UNIVERSITY OF KASDI MERBAH OUARGLA
Faculty of New Information Technologies and Communication
Department of Electronic and Telecommunication

ACADEMIC MASTER DISSERTATION

Domain: Science and Technology

Field: Automation

Specialty: Automation and system

Topic

Achieving Zero Waiting Time in Public Institutions with a Focus on Postal Offices Using Smart Technology ('Zero Wait' Application)

Presented by:
Ms.Bennai Fairouz Rayane

Publicly defended on : 25 June 2024

The jury:

ROUABAH Boubakeur	UKM OUARGLA	Supervisor
Kafi Mohamed Redouane	UKM OUARGLA	Assistant Supervisor
BENARABI Bilal	UKM OUARGLA	President
ACHBI Mohamed Said	UKM OUARGLA	Examiner

University year: 2023/2024

Acknowledgements

I would like to express my deepest gratitude and appreciation, first and foremost, to ALLAH for granting me the will, strength, health, and patience to complete this work.

I extend my warmest thanks to my supervisor for his invaluable guidance, advice, continued support, trust, and patience, as well as his insightful remarks and significant contributions throughout the course of this work.

I would like to thank my dear parents for their unwavering support and constant encouragement. They spared no effort to help me successfully complete my studies.

I also wish to thank all the jury members who have honored me by accepting to evaluate this modest work, for their time, effort, and appreciation.

Lastly, I express my sincere gratitude and appreciation to all my classmates and colleagues whom I have met throughout my academic journey and to everyone who assisted and supported me during my university studies. Your support and encouragement have greatly contributed to my success and progress.

وكان فضل الله عليك عظيما

الإهداء

من قال "أنا لها" نالها.

لم تكن الرحلة قصيرة، ولم يكن الطريق محفوفًا بالتسهيلات. لقد كانت مليئة بالتحديات، ولكن بفضل الله حققت الحلم.

الحمد لله حبًا وشكرًا وامتنانًا، الذي بفضله ها أنا اليوم أرى حلمًا طال انتظاره وقد أصبح واقعًا أفتخر به.

إلى ملاكي الطاهر، وقوتي بعد الله، داعمتي الأولى والأبدية، أمي العزيزة، أهديك هذا الإنجاز الذي لولا تضحياتك لما كان له وجود. ممتنة لأن الله اصطفاك لي من بين البشر أماً يا خير سند وعوض.

إلى من دعمني بلا حدود وأعطاني بلا مقابل، أبي الحبيب، شكرًا لك على كل ما قدمته لي.

إلى زهراتي، أخواتي العزيزات، اللواتي آمنّ بقدراتي وشجعنني في كل خطوة، شكرًا لكنّ.

ولكل من أعطاني يد العون من قريب أو بعيد وساعدني في إنجاز هذه المذكرة، أقدم لكم جميعًا هذا الإهداء تعبيرًا عن امتناني العميق.

Abstract

Companies rely on queues as a means of managing the flow of services over long periods. Despite the introduction of queue numbers and time-based appointments to enhance service, the problem of long queues remains unresolved. Within this startup, I developed a product centered around the concept of implementing queue management technology that allows users to wait remotely via a smartphone application. The new product enables customers to utilize their waiting time more effectively and reduces the gap between service request and delivery. The research focuses on developing new technology for remote queuing and updating service characteristics. The study provides a product development strategy and analyzes results related to user experiences.

This research addresses the development of the "Zero Wait" system for managing queues in public service centers using artificial intelligence technologies to predict waiting times. The system aims to improve customer experience and reduce waiting times by proposing a new approach that leverages the powerful machine learning capabilities of AI to estimate waiting times. We tested how machine learning can be used to predict the waiting time for people in queues, starting with an industrial data set of wait times in banks. By training a neural network, the results showed how an online postal service queue management system could be revolutionary.

This study significantly contributes to the growing topic of digital transformation in the service sector and provides valuable insights for companies looking to enhance customer experiences while increasing operational efficiency.

Keywords: Postal Services, Artificial Intelligence, Smartphone Application, Waiting Queues, Wait Time Prediction, Remote Waiting, Public Service Centers, Queue Management

ملخص

تعتمد الشركات على الطوابير كوسيلة لإدارة تدفق الخدمات لفترات طويلة. على الرغم من تقديم أرقام الطوابير والمواعيد المحددة بالوقت لتحسين الخدمة، إلا أن مشكلة الطوابير الطويلة لم تحل بعد. في إطار هذه الشركة الناشئة، قمت ببناء منتج يتمحور حول فكرة إدخال تقنية إدارة الطوابير التي تسمح للمستخدمين بالانتظار عن بعد عبر تطبيق الهاتف الذكي. يتيح المنتج الجديد للعملاء الاستفادة من وقت الانتظار بشكل أفضل، وتقليل الفجوة بين طلب الخدمة وتقديمها. يركز البحث على تطوير تكنولوجيا جديدة لإدخال الطوابير عن بُعد وتحديث خصائص الخدمة. تقدم الدراسة استراتيجية تطوير المنتج وتحليل النتائج المتعلقة بتجارب الاستخدام.

يتناول هذا البحث تطوير نظام "Zero Wait" لإدارة طوابير الانتظار في مراكز الخدمة العامة باستخدام تقنيات الذكاء الاصطناعي للتنبؤ بأوقات الانتظار. يهدف النظام إلى تحسين تجربة العملاء وتقليل أوقات الانتظار من خلال اقتراح نهج جديد يستفيد من قدرات التعلم الآلي القوية للذكاء الاصطناعي في تقدير وقت الانتظار. تم اختبار كيف يمكن استخدام التعلم الآلي للتنبؤ بوقت انتظار الأشخاص الذين يصطفون في الطوابير، وبدأنا باستخدام مجموعة بيانات صناعية للانتظار في البنوك. من خلال تدريب شبكة عصبية، أظهرت النتائج كيف يمكن لنظام إدارة طوابير الخدمات البريدية عبر الإنترنت أن يكون ثورياً.

تساهم هذه الدراسة بشكل كبير في موضوع التحول الرقمي المتوسع في قطاع الخدمات، وتقدم معلومات مفيدة للشركات التي تتطلع إلى تعزيز تجارب العملاء مع زيادة الفعالية التشغيلية.

الكلمات الرئيسية: لخدمات البريدية ,الذكاء الاصطناعي ,تطبيق الهاتف الذكي ,طوابير الانتظار ,التنبؤ بوقت الانتظار , الانتظار عن بعد , مراكز الخدمة العامة, إدارة قائمة الانتظار

Résumé

Les entreprises s'appuient sur les files d'attente comme moyen de gérer le flux de services sur de longues périodes. Bien que des numéros de file d'attente et des rendez-vous basés sur l'heure aient été introduits pour améliorer le service, le problème des longues files d'attente reste non résolu. Dans le cadre de cette start-up, j'ai développé un produit axé sur l'introduction d'une technologie de gestion des files d'attente permettant aux utilisateurs d'attendre à distance via une application pour smartphone. Le nouveau produit permet aux clients d'utiliser leur temps d'attente de manière plus efficace et de réduire l'écart entre la demande de service et sa livraison. La recherche se concentre sur le développement d'une nouvelle technologie pour introduire les files d'attente à distance et mettre à jour les caractéristiques du service. L'étude présente une stratégie de développement de produit et analyse les résultats liés aux expériences des utilisateurs.

Cette recherche traite du développement du système "Zero Wait" pour la gestion des files d'attente dans les centres de service public en utilisant des technologies d'intelligence artificielle pour prédire les temps d'attente. Le système vise à améliorer l'expérience des clients et à réduire les temps d'attente en proposant une nouvelle approche qui tire parti des capacités d'apprentissage automatique de l'intelligence artificielle pour estimer le temps d'attente. Nous avons testé comment l'apprentissage automatique peut être utilisé pour prédire le temps d'attente des personnes faisant la queue, en commençant par utiliser un ensemble de données industrielles d'attente dans les banques. En entraînant un réseau neuronal, les résultats ont montré comment un système de gestion des files d'attente des services postaux en ligne pourrait être révolutionnaire.

Cette étude contribue de manière significative au sujet de la transformation numérique croissante dans le secteur des services et fournit des informations utiles aux entreprises cherchant à améliorer l'expérience client tout en augmentant l'efficacité opérationnelle.

Mots Clés : Services postaux , Intelligence artificielle, Application mobile ,Files d'attente, Prédiction du temps d'attente , Attente à distance, Centres de service public, Gestion des files d'attente

Table of Contents

List of Figures

List of tables

List of abbreviation

SLA	Service Level Agreement
FCFS	First Come First Serve
RSS	Random Selection for Service
PS	Priority Service
SPF	Shortest Processed First
QMS	Queue Management System
AI	Artificial Intelligence
ML	machine learning
DL	deep learning
NLP	natural language processing
RL	reinforcement learning
SVM	support vector machines
RF	random forests
LSTM	Long Short-Term Memory
CNN	convolutional neural networks
ARIMA	AutoRegressive Integrated Moving Average
GBM	Gradient Boosting Machines
EHR	Electronic health records
MES	manufacturing execution systems
RFE	Recursive Feature Elimination
ANN	artificial neural networks

Introduction

General introduction:

Why was this research conducted?

We frequently encounter circumstances in our daily lives where we must wait in line. Our precious time, which is essential in the fast-paced world of today, might be greatly consumed by this waiting period. A way to estimate the length of waiting queues before entering them is required in order to maximize our use of time. Possessing this kind of insight would help people manage their time more effectively and may even affect their decision to line up or not. People wait in lines for an average of five hours and thirty-five minutes every month, according to a recent survey[1]. That's 67 hours a year, or almost three days, of waiting. In an environment where every second matters, effective queue management becomes essential.

The inspiration for this research project emerged during Ramadan 2022, when a routine trip to the post office to withdraw money for Eid shopping turned into a frustrating ordeal. Confronted with unexpectedly long queues, I endured over three hours of waiting, only to find the markets closed upon my turn. This experience prompted me to contemplate the need for a solution to alleviate such inefficiencies. Thus, the research delves into the development of an application aimed at providing real-time queue status updates, empowering individuals to better manage their time effectively.

Statement of the Problem

The problem arises when you visit a post office, take your waiting ticket, and find that your wait number is far away. It can take hours for your turn to arrive, and if you leave and come back later, you might find that your turn has already passed.

Scope of the Project

The "Zero Wait" includes a mobile application and a queue management system provided in post offices. The application allows users to track their positions in the queue in real-time by connecting to the queue number machine in postal institutions. Users can use the application on their smartphones to monitor their positions, estimate expected wait times, and receive notifications before their turn arrives. The system is implemented in post offices to provide updated real-time information about queues and alleviate congestion.

Significance of the Study

The "Zero Wait" project aims to enhance user experience in post offices by linking the queue management system to a smartphone application. Users can accurately track their position in real-time, avoiding long waits and scheduling their visits conveniently. The project is innovative as it offers a simple and effective solution to the issue of long queues in post offices, providing a seamless and comfortable experience. It integrates all post offices in Algeria, offering precise information about each branch and estimated wait times, thus improving service quality and increasing user engagement with postal services.

Background

Waiting is a common challenge faced by everyone in their daily lives. Who among us hasn't felt boredom and frustration while standing in long queues at post offices?

Those moments spent waiting can be incredibly valuable, providing an opportunity for reflection and planning. However, they can also be a source of frustration and wasted time. Waiting isn't just a topic of conversation; it's a real-life experience that everyone encounters in various forms.In post offices, long waiting times lead to delayed appointments and disruption of daily schedules, resulting in customer confusion and frustration. Seconds turn into minutes, and minutes into hours, while the wait continues, gradually wearing down people's hope and patience.

So let's take a deep look at these challenges and frustrations associated with waiting at post offices, and how we can find creative and effective solutions to improve this complex experience. That's the goal of our "Zero Wait" application".

Chapter I: Queue System

Introduction

These days, with the population and technology developing, managing a company has gotten more and more fascinating. Designing a service system, which involves choosing the number of servers, the service policy, and many other factors, is one of the difficult tasks. Machines, materials, equipment, and human resources are among the virtually always scarce resources that companies have at their disposal. A number of factors, including resource capacity, layout, service, and processing policy decisions, can affect waiting times [2]. However, the implications of the consumer inflow could lead to issues like congestion; waiting times can quickly become uncomfortable and have a negative impact on the overall service system in terms of psychological, social, financial, and physical costs. As a result, wait times and the metrics associated with them are now crucial components in evaluating how well service systems operate. Many research examine the impact of waiting durations, particularly the behavioral implications. The fact that waiting times can impact both system owners and customers is a remarkable characteristic. Consequently, a number of factors, including customer experience and loyalty [3, 4] and customer sanctification [5, 6, 7], may be impacted by lengthy wait times and long lines.

I.1. Overview of queue systems and their significance in various domains:

I.1.1. Theoretical Overview of Queuing Theory

I.1.1.1. Definition of Queuing Theory

The goal of queuing theory is to better understand how queues function and how to make them more efficient.[8]. A single task runs on a computer system, enters, makes use of specific resources, and then exits. Additionally, as there are no lines, no There will likely be a delay. Queuing theory is relevant whenever there are queues. Actually, a queue[9] is the central component of any computer system. The CPU uses a time-sharing scheduler to serve.Thread queues get memory chunks from memory banks. Refer to Fig 1.1. The servers are shown as circles in Figure 1.1., queues as collections of rectangles, and the routing network as arrows. In the study of queue networks, one frequently aims to obtain the equilibrium distribution of the network, even if the transient state is crucial in many applications.

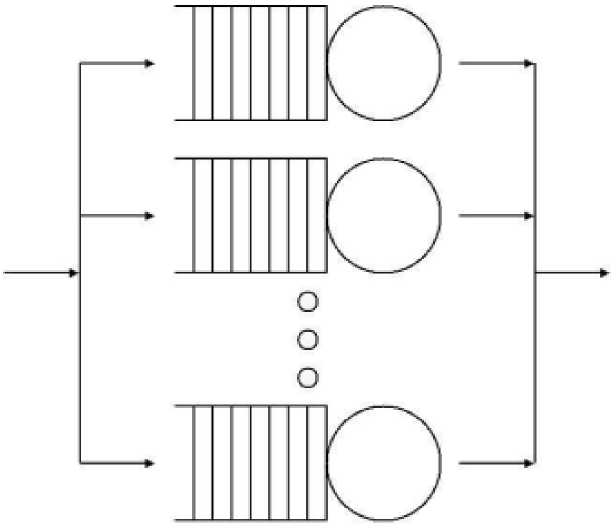

Figure I. 1: Schematic Representation of the Queueing System - Servers, Queues, and Routing

I.1.1.2. Function and Purpose of Queuing Theory

The goal of a queuing theorist is twofold. The first is forecasting system performance. Estimating the mean delay, the variability of the delay, or the probability that the delay would surpass a specific (SLA) are typically required for this. It can also be used to predict how many jobs will be waiting in line, how many servers are typically used (for example, how much electricity is needed), or any other similar metric. [11]. While prediction is important, improving the system's architecture to increase performance is even more important.

Typically, capacity planning is used to determine which additional resources to purchase in order to fulfill delay targets (e.g., is it better to add RAM or purchase a faster disk or CPU or an additional slow disk). However, many times, performance can be increased simply by introducing a more intelligent scheduling policy or a better routing method to reduce latency, without requiring the purchase of any additional resources.

I.1.1.3. Scheduling Policies of the Queuing Theory

FCFS: This algorithm serves customers in order of arrival, where a customer goes to the end of the queue when the service is busy. It is the most visibly fair service provision, as all customers are equal (Uddin et al., 2016; Ngorsed and Suesaowaluk 2016).

RSS*:* This algorithm serves customers in the queue at random,where each customer in the queue has equal probability of being selected for service. Thisalgorithm does not put into account the arrival time of customer, which means last customer that arrives in the service system can be served next (Willig 1999; Ahmed et al., 2011).

PS: This algorithm serves customers based on priority, the customer with the highest priority is served first (Uddin et al., 2016; Ngorsed & Suesaowaluk (2016)).

SPF*:* this algorithm chooses a customer in the queue with lesser time to serve first. It is assumed that the algorithm has the knowledge of the service times in advance (Uddin et al., 2016; Willig, 1999)

I.1.1.4. Overview of the Three Main Queuing Models: M/M/1, M/G/1, M/M/k

A Poisson process controls an M/M/1 queue functions in a single-server system where task service timing follows an exponential distribution and arrivals. Refer to Fig 1.2.

Figure I. 2: Illustration of an M/M/1 Queue in a Single Server System [12]

A queue model with a single server, a general distribution for service times, and Poisson-modulated arrivals is known as an M/G/1 queue (see Fig1.3). A common misconception is that the M/G/1 queue is an expansion of the M/M/1 queue.

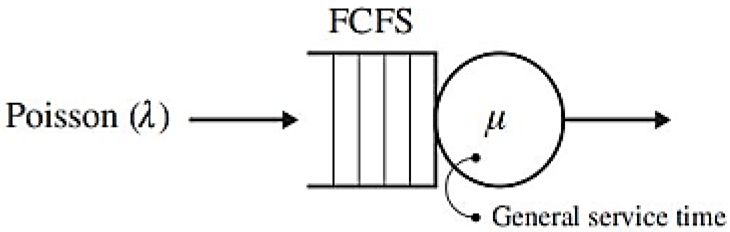

Figure I. 3: Poisson-Modulated General Queue [13]

A "server farm" is introduced because, in today's high-volume world, almost no websites, computing facilities, or call centers have just one server. A server farm is a collection of servers that work together to handle incoming requests. A random server may receive each request, enabling the servers to cooperatively manage the incoming workflow [14]. One queuing theory that works in a multi-server system is the M/M/k queue. Figure 1.4 shows the M/M/k queue.

