Reducing Doctor Idle Time with Machine Learning and Genetic Algorithms via Patient Attendance Classification..

by

Jonathan Hanley

This thesis has been submitted in partial fulfillment for the degree of Master of Science in Artificial Intelligence

in the
Faculty of Engineering and Science
Department of Computer Science

May 2023

Declaration of Authorship

This report, Reducing Doctor Idle Time with Machine Learning and Genetic Algorithms via Patient Attendance Classification.., is submitted in partial fulfillment of the requirements of Master of Science in Artificial Intelligence at Munster Technological University Cork. I, Jonathan Hanley, declare that this thesis titled, Reducing Doctor Idle Time with Machine Learning and Genetic Algorithms via Patient Attendance Classification..and the work represents substantially the result of my own work except where explicitly indicated in the text. This report may be freely copied and distributed provided the source is explicitly acknowledged. I confirm that:

- This work was done wholly or mainly while in candidature Master of Science in Artificial Intelligence at Munster Technological University Cork.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Munster Technological University Cork or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________

Date: ______________________________

MUNSTER TECHNOLOGICAL UNIVERSITY CORK

Abstract

Faculty of Engineering and Science
Department of Computer Science

Master of Science

by Jonathan Hanley

Optimal scheduling of patient appointments in healthcare settings is crucial to reducing wait times and improving patient experience. This thesis investigates the use of no-show appointment classification to improve appointment scheduling. The research question is: How can no-show appointment classification be used to improve appointment scheduling? The study aims to review existing literature on patient no-show predictions and optimal scheduling in healthcare, develop a predictive model using machine learning techniques, and evaluate its performance compared to existing methods. Additionally, an optimal scheduling algorithm will be developed and tested on real-world patient data to assess its effectiveness in reducing wait times. The results of this study have practical implications for healthcare providers, and potential areas for future research will be discussed. This thesis contributes to the ongoing effort to improve healthcare service delivery by leveraging predictive analytics to optimize patient scheduling.

Acknowledgements

I would like to extend my deepest appreciation to my supervisor, Dr. Alex Vakaloudis, for his unwavering support and expert guidance throughout my thesis journey. His valuable insights, constructive feedback, and encouragement were instrumental in shaping the direction of this work, and I am grateful for the opportunity to learn from him.

Contents

Declaration of Authorship **i**

Abstract **ii**

Acknowledgements **iii**

List of Figures **vi**

List of Tables **vii**

1 Introduction **1**
1.1 Motivation . . . 2
1.2 Contribution . . . 2
1.3 Structure of This Document . . . 3
1.4 Project Plan . . . 4
1.5 Risk Mitigation . . . 4

2 Literature Review **6**
2.1 Introduction . . . 6
2.2 Background . . . 6
2.3 Methods . . . 7
2.4 Results . . . 8
2.5 Conclusion . . . 11

3 Design **13**
3.1 Problem Definition . . . 13
3.2 Objectives . . . 14
3.3 Requirements . . . 14
3.4 Design . . . 15
3.4.1 Attendance Prediction Model . . . 15
3.4.2 Scheduling Model . . . 16
3.4.3 Model Integration . . . 17

4 Implementation **18**
4.1 Appointment Duration Dataset . . . 18

4.2 Baseline Scheduler 20
4.3 Overlapping Scheduler 20
4.4 Fitness/ Scoring Function 21
4.5 Attendance Classifier 22
4.5.1 Preprocessing 22
4.5.2 Class Imbalance 23
4.5.3 Feature Engineering 24
4.5.4 Model Creation 24
4.6 Genetic Algorithm 26
4.6.1 Mutation 26
4.6.2 Selection Operations 27
4.6.3 Genetic Algorithm 28

5 Testing and Evaluation **31**
5.1 Attendance Classifier 31
5.1.1 Model Generalization: Over-fitting and Under-fitting 31
5.1.2 Real World Accuracy 32
5.2 Scheduler 32
5.2.1 Scoring Function 33
5.2.2 Scoring of Non-Classification Algorithms 33
5.2.3 Schedule Creation Timing 34
5.3 Scheduler With Classifier 34
5.3.1 Fitness Score 35
5.3.2 Schedule Creation Time 35

6 Discussion and Conclusions **37**
6.1 Discussion 38
6.2 Conclusion 38
6.3 Future Work 40

Bibliography **42**

List of Figures

1.1 Gantt Chart for this Project 5

2.1 A number of papers matched the search term ”Al medical appointment attendance” over the past few years. 10

2.2 A number of papers matching the search term ”medical appointment scheduling AI” over the past few years. 10

3.1 Appointment Attendance Prediction Model Diagram 16

3.2 Diagram of the proposed Genetic Algorithm for Scheduling 16

3.3 Diagram For Integrating Classification Model with GA Scheduler 17

4.1 Appointment Duration Probabilities 20

4.2 Frequency of appointment day 23

4.3 Neural Network containing 1 input layer, 3 hidden layers and 1 output layer 25

5.1 Model Training History 32

5.2 Fitness Scores Without Classifier 34

5.3 Fitness Timing Without Classifier 34

5.4 Fitness Score With Classifier 35

5.5 Fitness Timing With Classifier 36

5.6 Fitness Timing With Classifier 36

List of Tables

2.1 Table with paper information, year, accuracy, dataset name, dataset size, and algorithms used. 12

4.1 Accuracy of Machine Learning Algorithms 25

For my mother and late father. . .

