used for data analytics are: **Rstudio, Jupyter, Google Colab, Orange**

Libraries used in the implementation of this work are:

Rstudio: tidyverse, dpylr, lubridate , stringr

Jupyter or Colab: sklearn (MinMaxScaler, Train_Test_Split, KFolds,RandomForestRegressor, DecisionTreeRegressor, LinearRegression, Adaboost, NeuralNetwork, Support Vector Machine, K Nearest Neighbors, Isotonic Regressor), pandas, numpy, matplotlib, Tensorflow, Keras(Dense Layer, LSTM, CuDNNLSTM)

Orange: (RandomForestRegressor, DecisionTreeRegressor, LinearRegression, Adaboost, NeuralNetwork, Support Vector Machine, K Nearest Neighbors, Isotonic Regressor)

Error metrics: R2 score, Mean Square Error(MSE). Root Means Square Error(RMSE), Mean Absolute Error(MAE)

CHAPTER 3

PREDICTING THE INCOMING RATES OF THE HOSPITAL

The incoming rates of patients explain how many patients come to emergency departments at a particular hour of the day. The hospital has to allocate the staff according to the demand of the patient's arrival to the hospital. Many stochastic algorithms provide the optimized schedules for the staff of the hospital, based on the incoming flow of the patients. The critical factor in generating the perfect schedule for the hospital staff is the incoming patient rate.

The incoming patient rate will be affected by the calendar parameters (hour of the day, day of the week, the season of the year), weather parameters(Temperature of the hour/day, Relative Humidity, etc.), and holidays.

3.1 Processing of Dataset

The prediction of incoming rates of patients to the emergency department requires all three datasets, as mentioned. Since the prediction of the incoming patient's rate is hourly, we have to arrange the data in chronological order from the start date to the end date of the dataset. The number of patients incoming at a particular hour of the day, month, and year are drawn from the hospital's dataset, and it is appended to the sorted dataset. The weather parameters and the holiday parameters are appended to the dataset, which is sorted by year, month, day, and hour of the day. Now, the created dataset is in chronological order of year, month, day of the week, day and hour of the day and weather parameters, holiday parameters, and the patient incoming counts, respectively.

3.2 Data Analysis

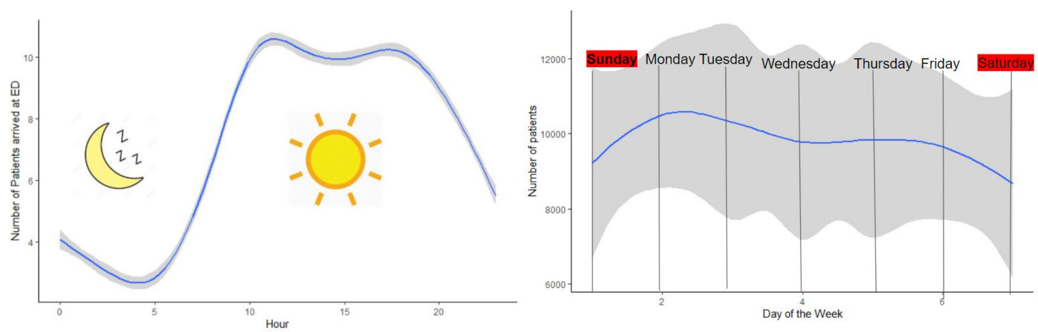

Figure 3.1

Effect of hour and week on patient's arrival count

Before the modeling, we want to know what are the most related features that will affect the number of patients count to ED. The Pearson correlation factor will explain this by a value how related the features are: 0 indicates not related, 1 indicates positively related and -1 indicates negatively related. The Pearson correlation among the hour and the number of patients results to be 0.444, and the relationship among the temperature and number of patients was 0.248.

The Figure 3.1 explains the relation between the number of patients and the hour of the day , day of the week. In above graphs the figure shows the smooth curves drawn with respective parameters again number of patients and the grey portion indicates the distribution of data across the line. The exciting facts from the graphs are: the patient's count is high in day time compared to night time which is matter of fact where most of the people will be awake in the morning rather night. The patient's arrival count on weekends is less compared to weekdays, and the first couple days of the week might expect the high count compared to the rest of the week. This result slightly leads to think about the effect of holidays on the number of patients arrival count.

Figure 3.2 explains how each holiday affects the count of patients coming to emergency departments. The patients incoming rate at every hour is drawn throughout the year and faceted by month. Holidays are marked in red dotted lines. We can observe in some of the holiday periods that the count of incoming patients will fall before and rise after the holiday. That demonstrates the significant effect the holiday has on the incoming rate of patients. The holidays where this effect is seen are as follows: Thanksgiving, Christmas, New Year's Eve, Mother's Day, Memorial Day, Father's Day, Independence Day, Labor Day.

3.3 Unsupervised analysis

Before performing the training, we have implemented an unsupervised learning method called clustering to find out the similar clusters in the dataset. Unsupervised learning may bring out the hidden pattern from the data which are not even expected. Since the data has seasonal features, we don't want to break the clusters losing them, and clustering has to consider the sequence of input features. Toeplitz Inverse Covariance-Based Clustering(TICC) is used to discover the repeated patterns in the data. We have removed the day,