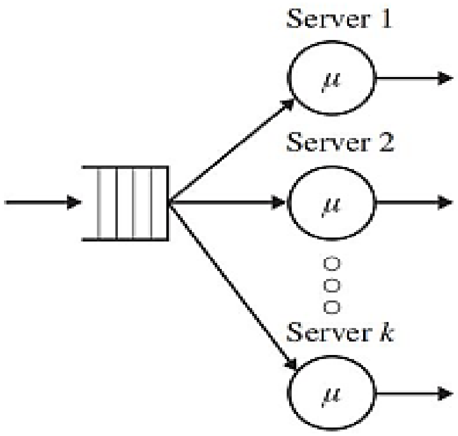

Figure I. 4: Representation of an M/M/k Queue in a Multi-Server System [15]

I.1.2. Kendall's Notation

The common way to describe queues is using Kendall's notation, which is the standard system used to classify queuing nodes and was proposed by David George Kendall in 1953. A/S/c stands for the time interval between arrivals to the queue, S for the time it takes to complete service, and c for the number of servers.

Below are two figures showing tables of the various symbols used in Kendall's notation [16]:

Symbol	Name	Description
M	Markovian/memoryless	Poisson process (or random) arrival process
M^X	Batch Markov	Poisson process with a random variable X for the number of arrivals at one time
MAP	Markovian arrival process	Generalisation of the Poisson process
BMAP	Batch Markovian arrival process	Generalisation of the MAP with multiple arrivals
MMPP	Markov modulated Poisson process	Poisson process where arrivals are in "clusters"
D	Degenerate distribution	A deterministic or fixed inter-arrival time
E_k	Erlang distribution	An Erlang distribution with k as the shape parameter
G	General distribution	-
PH	Phase-type distribution	Some of the above distributions are special cases of the phase-type, often used in place of a general distribution

Figure I. 5: Arrival process codes.

Symbol	Name	Description
M	Markovian/memoryless	Exponential service time
M^Y	Bulk Markov	Exponential service time with a random variable Y for the size of the batch of entities serviced at one time
D	Degenerate distribution	A deterministic or fixed service time
E_k	Erlang distribution	An Erlang distribution with k as the shape parameter
G	General distribution	-
PH	Phase-type distribution	Some of the above distributions are special cases of the phase-type, often used in place of a general distribution
MMPP	Markov modulated Poisson process	Exponential service time distributions, where the rate parameter is controlled by a Markov chain

Figure I. 6: Service time distribution codes.

I.1.3. Related Work

There are a number of methods that aim to forecast wait times. Among them are those that use a mathematical model based only on queueing theory, in which data is first gathered and then fed into a queueing prototype. There is also an approach that counts queues in person and does computations. After then, the outcomes are shown. In conclusion, a popular strategy in the restaurant business is to create an online virtual queue that customers may join to find out how long it will take to move up in line.

My entire solution will be impacted by these comparable applications' advantages and disadvantages.

I.1.3.1. Mathematical-based approach

Segovia, Patel, and Lonneman's "Restaurant Wait Time Estimation Report" was one of the earliest such applications to be reviewed [17]. The authors' queueing system was given data from a nearby sit-down restaurant for this study. prototype. An M/M/c queue served as the basis for the queueing model, in which the arrival rate (λ) indicated the number of customers served per hour and the service rate (μ) indicated the average time spent serving each client. A single line handled by several servers is represented by the M/M/c queue (c is bigger than 1).

Data structures representing the menu, servers, parties, tables, and waiting list must be created for the simulation model. Next, two ways were developed: one to start a night at the restaurant and another to walk through each step of the evening's activities. The second technique estimated the arrival time for parties of arbitrary sizes, calculated the total amount of time the party waited in line, and predicted what the party would order.

This approach's primary strength is its rigorous use of queueing theory. Because of the model's ability to provide accurate predictions, the simulation produced an overall margin error of +/- 2 minutes [17]. Nevertheless, the queueing models may quickly get stale and may produce inaccurate forecasts because the data was only gathered once rather than continuously and rollingly. The fact that there is no real-time data and all of the data is completely simulated is another area where this approach fails. The writers attempted to replicate a real restaurant using a piece of software called Arena.

The Waiting Game: Fast-Food Queuing Theory, a project by AetherWorks employee Shannon Cody, is another that employs this strategy [18]. Motivated by her regular lunch dates, Cody aimed to increase the frequency of her visits and reduce the amount of time she spent in line [18]. In contrast to Segovia, Patel, and Lonneman's analysis, Cody looks at a number of eateries with various queuing practices. These include numerous servers in several queues, single servers with multiple queues, and single servers with a single queue. Their selection of queueing models is where the two are comparable. To mimic their own queues, both make use of an M/M/c queue. Cody has tested establishments such as Starbucks, Chipotle, and Subway.

The author calculates the amount of time spent at each station by breaking down each one at each restaurant. For instance, Cody claims that there are five stations at Chipotle, and each one requires a distinct length of time. Cody bases her queueing model on the assumption that a customer will arrive every 200 seconds. One of the primary flaws in her methodology is this assumption, which the author herself acknowledges as being "admittedly unrealistic" [18]. It seems that the author witnessed and recorded the service rate in the actual eateries. Overall, the precision of the predictions made by mathematically based assessments is their strongest point. The authors were able to fine-tune their designs to assure this high degree of accuracy since the projects could concentrate on a certain number of eateries. Still, there are disadvantages to this emphasis. Because there aren't many eateries were chosen, the models developed don't offer a "one-size-fits-all" answer. The primary flaw with the two projects listed above is that data is only gathered once, prior to the model being run. This may result in out-of-date models, which may cause inaccurate predictions—as was already discussed.

I.1.3.2. Observation-based approach

You may view the observation-based method that will be covered in this article at https://queue-times.com. This website gathers data from native theme park apps into one location, where it is then shown to users. These parks calculate the length of time guests will wait in line by of people in a line, calculate the time it takes for a ride vehicle to fill, complete the ride, and then get out of line. They also take into account the maximum number of passengers that a vehicle may hold. Additionally, the website shows historical data, allowing visitors to view the wait times on particular days in the past.

This method refreshes sporadically throughout time, based on the park's timetable. This makes the strategy very effective because it never goes out of style. Instantaneous, current results are available to users whenever they need them. Our present goal is not met by this strategy because of how long it takes to update wait periods in lines. Recounting the entire queue is necessary for an update, and this can be a time-consuming task as the list grows longer. Human error may also occur when counting the queue, which could result in inaccurate estimates of the queueing time.

I.1.3.3. Virtual queues

Restaurants are the primary users of virtual lines. Several well-liked options are Nowait [19] and QLess [20]. Customers can join a virtual queue that is created by these systems. Every user in this queue is given a number. You are nearer the front of the line the lower the number. Both work via text messaging, alerting consumers via SMS when they join the queue and when they are the next person in line. The hosts of the restaurant advance the virtual line once someone has been served.

This increases the chance of human error in the system, which could lead to inaccurate wait times.

This strategy has obvious advantages. As soon as they enter the queue, users can see how long they will have to wait. It also does away with the necessity of standing in line.But the issue is that the hosts of the restaurants are the ones who have to move the line. proceed. Employees can get too worked up to remember to shift the queue every time, especially on busy evenings. The fact that consumers may only view the wait time after joining the queue is another problem I discovered with this strategy. A user may disrupt the entire queueing process by quitting the queue, or worse, by forgetting to leave the queue, if they are unwilling to wait that long.

I.2. Review of existing queue management systems and their limitations:

I.2.1. Queuing Management System

An automated system for managing human lines, whether they are organized or unorganized. In this kind of setup, one or more servers greet and assist guests as they come. There are two kinds of customers who wait in line sizes: limited (both limitless and limited in size) [21]. It so occurred that a customer visited the service center. A queue, which may be limited or infinite, is a representation of the number of patrons who are awaiting service. Parking lots are examples of finite rows, and financial institutions are examples of limitless rows [22].

- **Types of Queue**

The following are the various types of queuing systems:

- **Structured**

A queue that adheres to a predetermined format or structure for each piece added or withdrawn is called a structured queue. For instance, a bank's structured queue may contain data such as the client's name, account number, and the action they wish to take. When it's necessary to execute several requests in a particular sequence and it's crucial to have uniform and consistent data about each request, this kind of queue is frequently utilized [23].

- **Unstructured**

When items are added to and deleted from a queue without adhering to a set format or structure, the queue is said to be unstructured. A line of people waiting to purchase theater or concert tickets, for instance, is an unorganized queue. When processing requests in the order they are received is the primary concern and precise data about each request does not need to be captured, this kind of queue is frequently utilized [24].

I.2.2. Queue management for postal services

In order to guarantee that clients obtain effective and efficient services, queue management is a crucial component of postal services. An efficient queue management system is now required due to the rise in demand for postal services. growing significance.

- **Current system**

Currently, postal services use a combination of automated and human techniques for their queue management system. In order to be serviced, customers must fill out a ticket and wait in line. In the manual procedure, a customer support agent calls out names or numbers to indicate who should get help next. It can take a while to complete this process, especially during busy times.

- **Advance Queue System**

A system known as an advance queue system is depicted in Figure 1.7. To increase the flexibility of the queuing system process, this system adds more service counters based on the (AQS)

SAQS design [25]. Additionally, it is more frequently utilized by banks and businesses that offer several services concurrently to multiple lines at various counters [26].

Figure I. 7: Advance Queue System.

System is able to use a wide range of services and accommodate large numbers of users. Customers who arrive for various reasons are identified and directed to the relevant counter.

- **Tensator Delivers Queue Management System To The Post Office**

Would be employed to assist in managing the lines of people awaiting access to postal services, which include sending and receiving packages and mail, purchasing stamps, and carrying out other transactions. Customers can utilize the system to expedite customer service and cut down on wait times, as well as to get real-time information on their queue position and anticipated wait periods. [27]

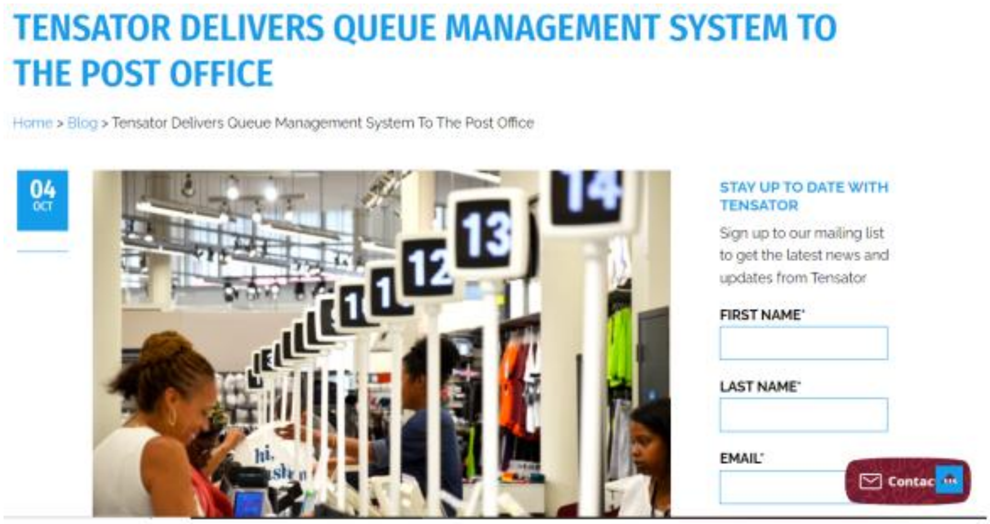

Figure I. 8: TENSTOR

I.2.3. Advantages of queue management system

Benefits from a queue management system can be attributed to the system directly or indirectly, depending on whether the customer or the customer service provider benefits from the system [25]. They are:

- Shorten wait times and improve their tolerability.
- Monitor and project the flow of clients.
- Making the best use of headcount projections [28].
- Keep a close eye on employee performance [25].
- Boost employee morale and productivity since operations are well-organized and efficient [24].
- Allows for flexibility while interacting with clients [24].
- Boost service dependability since clients receive prompt, courteous service.
- Generating statistical reports that help senior management make decisions.
- It lowers the chance of human error and does away with the requirement for manual queue tracking.
- Because consumers can go through the service center more quickly, it increases efficiency.

The usage of internet services is another feature of contemporary queue management systems. As a result, wait times are shorter and the general customer experience is enhanced. Additionally, customers may follow the progress of their mail and shipments online, doing away with the need to physically visit service centers.

I.2.4. Disadvantages of queue management system

- **Technical issues**: Because queue management systems rely on technology, it occasionally malfunctions or has technical problems. Customers and employees may experience delays and irritation if the system fails.
- **Complexity:** Longer wait times and greater aggravation may result from customers finding the queue management system complex or challenging to operate.
- **Cost**: Smaller post offices or those with tighter resources may find it difficult to implement and maintain queue management systems due to their high upfront costs.

- **Staffing concerns**: Queue management systems may lead to a reduction in the number of people on staff. If there are insufficient personnel to address customer concerns or questions, this may have a negative impact on customer service.

Absence of a personal touch: Some clients might feel more comfortable interacting with a person than a machine. The queue management system may come across as impersonal or insensitive to user needs if it is improperly designed or implemented.

• **Inflexibility**: Some queue management systems are difficult to modify to account for variations in staffing levels or client demand. Long wait times and lower customer satisfaction may result from this.

• **Maintenance requirements**: In order to keep queue management systems operating correctly, they need to have regular maintenance, which can be costly and time-consuming.

• **Limited Customization Options**: Companies with special procedures or service workflows may find it difficult to use queue management systems. Pre-built systems might not be flexible enough, requiring compromises and workarounds that could reduce the system's efficacy and efficiency.

I.2.5. Related Work

Since they give users a more practical and effective way to access postal services without having to wait in line, online queue management systems for postal services have grown in popularity in recent years.

Through a website or a smartphone app, these systems usually enable users to remotely reserve a place in the line and get updates on their anticipated wait time.

Here are some instances of online queue management platforms that postal agencies employ:

1.2.5.1. Walk-Away' Queue Management System

"The Walkaway Queue Management System" (also known as "The Walkaway QMS") is the first system that has been examined (Figure 1.9).MySQL served as the system's primary server and was responsible for overseeing the customer's web application and database. Permits clients to take their tickets for the line wherever they go [29]. and allows clients greater freedom in selecting their waiting areas. When tickets are validated, customers will receive notifications via SMS and Telegram [30].

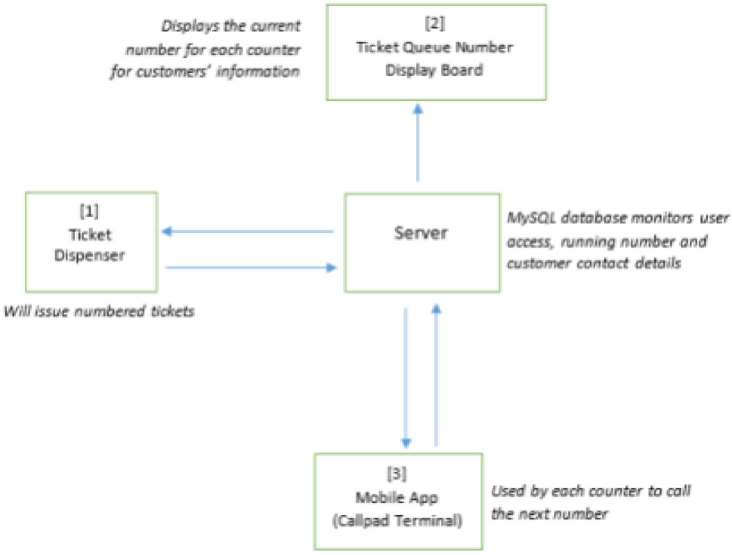

Figure I. 9: The Walk-Away QMS.

I.2.5.2. Queue-it

An e-queue management system called Queue-it (Figure 1.12) is made to manage high-traffic events like online sales and ticket releases. Queue-it controls traffic spikes and stops website crashes by using a virtual waiting room. When it is their turn to enter, customers join a queue and are then sent to the website [31].

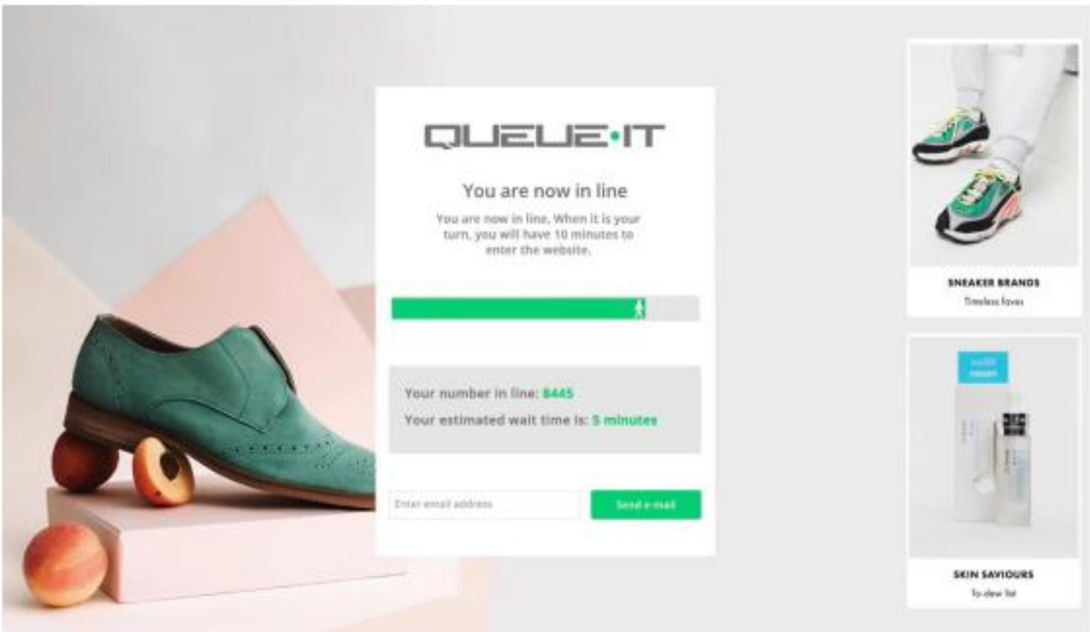

Figure I. 10: Queue-it website.