Chapter 1

Introduction

In recent years, healthcare organizations have been facing a significant challenge regarding patient no-shows, which has become a significant issue that can negatively impact the quality of care provided. Patient no-shows can lead to increased wait times, wasted resources, and decreased patient satisfaction. Despite the various interventions that have been used to reduce no-shows, such as reminder systems and patient education, the effectiveness of these methods has been limited.

To address this issue, machine learning techniques have emerged as a promising solution to predict patient no-shows and improve scheduling in healthcare. Therefore, this thesis aims to investigate the potential of no-show appointment classification in improving appointment scheduling. Specifically, this research seeks to develop a predictive model using machine learning techniques that can accurately predict patient no-shows. The performance of the developed model will be evaluated, and its effectiveness will be compared with existing methods.

Moreover, this study aims to develop an optimal scheduling algorithm that considers patient no-show predictions and minimizes wait times for patients and their family members. The algorithm will be tested on real-world patient data to assess its effectiveness in reducing wait times. To achieve these goals, this thesis will conduct a comprehensive review of the existing literature on patient no-show predictions and optimal scheduling in healthcare.

This review will help provide an understanding of the state of the art in this field and identify gaps in existing research. The findings of this literature review will then be used to develop the predictive model and optimal scheduling algorithm, which can have significant practical implications for healthcare organizations. The results of this research can help healthcare providers reduce patient no-shows and improve scheduling efficiency, leading to better patient outcomes.

Additionally, this study can contribute to the growing body of research on machine learning techniques in healthcare. The results of this research can also suggest potential areas for future work in this field. In conclusion, this thesis aims to develop a predictive model and optimal scheduling algorithm that can help healthcare organizations improve patient outcomes by reducing no-shows and minimizing wait times.

1.1 Motivation

Having witnessed the struggles of family members battling cancer, I have developed a profound motivation to explore and address the challenges faced by patients during their healthcare journey. The issue of long wait times for appointments and treatments is particularly distressing for patients and their loved ones, and when patients fail to show up for their appointments, it only exacerbates the problem. This situation can lead to increased stress and anxiety due to uncertainty and delays, which can negatively impact the patient's mental and physical health. Thus, my desire to use patient no-show predictions to optimize scheduling in healthcare is driven by the opportunity to improve patient outcomes, reduce wait times, and alleviate the stress and anxiety associated with the uncertainty and unpredictability of the scheduling process

1.2 Contribution

1. Investigate existing literature on patient no-show predictions and optimal scheduling in healthcare.

2. Develop a predictive model for patient no-shows using machine learning techniques.

3. Evaluate the performance of the predictive model and compare it to existing methods.

4. Develop an optimal scheduling algorithm that takes into account patient no-show predictions and minimizes wait times for patients and family members.

5. Test the optimal scheduling algorithm on real-world patient data and evaluate its effectiveness in reducing wait times.

6. Discuss the practical implications of the research and suggest potential areas for future work.

1.3 Structure of This Document

This thesis is structured into five main chapters. Chapter 1 provides an introduction to the research topic, problem statement, the research aims, objectives, and research questions. Chapter 2 provides a literature review that identifies and evaluates the existing studies and theories related to the research topic. Chapter 3 presents the design of the proposed solution, including the problem definition, objectives, requirements, and the detailed design of the proposed models. Chapter 4 describes the implementation of the proposed models, including data preprocessing, model development, and integration. Finally, Chapter 5 presents the testing and evaluation of the proposed models, including the experimental setup, results, and discussion.

Chapter 2 starts with an introduction that presents the research aims and objectives. It then proceeds with a background that provides an overview of the research topic, its history, and its relevance. The chapter then describes the methods used to review the literature and select the relevant studies. Finally, the chapter presents the results of the literature review and identifies the research gaps that the proposed research aims to address.

Chapter 3 presents the design of the proposed solution, starting with the problem definition, followed by the objectives and requirements. The chapter then describes the detailed design of the proposed models, including the attendance prediction model, scheduling model, and model integration.

Chapter 4 describes the implementation of the proposed models. It starts with the description of the appointment duration dataset, followed by the development of the baseline and overlapping schedulers. The chapter then presents the fitness/scoring function and the attendance classifier, including data preprocessing, class imbalance, feature engineering, and model creation. Finally, the chapter describes the genetic algorithm used to optimize the scheduler.