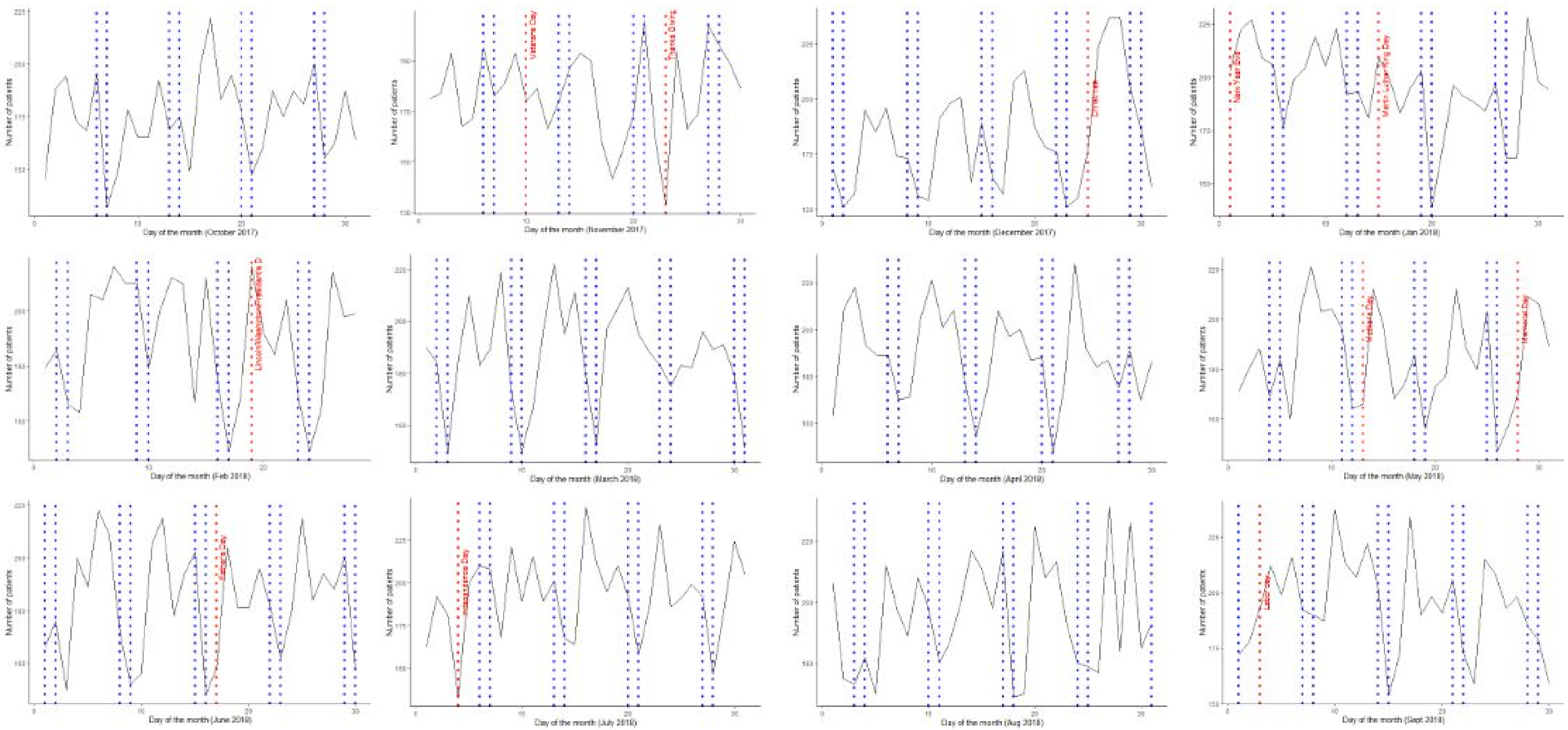

Figure 3.2

Effect of Holidays

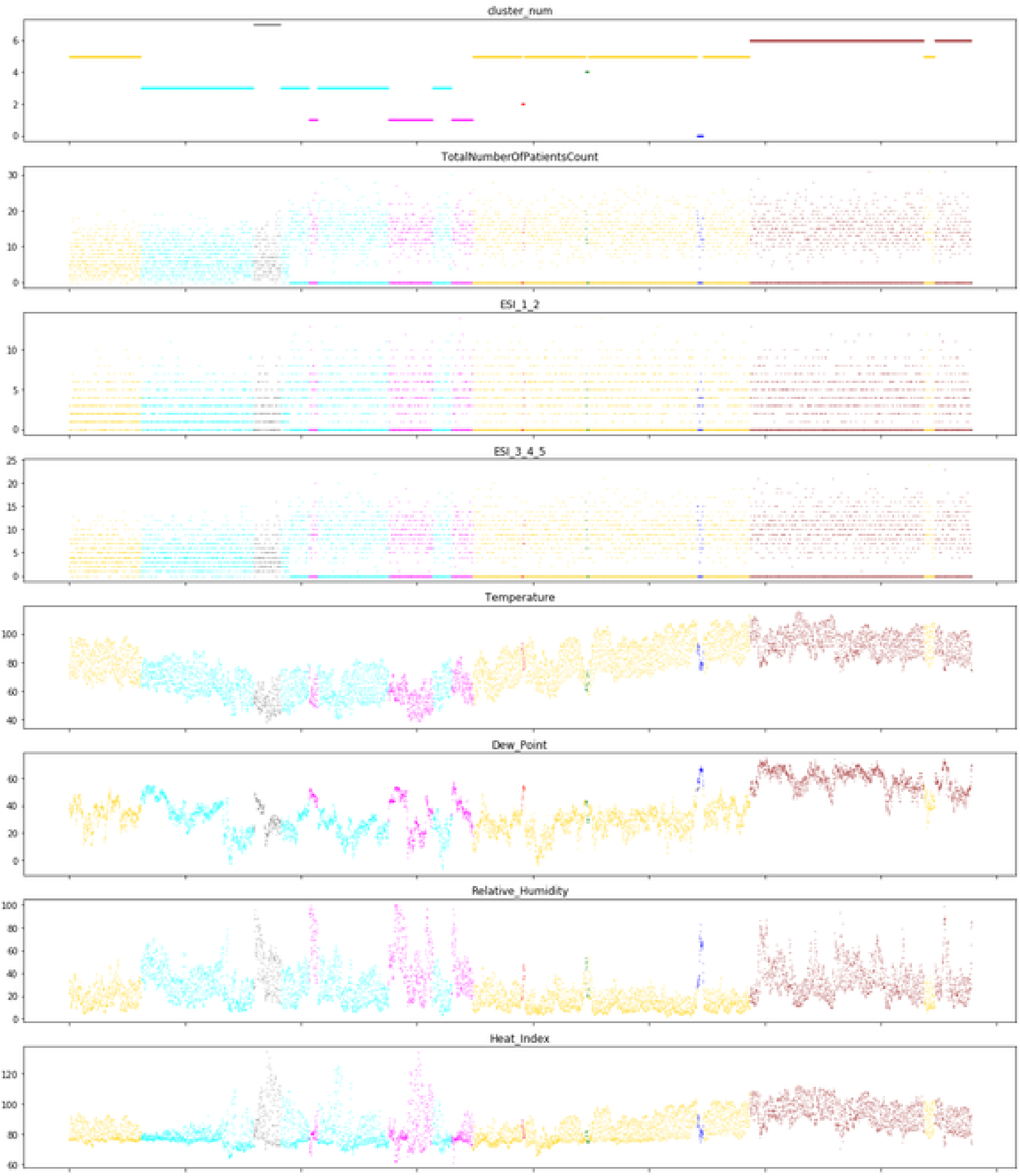

Figure 3.3

TICC Clustering

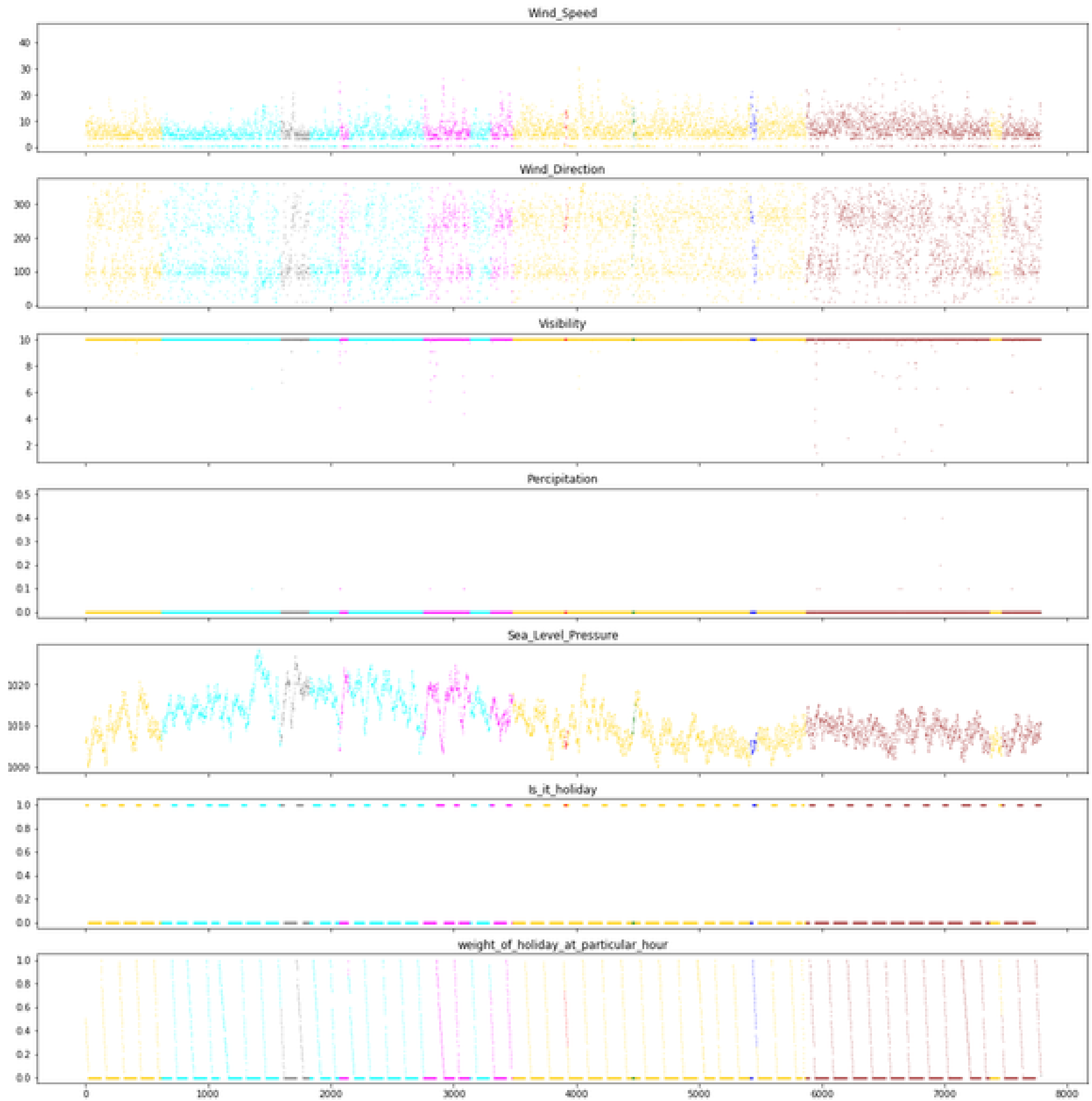

Figure 3.4

TICC Clustering Contd..

month, year, and day of the week parameters from the dataset, and the clustering is performed on the remaining data to see how TICC will capture the patterns and cluster them without losing any seasonal trends. Finally, TICC has performed exceptionally well, and it followed the seasonal patterns and clusters accordingly without losing the patterns. The Figure 3.3, Figure 3.4 explains the clustering on every parameter of the dataset considering the sequence of input features.