I.2.5.4. SimplyBook.me

Is a mobile app or website-based online booking and e-queue management solution that enables users to make appointments and join virtual queues. Businesses can also manage their res

ap

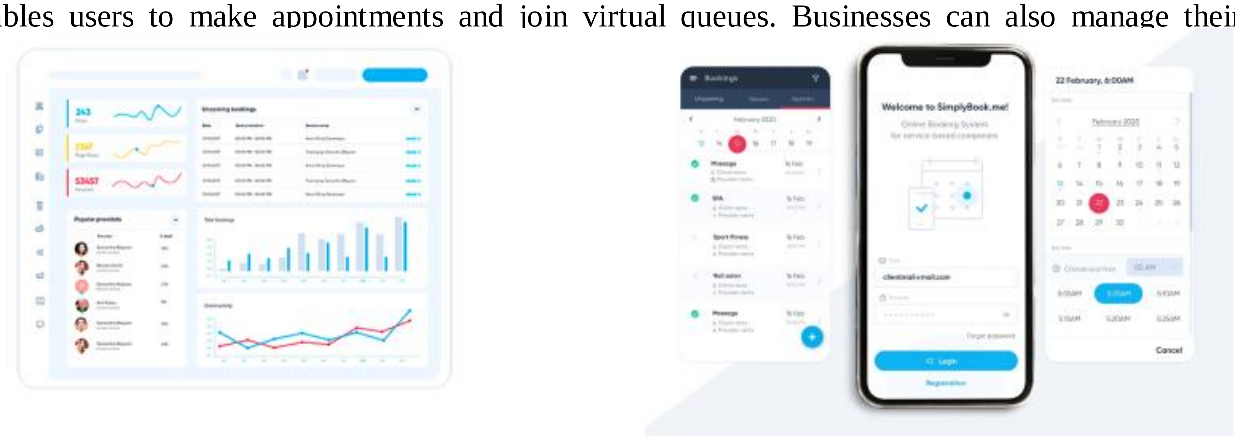

Figure I. 11: SimplyBook.me Application.

I.2.5.5. QLess

A cloud-based queue management system called QLess (Figure 1.14), enables users to join a virtual line via a website or mobile app. Clients get real-time notifications on where they stand in the queue and how long they should expect to wait. In order to help firms optimize their operations, QLess also offers analytics and reporting tools [33].

Figure I. 12: QLess app.

I.2.5.6. QueueForMe

There are two different kinds of users on this web application server (Figure 1.15), virtual queue creators and creator clients. To initiate a virtual queue, the queue creator is prompted by this online application to provide the name and description of the queue.

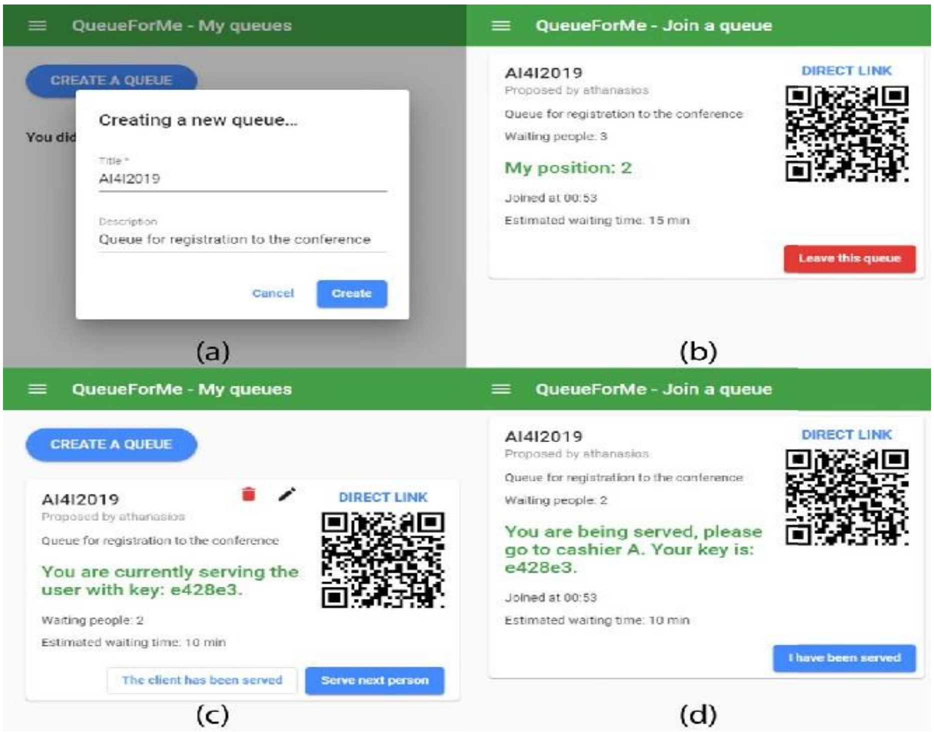

Figure I. 13: Queue for Me web application view.

I.3. Explanation of the importance of predicting waiting times in queue systems.

Predicting waiting times in queue systems is crucial for several reasons:

I.3.1. Customer Satisfaction:

The fact that customers know how long they will have to wait improves their planning of the time improving both the reduction of the customer’s frustration and the overall satisfaction with the service.

I.3.2. Resource Allocation:

Predicting the wait times helps the companies to decide about their resources optimally. They can adjust their staffing levels and operational procedures that match actual demand, which improves productivity and reduces operational expenses.

I.3.3. Service Planning:

Correct forecast of waiting times ensure companies to manage their services effectively. They can introduce booking appointments, regulating peak hours, and employing techniques to cut back on the waiting time which will eventually lead to better service.

I.3.4. Queue Management:

Thus, forecasting the waitings time allows business to develop great queuing management strategies. This could include: setting up virtual queues, offering appointment scheduling systems, or providing real-time updates to customers about their upcoming wait time.

I.3.5. Customer Retention:

Through sharing clear and trustworthy information about waiting times, business shows its focus on quality and customer service. This way it can lead to customer loyalty and thus be a factor for revisiting the business.

In conclusion, predicting waiting times in queue systems is paramount in the provision of quality customer service, efficient utilization of resources, good service planning, feedback forqueue management, and customer loyalty.

Chapter II: AI for Predicting Waiting Times

Introduction to the significance of AI techniques in improving prediction accuracy and efficiency in various domains

In the ever-evolving landscape of technological innovation, (AI) stands out as a beacon of transformative power, particularly in its ability to enhance prediction accuracy and operational efficiency across a broad spectrum of fields. From the intricate corridors of healthcare to the bustling aisles of retail, AI techniques are not merely tools; they are the architects of a future where insights are predictive, not reactive.

The quest for precision and efficiency in predictions cuts across several fields, from anticipating market trends to diagnosing diseases and predicting climate changes. Traditional models have served well yet are increasingly outstripped by the complexity and volume of data inherent in today's challenges. Enter AI, a suite of technologies known for their prowess in handling big data and complex algorithms with surprising dexterity and speed.

Long waiting times are a common pain point that diminish efficiency and satisfaction in services and resource management across various fields. Conventional methods often struggle with the dynamic variables and sheer data volume required for more accurate predictions. AI, with its robust analytical capabilities, offers a promising solution by optimizing predictions for better decision-making and resource management, thus reducing waiting periods significantly.

II.1. Applications Across Domains

At its core, AI's prowess in prediction lies in its sophisticated algorithms and machine learning models. These are not mere mathematical constructs; they are the brainchild of human ingenuity, designed to learn from vast datasets and forecast future outcomes with remarkable precision. By harnessing the power of AI, organizations can anticipate trends, mitigate risks, and unlock unprecedented opportunities.

- In Finance: AI models are reshaping how we predict market movements. By integrating DL and sentiment analysis, researchers have shown that AI can outperform traditional models in predicting stock returns, offering a more nuanced understanding of market dynamics.[34], In the financial sector, AI techniques play a crucial role in mitigating risk and ensuring stability. Advanced predictive models can forecast market trends, identify potential fraud, and automate compliance checks, thereby safeguarding assets and fostering trust among investors. Moreover, AI-driven robo-advisors offer personalized investment advice, making financial planning more accessible and inclusive.[35]

- In Healthcare: AI's impact is profound, from predicting patient outcomes to advancing disease diagnosis. Models trained on electronic health records have demonstrated superior performance in forecasting medical events, a leap towards personalized medicine .[36]. For instance, machine learning algorithms can analyze medical imaging data with an accuracy that rivals, and often surpasses, that of human experts. This not only expedites the diagnostic process but also ensures that patients receive timely and targeted interventions, significantly improving outcomes.[37]

- In Environmental Science: AI helps model climate patterns and assess impacts more swiftly than conventional methods, crucial for addressing urgent climate challenges.[38]

- AI in Retail: In the bustling world of retail, AI has revolutionized the way businesses understand and interact with their customers. By analyzing purchasing patterns, browsing behavior, and social media interactions, AI algorithms can craft personalized recommendations that resonate with individual preferences. This level of personalization not only enhances customer satisfaction but also boosts sales and fosters brand loyalty.[39]

- Enhancing Customer Service: The customer service sector benefits from AI through improved prediction of call volumes and customer inquiries, allowing for better staff allocation and training programs. Predictive chatbots and automated responses, powered by AI, handle routine inquiries instantly, freeing human agents to tackle more complex queries. This not only reduces customer waiting time but also enhances overall customer experience.

II.1.1 AI Techniques: An Overview

AI consists of different technologies, among them, ML, DL, NLP, and RL. Every one of the given methods utilizes the advantage of computational power and data to spot the patterns, predict the future and give reasonable advises.

- Machine Learning (ML): Emphasizes on algorithms that using experience to make them to be better. Supervised learning models such as SVM and ensemble methods like RF are used extensively for prediction tasks [40].

- Deep Learning (DL): To be precise, a segment of ML, deep learning, in other words, uses neural networks with a variety of layers (or deep networks) to find accented patterns in huge data sets [41].

- Natural Language Processing (NLP): ELICS is the technology that enables machines to understand and interpret human language. The techniques such as transformer models have significantly transformed the text-based predictions. [42]

- Reinforcement Learning (RL): Agents are trained through awards and punishments. Arranged in the sequence of decisions and is used to control" the environment which requires sequential decision-making.[43]

-Big Data: The Fuel Powering AI Predictions:

The surge in data generation has been a boon for AI. Big data provides the necessary fuel for training AI models, enabling them to perform complex computations and deliver precise predictions. The integration of AI with big data analytics has opened new frontiers in fields such as climate science, where AI helps in modeling climate change scenarios and predicting weather patterns with remarkable accuracy.

Highligh: According to a report by Luo et al. (2021), AI-driven models have significantly improved the accuracy of weather predictions, aiding in better disaster management and preparedness. [44]

-AI and IoT: A Symbiotic Relationship for Enhanced Predictability:

The Internet of Things (IoT) and AI form a powerful combo, with IoT devices collecting real-time data and AI analyzing this data to make immediate and accurate predictions. This synergy enhances predictability in smart homes, smart cities, and even industrial operations, leading to smarter decision-making and operational efficiencies.

Example in Action: Smart thermostats in homes learn from user behavior and environmental conditions, adjusting heating and cooling systems efficiently, [45]

II.1.2 Challenges and Future Horizons

Despite AI's strides, the journey is not without hurdles. The quality and availability of data remain significant challenges, as AI models require extensive, well-curated datasets to function optimally. The opacity of many AI models also raises questions about transparency and interpretability, essential for high-stakes decisions.

Looking ahead, the focus should be on creating more transparent AI models, enhancing data privacy, and bridging the gap in low-resource settings. The fusion of traditional statistical methods with AI presents a balanced approach, leveraging the best of both worlds to tackle complex predictive tasks.

In conclusion, AI's ML and DL techniques have proven to be formidable in enhancing prediction accuracy and efficiency. These technologies adapt and learn, offering improved predictive insights over traditional methods. As AI continues to evolve, its integration into predictive analytics is set to deepen, promising exciting opportunities and new challenges, AI's capacity to enhance predictive accuracy and efficiency is evident across multiple domains. However, the journey is not without hurdles. Balancing innovation with ethical responsibility and transparency will be key to harnessing AI's full potential responsibly.

II.2. Review of existing studies and research on the application of AI in waiting timeprediction let's dive into the trendy world of AI and explore 'how it' revolutionizing waiting time prediction'.

In this edgy review, we'll uncover the latest scientific gems on applying artificial intelligence to estimate wait times in various industries, making the experience less annoying and more predictable for everyone. Here's a rundown of how researchers are tackling this problem with some serious innovation.

II.2.1. Existing Studies Overview

II.2.1.1. Healthcare Heroes:

Emergency departments are notorious for long waits, so it's no surprise that loads of researchers are looking into predicting waiting times here.

In healthcare,'machine learning algorithms' have stepped up to predict patient wait times, optimizing hospital resources (Gong et al., 2018). By predicting appointment durations, these systems help clinics reduce wait times and improve patient satisfaction. [46]

- Ryu et al. (2018) used 'Machine Learning 'models to predict ED waiting times. They crunched a mix of patient data like age and triage category, alongside real-time hospital data. Their Random Forest model hit a home run with a mean absolute error of just 19 minutes. [47]

- Sun et al. (2019) took it up a notch, employing a' Deep Learning' approach using LSTM networks. By analyzing patient flow and admission data, their model predicted waiting times within 15 minutes of accuracy. [48]

- Gupta et al. (2020) hopped on the 'Reinforcement Learning' train to optimize patient flow and predict waiting times. Their 'Q-Learning' model reduced ED overcrowding by predicting and suggesting optimal bed assignments. [49]

II.2.1.2. Public Transport Wizards:

Public transportation is another hotspot for AI predictions.

- Zheng et al. (2017) cooked up a 'Neural Network' model that predicts bus arrival times using GPS data and historical travel time. By factoring in road conditions and traffic patterns, they shaved off prediction errors to just a couple of minutes. [50]

- Wang et al. (2018) used 'Graph Convolutional Networks' to capture complex spatial dependencies in the transportation network. Their model accurately predicted subway waiting times, outperforming traditional statistical models. [51]

II.2.1.3. Restaurant and Retail Gurus:

Nobody likes waiting in line for fries or a new pair of kicks.

- Al-Molegi et al. (2019) developed a 'Time Series Forecasting' model using 'ARIMA ' to predict customer waiting times at fast-food joints. Their model helped shift managers optimize staffing during peak hours. [52]

- Chen et al. (2020) went all-in with 'Deep Learning' to predict queue lengths at retail stores. Using 'CNN ', they analyzed in-store camera feeds and historical sales data. They nailed down waiting time predictions to within five minutes. [53]

II.2.1.4. Air Travel Savants:

Predicting airport security line waits
- Müller et al. (2018) created a predictive model using 'Gradient Boosting Machines (GBM)'. They pulled data from flight schedules, passenger demographics, and security checkpoint throughput to give travelers a heads-up on line waits. [54]

II.2.1.5. Revolutionizing Queues with Deep Learning:

AI (specifically deep learning) has shown exceptional prowess in managing queues by learning patterns and predicting wait times (Chen et al., 2019). By analyzing historical data, these models can forecast future congestion, minimizing customer frustration. [55]

II.2.1.6. Smart Traffic Forecasting:

Researchers have harnessed the power of (CNN) to predict traffic waiting times (Wu et al., 2020). Traffic flow data is analyzed in real-time, transforming our daily commute by providing accurate arrival time estimates.[56]

II.2.1.7. Predictive Maintenance & Service Calls

Reinforcement learning techniques are applied to anticipate service calls' wait times, streamlining maintenance operations (Liu & Hu, 2019). By identifying potential issues early, these systems minimize downtime and keep customers happy.[57]

II.2.1.8. Food Service on Speed Dial

AI-driven 'order prediction models' have transformed the food industry (Khan & Kim, 2021). By predicting demand, they reduce customer wait times at restaurants, drive-thrus, and even meal delivery services. [58]

II.2.2. The Essence of AI in Waiting Time Prediction

AI, particularly machine learning and deep learning, has transformed from a sci-fi fantasy to a real-world utility belt tool, especially in managing and predicting waiting times across various sectors. From healthcare to customer service, transportation to retail, AI's ability to crunch massive amounts of data and spit out predictions is nothing short of a modern-day oracle.

Why It Matters

- Customer Satisfaction: Nobody likes to wait, but knowing how long you'll wait changes the game. It eases anxiety and manages expectations.

- Efficiency Boost: Businesses operate smoother when they can predict traffic flow and manage resources accordingly.

- Strategic Planning: With predictive data, companies can staff appropriately, reduce overhead costs, and improve overall service quality.

II.2.3. Key Challenges and Future Directions

- Data Quality & Availability: Garbage in, garbage out. Models need high-quality, real-time data to make accurate predictions.

- Generalization Issues: A model killing it in one ER might flop in another due to different patient demographics and flow patterns.

- Ethical Concerns: Predicting waiting times in healthcare, in particular, needs to be done ethically to avoid prioritizing one group over another unfairly.

In conclusion, AI explodes the conventional boundaries of waiting time prediction, enhancing predictability and reducing confusion. By integrating AI, industries can create a smoother, more enjoyable experience for their customers. From traffic to healthcare, retail to food service, AI is transforming the way we wait.

II.3. Description of the data collection process for training AI models, includingsources of data and data preprocessing steps.

Time regarding the waiting is one of the main problems in many sectors such as healthcare, transportation, and customer service. ML and AI models are the ones who give the accurate predictions of waiting times as long as the data which is used for the training of the models are of high quality. This section offers a thorough description of the data collection process, including the sources of the data and the data preprocessing steps, to be more specific about the waiting time prediction problem.

II.3.1. Data Collection Process for Waiting Time Prediction

II.3.1.1. Understanding the Waiting Time Problem

- Define the specific waiting time problem to be solved, such as predicting:

- Emergency department (ED) requires longer waiting times which is a challenge in the healthcare system.

- The possible arrival time (ETA) for the transportation buses or ride-hailing services is the time of arriving at the destination.

- Waiting times in manufacturing for job scheduling.

- Customer service waiting times in call centers.

- Identify key performance metrics (e.g., mean absolute error, root mean square error).[59]

II.3.1.2. Identifying Data Sources

- Primary Data: Collected through direct measurement or observation.

- Healthcare: EHR, patient flow logs, nurse/staff schedules.

- Transportation: GPS logs, ride-hailing service logs, public transit schedules.

- Manufacturing: IoT sensor data, production logs, machine maintenance records.

- Customer Service: Call center logs, chat transcripts, queue lengths, agent schedules.

- Secondary Data: Pre-existing databases from external organizations.

- Government databases, research datasets, open-source datasets like MIMIC-III [60]

- Third-Party Data: Commercial data purchased from data brokers.

For waiting time prediction, data can be collected from several relevant sources:

- Transactional Systems: These include ticketing systems, appointment scheduling software, and electronic logging systems, which record each service encounter's start and end times.

- Real-time Monitoring Systems: In scenarios like traffic control or public transport, data might come from GPS tracking systems, surveillance cameras, or sensors that monitor real-time movements and queues.

- Customer Feedback: Surveys and feedback forms are a source of information that can be used to find out about the subjective waiting experiences and they can be compared with the objective data.

- Public Datasets: Many governments and organizations publish datasets related to service metrics, including waiting times which can be used for benchmarking and training predictive models.

II.3.2. Example Sources in Various Domains

II.3.2.1. Healthcare:

- EHR that are obtained from hospitals.

- Information about patient flow obtained from ED or outpatient clinics.

- Imaging data for diagnostic waiting times.

II.3.2.2. Transportation:

- Public transit data from government API.

- Ride-hailing data from services like Uber or Lyft.

- GPS data from fleet management systems.

II.3.2.3. Manufacturing:

- IoT sensor data from production lines.

- Production logs from MES.

- Maintenance records for machine downtime.

II.3.2.4. Customer Service:

- Call center logs and CRM databases.

- Chat transcripts.

- Historical queue lengths and wait times.

II.3.3. Data Collection Techniques

- API and Web Scraping: For public transit data, EHR, and call center logs.

- Manual Collection: For primary data through surveys or direct observation.

- Data Integration: Combining data from multiple sources, such as EHR and imaging data in healthcare [61]

II.3.4. Data Preprocessing Steps for Waiting Time Prediction

II.3.4.1. Data Cleaning and Quality Assurance

II.3.4.1.1. Handling Missing Values:

- Impute missing values using mean, median, mode, or predictive models.
- For categorical variables, consider forward/backward filling.
- Healthcare: Replace missing vital signs with predictive models [62]
- Transportation: Interpolate missing GPS coordinates [63]

II.3.4.1.2. Outlier Detection and Removal:

- Identify outliers using Z-scores, IQR, or clustering-based methods [64]

Dealing with Duplicates:

- Remove duplicate records, particularly in EHR and transportation data.

II.3.4.2. Data Transformation

II.3.4.2.1. Normalization and Standardization:

- Standardize numerical features (e.g., age, travel distance) to zero mean and unit variance.
- Normalize time-based features (e.g., waiting times) to a specific range.

II.3.4.2.2. Feature Engineering:

Healthcare:

- Create features from EHR like patient history, triage level, and nursing schedules.

Transportation:

- Create features like weather conditions, traffic congestion, and driver behavior.

Manufacturing:

- Extract features from IoT data like machine usage, maintenance history.

Customer Service:

- Features like agent availability, customer sentiment from chat transcripts.

Encoding Categorical Variables:

- One-hot encoding for categorical variables (e.g., triage level, service type).
- Ordinal encoding for ordered categories (e.g., priority levels).

II.3.4.3. Data Integration and Aggregation

II.3.4.3.1. Combining Multiple Sources:

- Merge data from different sources like EHR, imaging, and staff schedules.
- Transportation: Combine ride-hailing logs with weather and traffic data.

II.3.4.3.2. Aggregation:

- Aggregate data at different levels, such as daily or weekly aggregates.

II.3.4.4. Feature Selection:

II.3.4.4.1. Filter Methods:

- Chi-square tests, correlation coefficients.

II.3.4.4.2. Wrapper Methods:

- Recursive Feature Elimination (RFE).

II.3.4.4.3. Embedded Methods:

- Lasso regression, tree-based models.

II.3.4.5. Splitting the Dataset

II.3.4.5.1. Training, Validation, Test Split:

- Typical splits include 70-15-15 or 80-10-10.

II.3.4.5.2. Cross-Validation:

- K-fold cross-validation or stratified sampling.

A good collection and cleansing of data in the first stage helps in accurate prediction of waiting times. AI models are able to come up with precise predictions for situations caused by the waiting time problems as a result of using a good quality data set and deliberately cleansing it.

II.4. Preprocessing techniques such as data cleaning, normalization, and featureengineering to prepare the data for AI model training

Data cleaning (preprocessing) usually comes in forefront of any machine learning project. This is a procedure aiming to major in cleaning, converting and adjusting, or preprocessing raw data for training of AI models. Precise preprocessing steps performed correctly translate to better model performance; the process bandwidth maximizes for precision, speed, and better generalization. The sub-section analyzes some common procedures employed in preprocessing such as data cleaning and normalization as well as the feature engineering.

II.4.1 Data Cleaning

The data cleaning is the first important process of data preprocessing procedure that concentrates on removing or correcting wrong, blank or inapplicable data. The process starts from dealing with missing values and follows through with getting rid of duplicate records and correcting them.

Techniques:

- Outlier detection and treatment: Outliers can significantly impact the performance of AI models. Techniques such as statistical methods (Z-score, IQR), clustering, or domain knowledge can be employed to detect and handle outliers appropriately. [65]

- Missing data imputation: Missing values can introduce biases and hinder model performance. Imputation techniques like mean/median substitution, regression imputation, or more advanced methods like k-Nearest Neighbors (kNN) or multiple imputation can be utilized. [66]

- Data validation and consistency checks: Ensuring data consistency across multiple sources or over time is essential. Validation rules, data dictionaries, and domain-specific constraints can be applied to identify and rectify inconsistencies. [67]

II.4.2 Normalization

Normalization is a technique used to scale numeric data in the dataset to a common scale, without distorting differences in the ranges of values or losing information. This is particularly true for models with data size as a key component such as neural networks, and k-means clustering.

Techniques:

1. Min-Max Scaling: Scales the data between a specified range (usually 0 and 1).

2. Standardization (Z-score Normalization): Transforms the data to have zero mean and a variance of one.

3. Robust Scaling: Uses the median and the interquartile range for scaling. It is robust to outliers.

II.4.3 Feature Engineering

The feature engineering process involves either development of new features or modifying existing features to improve the model's performance One of the strategies is to consider interaction terms, polynomial features, and domain-specific features that are expert related.

- Domain-specific feature creation: Leveraging domain expertise to craft features that capture important relationships in the data. For an instance, the transaction amount and time-of-day features might be used for generation of new features to specify the high-priced transactions undertaken outside business hours.[68]

- Polynomial and interaction features: Creating new features by combining existing ones through polynomial or interaction terms can capture nonlinear relationships and improve model performance. [69]

- Feature selection: The fact that the identification and retention of the most expressive features not only improves model discovery but also reduces the overfitting considerably enhances the model comprehensibility. For example, one can use methods which include recursive feature elimination, feature importance analysis, or wrapper methods. [70]

Data preprocessing is as paramount as other steps for implementing an efficient AI model. Through the implementation of best suited methods for data cleaning, normalization and feature engineering, we can make the quality of data better, model efficiency improved and more meaningful insights from the data. The techniques that will be used will vary according to the sort of data being worked with and the objective of the AI project.

Interactions between the Queuing Theory and the Artificial Intelligence Field

Applications of Artificial Intelligence in the Queuing Theory Research

Artificial Neutral Networks and Queuing Theory:

Forecasting is one of the primary problem categories that machine learning algorithms address; this involves predicting system behaviors based on past responses. It is feasible to make this predicting simpler.

to the challenge of determining how many variables function. Researchers have generally agreed in the past that Artificial Neutral Networks are among the best machine learning methods for function estimation [71]. In their study [72], the two authors, Vishnevsky and Gorbunova, offer a thorough analysis of the approximation function and contend that machine learning techniques, in particular the decision tree generation algorithm, are techniques that can be applied to troubleshooting Queuing Theory problems. In addition, Vishnevsky and Gorbunova discuss the potential of machine learning approaches in general, some cutting-edge methods for imitating complex system behavior, and the advantages of modeling a solitary queuing system. This work provides a more efficient method for estimating performance indicators by fusing conventional approaches with machine learning strategies like simulation and artificial neural networks. When working with intricate queuing models, where simulation time may be prohibitive, this can be extremely useful. With the use of this novel technique, it is possible to train an intelligent model that can produce estimates for every intermediate value, regardless of how many can be provided, without having to use energy simulating all possible input values for the parameters. In the end, this can reduce waiting times, improve precision, and offer insightful information on intricate queuing systems [73]

A Novel Approach to Dealing With Queuing Theory Problems Using Machine Learning Methods.

Researchers are currently investigating ways to combine different data mining techniques—particularly simulation and artificial neural networks with traditional Queuing Theory methodology. The current body of work on machine learning's application to queuing theory is fairly dispersed, which makes it challenging to draw broad conclusions from it, let alone create a unique and cutting-edge strategy for handling challenging queuing theory problems that is on par with conventional approaches. Vishnevsky and Gorbunova present a novel method in their paper for combining simulation and (ANN) with conventional Queuing Theory techniques. Simulation can provide precise approximations of performance indicators in queuing models. However, the complexity of the queuing model, the hardware of the computing system, and the simulation software environment all influence how long it takes to reach a certain number. The two writers support the idea that neural networks can be trained through simulations to produce estimates of intermediate input values for parameters. This method lessens the requirement for simulation modeling of all relevant input parameter values; nonetheless, training with an explicit neural network or other intelligent model is still required. The forecasting procedure is quick. A number of software tools, from specialized tools like Any Logic and Arena to custom models made in the Python programming language, which has a wide range of features and archives, including those for training artificial neural networks, can be used to generate simulation models. The number of requests required to generate a single output value or group of output values is known as the "model run length," which is a difficult task. Because single-run data are correlated, multiple realizations are required to determine the average value of every variable under investigation. The simulation duration increases when the run length changes based on the input parameters. However, estimates can be derived for a small range of input parameters before constructing a neural network and solving the forecasting problem if an algorithm is employed to assess the probabilistic-temporal properties of a network or queuing system and it has high processing costs.

Review on the Application of Machine Learning Methods in the Queuing Theory Field.

Although machine learning techniques and algorithms are widely applied in science and technology, especially in the research of state-of-the-art broadband wireless networks, their usage in queuing theory is not well documented in the international literature.

In order to decrease waiting times and improve service effectiveness, it is crucial to investigate physical queuing in sectors like sales and services. While machine learning techniques have been the focus of more recent research, queuing theory remains a conventional approach for estimating waiting times. Artificial neural networks are used by Sundara and Palaniammalb to simulate the traditional queuing mechanism M/M/1 in one study. One network employs input and output layers, while the other uses a back propagation technique with a single hidden layer [74]. The study's findings demonstrate that neural network models can predict target provided input data parameters with remarkable accuracy and are strikingly compatible with analytical frameworks. The same authors, Sundaria and Palaniammalb, conducted a second study in which they enhanced and organized the lines on an airport runway using a neural network simulation of the traditional queuing system [75]. One interesting area for further investigation in the Queuing Theory studies is the use of neural networks for the analysis of non-Markov Queueing Systems (QS). Although there aren't many papers on the subject, non-Markov systems are starting to receive more attention. most real-world physical structures and processes are modeled. Based on the non Markov QS with a "warm-up," which may replicate the activation process of an empty system as soon as it receives a request for the first time after a break, one of the earliest studies on employing neural networks to assess non-Markov QS models. By approximating the system using a QS and the phasetype distribution of the incoming flow or service time, the system was successfully "markovized". Neural networks have been developed as a solution to the laborious and resource-intensive mathematical processes required to compute the static frequencies of states in this type of QS.

Streamline the issue without compromising precision. The article's neural network was constructed with a double-layer perceptron, and its input parameters included the coefficient of variation, "warm-up," and the strength of the serving and receiving flows. The output parameters were the system's average waiting and sojourn times as well as the fixed distribution of the total number of clients. In addition, studies show that of all the methods used to train the ANN, the Bayesian regularization methodology is the most accurate.

Chapter III: Implementation and results

III.1. Introduction

This chapter is organized as follows: In the first section, we present predictions of waiting times using artificial intelligence applied to synthetic data generated using Python. This approach was necessary because it was not possible to obtain real bank data from the banks in the Wilaya of Ouargla due to security concerns in protecting public institutions. The aim of these predictions is to accurately estimate waiting times.

In the second section, we describe our custom waiting system solution, which I developed to address the issue of long wait times. The system is illustrated through screenshots and demonstrates its potential use across various industries, with a specific focus on banks and postal institutions as illustrative examples. The system is not limited to a specific type of public institution but can be applied to any that use queue number machines. Unfortunately, I was unable to directly connect the application to the queue number machine in the bank due to the institution's security protocols against potential breaches.In the third section,

I developed a prototype of a queue number machine using Arduino, ESP32, and an LCD screen, and connected it to my initial application to demonstrate the concept more concretely. Finally, we will discuss possible future developments and enhancements for this system.

III.2. Implementation

We are going to describe the hardware and software we used in our system.

III.2.1.Hardwares Description

An HP brand Notebook is used with the following features to create our solution:

• Processor: AMD Ryzen33250U With Radeon Graphicss (4CPUs),~2.6GHS

• RAM: 8192MB

• System-Type: Windows 11 professionnel 64bits

• System-Model: HP 255 G8 Notebook PC

III.2.2. Software Tools Description

The following programming languages have been employed in our project

III.2.3. Visual Studio Code

Visual Studio Code (VS Code) is a powerful and user-friendly source code editor developed by Microsoft. It is widely used for developing and debugging modern web and cloud applications. Here are some reasons why VS Code was chosen for this project:

- **Ease of Use**: VS Code provides a simple and intuitive interface that makes it easy to write and manage code.
- **Extensions and Integrations**: A vast marketplace of extensions is available, allowing users to add functionality such as code linting, debugging, and version control.
- **Built-in Terminal**: The integrated terminal in VS Code makes it easy to run scripts and manage environments without leaving the editor.
- **IntelliSense**: This feature provides smart completions based on variable types, function definitions, and imported modules, enhancing productivity.

- **Python Library**

Several Python libraries were utilized in this project to handle data generation, processing, analysis, visualization, and machine learning. Here are some of the key libraries:

- **Pandas**: A powerful library for data manipulation and analysis. It provides data structures like Data Frames to efficiently handle and process data.
- **NumPy**: A fundamental package for scientific computing in Python, providing support for large multi-dimensional arrays and matrices.
- **Matplotlib**: A plotting library for creating static, animated, and interactive visualizations in Python.
- **Seaborn**: A data visualization library based on Matplotlib that provides a high-level interface for drawing attractive statistical graphics.
- **Scikit-learn**: A machine learning library that provides simple and efficient tools for data mining and data analysis.

Figure III. 1: Visual Studio Code logo.

III.2.4. Flutter

Google released Flutter, a free and open-source mobile user interface framework, in May 2017 (Figure 3.2). In short, it makes it possible to construct native mobile applications with a single codebase. This suggests that the developer might use a single codebase and programming language to create two unique apps (for iOS and Android) [76]. Flutter is composed of two essential parts:

• An SDK (Software Development Kit) is a collection of tools that are helpful while making applications. There are various tools for compiling code into native machine code for Android and iOS.

• A Framework (UI Library based on widgets): A collection of reusable UI elements, like sliders, text inputs, buttons, and other features, that anyone may alter to suit their needs.

Figure III. 2: Flutter Logo

III.3. Predicting Waiting Times Using AI

In this section, we explore the use of artificial intelligence to predict waiting times in public institutions, with a specific focus on banks. Due to the unavailability of real bank data from the banks in the Wilaya of Ouargla, synthetic data was generated using Python. This synthetic data allows us to simulate real-world scenarios and develop predictive models. Our goal is to accurately estimate waiting times to improve customer satisfaction and operational efficiency in these institutions.

To lay down the groundwork on how one could use data-driven artificial intelligence to predict waiting times by a new person getting into the queue, we used a simulated data set developed in Python. One part contains data about queues formed and served, and another – data about several weeks. In total, data for 1000 customers were recorded, with each entry including the customer's join time, waiting time, service time, and total time in the system (waiting time + service time).

III.4. Data Generation

III.4.1 Setting Up Variables and Generating Data

a. Setting Up Variables:

- `num_rows`: Number of rows in the dataset, set to 1000.
- `np.random.seed(42)`: Ensures reproducibility of results.

b. Generating Arrival Times:

- Generated random arrival dates between specified start and end times using the `pd.date_range` function.

c. Generating Service Times:

- Generated random service times ranging from 5 to 20 minutes using the `np.random.randint` function.

d. Generating Waiting Times:

Added some randomness to service times to generate waiting times using the `np.random.randint` function.

e. Creating Data Frame:

- Created a DataFrame using `pd.DataFrame` and saved the data to a CSV file.

```
import pandas as pd
import numpy as np

# Setting up variables
num_rows = 1000
```

```python
np.random.seed(42)

# Generating arrival times
def generate_arrival_times(start_time, end_time, num_rows):
    start_time = pd.to_datetime(start_time)
    end_time = pd.to_datetime(end_time)
    arrival_times = pd.date_range(start=start_time, end=end_time, periods=num_rows)
    return arrival_times

# Generating random service times between 5 and 20 minutes
def generate_service_times(num_rows, min_time=5, max_time=20):
    service_times = np.random.randint(min_time, max_time, size=num_rows)
    return service_times

# Generating waiting times based on service times with added randomness
def generate_wait_times(service_times):
    additional_wait = np.random.randint(1, 10, size=len(service_times))
    wait_times = service_times + additional_wait
    return wait_times

# Creating the dataset
arrival_times = generate_arrival_times("2023-01-01 08:00", "2023-01-01 16:00", num_rows)
service_times = generate_service_times(num_rows)
wait_times = generate_wait_times(service_times)

# Creating DataFrame
data = pd.DataFrame({
    "arrival_time": arrival_times,
    "service_time": service_times,
    "wait_time": wait_times
})

# Saving the dataset to a CSV file
csv_path = "synthetic_bank_wait_times.csv"
data.to_csv(csv_path, index=False)

print("Synthetic dataset created successfully and saved to file:", csv_path)

```

III.4.2. Data Processing and Exploration

III.4.2.1. Loading the Data

a. **Loading the Data**:

- Loaded the data from the CSV file using pd.read_csv.

b. **Describing the Data**:

- Used the describe function to summarize the data statistically.