3.4 Methodology

Recurrent Neural Networks, or RNNs, were used to predict the sequences. There are two types of recurrent neural networks; those are Long Term Short Memory network and Gated Recurrent Units. These neural networks used memory to capture the sequence among the inputs and features.

According to our dataset, there are some features where we need to capture the sequence of the flow of inputs. Example: Weather parameters, the current hour weather is related to the previous hours, and the next hour weather also depends on the current hour weather. To capture all sorts of patterns in the data, the RNN is applied to the dataset.

3.4.1 The Recurrent Neural Network Architecture

Our RNN was trained on the dataset with output as incoming rate and inputs as weather parameters, calendar parameters, and holiday parameters. RNN was implemented using Keras 2.3.0 library.The training had a train-validation-test split of 64% training, 16% validation, and 20% testing and was trained for 100 epochs with a batch size of 100. The loss function was ”means square error” and an adam optimizer was used. The architecture of the neural network consists of 4 hidden layers with input and output layers. The first layer of the RNN is the LSTM layer and the rest being densely connected layers. The activation function used was ”tanh”. The neurons dimensions to each layer follow like 16, 130, 90, 48, 16, and 1.

3.5 Results and Discussion

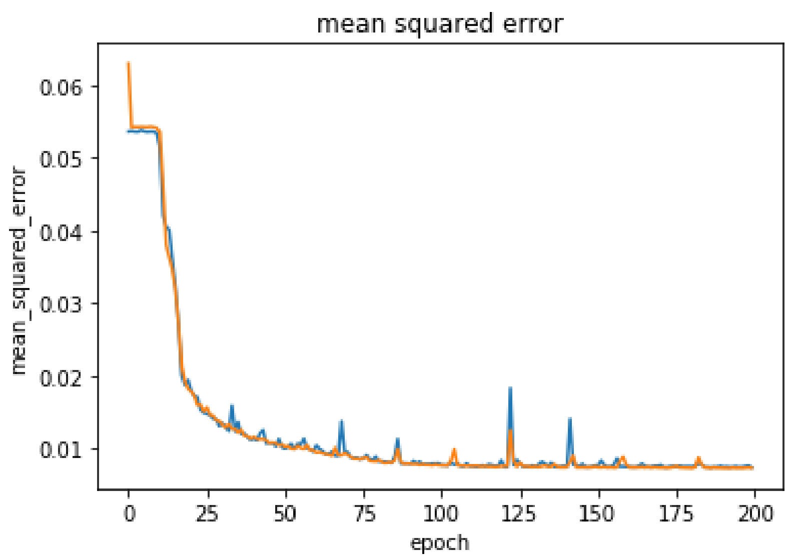

Figure 3.5

Mean Square Error of RNN

As mentioned earlier, we removed the calendar attributes (temporarily) from our and performed clustering with the best performing number of clusters being 7. Figure 3.3, Figure 3.4 shows how the weather parameters bring out the seasonal clusters with out the inputs of month and year.

While not a one to one mapping onto yearly seasons in the US this is closely similar and shows a strong seasonality to the data, one likely to be picked up on by a neural net even if the data points' clustered identity were not, as they were with us after this point, explicitly labeled. We reassembled the data, adding back the calendar attributes and adding the final cluster identity attribute to each row and then prepared for regression, as performed by our RNN, by normalizing the data. For the normalization we used the sklearn library's "MinMaxScaler" function and when given the data to train and test on our RNN performed well. MinMaxScaler subtracts the minimum value in the feature and then divides by the range (the difference between the maximum value and the minimum value of the feature).

Our mean squared error (MSE) was only .007 while our r2 score was .86, exhibiting an 86% accuracy in predicting the real world results. Figure 3.5 shows the MSE's inverse relationship to the progression of training.

The predicted incoming rates can be used for the realtime schedule generation, and it can be more accurate than the schedules that are generated with the averaged incoming rates.

CHAPTER 4

PATIENT ADMITTANCE PREDICTORS

In an emergency department admittance follows, when successful, a standard sequence: a patient waits for x minutes in the waiting-room/queue after which point they are admitted into the ED proper for treatment. Predicting which patient is selected out of the waiting-room/queue for the treatment is the goal for the patient admittance predictors. In this work, we introduce and compare two prediction systems on the task of replicating the human decisions regarding patient admittance in a typical American emergency department.
The purpose of modeling and constructing an autonomous patient admittance predictor in this work is thus primarily to improve the quality of the aforementioned simulations. Obviously the more accurate are the components that make up the simulation then the better the simulation represents the real world situation and the better will be the decisions made on the basis of that simulation.

4.1 Processing of Dataset

The dataset used in this work contained around 65,000 records, though of them only 39,130 records presented data describing queues containing more than one patient (those describing single person queues were removed since a queue that only ever consists of one person, where that person is then called in, does not require prediction) and was collected at a private US hospital for one year. Each patient has two major parameters waiting for time and ESI level which need to combine to bring the metric.

4.1.1 Calling Probabilities

The actual dataset allows us to reconstruct the waiting- room queue as it evolves across a given day and we can use the time a patient waited before being admitted to the ED proper, and their ESI level, to calculate a derived value, the probability a patient will be called at the end of their x^{th} waited minute. This metric, a patient's "calling-probability", is produced from the dataset using the random forest as a function approximator. The function in question can be seen in Equation 1 but only works for combinations of ESI and waiting time extant in the dataset. Given each patient's calling-probability we can compare those patients which have the highest calling-probability in any given queue with the patients that were, in fact, admitted out of that queue, and see how closely they map to one another. This work concerns the use of the above-described dataset to train and/or test systems that predict which patient is called into the ED from the waiting room at any given calling occurrence.

$$P(C_{L,W}) = \frac{c_{L,W}}{(c_{L,W} + nc_{L,W})} \tag{4.1}$$

$P(C_{L,W})$ (the calling probability) for a patient of ESI level L, with a wait time of W, equals $c_{L,W}$ (a count of observed patients of that ESI level and w-time that were called in on their W^{th} waited minute) divided by the total number of patients witnessed in the dataset with those same characteristics, called in ($c_{L,W}$) or not ($nc_{L,W}$) on the W^{th} minute. Calling-probabilities are specific to ESI level and are calculated as though a patient's likelihood of being admitted is independent of the other patients in the queue. Figure 4.1 shows the plotted calling-probabilities against time after AIWR for ESI levels 1 (crosses) and 5 (circles).