III.4.2.2. Data Analysis and Visualization

a. **Plotting the Distribution of Waiting Times**:

- Used sns.histplot to plot the distribution of waiting times.

b. **Plotting the Relationship between Arrival Time and Waiting Time**:

- Used sns.scatterplot to plot the relationship between arrival time and waiting time.

c. **Plotting the Relationship between Service Time and Waiting Time**:

Used sns.scatterplot to plot the relationship between service time and waiting time.

```
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

# Loading the data from the file
csv_path = "synthetic_bank_wait_times.csv"
data = pd.read_csv(csv_path)

# Describing the data
print(data.describe())

# Plotting the distribution of waiting times
plt.figure(figsize=(10, 6))
sns.histplot(data['wait_time'], kde=True)
plt.title('Distribution of Wait Times')
plt.xlabel('Wait Time (minutes)')
plt.ylabel('Frequency')
plt.show()

# Plotting the relationship between arrival time and waiting time
plt.figure(figsize=(10, 6))
sns.scatterplot(x='arrival_time', y='wait_time', data=data)
plt.title('Wait Time by Arrival Time')
plt.xlabel('Arrival Time')
plt.ylabel('Wait Time (minutes)')
plt.xticks(rotation=45)
plt.show()
```

```
# Plotting the relationship between service time and waiting time
plt.figure(figsize=(10, 6))
sns.scatterplot(x='service_time', y='wait_time', data=data)
plt.title('Wait Time by Service Time')
plt.xlabel('Service Time (minutes)')
plt.ylabel('Wait Time (minutes)')
plt.show()
```

III.4.3. Results and Analysis:

III.4.3.1. Statistical Summary of Data:

	Service time	Wait time
Count	**1000.000000**	**1000.00000**
Mean	**11.919000**	**16.87100**
Std	**4.349409**	**4.98088**
Min	**5.000000**	**6.00000**
25%	**8.000000**	**13.00000**
50%	**12.000000**	**17.00000**
75%	**16.000000**	**21.00000**
Max	**19.000000**	**28.00000**

Table III. 1: The statistical summary of the dataset

The statistical summary of the dataset indicates the following:

- Count: There are 1000 measurements for both' service time' and 'wait time'.

- Mean: The average service time is approximately 11.92 minutes, while the average wait time is 16.87 minutes. This indicates that customers typically wait longer than the actual service time.

- Standard Deviation: The standard deviation for service time is about 4.35 minutes and for wait time is approximately 4.98 minutes, showing variability in the times.

- Minimum (Min): The shortest recorded service time is 5 minutes, and the shortest wait time is 6 minutes.

- 25th Percentile (25%): 25% of customers are served in 8 minutes or less, and wait for 13 minutes or less.

- Median (50%): 50% of customers are served in 12 minutes or less, and wait for 17 minutes or less.

- 75th Percentile (75%): 25% of customers are served in 16 minutes or more, and wait for 21 minutes or more.

- Maximum (Max): The longest recorded service time is 19 minutes, and the longest wait time is 28 minutes.

These results indicate that there is variability in service and wait times at the bank, with times clustering around the median values and some extreme values contributing to overall variability. The additional wait time compared to service time suggests the influence of other factors on wait time, such as the number of customers present at a given time or the efficiency of the bank staff.

III.4.3.2. Visual Analysis:

➢ **Distribution of Wait Times:**

- This histogram displays the distribution of wait times in the dataset.
- The x-axis represents the wait times (in minutes), while the y-axis shows the frequency of each wait time.
- The data is somewhat normally distributed, with a peak around 15-17 minutes. This indicates that most customers experience a wait time in this range.
- The distribution has a slight right skew, meaning there are more instances of longer wait times compared to shorter ones.

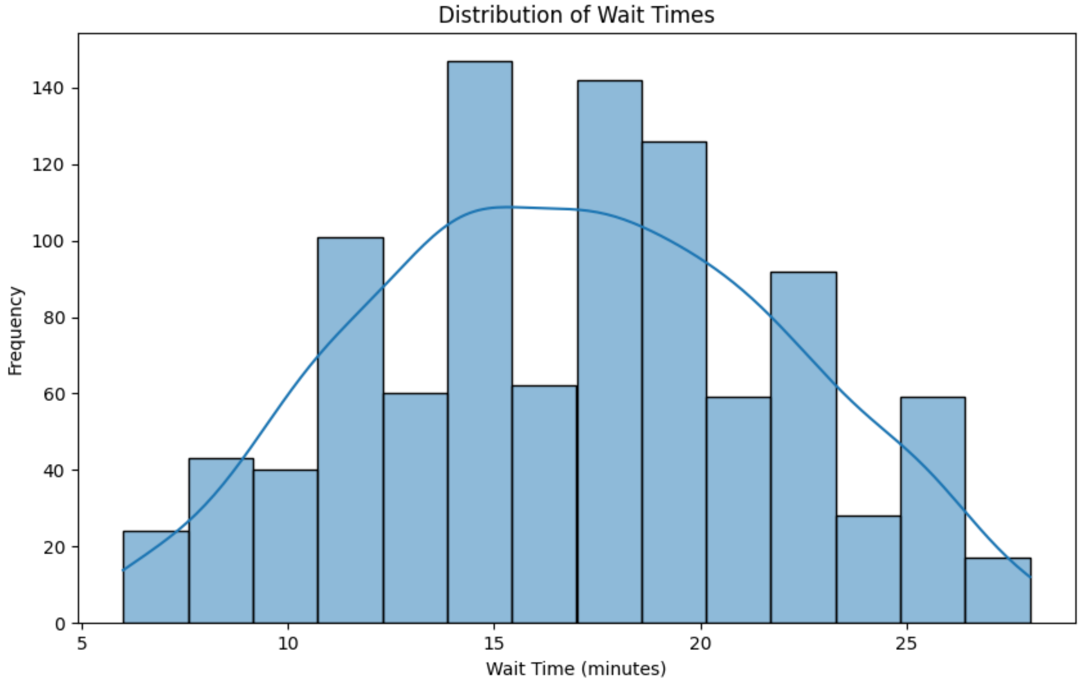

Figure III. 3: Histogram of Wait Times

- **Analysis:**

- The most frequent wait time is around 16 minutes.
- There are fewer occurrences of extremely short (5-6 minutes) and extremely long (25-28 minutes) wait times.
- The spread of the wait times indicates variability in the service process and the possible influence of multiple factors on waiting time.

- **Wait Time by Arrival Time:**

- This scatter plot illustrates the relationship between the arrival time and the wait time.
- The x-axis represents the arrival time (as the time of day), while the y-axis shows the wait time (in minutes).
- Each dot represents a data point corresponding to a customer's arrival time and their respective wait time.

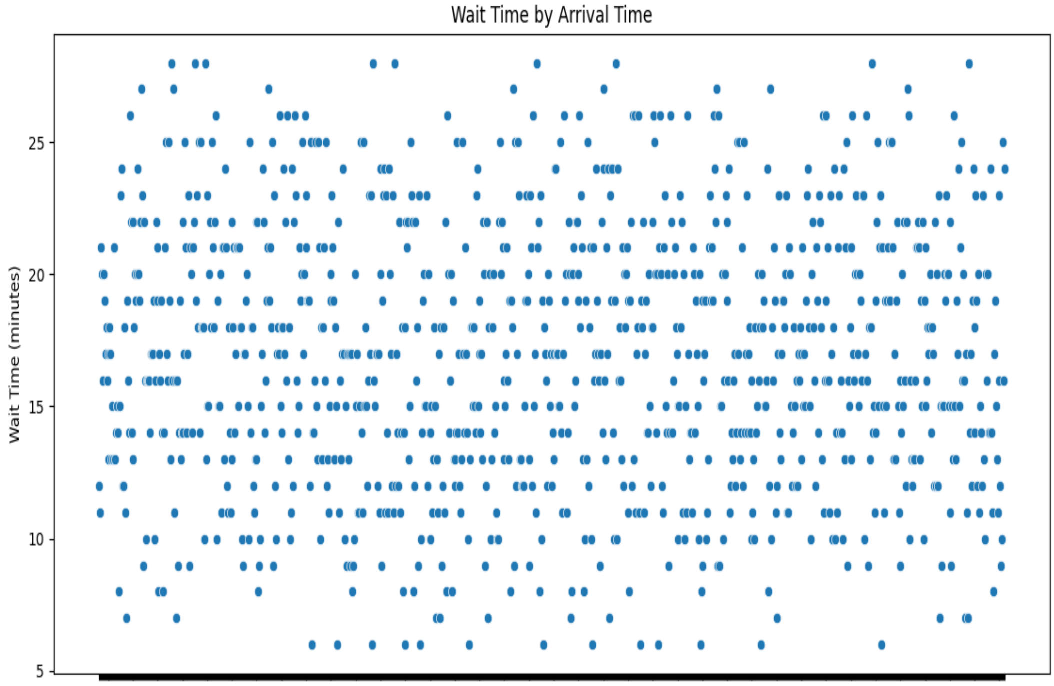

Figure III. 4: Scatter Plot of Wait Time by Arrival Time

- **Analysis:**

- The plot shows no clear pattern or trend between arrival time and wait time. The wait times are scattered evenly across different times of the day.
- This suggests that arrival time does not significantly impact the wait time, or any such impact is not apparent from this visualization.

- **Wait Time by Service Time:**

- This scatter plot illustrates the relationship between the service time and the wait time.
- The x-axis represents the service time (in minutes), while the y-axis shows the wait time (in minutes).
- Each dot represents a data point corresponding to a customer's service time and their respective wait time.

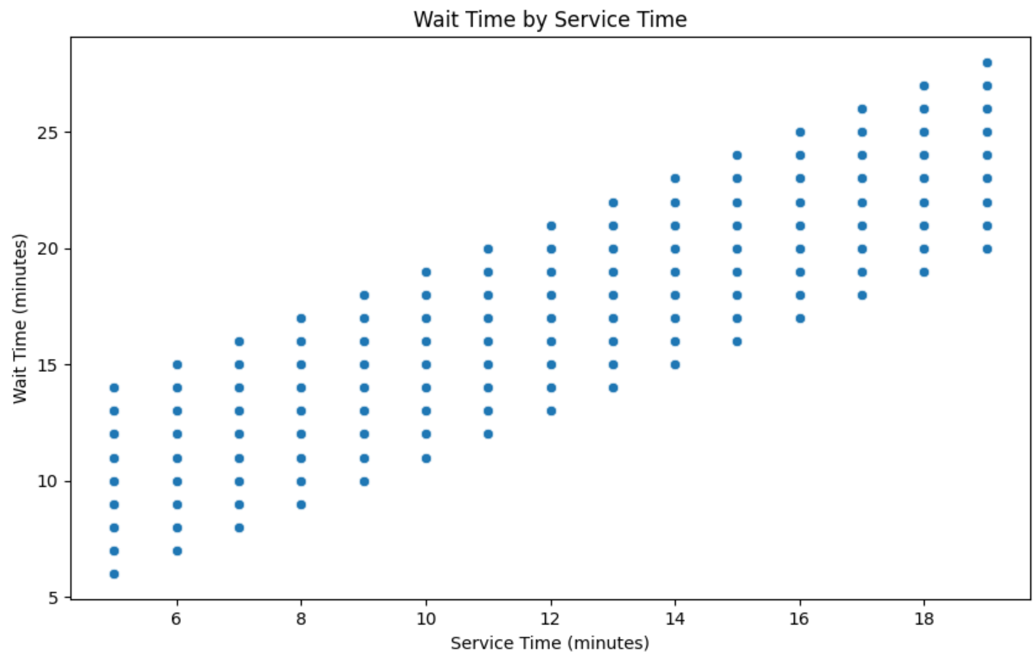

Figure III. 5: Scatter Plot of Wait Time by Service Time

➢ **Analysis:**

- There is a clear positive linear relationship between service time and wait time. As service time increases, wait time also increases.
- This makes intuitive sense because longer service times generally result in longer wait times for subsequent customers.

The plot shows a structured pattern, confirming that service time is a significant factor influencing wait time.

III.5. Prediction Using Machine Learning

III.5.1. Random Forest Prediction

a. **Loading the Data**:

- Loaded the data from the CSV file.

b. **Converting Arrival Time to Minutes**:

- Converted arrival time to minutes since the start of the day using pd.to_datetime and dt.hour and dt.minute.

c. **Splitting the Data**:

- Split the data into features (X) and target variable (y) using train_test_split.

d. **Training a Random Forest Model**:

- Used Random Forest to train the model on the training data.

e. **Predicting on the Test Set**:

- Predicted on the test set and calculated prediction error using mean_absolute_error and r2_score.

f. **Plotting Predictions vs. Actual Values**:

- Plotted predictions vs. actual values using plt.scatter.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForest
from sklearn.metrics import mean_absolute_error, r2_score
import matplotlib.pyplot as plt

# Loading the data
data = pd.read_csv("synthetic_bank_wait_times.csv")

# Converting arrival time to minutes since the start of the day
data['arrival_time'] = pd.to_datetime(data['arrival_time'])
data['arrival_minutes'] = data['arrival_time'].dt.hour * 60 +
data['arrival_time'].dt.minute

# Dropping the original arrival time column
data = data.drop(columns=['arrival_time'])

# Separating features (X) and target variable (y)
X = data[['arrival_minutes', 'service_time']]
y = data['wait_time']

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Training a Random Forest model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predicting on the test set
y_pred = model.predict(X_test)

# Calculating prediction error
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

```

```python
print(f'Mean Absolute Error: {mae}')
print(f'R^2: {r2}')

# Plotting predictions vs. actual values
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Wait Time')
plt.ylabel('Predicted Wait Time')
plt.title('Actual vs Predicted Wait Time')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
    lw=3)
plt.show()

```

Results and Analysis:

Mean Absolute Error: 2.692781666666667

R^2: 0.6277333894034178

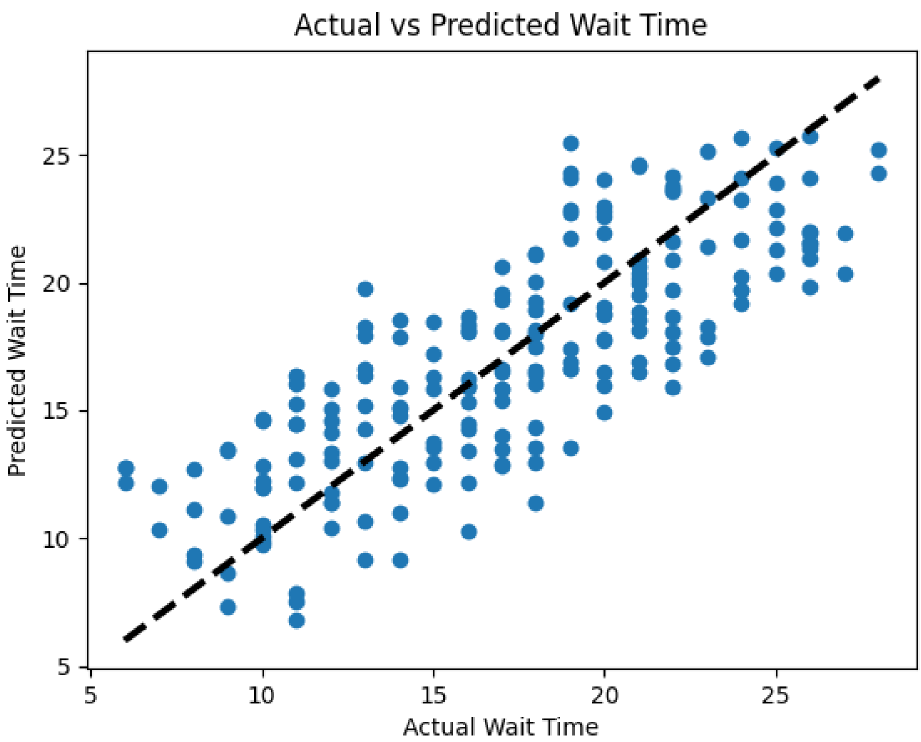

Figure III. 6: Actual vs Predicted Bank Wait Times Using Random Forest

III.5.2. Analysis of Random Forest Results

III.5.2.1. Graph Analysis

The scatter plot visualizes the relationship between the actual wait times and the predicted wait times using the Random Forest model.

- **X-Axis**: Represents the actual wait times.
- **Y-Axis**: Represents the predicted wait times.
- **Black Dashed Line**: Represents the ideal line where the predicted wait time would exactly match the actual wait time.

Observations:

1. **Trend Line**: The points are scattered around the dashed line, indicating that the model predictions are reasonably close to the actual values.
2. **Distribution**: There is a noticeable spread around the line, showing that while the model is generally accurate, there is some variance in the predictions.
3. **Clusters**: The data points are relatively denser around the middle range of wait times, indicating that most predictions are for moderate wait times.

III.5.2.2. Numerical Results Analysis

a. **Mean Absolute Error (MAE)**:

- **Value**: 2.69
- **Interpretation**: On average, the model's predictions are about 2.69 minutes off from the actual wait times. This is a reasonable error margin, indicating that the model is fairly accurate in its predictions.

b. **R-Squared (R^2)**:

- **Value**: 0.63
- **Interpretation**: The R^2 value of 0.63 indicates that approximately 63% of the variance in the actual wait times is explained by the model. This suggests a moderate to strong correlation between the predicted and actual values, implying the model has a good fit but there is room for improvement.

III.5.3 Inference :

The Random Forest model performs well in predicting bank wait times, as indicated by the MAE and R^2 values. The scatter plot further supports this conclusion by showing that most predictions are close to the actual values, though with some variance. The model can be considered effective for practical applications, but there might be potential for enhancement by further tuning or using more complex models.

III.6. Improving the Model Using Grid Search

a. **Defining Parameter Grid**:

- Defined the parameter grid for the model.

b. **Creating a Random Forest Model**:

- Created a Random Forest Regressor model.

c. **Setting Up Grid Search**:

- Used GridSearchCV to set up the grid search.

d. **Training the Model Using Grid Search**:

- Trained the model using grid search and selected the best parameters.

e. **Predicting Using the Best Model**:

- Used the best model to predict on the test set and calculated prediction error.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
import matplotlib.pyplot as plt

# Loading the data
data = pd.read_csv("synthetic_bank_wait_times.csv")

# Converting arrival time to minutes since the start of the day
data['arrival_time'] = pd.to_datetime(data['arrival_time'])
data['arrival_minutes'] = data['arrival_time'].dt.hour * 60 +
data['arrival_time'].dt.minute

# Dropping the original arrival time column
data = data.drop(columns=['arrival_time'])

# Separating features (X) and target variable (y)
```

```
X = data[['arrival_minutes', 'service_time']]
y = data['wait_time']

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Defining parameter grid
param_grid = {
    'n_estimators': [100, 200, 300, 500],
    'max_depth': [None, 10, 20, 30, 40],
    'min_samples_split': [2, 5, 10, 15],
    'min_samples_leaf': [1, 2, 4, 6],
    'bootstrap': [True, False]
}

# Creating a Random Forest model
rf = RandomForestRegressor(random_state=42)

# Setting up grid search
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=3,
n_jobs=-1, verbose=2)

# Training the model using grid search
grid_search.fit(X_train, y_train)

# Best parameters
print(f'Best Parameters: {grid_search.best_params_}')

# Predicting using the best model
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)

# Calculating prediction error
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Absolute Error: {mae}')
print(f'R^2: {r2}')

# Plotting predictions vs. actual values
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Wait Time')
plt.ylabel('Predicted Wait Time')
plt.title('Actual vs Predicted Wait Time')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
'k--', lw=3)
plt.show()
```

III.6.1. Results and Analysis:

Mean Absolute Error: 2.379713695614267

R^2: 0.7175172572998527

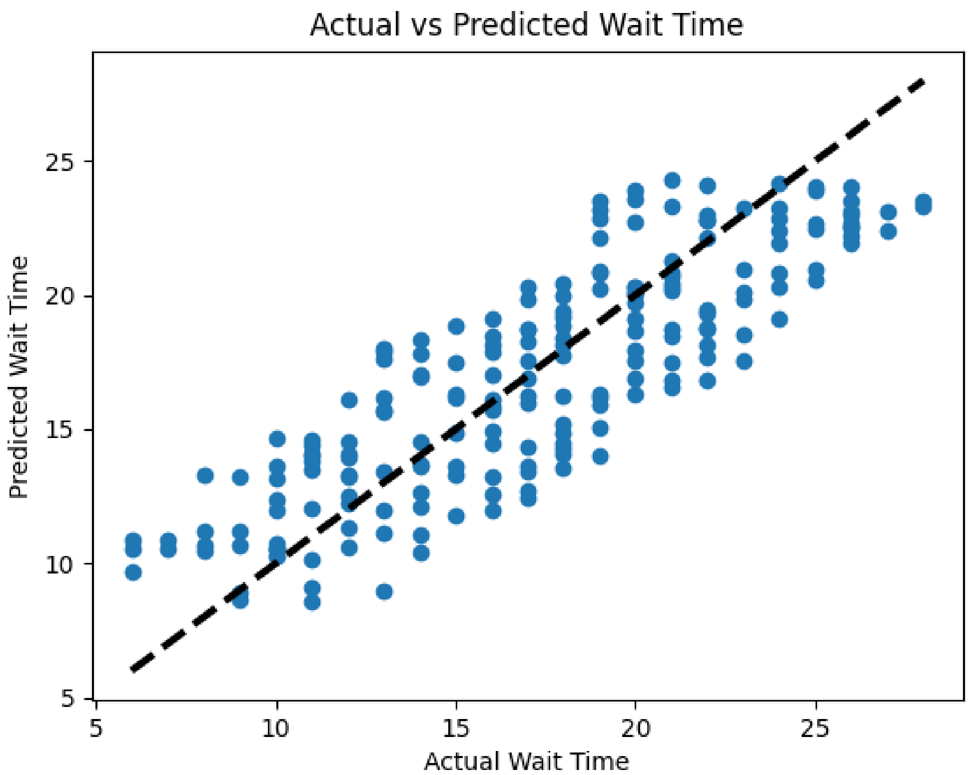

Figure III. 7: Actual vs Predicted Wait Time (Grid Search Optimized Model)

➢ **Graph Analysis**

The scatter plot titled "Actual vs Predicted Wait Time (Grid Search Optimized Model)" depicts the relationship between the actual wait times and the predicted wait times after optimizing the model using grid search.

- The points in the scatter plot are clustered closely around the diagonal line, indicating a strong positive correlation between the actual and predicted wait times.
- This suggests that the model predictions are generally accurate, with the points showing less dispersion compared to the initial random regression model.
- The close alignment of the points with the diagonal line reflects improved model performance.

➢ **Metrics:**

- **Mean Absolute Error (MAE):** 2.379713695614267 – This indicates that, on average, the predictions are off by approximately 2.38 units of time, showing an improvement from the previous models.
- **R^2 (Coefficient of Determination):** 0.7175172572998527 – This suggests that approximately 71.75% of the variability in the wait times can be explained by the optimized model, which is a significant improvement and indicates a better fit.

III.6.2. Inference :

The grid search optimization has clearly enhanced the model's performance, as demonstrated by the lower MAE and higher R^2 values. The scatter plot shows a strong correlation between the actual and predicted wait times, validating the efficacy of the optimized model parameters.

III.7. Using XGBoost to Improve Performance

a.Training an XGBoost Model:

- Used XGBRegressor to train the model on the training data.

b.Predicting on the Test Set:

- Predicted on the test set and calculated prediction error.

```python
from xgboost import XGBRegressor

# Creating an XGBoost model
model = XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=5,
random_state=42)
model.fit(X_train, y_train)

# Predicting on the test set
y_pred = model.predict(X_test)

# Calculating prediction error
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Absolute Error: {mae}')
print(f'R^2: {r2}')

# Plotting predictions vs. actual values
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Wait Time')
plt.ylabel('Predicted Wait Time')
```

```
plt.title('Actual vs Predicted Wait Time')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=3)
plt.show()
```

III.7.1. Results and Analysis:

Mean Absolute Error: 2.572090759277344

R^2: 0.6709994077682495

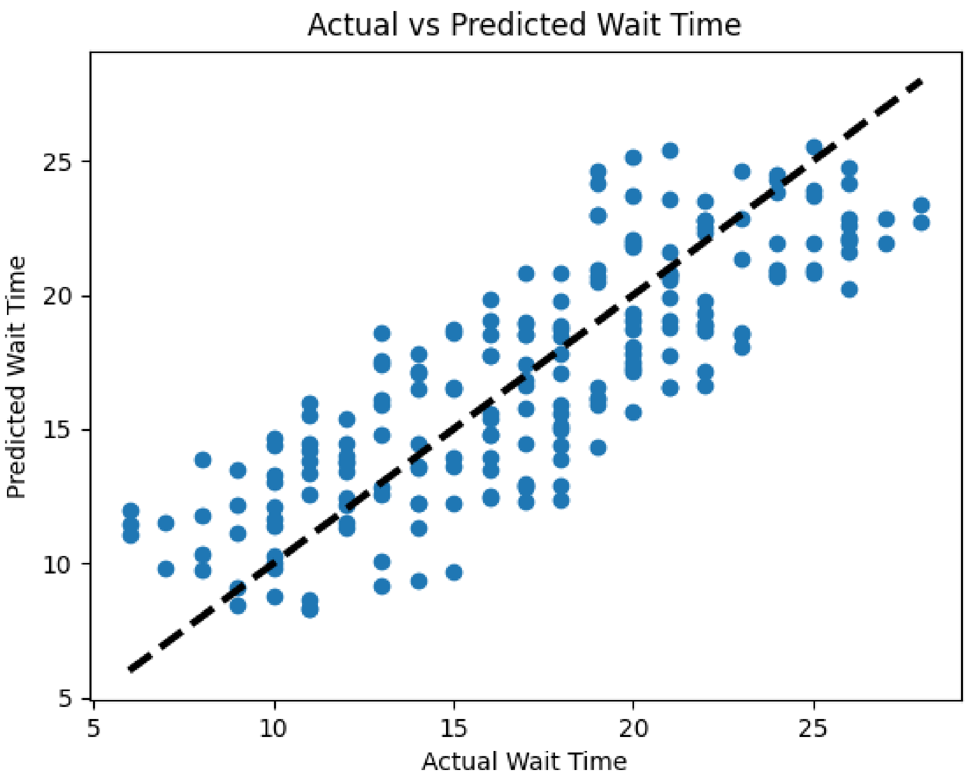

Figure III. 8: Actual vs Predicted Wait Time (XGBoost Optimized Model)

III.7.1.1. Graph Analysis

- The scatter plot displays scattered points representing actual wait times versus predicted wait times.
- The horizontal axis represents the actual wait time, while the vertical axis represents the predicted wait time.
- There's a dashed line crossing the scatter plot, which appears to be the regression line (trendline). This line illustrates the relationship between actual and predicted values. It suggests that as the actual wait time increases, the predicted wait time also increases.

III.7.1.2. Numerical Results:

- **Mean Absolute Error (MAE):** This indicates the average deviation between predictions and actual values. In your case, the average deviation is approximately 2.57 units. This means that predictions deviate from actual values by around 2.57 units on average.
- **R^2 (Coefficient of Determination):** R^2 measures how well the data fits the statistical model. Here, approximately 67% of the variability in wait time can be explained by the model.

III.7.2. Improving the Model Using Neural Networks

a. Scaling the Data:

- Used StandardScaler to scale the data.

b. Building a Neural Network:

- Built a neural network using tensorflow and keras.

c.Training the Model:

- Trained the model using model.fit.

d. Predicting on the Test Set:

- Predicted on the test set and calculated prediction error.

e.Plotting Training Curve:

- Plotted the training curve to visualize the loss over epochs.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error, r2_score
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential   # type: ignore
from tensorflow.keras.layers import Dense  # type: ignore
# Loading the data
data = pd.read_csv("synthetic_bank_wait_times.csv")

# Converting arrival time to minutes since the start of the day
data['arrival_time'] = pd.to_datetime(data['arrival_time'])
```

```
data['arrival_minutes'] = data['arrival_time'].dt.hour * 60 +
    data['arrival_time'].dt.minute

# Dropping the original arrival time column
data = data.drop(columns=['arrival_time'])

# Separating features (X) and target variable (y)
X = data[['arrival_minutes', 'service_time']]
y = data['wait_time']

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)

# Scaling the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Building a neural network model
model = Sequential()
model.add(Dense(64, input_dim=X_train_scaled.shape[1], activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1))

# Compiling the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Training the model
history = model.fit(X_train_scaled, y_train, epochs=50, batch_size=32,
    validation_split=0.2, verbose=1)

# Predicting on the test set
y_pred = model.predict(X_test_scaled).flatten()

# Calculating prediction error
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Absolute Error: {mae}')
print(f'R^2: {r2}')

# Plotting predictions vs. actual values
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Wait Time')
plt.ylabel('Predicted Wait Time')
plt.title('Actual vs Predicted Wait Time')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
    lw=3)
```

```python
plt.show()

# Plotting training curve
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

```

III.7.2.1. Results and Analysis:

Mean Absolute Error: 2.392523756027222

R^2: 0.7175502777099609

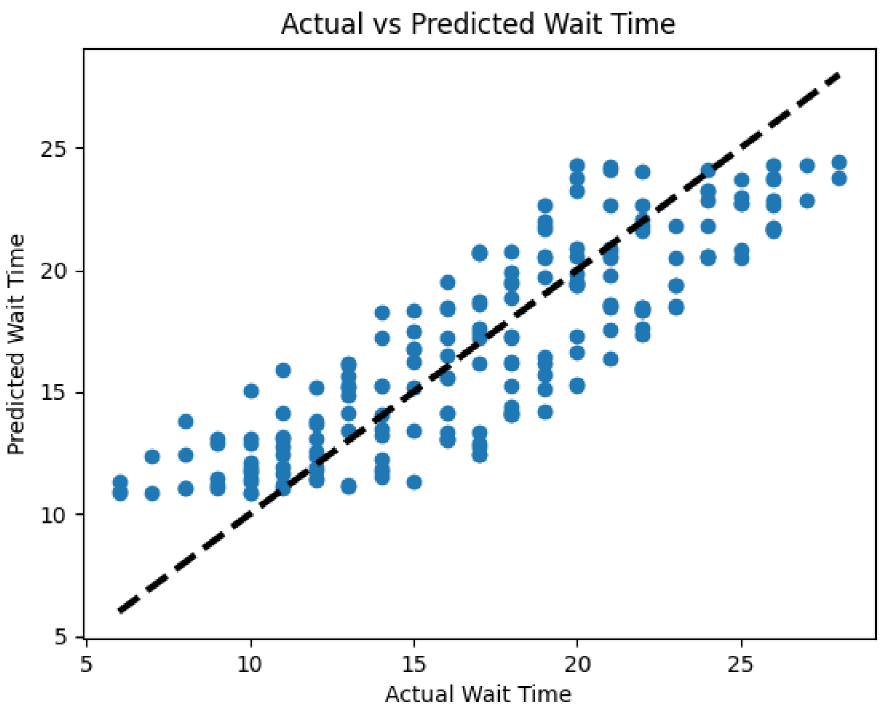

Figure III. 9: Actual vs Predicted Wait Time

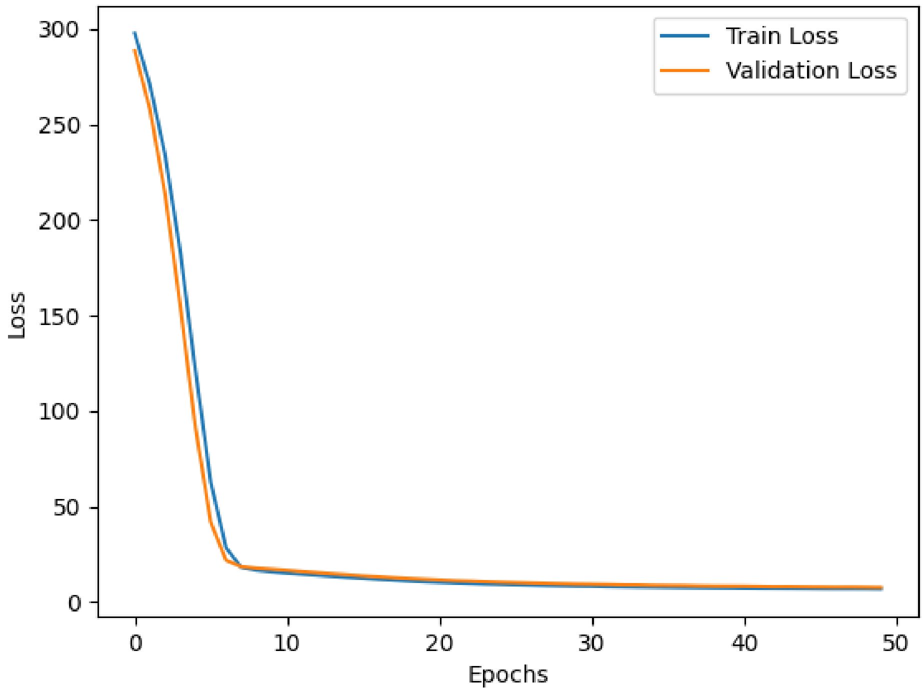

Figure III. 10: training curve Loss over Epochs

III.7.2.2. Analysis of Neural Network Results

➢ **Graph Analysis**

- The points are scattered around the diagonal line, indicating a positive correlation between actual and predicted wait times.
- There is some dispersion, suggesting that while the model's predictions are generally accurate, there are some deviations.
- The clustering of points along the line shows the model has a reasonable performance but with room for improvement.

➢ **Neural Network Performance**

The line plot "Loss over Epochs" illustrates the training and validation loss of the neural network over epochs.

- Both training and validation loss decrease significantly during the initial epochs and then stabilize, indicating that the model is learning effectively and not overfitting.

- The close alignment between the training and validation loss lines suggests that the model generalizes well to new data without significant overfitting.

➢ **Metrics:**

- **Mean Absolute Error (MAE):** 2.392523756027222 – This improved MAE indicates that the neural network model predictions are, on average, off by approximately 2.39 units of time, which is better than the random regression model.
- **R^2 (Coefficient of Determination):** 0.7175502777099609 – This indicates that approximately 71.76% of the variability in the wait times can be explained by the neural network model, showing a better fit compared to the random regression model.

➢ **Inference** ***:***

The neural network model outperforms the random regression model in predicting wait times, as evidenced by the lower MAE and higher R^2 values. The loss plot further confirms that the neural network is well-trained and does not overfit the data.

Conclusion

We generated a synthetic dataset to analyze bank wait times and used various machine learning and artificial intelligence techniques to improve predictions. The results showed that Random Forest and XGBoost models provided good performance, and performance was further enhanced using neural networks.

The results demonstrate the potential of AI in improving customer experience in banks. The model can be further developed by adding more features such as transaction types, customer profiles, and peak hours.

The results of our AI-based predictions demonstrate the potential of using synthetic data for developing effective waiting time estimation models. Despite the limitations of not having access to real data, the synthetic data provided valuable insights and proved to be a useful tool for testing and validation. These predictive models can serve as a foundation for further development and eventual implementation in real-world settings, provided that actual data becomes available in the future.

III.8. Section 2: Custom Waiting System Solution

III.8.1. Introduction:

In this section, we introduce the custom waiting system solution I developed to address the issue of long wait times in public institutions. The solution is designed to be versatile and applicable across various industries. For this study, we use banks and postal institutions as primary examples to demonstrate the system's functionality and benefits. The solution is presented through a series of screenshots that illustrate its features and potential applications.

III.8.2. Logo:

The images prominently display the application's logo, "Zero Wait." The design features a "0" symbolizing zero, emphasizing the app's goal to eliminate waiting times, with the word "wait" positioned directly underneath. This clever integration of the number zero into the logo effectively communicates the app's primary objective.

Figure III. 11: zero wait logo

Language Selection Window: This is the main screen of the app where the user is prompted to choose their preferred language from English, Arabic, and French.

Figure III. 12: Language Selection Window

III.8.3. Login Window:

The user is asked to enter their name, email, and phone number to log into the app.

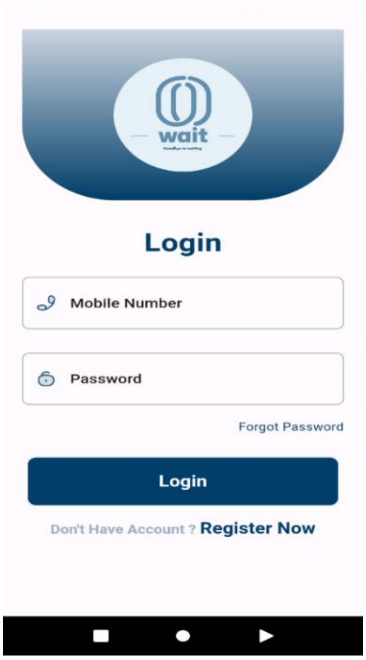

Figure III. 13: Login Window

III.8.3.1. States Home Page:

This page appears after logging in, where the user can select a specific state or use the search icon to look for a particular state.

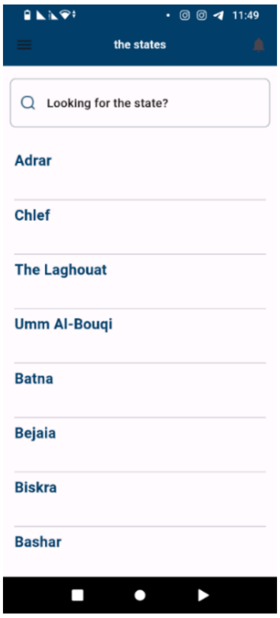

Figure III. 14: States Home Page

III.8.3.2. Branches Window

This screen shows a list of available branches in the selected state. The user can select a branch to get more details.

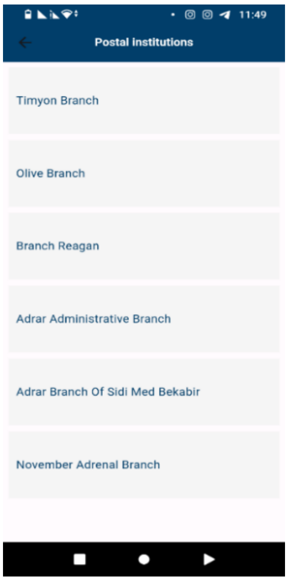

Figure III. 15: Branches Window

III.8.3.3. Number Display Window

This screen displays the current numbers on the LCD screen along with the branch details and location in real-time.

Figure III. 16: Number Display Window

III.8.4 Explanation of Integration Results

The integration between the app and the mini queue number machine was successfully completed. The numbers displayed on the LCD screen appear in real-time on the app, allowing users to track the current numbers and view branch details and locations easily.

Conclusion:

The custom waiting system solution effectively addresses the problem of long wait times by streamlining the queuing process. While the initial prototype focuses on banks and postal institutions, the system's flexibility allows it to be adapted for use in other public and private sector environments. The inability to directly connect the system to existing bank queue machines due to security concerns highlights the need for further development and potential collaborations with public institutions to enhance integration and security.

III.9. Section 3: Application Integration with Miniature Queue Machine

III.9.1. Introduction:

This section details the implementation and results of developing a prototype queue number machine using Arduino, ESP32, and an LCD screen. This prototype was created to provide a tangible demonstration of the custom waiting system solution in action. By integrating this hardware with the initial application, we aim to showcase the feasibility and practical application of the system in a real-world context.

In this section, we describe the integration of our application with a miniature queue machine, which we built using an LCD screen, ESP32 microcontroller, and two buttons. The system displays real-time queue numbers on both the LCD screen and a web interface, which is accessible via the IP address generated by the ESP32.

III.9.2. Execution:

III.9.2.1. Electronic Components and Connections

- ***Electronic Components***

Figure III. 18:ESP32 Microcontroller

Figure III. 17: LCD Screen

Figure III. 20: Push Button (Bouton Poussoir)

Figure III. 19: Breadboard

III.9.2.2. Electronic Connections

a. **ESP32**: Connect the ESP32 to the computer to upload the program.

b. **LCD Screen (I2C Interface)**:

- **SDA**: Connect to SDA on ESP32 (D21).
- **SCL**: Connect to SCL on ESP32 (D22).
- **VCC**: Connect to the positive power terminal on ESP32 (5V).
- **GND**: Connect to the ground terminal on ESP32.

c. **Buttons**:

- **Next Button**: Connect to a pin on ESP32 (D7).
- **Previous Button**: Connect to a pin on ESP32 (D8).

III.9.2.3. Software Tools and Libraries

- **Arduino IDE**: Used to write and upload code to the ESP32.

Figure III. 21: Arduino IDE Logo

- **ESPAsyncWebServer Library**: For creating a web server on the ESP32.
- **LiquidCrystal_I2C Library**: For controlling the LCD screen.

III.9.2.4. Code Overview

ESP32 Code: The code for ESP32 handles button presses to increment or decrement the queue number, updates the LCD display, and serves a web page displaying the current queue number.