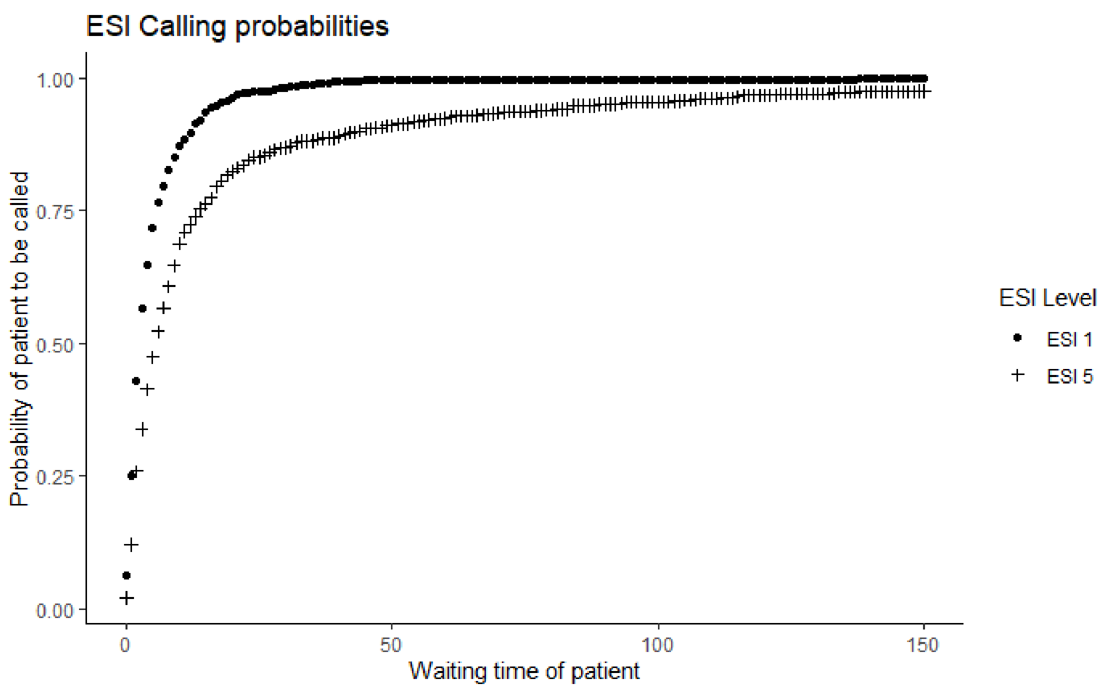

Figure 4.1

Calling-probabilities against time for ESIs 1 and 5.

Because we needed a continuous function and our formula only works for combinations of ESI and wait-time extant in our dataset we needed to use a function approximator and selected a random forest as described below. The approximated function was then used to calculate calling-probabilities. The curves in Figure 4.1 do look similar to log curves suggesting the application of logistic regression when trying to model them, however, this is more suitable for classification problems than regression problems.[23] For the uninitiated: "A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting."[19] To build our random forest we used the sklearn library version 3.2.4.3.2 and achieved an accuracy, averaged over all ESI levels, of 96%. During training, we used a test-train-split of 80% train and 20% test and used a max depth of 3 and the number of estimators (Number of trees in the forest) set to 10. The models were then exported and were the functions used to calculate the calling-probabilities to the dataset as it was transformed.

4.1.2 Conversion of the dataset to the required form

In reference to Figure 4.2 the transformation of the dataset from its original form to a form acceptable for training the RNN was straightforward.

a) This is the original patient-centered form of the data. Each record describes a patient's visit to the ED with their ID, a time-stamp of when they arrived in the waiting-room, their ESI level, etc...

b) From a. the queues are reconstructed and in the queue reconstruction form each record documents an instance of a patient in the waiting-room/queue being called and admitted into the ED proper. The values of how long a patient waited in the waiting-room, and what their ESI level was, are combined in form c via the function approximated by our random forest. It should be noted that the first attribute indicates the time since the "start-of-day" that has elapsed when a patient was called from the queue. Thus, if the start-of-day is 8:00

a.

Patient ID	Time arrived in waiting room	ESI level	....	Time spent in waiting room (in minutes)	Time Departed from hospital
1	8:00	2	....	2	8:59
4	8:50	2	....	1	9:17
5	8:51	3	....	8	9:55
6	9:00	3	....	60	10:00
7	9:02	3	....	0	11:00

b.

Number of minutes after "start_of_day" when a patients was admitted into ED proper	Queue (Patient Id's)	**How long patients have waitied in waiting room**	**ESI of patients**	ID of patient called into ED Proper
2	1	2	2	1
51	4,5	1,0	2,3	4
59	5	8	3	5
62	6,7	2,0	3,3	7
82	6,8,9	22,11,7	3,3,3	8

c.

Number of minutes after "start_of_day" when a patients was admitted into ED proper	Queue (Patient Id's)	**Calling Probability (CP) of patients**	Patient called for the treatment	Calling Probability (CP) of called patient
2	1	0.2	1	0.2
51	4,5	0.25,0.05	4	0.25
59	5	0.04	5	0.04
62	6,7	0.07,0.05	7	0.05
82	6,8,9	0.4,0.1,0.07	8	0.1

d.

Number of minutes after "start_of_day" when a patients was admitted into ED proper	Calling Probability (CP) of patients	Calling Probability (CP) of called patient
2	0.2	0.2
51	0.25,0.05	0.25
59	0.04	0.04
62	0.07,0.05	0.05
82	0.4,0.1,0.07	0.1

e.

Number of minutes after "start_of_day" when a patients was admitted into ED proper	Calling Probability (CP) of Patient 1	Calling Probability (CP) of Patient 2	--------	Calling Probability (CP) of Patient n	Calling Probability (CP) of called patient
2	0.2	0	---------	0	0.2
51	0.25	0.05	----------	0	0.25
59	0.04	0	----------	0	0.04
62	0.07	0.05	---------	0	0.05
82	0.4	0.1	----------	0	0.1

Figure 4.2

Conversion of original dataset to records of queues am and a patient is called out of the queue at 9:02 am, the entry for this attribute would be 62 minutes.

c) The calling-probability form of the data introduces the calling-probabilities which are ultimately used to judge and/or calculate the patient admission predictions the prediction systems tested to make.

d) We drop attributes that are extraneous to the neural network and then expand the data in e., achieving the final form.

e) The neural-network form of the data is that which our neural net accepts as input. Each record is one labeled input that the neural net can be trained or tested with. Since our neural net required fixed length inputs the variable nature of the size of the queue is expressed in the occupation of a queue with an upper limit of n patients where n = 49. Each non-zero calling-probability in the array represents the patient in the queue while absences in queues below 49 patients are indicated as 0. The label for the input is not a patient ID but instead a calling probability indicating that the selection, by a prediction system, of any patient with that calling probability would be a "correct" prediction.

4.2 Methodology

In this study, we have applied the data on two algorithms and those are:

4.2.1 Pick Most Severe Algorithm

This algorithm, drawn from industry used simulations, nearly implements the obvious heuristic of picking the patient with the lowest ESI first but mildly modifies it with wait time considerations. Increasingly large weights, scaling up as ESI falls, are multiplied by each patient's wait time to produce a score and the patient with the highest score gets selected. The weights on the lowest ESI levels are so large as to dwarf the effect of even the longest reasonable waiting times for higher ESI levels.

4.2.2 The Recurrent Neural Network Architecture

Our RNN was trained on the transformed version of the dataset described earlier and was implemented using the Keras 2.3.0.The training had a train-validation-test split of 64% training, 16% validation, and 20% testing and was trained for 100 epochs with a batch size of 100. The loss function was " means squared error" and the optimizer was the popular adam optimiser, The architecture consisted of 4 computation layers, the first being a "long short-term memory" (LSTM) layer with the rest being densely connected layers. A 25% dropout was used between layers one and two and layers two and three. The activation function used was the hyperbolic tangent function (tanh) because of the nature of our input. To retain our input's expected dimensions, we almost always had to pad them with 0s (padding was needed when there were fewer than 49 people in the queue) and we needed these null value's effects to be propagated forward. Tanh, unlike sigmoid or softmax, allows 0 as a neural value. Our layer dimensions, starting from the input and ending in output, are, in order: 49, 130, 65, 32, 16, and 1. While not meaningfully changing the achieved accuracy a 50% speedup in training was achieved using a CuDNNLSTM layer, a CUDA implemented version of an LSTM layer able to be run using a GPU.