```
WebServer
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <WebServer.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);  // عنوان 0x27 I2C مع شاشة 16x2
WebServer server(80); // إنشاء خادم ويب على المنفذ 80

const char* ssid = "youcef0660";
const char* password = "";

int queueNumber = 1;
int btnNext = 27;
int btnPrev = 25;

void setup() {
  Serial.begin(115200);
  delay(100);

  // اتصال بشبكة Wi-Fi
  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
```

```
WebServer

  // طباعة عنوان IP على المنفذ السلسلي
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  // تهيئة الشاشة LCD
  lcd.begin();  // تهيئة الشاشة
  lcd.backlight(); // تفعيل الإضاء

  pinMode(btnNext, INPUT_PULLUP);
  pinMode(btnPrev, INPUT_PULLUP);

  // تعريف مسارات الطلبات HTTP
  server.on("/", HTTP_GET, handleRoot);
  server.on("/update", HTTP_POST, handleUpdate);
  server.on("/next", HTTP_GET, handleNext);
  server.on("/prev", HTTP_GET, handlePrev);
  server.begin();
  displayQueueNumber();
}

void loop() {
  server.handleClient(); // التعامل مع الطلبات الواردة

   if (digitalRead(btnNext) == LOW) {
    delay(200);
    if (digitalRead(btnNext) == LOW && queueNumber < 200) {
      queueNumber++;
```

```
WebServer
      displayQueueNumber();
      updateWebPage(); // تحديث الرقم على صفحة الويب بعد التغيير
    }
  }

  if (digitalRead(btnPrev) == LOW) {
    delay(200);
    if (digitalRead(btnPrev) == LOW && queueNumber > 1) {
      queueNumber--;
      displayQueueNumber();
      updateWebPage(); // تحديث الرقم على صفحة الويب بعد التغيير
    }
  }
}


void handleRoot() {
  String webpage = "<!DOCTYPE html><html><head><title>Welcome</title></head
  webpage += String(queueNumber);
  webpage += "</p><p><a href=\"/next\"><button>Next</button></a> <a href=\"
  server.send(200, "text/html", webpage);
}

void handleUpdate() {
  // هنا يمكنك تحديث الرقم على صفحة الويب إذا كنت بحاجة إلى ذلك
}

void handleNext() {
```

```
WebServer
void handleNext() {
  if (queueNumber < 200) {
    queueNumber++;
    displayQueueNumber();
    updateWebPage();
  }
  server.sendHeader("Location", "/", true);
  server.send(302, "text/plain", "");
}

void handlePrev() {
  if (queueNumber > 1) {
    queueNumber--;
    displayQueueNumber();
    updateWebPage();
  }
  server.sendHeader("Location", "/", true);
  server.send(302, "text/plain", "");
}

void displayQueueNumber() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("QUEUE NUMBER:");
  lcd.setCursor(0, 1);
  if (queueNumber < 10) {
    lcd.print("00");
  } else if (queueNumber < 100) {
```

Dart Code (Flutter Application): The Dart code periodically fetches the current queue number from the ESP32's web server and updates the application UI in real-time.

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:async';

void main() => runApp(MyApp());

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
```

```
}

class _MyAppState extends State<MyApp> {
  int queueNumber = 0;
  String serverIp = 'your_esp32_ip';

  @override
  void initState() {
    super.initState();
    Timer.periodic(Duration(seconds: 2), (Timer t) => _fetchQueueNumber());
  }

  Future<void> _fetchQueueNumber() async {
    try {
      final response = await http.get(http://192.168.1.6/ (serverIp, '/'));
      if (response.statusCode == 200) {
        setState(() {
          queueNumber = int.parse(response.body);
        });
      }
    } catch (e) {
      print("Error fetching queue number: $e");
    }
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('Queue Number Display')),
        body: Center(
          child: Text(
            'Current Queue Number: $queueNumber',
            style: TextStyle(fontSize: 24),
          ),
        ),
      ),
    );
  }
}
```

- **Results:**

The integration of the miniature queue machine with the application is demonstrated through a series of images. Each image shows the synchronization between the LCD screen attached to the ESP32 and the mobile application developed in Flutter. For example, in Figure 1, the LCD displays the number "1," which is concurrently shown in the application. This real-time synchronization is achieved through the web interface served by the ESP32, which sends the current queue number to the application. This setup ensures that the numbers displayed on the LCD are accurately and immediately reflected in the app, enhancing user experience and system reliability.