4.3 Results and Discussion

These tests performed to examine the success of each prediction system are straightforward with the prediction system's patient predicted for admittance (predicted patient) compared to the patient admitted in the data, representing a human decision, (actual patient) which is considered our "expected/accurate" choice. The industry algorithm achieved a baseline accuracy of 44.04%, meaning it replicated 44.04% of the choices the human-made. In several cases, the calling probabilities of the actual and predicted patients were rather close and if we broaden the notion of accuracy for "agreement" between actual and predicted patients we can see that accuracy, naturally, improves.

Table 4.1

Accuracy measurements with algorithm

Diff in calling-probability of actual and predicted patients deemed "accurate"	Accuracy of the algorithm's predictions per level of acceptable error
== 0.0	40%
≤ 0.1	49%
≤ 0.2	58%
≤ 0.3	65%
≤ 0.5	80%

Table 4.1 shows the accuracy archived by the algorithm as the allowed difference in calling probability between the actual and predicted patient is increased. The allowed difference is the number of percentage points the actual and predicted patients' calling probabilities are allowed to differ by and still have the prediction count as accurate. In the first

row, with the intuitive level of no allowed difference, the algorithm is only accurate if it selects the actual patient or one with an identical calling-probability. An accuracy of 80% is achievable but only at the cost of widening the notion of accuracy to an unacceptable 50 percentage points. All such results are achieved on the raw dataset, which is, of course, formed for neural input but has none of the so-called "anomalous cases" removed.

The neural net, which is trained on this same dataset, and variants thereof, which will be explained, performs, with a return to a strict definition of an accurate choice, at a level of 44.15% accuracy (averaged across the results of a 5-fold cross-validation), which is hardly an improvement on the industry algorithm, which suggested to us, at the point we got that result, that the algorithm's poor performance might not be due to its simplicity.

What we mean by anomalous cases needs to be explained before the rest of our results are presented. What we call an anomalous case is any admittance witnessed in the dataset where there is at least a 50 percentage point difference between the calling probability of the actually admitted patient and the patient predicted to be admitted by the industry algorithm. The fundamental identity of these anomalous cases is unknown to us as the dataset does not characterize them enough to describe them meaningfully. We do not know how many varieties there are, what conditions in the hospital produce them, and indeed they are known to us in only a mathematical sense, and an assumed one at that, constructed to investigate our models' largest sources of error. We suspect these cases, based on the conversation with an industry expert, to be made up of fast-track-bed cases or those admission anomalies caused by the availability of medical staff or other resources optimized for treating a patient causing that patient to be admitted earlier or later than was predictable.

A variant of the dataset which removes these anomalous cases allows the RNN to perform at an accuracy of 75.29%, demonstrating a significant amount of the network's error came from them but that a significant amount more remains. The thus removed cases amount to 20.5% of the dataset and cannot, we originally supposed, be dismissed as outliers, at least en masse, though an industry expert assured us that they can be considered as such. The industry algorithm achieved accuracy on this reduced dataset of 47.01% demonstrating that it is indeed sub-optimal, at least when tested with this dataset, even in the absence of obviously tough to predict outcomes.

Comparing these two results we can tentatively advance the RNN as a superior prediction system, not that it is surprising that the more expressive neural network can outperform a heuristic. If we assume all anomalous cases have an explanation (one was a fast-track-bed case, another was a burn case and the staff specializing in burns had nothing else to do just then so the patient was admitted very early, etc...) which could somehow be made present/indicated in the data a predicting system receives as input then we can assume that a more sophisticated dataset with these signals would allow the NN to perform better on the whole dataset than the industry algorithm simply by perceiving and computing on these additional characteristics. Lacking knowledge of what exactly these anomalous cases were we augmented our data by simply adding a binary flag to each input indicating whether it was or was not input for an anomalous case. This boosted the accuracy to 84.97% which is interesting in that such a performance boost was achieved without actually being able to characterize any anomalous case but by its presence. This suggests to us that perhaps even something as simple as one new input flag, if producible as part of a patient's triage, perhaps by some kind of system, or personnel, aware of resource availability issues at the time of their admittance, might be enough to let a NN achieve usefulness in simulation, and perhaps outside of it. Unfortunately, it was beyond the scope of this work to attempt to interrogate the NN to learn how it was using this additional information.

Since our only way of detecting anomalous cases is an after-the-fact (anomalous cases are only known after a choice has been made which can seem anomalous) significant discrepancy between the algorithmically and selected patients we cannot at this time produce a system that performs on real-world data at more than an accuracy of 44.15%. However, if, as we were told, the anomalous cases do indeed represent outliers, we can report a system that performs at the much more useful 75.29%; much depends on whether the anomalous cases can indeed be considered outliers. Pursuing a more expressive dataset, and reevaluating our methods using it, is part of our future work.

Table 4.2

Accuracy and error metrics for the RNN

Data Category	R2	RMSE	MSE	MAE
Raw dataset	44.15%	0.231094	0.053404	0.162643
Anomalous cases removed	75.29%	0.122312	0.014960	0.081476
Anomalous cases flagged	84.97%	0.106931	0.011434	0.058311

Table 4.2 presents the performance of the neural network in the form of the averaged r2 (accuracy) scores, mean square errors, root mean square errors and mean absolute errors. Given the failure to perform well on the raw dataset these results are presented not as a

useful measure of what can be practically achieved on real-world data but to demonstrate the superior performance of the RNN when compared to the industry algorithm, especially if the anomalous cases are true to be disregarded. The last dataset, again, flags anomalous cases and returns them to the dataset. The reader should consider the results on the last dataset only theoretically achievable with a dataset more descriptive of the patients and the circumstances within the hospital, including a focus on available resources. Figure 4.3 shows the drop in the loss as the training progressed for each dataset variant.

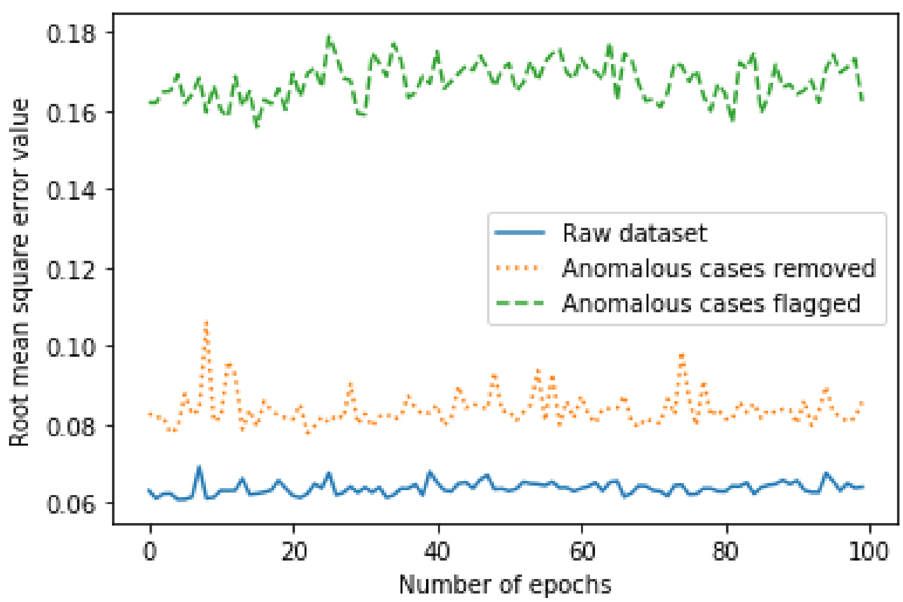

Figure 4.3

Root mean squared error per dataset variant across epochs

What are we to make of these results? The naive algorithm insufficiently models the real-world data (though this does not mean it's guiding rule is itself flawed, we do not comment on the relative superiority of its rule or whatever rules the real-world data may be the result of). The RNN performs well enough to be a step up as a part of a simulator but only if the problematically uncharacterized anomalous cases are eliminated, which may or may not be unrealistic. Though the confidential nature of healthcare data may impede such work we feel that it would be possible to achieve greater accuracy if more information about what we generically call anomalous cases were known, and not in the purely mathematical way we detect them now. Further, there is a wide gulf between a difference of 50 percentage points (the threshold between the calling probabilities of expected and actual we take to be indicative of an anomalous case) and a difference of 0. Characterizing that gap, especially when the difference rises beyond +- 10 percentage points, is likely necessary to improve accuracy further, likely in the form of more detailed data. We hope to continue our work on this dataset's more detailed decedents.

That the RNN performs as well as it does (a 2x boost to accuracy) when the anomalous cases are flagged is intriguing and suggests there is some rule yet to be recognized by us which could be of use even in the presence of the barest indication of a patient destined to be admitted anomalously.

CHAPTER 5

PREDICTING THE LIKELIHOOD OF PATIENTS DESERTION

The Emergency Departments' rate of service exceeds if there is an influx of patients, and then the patients are inserted into a queue and asked to wait given that their condition is not deemed urgent or emergent. Usually, ESI levels 4, 5, and a select number of ESI 3 patients fall into this category. The patient's position in the queue is dynamic and depends on the ESI score of all patients in the queue and how long each patient has waited for treatment. Those patients who receive treatment will be referred to as having "remained," and those that leave without being seen will be referred to as having "left" (or "LWOT," meaning "left without treatment").

To know, What is the probability that a given patient will leave having waited m minutes for treatment since arrival? And What is the wait time with the peak probability of leaving without treatment (LWOT)?

We must first answer two preliminary questions for each ESI: What is the distribution of wait times among the patients who elected to leave? And What is the distribution of wait times among the ones who remained?

After removing all incomplete records (which contains null values or missing values), we were left with approximately 95% of our original dataset, the distribution of which is shown in Table 5.1. In this work, we use the term "w-time" to refer to the time a patient waited before her/his visit was "resolved", either by being seen by a doctor, or by leaving the ED. w-time is not an attribute in the dataset but is easily derived from the provided attributes.

Table 5.1

Data-set broken down by patient type

ESI Level	Patients Remained	Patients left
1	975	0
2	19399	120
3	32173	1485
4	9698	517
5	614	54

.

When a patient leaves without treatment, the w-time is simply the time of departure minus the time of arrival. When a patient remains, the first step could be assigning the patient to a bed or having a doctor examine them. The first step of this process, whatever it is, counts as the patient's time of "admittance" and the w-time for a remainder is the admittance time minus the arrival time.

Figure 5.1 shows a visualization of the ESI level 3 LWOT subset with a line of best fit superimposed on it (how this line of best fit is produced is discussed in the later sections),

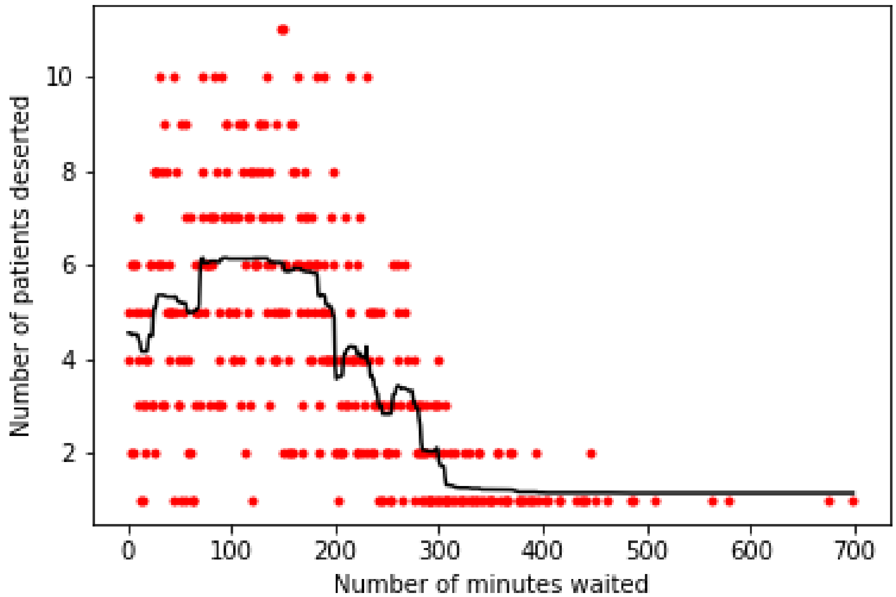

Figure 5.1

Patients left at each minute at ESI level 3

Table 5.2

Approximate peak leaving time per ESI level

ESI Level	Approximate peak leaving time
2	67 minutes
3	133 minutes
4	108 minutes
5	85 minutes

notice the peak at a w-time of approximately 133 minutes. The range around this peak contains the most frequently occurring w-times for LWOTs for ESI 3 patients. Each ESI level for the LWOT subset had such a peak, an approximate of which is shown in Table 5.2 except for ESI level 1. ESI level 1 patient are critical patients and no record in our dataset shows any of them leaving without treatment.

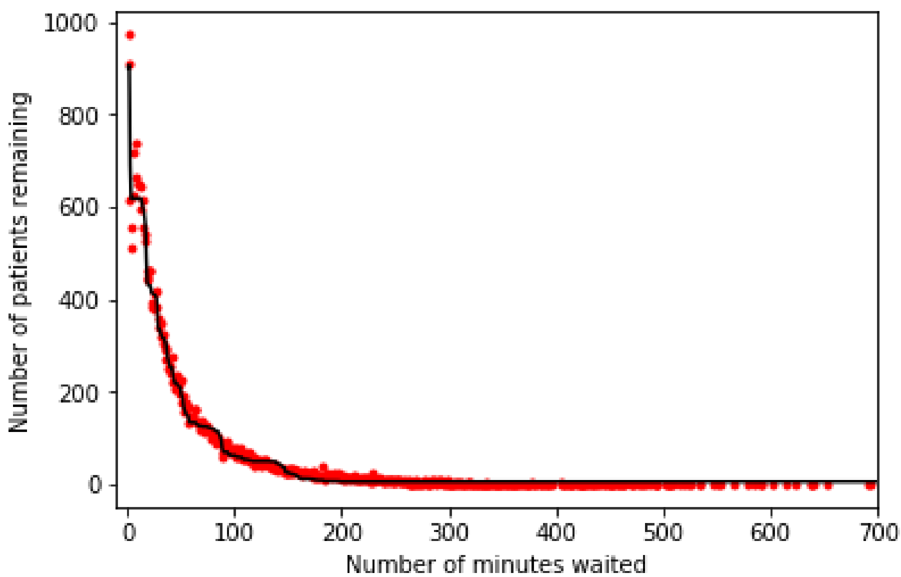

Figure 5.2

Patients remaining at each minute at ESI level 3

Figure 5.2 illustrates the ESI level 3 remainder subset. Once again, there is a salient value, in this case, the elbow point located at approximately 60 minutes. A vast majority of remainers at this level were seen with w-times at or below this point and thus this is likely to be the largest w-time any given ESI level 3 patient could be expected to have before

Table 5.3

Longest w-time likely per ESI level

ESI Level	Longest w-time likely
1	20 minutes
2	40 minutes
3	60 minutes
4	75 minutes
5	65 minutes

being seen. The elbows of this kind in the other ESI levels' remainer subsets are seen in Table 5.3.

5.1 Methodology

The Equation 5.1, seen below, we have calculated the percent likelihood of a patient leaving for all ESI/w-time pairs that exist in our dataset.

$$P(D_{L,W}) = \frac{d_{L,W}}{(d_{L,W} + r_{L,W})} \tag{5.1}$$

$P(D_{L,W})$ is the probability that a patient of ESI level L, with a w-time of W, will leave. $d_{L,W}$ is a count of observed patients of that ESI level and w-time that did leave. $r_{L,W}$ is a count of observed patients, with those same characteristics, which remained.