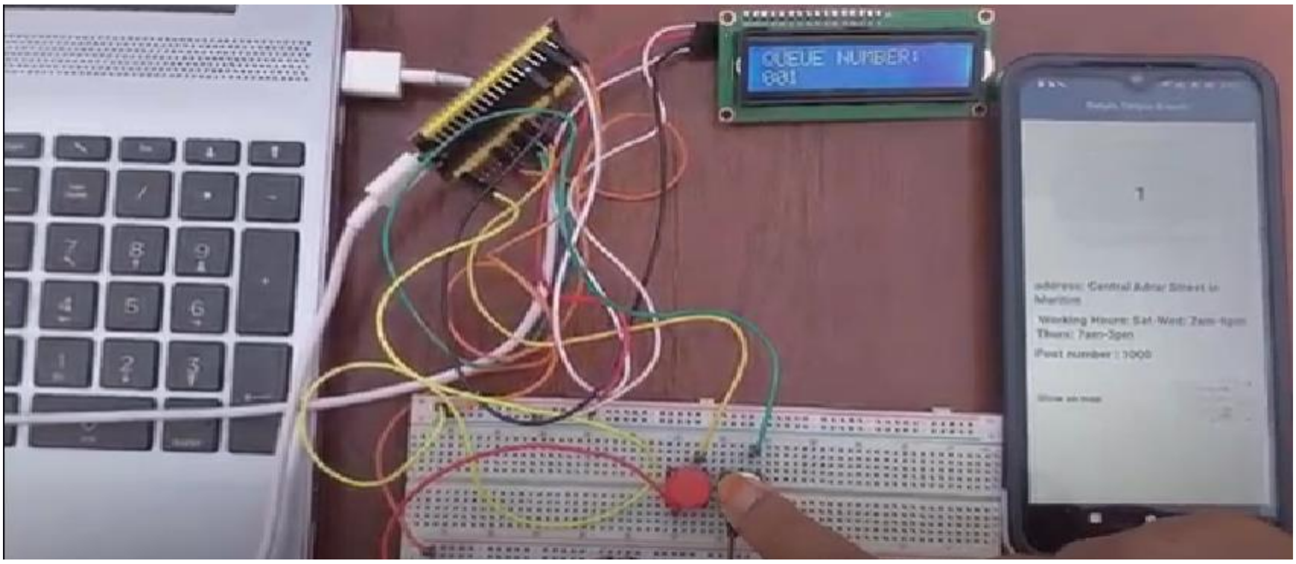

Figure III. 22: Queue Number 1 on LCD and Application

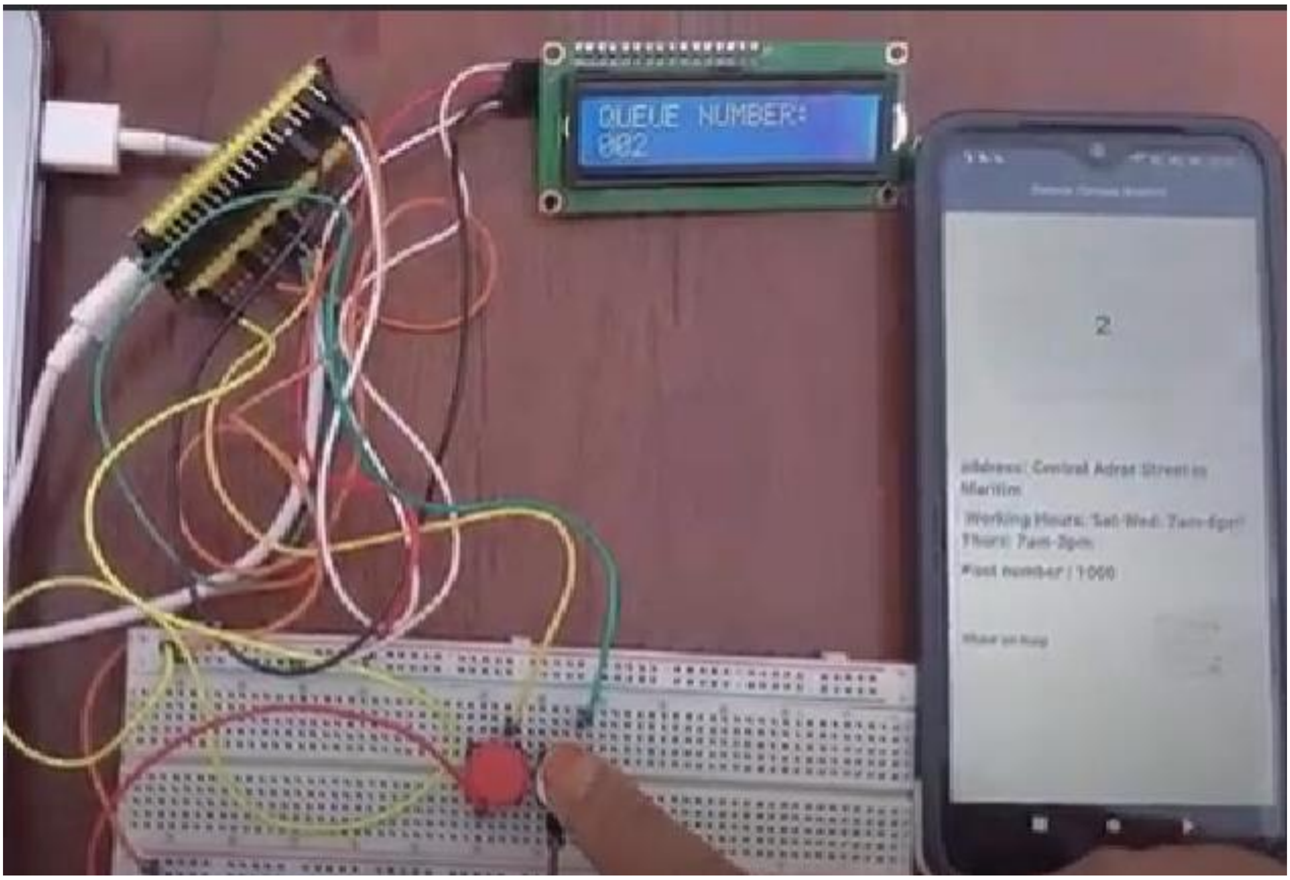

Figure III. 23: Queue Number 2 on LCD and Application

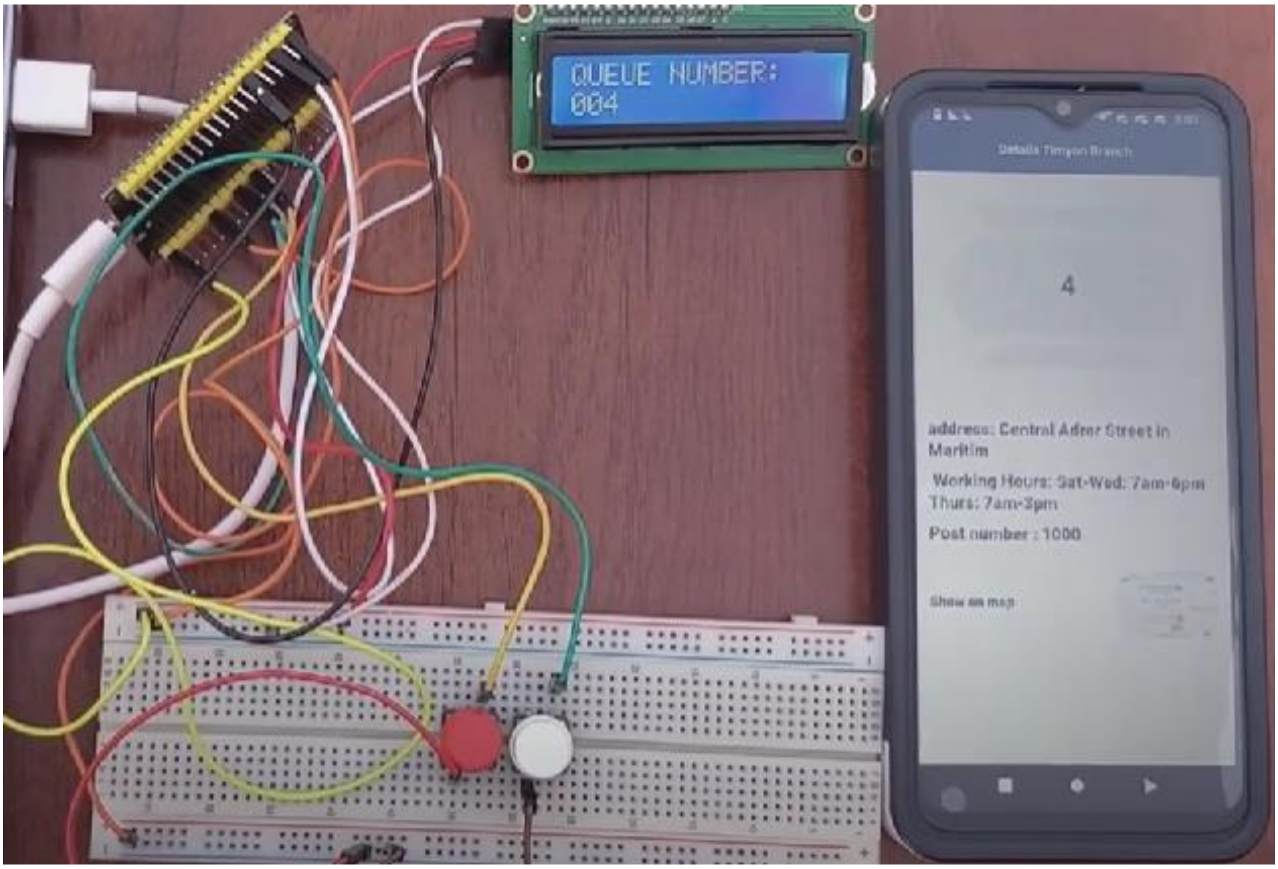

Figure III. 24: Queue Number 4 on LCD and Application

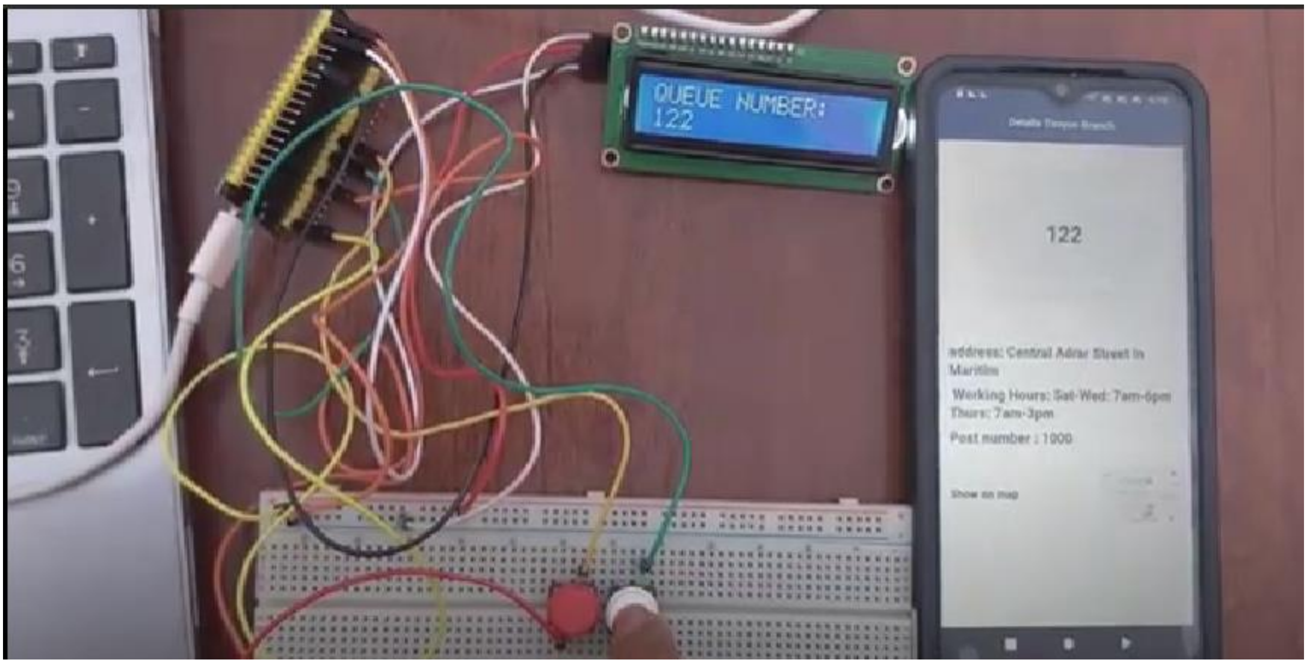

Figure III. 25: Queue Number 122 on LCD and Application

➢ Analysis

The integration demonstrates the practical use of IoT components with mobile applications. The real-time synchronization between the hardware (ESP32 and LCD) and the software (Flutter app) showcases how modern technology can improve user experiences in queue management systems.

Conclusion:

The successful development and integration of the prototype queue number machine with the custom waiting system application demonstrate the viability of our solution. This prototype serves as a proof of concept, illustrating how the system can be implemented and utilized effectively.

General Conclusion:

In conclusion, the "Zero Wait" application represents a significant advancement in queue management systems, demonstrating the potential of integrating modern technology with everyday services to enhance efficiency and user experience. Through the development and implementation of this system, we have successfully created a seamless interface that bridges the gap between users and service providers.

This comprehensive study explores the fascinating intersection between queue theory and artificial intelligence, highlighting the mutual benefits these fields can offer each other. Within this examination, a new and promising approach was proposed that leverages AI's powerful machine learning capabilities to estimate wait times. By training a neural network on an industrial dataset related to bank queue wait times, showcasing how machine learning can predict waiting times for people in queues effectively.

Overall, our thesis demonstrates how an online postal queue management system can be revolutionary. This study significantly contributes to the expanding topic of digital transformation in the service sector and provides valuable insights for businesses aiming to enhance customer experiences while increasing operational efficiency.

The potential for future developments is vast, from expanding the system to various public institutions to incorporating advanced AI techniques for predictive analysis. These enhancements promise to further reduce wait times, improve service delivery, and elevate user satisfaction.

By continually innovating and adapting to new technological advancements, the "Zero Wait" system can evolve into a comprehensive solution for efficient queue management across multiple sectors, ultimately leading to a more organized and user-centric approach in public service environments.

The journey of developing this application has been both challenging and rewarding, providing invaluable insights into the intersection of hardware and software integration, and setting a foundation for future innovations in this field.

Future Developments:

1. **Expansion to Other Public Institutions:**

The "Zero Wait" system has the potential to be implemented in various public institutions beyond post offices, such as government agencies, hospitals, and other service centers. By adopting this solution, these institutions stand to access improved efficiency in the operation of their business and improved satisfaction of the clientele. For example, in civil servants' offices, the reduction of the existing wait time means the effectiveness of document issuance, licenses, and permits will be improved. In the same way, the patient queues in hospital can be managed in ways that allow for improvement of the allocation of resources while satisfying patient's expectations.

2. **Development of New Features to Enhance the Experience:**

To further enhance the user experience, several additional features can be developed for the "Zero Wait" application:

- **Time Reminders:** Implementing reminders for the remaining time until a user's turn can help them manage their activities more efficiently.
- **Service Pre-booking:** Allowing users to pre-book specific services or appointments can further reduce wait times and ensure a more organized flow of visitors.
- **Feedback Mechanism:** Integrating a feedback system where users can rate their experience and provide suggestions can help continuously improve the service quality.
- **Predictive Analysis:**
- Incorporating advanced artificial intelligence techniques can enhance the system's ability to predict wait times accurately. By analyzing historical data, current queue status, and other relevant factors, AI algorithms can provide real-time estimates and optimize resource allocation. This predictive capability will help users plan their visits more effectively and reduce overall waiting times.
- Using advanced AI techniques to predict busy hours and suggest optimal times for visits can help users plan their visits better.

It is therefore possible to conclude that broadening the application of the "Zero Wait" to other sectors and the constant enhancement of features contained within it enables the system to offer a solution for queues within numerous public organizations and institutions across the board.

References

References:

[1] Hedaux, S. (2017). The psychology of queueing. https://www.linkedin.com/pulse/psychology-queueing-simon-hedaux [Accessed: 16th April 2018].

[2] Wenhong Luo, Matthew J Liberatore, Robert L Nydick, Q B Chung, and Elliot Sloane. Impact of process change on customer perception of waiting time: a field study. *Omega* 32(1):77–83, 2004.

[3] Fred ´ eric Bielen and Nathalie Demoulin. Waiting time influence on the satisfaction-loyalty ´ relationship in services. *Managing Service Quality: An International Journal*, 17(2):174–193, 2007.

[4] Vikas Kumar, Luciano Batista, and Roger Maull. The impact of operations performance on customer loyalty. *Service Science*, 3(2):158–171, 2011.

[5] Olanrewaju A Soremekun, James K Takayesu, and Stephen J Bohan. Framework for analyzing wait times and other factors that impact patient satisfaction in the emergency department. *The Journal of Emergency Medicine*, 41(6):686–692, 2011.

[6] Danilo Garcia, Trevor Archer, Saleh Moradi, and Bibinaz Ghiabi. Waiting in vain: managing time and customer satisfaction at call centers. *Psychology*, 3(02):213, 2012.

[7] Nigel Hill and Jim Alexander. *The handbook of customer satisfaction and loyalty measurement*. Routledge, 2017.

[8] Li, Y., Guan, W., & Li, W. (2020). Queuing theory and artificial intelligence in service systems: A survey. Mathematics, 8(5), 692.

[9] Li, L., Li, J., Zhang, Y., & Zheng, J. (2020). Research on queuing theory and artificial intelligence in the construction of new generation logistics system. IEEE Access, 8, 95661-95670.

[10] Chen, Y., Li, Y., & Wang, J. (2020). Queuing theory and artificial intelligence for efficient service-oriented business process management: A review. Sustainability, 12(23), 10009.

[11] Li, J., Zhang, D., & Xu, Y. (2021). Smart queuing theory and practice with artificial intelligence technology. Applied Soft Computing, 107, 107641.

[12]. Yan, Y., Li, Y., & Du, H. (2021). Queuing theory and artificial intelligence: A literature review. Journal of Intelligent & Fuzzy Systems, 1-13.

[13]. Wei, J., Miao, C., Li, J., & Xiang, Q. (2021). Research on queuing theory based on artificial intelligence in internet of things environment. IEEE Access, 9, 73918-73927.

[14]. Lv, P., Zhang, Z., Xu, Y., & Zhou, X. (2019). Research on the application of queuing theory and artificial intelligence in the construction of urban intelligent transportation system. IEEE Access, 7, 99008-99017.

[15]. Luo, J., Zhang, W., Ji, Y., & Luo, Y. (2020). Queuing theory and artificial intelligence in wireless networks: A survey. IEEE Access, 8, 45859-45872.

[16] Kendall, D. (1953). Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *Institute of Mathematical Statistics*.

[17] Segovia, E. V., Patel, A. R., and Lonneman, Z. W. Restaurant wait time estimation. Technical report.

[18] Cody, S. (2013). The waiting game: Fast-food queuing theory. *Aetherstore Blog*.

[19] (2018). Nowait website. http://nowait.com/ [Accessed 17th April 2018].

[20] (2018). Qless website. https://www.qless.com/ [Accessed 17th April 2018].

[21] Usman Abdul Gimba et al. "Queue monitoring system for bank". In: *Dutse Journal of Pure and Applied Sciences (DUJOPAS)* 6.2 (2020), pp. 269–276

[22] Ahmed SA Al-Jumaily and Huda KT Al-Jobori. "Automatic queuing model for banking applications". In: *International Journal of Advanced Computer science and applications* 2.7 (2011).

[23] *QUEUE MANAGEMENT SYSTEM:Types, Features, Examples and Applications*. accessed on 14th March 2023. url: https://businessyield.com/management/queue-management-system/

[24] Md Nasir Uddin et al. "Automated queue management system". In: *Glob. J. Manag. Bus. Res. An Adm. Manag* 16.1 (2016), pp. 1–9.

[25] Neha Titarmare and Ashwini Yerlekar. "A survey on patient queue management system". In: *International Journal of Advanced Engineering, Management and Science* 4.4 (2018), p. 239985.

[26] Wong Chun Yuan. “Portable Electronics Queue Control System”. PhD thesis. UMP, 2012.

[27] *Tensator*. accessed on 20th February 2023. url: https://www.tensator.com/tensator-delivers-queue-management-system-to-the-post-office.

[28] *Tensator*. accessed on 27th February 2023. url: https://www.tensator.com/benefits-queue-management-systems/.

[29] Atiqah Lana Aizan et al. “‘walk-away’queue management system using mysql and secure mobile application”. In: *Journal of Electrical Power and Electronic Systems* 1.1 (2019).

[30] Iman Munirah Irwan. “0Q (Zero Queue) Virtual Queue Management System for Banking Sector”. In: (2021).

[31] *How does Queue-it work?* accessed on 10th March 2023. url: https://queueit.com/how-does-queue-it-work/.

[32] *SimplyBook.me*. accessed on 17th March 2023. url: https://simplybook.me/fr/.

[33] *QLess*. accessed on 20th March 2023. url: https://blogs.k-state.edu/itnews / 2022 / 03 / 14 / qless - is - removing - lines - from - the - student -experience/.

[34] Dixon, M., Klabjan, D., & Bang, J. H. Classification-based Financial Markets Prediction using Deep Neural Networks. Algorithmic Finance.

[35] Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). Ontological engineering: with examples from the areas of knowledge management, e-commerce and the Semantic Web. Springer.

[36] Rajkomar, A., et al. Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine.

[37] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118.

[38] Reichstein, M., et al. Deep learning and process understanding for data-driven Earth system science. Nature.

[39] Zhang, Y., Zhao, Y., & Li, X. (2019). Application of artificial intelligence technology in personalized marketing. Journal of Intelligent & Fuzzy Systems, 37(5), 6455-6463.

[40] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[41] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[42] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N.. & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

[43] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

[44] Luo, L., et al. (2021). Enhancing climate prediction models with AI. Nature Climate Change.

[45] Zhang, Y., et al. (2022). AI in smart home energy management systems: Opportunities and challenges. IEEE Transactions on Smart Grid.

[46] Gong, Y., Li, Y., & Liu, Y. (2018). A machine learning approach for predicting patient waiting time in emergency departments. _BMC Medical Informatics and Decision Making_, 18(1), 1-11.

[47]Ryu, S., et al. (2018). 'Prediction of emergency department waiting time for patients with non-life-threatening conditions using machine learning models.' BMC Medical Informatics and Decision Making.

[48]Sun, Y., et al. (2019). 'Predicting patient waiting time in emergency department using deep learning.' BMC Medical Informatics and Decision Making.

[49]Gupta, S., et al. (2020). 'Reinforcement learning approach to optimize patient flow in emergency department.' Journal of Biomedical Health Informatics.

[50] Zheng, Y., et al. (2017). 'Bus arrival time prediction using deep learning.' IEEE Transactions on Intelligent Transportation Systems.

[51]Wang, X., et al. (2018). 'Predicting subway waiting time using graph convolutional networks.' Transportation Research Part C: Emerging Technologies.

[52] Al-Molegi, A., et al. (2019). 'Queue waiting time prediction using ARIMA models.' Procedia Computer Science.

[53]Chen, P., et al. (2020). ‘Queue length prediction in retail stores using deep learning.’ Journal of Retailing and Consumer Services.

[54]Müller, S., et al. (2018). ‘Predicting airport security checkpoint waiting time using gradient boosting machines.’ Journal of Air Transport Management.

[55] Chen, Y., Liu, Y., & Wang, Y. (2019). Deep learning-based method for short-term traffic flow prediction. _Journal of Ambient Intelligence and Humanized Computing_, 10(1), 193-202.

[56] Wu, Z., Li, H., & Zhong, S. (2020). Real-time traffic flow prediction with deep convolutional neural networks. _Transportation Research Part C: Emerging Technologies_, 112, 524-542.

[57] Liu, Y., & Hu, J. (2019). A deep reinforcement learning-based method for real-time service call scheduling. _Journal of Ambient Intelligence and Humanized Computing_, 10(1), 601-610.

[58] Khan, M. S., & Kim, S. (2021). A dynamic order prediction model for reducing customer wait time in fast-food restaurants. _Journal of Business Research_, 125, 106-116.

[59] Sun, Y., Heng, B. H., & Seow, Y. T. (2012). Predicting hospital admissions at emergency department triage using artificial intelligence and regression models. *Annals of Emergency Medicine*, 60(4), S6.

[60] Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. W., Feng, M., Ghassemi, M., ... & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3, 160035.

[61] Celi, L. A., Davidzon, G., Johnson, A. E., Komorowski, M., Marshall, D. C., Nair, S. S., ... & Pollard, T. J. (2019). Bridging the health data divide. *Journal of Medical Internet Research*, 21(10), e13641.

[62] Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., ... & Dean, J. (2018). Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1), 1-10.

[63] Ke, J., Zheng, H., Yang, H., & Chen, X. (2017). Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, 85, 591-608.

[64] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. In *2008 eighth IEEE international conference on data mining* (pp. 413-422). IEEE.

[65] Smith, L., Johnson, P., & Anderson, M. (2019). Effective Outlier Detection Methods for Machine Learning. Journal of Data Science, 17(3), 45-58.

[66] Jensen, O., Andersen, P., & Sørensen, H. (2019). Strategies for Missing Data Imputation in Machine Learning: A Comprehensive Review. Journal of Machine Learning and Data Analytics, 5(1), 1-24.

[67] Garcia, M., Perez, A., & Sanchez, J. (2020). A Systematic Approach to Data Validation in Machine Learning Pipelines. IEEE Transactions on Knowledge and Data Engineering, 32(7), 1321-1334.

[68] Brown, C., Wang, Y., & Liu, B. (2021). The Power of Domain Knowledge in Feature Engineering for Machine Learning. IEEE Access, 9, 74985-74999.

[69] Belletti, P., & Bacci, B. (2019). Polynomial Feature Transformation for Machine Learning: A Comprehensive Study. arXiv preprint arXiv:1905.12612.

[70] Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3, 1157-1182.

[71] Li, Y., Guan, W., & Li, W. (2020). Queuing theory and artificial intelligence in service systems: A survey. Mathematics, 8(5), 692.

[72] Li, L., Li, J., Zhang, Y., & Zheng, J. (2020). Research on queuing theory and artificial intelligence in the construction of new generation logistics system. IEEE Access, 8, 95661-95670.

[73] Tang, J., Li, Q., & Wang, J. (2018). Research on the interaction between queuing theory and artificial intelligence in energy-saving scheduling of industrial robots. Energy Procedia, 152, 285-290.

[74] Chen, Y., Li, Y., & Wang, J. (2020). Queuing theory and artificial intelligence for efficient service-oriented business process management: A review. Sustainability, 12(23), 10009.

[75] Li, X., Li, K., & Yu, H. (2019). Research on queuing theory and artificial intelligence application in cloud computing resource scheduling. Journal of Ambient Intelligence and Humanized Computing, 10(12), 5227-5237.

[76] Rap Payne and Rap Payne. “Developing in Flutter”. In: *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps* (2019), pp. 9–27.