So for example, if, for a given w-time and ESI level, there were 7 LWOT and 77 remainers the probability calculated by the above formula would be 0.08, quite low. This operation is committed for each w-time seen in each of the ESI level subsets, excluding ESI level 1.

5.2 Results and Discussion

The above operations produce a derived subset for each ESI level. For each derived dataset, the independent variable is the w-time and the dependent (target) variable is the probability of a patient leaving. It is to these derived subsets that our regression analysis was applied. In general, and by way of an introduction for the uninitiated, regression analysis is a kind of statistical modeling where one or more independent variables are used to predict one dependent, target, variable.

Being so well studied, we were able to select a number of off-the-shelf implementations of regression analysis methods to evaluate for the best performance on our data. The Orange data mining toolkit [9], and the python library "Scikit-learn" [27] provided us with the implementations of the selected regression models, each of which is presented above with a brief description.

Each regression model was trained with 80% and tested with 20% of the derived data. The accuracy of each regression method is then averaged across the ESI level datasets, to which it was applied, to get the final, overall, the measure of their accuracy. Table 5.4 presents the accuracy of each method implemented with Scikit-learn, both on the individual derived ESI datasets and the aggregated values across all four ESI levels.

It can be seen that Scikit-learn's AdaBoost algorithm performed best overall, followed by Isotonic regression implementation. It is not completely surprising that AdaBoost performed better overall than anyone random forest or decision tree since it uses ensembles of machine learning algorithms like them to perform a more sophisticated analysis.

Table 5.4

Sklearn Implementations

Regression Algorithms	Accuracy (ESI 2)	Accuracy (ESI 3)	Accuracy (ESI 4)	Accuracy (ESI 5)	Overall Accuracy
Linear Regression	0.681	0.605	0.560	0.142	0.497
Decision Tree Regression(max-depth:1)	0.773	0.904	0.787	0.04	0.626
Decision Tree Regression(max-depth:3)	0.949	0.971	0.925	0.20	0.761
Decision Tree Regression(max-depth:5)	0.950	0.986	0.949	0.19	0.768
Random Forest Regression(max-depth:1)	0.859	0.905	0.800	0.20	0.691
Random Forest Regression(max-depth:3)	**0.958**	0.977	0.947	0.22	**0.775**
Random Forest Regression(max-depth:5)	0.951	**0.988**	**0.951**	0.16	0.762
K Nearest Neighbours(n_neighbours:50)	0.956	0.986	0.937	0.176	0.763
SVR	0.788	0.730	0.712	0.09	0.58
Isotonic Regression	**0.958**	0.984	**0.951**	0.255	**0.787**
AdaBoost Regression	0.957	0.972	0.940	**0.285**	**0.7885**

Table 5.5

Orange Tool Analysis

Regression Algorithms	Accuracy (ESI 2)	Accuracy (ESI 3)	Accuracy (ESI 4)	Accuracy (ESI 5)	Overall Accuracy
Linear Regression	0.749	0.561	0.534	0.077	0.480
Decision Tree Regression(max-depth:3)	0.930	0.980	0.935	-0.066	0.696
Random Forest Regression(max-depth:3)	0.936	**0.984**	**0.943**	0.104	**0.741**
K Nearest Neighbours(n_neighbours:50)	**0.951**	0.979	0.915	0.188	**0.758**
Neural Network	0.925	0.964	0.924	**0.281**	**0.773**
AdaBoost Regression	0.899	0.975	0.914	-0.357	0.607

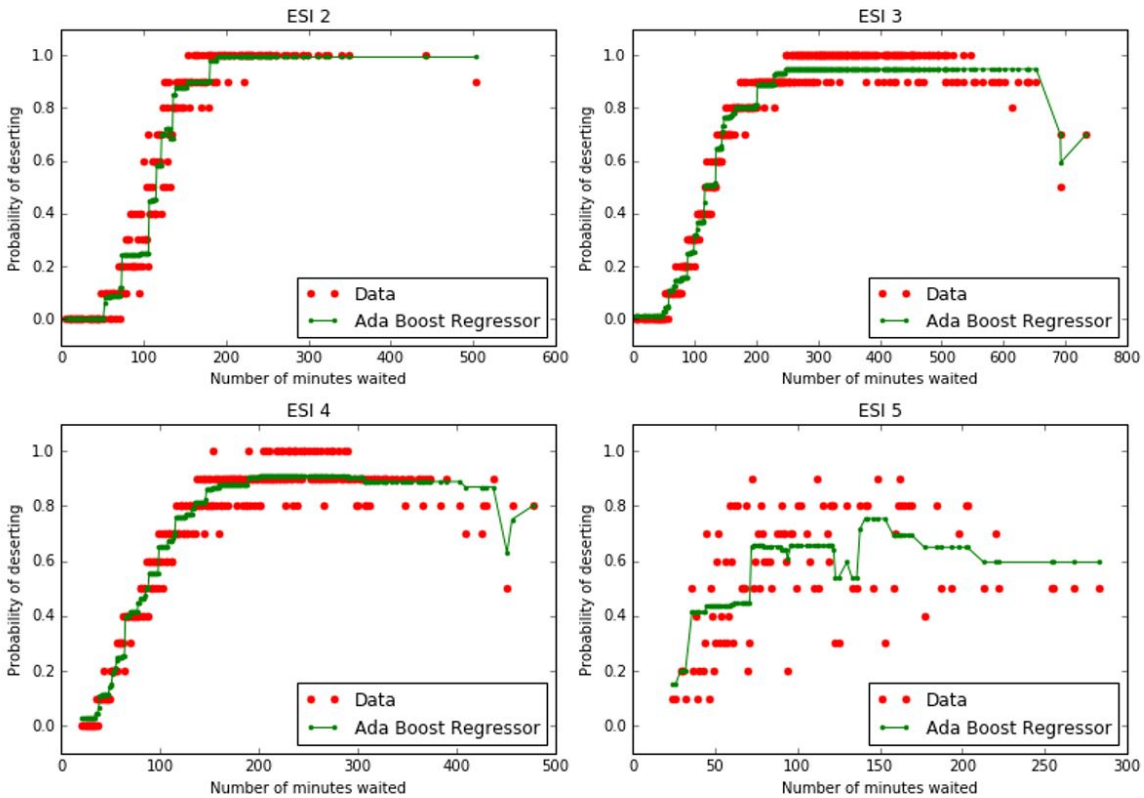

Figure 5.3

Adaboost Regression

Orange's implementations give us a different ranking for similar parameter settings. Here the neural network placed first, followed by KNN with k = 50 as the second best. Orange's results are shown in Table 5.5.

The stochastic nature of the generation of the training and the test sets is undoubtedly responsible for some variation of performance among analogs regression models when comparing the Orange and SciKit-learn results, but we suspect there is more to the variation than this alone.

A KNN implementation in each suite had k = 50 and the AdaBoost instances both had 100 estimators so at least these hyperparameters cannot be the source of the variation. Investigating the reasons for these variations would require more study and, possibly, should be done by independently developing the above algorithms outside the aforementioned packages. As important as they may be, these tasks do not fit in the scope of this paper.

Without a deeper knowledge of the architecture of the ambiguously titled "neural network", more thorough knowledge of the sub-learners used by both suites' AdaBoost implementations, and an execution of a k-fold cross-validation for all experiments, our only response to the ambiguity as to which regression models is superior is to appeal to raw numbers: Scikit-learn's AdaBoost has the highest overall performance of any fielded implementation. Figure 5.3 illustrates the regression curves produced by Scikit-learn's implementation of AdaBoost for each ESI level.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this work, we have observed the three case studies which are very common at the emergency departments. In the emergency department, patient's incoming rate are significantly affected by the weather of the day, an hour of the day, seasonal and public holidays. The calling procedure of the hospital from the waiting room to the treatment can be mimicked but it can be more accurate if we could get more data on the anomalies like the reason behind the case. As of now, the model is not accurate enough to replicate the current scenario with the produced data, but we can improve the accuracy by getting more information on the anomalies. The patients can be stopped by leaving the hospital without being seen by the doctor using the likelihood prediction. Suppose the patient leaving probability crosses 80%, the staff can prioritize the patient from the queue and then it reduces the patient leaving the hospital from which hospitals and patients benefit.

According to the future work, Regarding the prediction of incoming rates, the plan is to find the data regarding the virus index in the region of the reputed hospital. Because the virus index will also play a crucial role in the prediction of the incoming rates. Regarding Patient admittance predictors, our current plan is to consult subject matter experts, first to attempt to characterize what we have been calling anomalous cases and then to see if we can be given an even mildly augmented dataset marking them (assuming one is producible). Another possible step forward is to make a patient's calling-probability calculated in a way so that it is considered dependent on the other occupants of the waiting room at that time. Outside of ED queue work, we can construct further components to be added to the simulations used in research. One such project concerns the prediction of rates of patient arrival to the ED based on the current weather, the season of the year, and any recent public holidays. This work could facilitate better scheduling of medical staff and resources to cope with rises and falls in demand. Regarding LWBS, the queue in an ED could be dynamically reordered whenever a new patient arrives, and as the waiting time of each patient increases, to minimize the net probability of the LWOT ratio.

REFERENCES

[1] *Innovations in smart cities applications edition 3 . The proceedings of*, Springer nature, 2020.

[2] G. Aurelien, *Hands-on machine learning with scikit-learn tensorflow: concepts, tools, and techniques to build intelligent systems*, OReilly Media, Inc., 2017.

[3] M. Badri and J. Hollingsworth, "A Simulation Model for Scheduling in the Emergency Room," *International Journal of Operations Production Management*, vol. 13, 12 1993, pp. 13–24.

[4] J. Billings, T. Georghiou, I. Blunt, and M. Bardsley, "Choosing a model to predict hospital admission: an observational study of new variants of predictive models for case finding," *BMJ Open*, vol. 3, no. 8, 2013.

[5] Blasak, Armel, Starks, and Hayduk, "The use of simulation to evaluate hospital operations between the emergency department and a medical telemetry unit," *Proceedings of the 2003 Winter Simulation Conference, 2003.*, Dec 2003, vol. 2, pp. 1887–1893 vol.2.

[6] J. Boyle, M. Jessup, J. Crilly, D. Green, J. Lind, M. Wallis, P. Miller, and G. Fitzgerald, "Predicting emergency department admissions," *Emergency medicine journal : EMJ*, vol. 29, 06 2011, pp. 358–65.

[7] M. Centeno, R. Giachetti, R. Linn, and A. Ismail, "Emergency departments II: a simulation-ilp based tool for scheduling ER staff.," 01 2003, pp. 1930–1938.

[8] A. J. Clarey and M. W. Cooke, "Patients who leave emergency departments without being seen: literature review and English data analysis," *Emergency Medicine Journal*, vol. 29, no. 8, 2012, pp. 617–621.

[9] J. Demšar, T. Curk, A. Erjavec, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, and et al., "Orange: Data Mining Toolbox in Python,", Jan 1970.

[10] G. W. Evans, T. B. Gor, and E. Unger, "A simulation model for evaluating personnel schedules in a hospital emergency department," *Winter Simulation Conference*, 1996.

[11] S. Fomundam and J. Herrmann, "A Survey of Queuing Theory Applications in Healthcare," 09 2007.

[12] B. Graham, R. Bond, M. Quinn, and M. Mulvenna, "Using Data Mining to Predict Hospital Admissions From the Emergency Department," *IEEE Access*, vol. 6, 02 2018, pp. 1–1.

[13] M. M. Gunal and M. Pidd, "Understanding Accident and Emergency Department Performance using Simulation," 12 2006.

[14] D. Hallac, S. Vare, S. Boyd, and J. Leskovec, "Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data,", 2017.

[15] C. Harrell and R. Price, "Healthcare simulation modeling and optimization using MedModel," 12 2000, vol. 1, pp. 203–207.

[16] A. M. Hay, E. C. Valentin, and R. A. Bijlsma, "Modeling Emergency Care in Hospitals: A Paradox - The Patient Should not Drive the Process," *Proceedings of the 2006 Winter Simulation Conference*, Dec 2006, pp. 439–445.

[17] W. S. Hong, A. D. Haimovich, and R. A. Taylor, "Predicting hospital admission at emergency department triage using machine learning," *PLOS ONE*, vol. 13, no. 7, 07 2018, pp. 1–13.

[18] A. Komashie and Ali Mousavi, "Modeling emergency departments using discrete event simulation techniques," *Proceedings of the Winter Simulation Conference, 2005.*, Dec 2005, pp. 5 pp.–.

[19] Krishni, "A Beginners Guide to Random Forest Regression,", Jun 2019.

[20] A. P. Kumar and R. Kapur, "Discrete Simulation Application-Scheduling Staff for the Emergency Room," *Proceedings of the 21st Conference on Winter Simulation*, New York, NY, USA, 1989, WSC '89, p. 1112–1120, Association for Computing Machinery.

[21] M. A. LaMantia, T. F. Platts-Mills, K. Biese, C. Khandelwal, C. Forbach, C. B. Cairns, J. Busby-Whitehead, and J. S. Kizer, "Predicting hospital admission and returns to the emergency department for elderly patients,", Mar 2010.

[22] D. R. Li, J. J. Brennan, A. A. Kreshak, E. M. Castillo, and G. M. Vilke, "Patients Who Leave the Emergency Department Without Being Seen and Their Follow-Up Behavior: A Retrospective Descriptive Analysis," *The Journal of Emergency Medicine*, vol. 57, no. 1, 2019, pp. 106 – 113.

[23] S. Narkhede, "TowardsDataScience: Understanding Logistic Regression,", May 2019.

[24] E. A. Z. Noughabi, B. Raahemi, A. Albadvi, and B. H. Far, *Handbook of research on data science for effective healthcare practice and administration*, Medical Information Science Reference, 2017.

[25] F. M. Pajouh and M. Kamath, “”Applications of Queueing Models in Hospitals” by Foad Mahdavi Pajouh and Manjunath Kamath,”, 2010.

[26] S. Paul, M. Reddy, and C. Deflitch, “A systematic review of simulation studies investigating emergency department overcrowding,” *Simulation*, vol. 86, no. 8-9, 8 2010, pp. 559–571.

[27] Pedregosa, Fabian, Varoquaux, Gaël, Alexandre, Michel, Vincent, Thirion, Bertrand, Grisel, and et al., “Scikit-learn: Machine Learning in Python,”, Jan 1970.

[28] K. Roche and J. Cochran, “Improving patient safety by maximizing fast-track benefits in the emergency department - A queuing network approach,” *IIE Annual Conference and Expo 2007 - Industrial Engineering’s Critical Role in a Flat World - Conference Proceedings*, 01 2007, pp. 619–624.

[29] B. H. Rowe, P. Channan, M. Bullard, S. Blitz, L. D. Saunders, R. J. Rosychuk, H. Lari, W. R. Craig, and B. R. Holroyd, “Characteristics of Patients Who Leave Emergency Departments without Being Seen,” *Academic Emergency Medicine*, vol. 13, no. 8, 2006, pp. 848–852.

[30] D. Sinreich and Y. Marmor, “A simple and intuitive simulation tool for analyzing emergency department operations,” 01 2005, vol. 2, pp. 1994 – 2002 vol.2.

[31] S. Takakuwa and H. Shiozaki, “Functional analysis for operating emergency department of a general hospital,” *Proceedings of the 2004 Winter Simulation Conference, 2004.*, Dec 2004, vol. 2, pp. 2003–2011 vol.2.

[32] K. W. Tan, C. Wang, and H. C. Lau, “Improving patient flow in emergency department through dynamic priority queue,” *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2012, pp. 125–130.

[33] V. Tucci, S. Ahmed, D. Hoyer, S. Greene, and N. Moukaddam, “Stabilizing Intentional Overdoses in Freestanding Emergency Departments: A Good Idea?,” *Journal of General and Emergency Medicine*, vol. 2, 01 2017, p. 007.