Chapter 5 presents the testing and evaluation of the proposed models. It starts with the evaluation of the attendance classifier, including the model generalization, over-fitting, and under-fitting. The chapter then presents the evaluation of the scheduler, including the scoring function and real-world accuracy. Finally, the chapter concludes with a discussion of the results and their implications.

1.4 Project Plan

Effective project management is essential for the successful and timely completion of a research project, and one of the key aspects of project management is the management of project timelines and dependencies. The use of a Gantt chart is a widely accepted and highly effective tool for project management, as it facilitates the tracking of progress, identification of potential delays and obstacles, and enables timely interventions to be made to address these issues. The visualization of the entire project timeline also helps to manage the overall project and ensure that milestones and deadlines are met.

To further enhance the efficiency and effectiveness of the research project, a sprint-based approach will be used, dividing the project work into one-week sprints aimed at achieving specific aspects of the writing and implementation process. This approach ensures that project tasks are well-defined and progress towards the research goals is continuous and monitored. A solo worker can benefit from a sprint-based approach as it helps to break down a project into smaller, manageable tasks that can be achieved within a specific timeframe. This enables the worker to focus on specific tasks and make progress towards achieving their goals, while also ensuring that their time and resources are used efficiently. By setting clear objectives and timelines for each sprint, a solo worker can better manage their workload and ensure that they are making consistent progress towards their overall research goals.

In summary, the use of a Gantt chart and the implementation of a sprint-based approach are critical components of the project management plan for this research project. These tools ensure that the project is completed within the planned timeline and that all tasks and dependencies are managed effectively to achieve the desired research outcomes. The Gantt chart for this project is presented in Figure 1.1. The combination of these tools provides a solid framework for managing the project and helps to ensure its success.

1.5 Risk Mitigation

This research project has identified several risks that could affect its progress. The foremost risk is related to the availability of a suitable dataset. The absence of such a dataset could hinder the development of a classifier to predict patient attendance and impede the creation of a realistic scheduling algorithm. To mitigate this risk, extensive efforts were undertaken before the project started to ensure that appropriate datasets for both classification and scheduling were identified and procured.

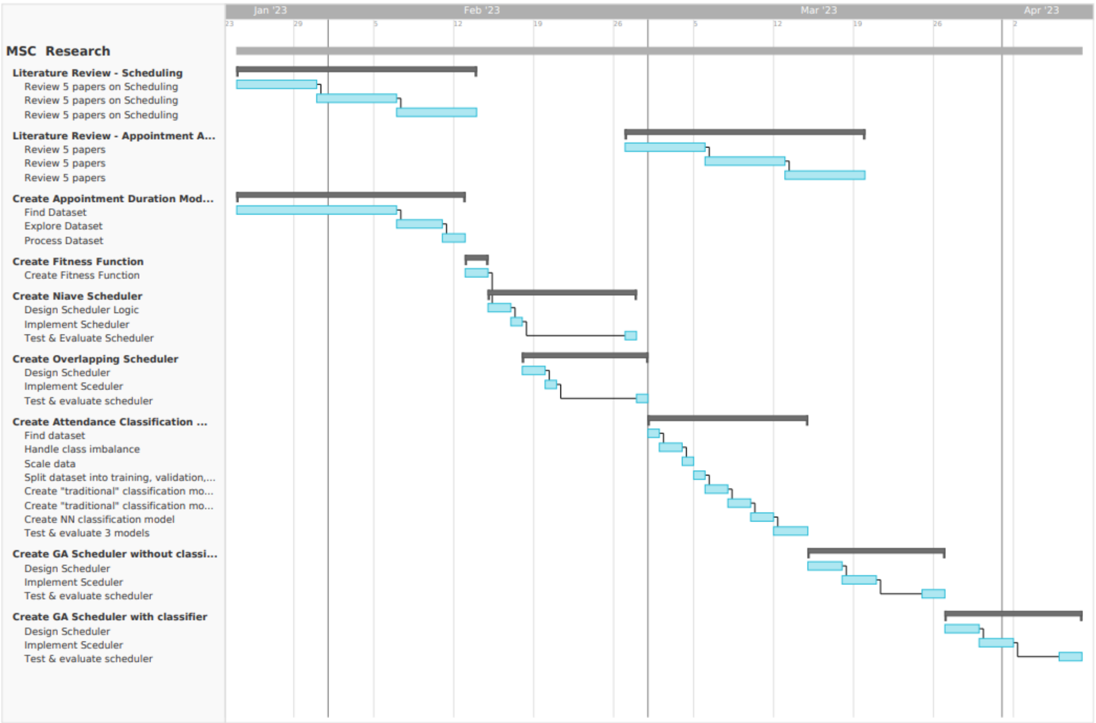

FIGURE 1.1: Gantt Chart for this Project

The second significant risk concerns the potential loss of access to academic papers for an extended period, which could cause delays in the research process. To mitigate this risk, the researcher intends to leverage their prior access to academic papers through their undergraduate login if such access is no longer available.

Finally, the third risk pertains to the possibility of the research machine used for developing the solution breaking down. Although the risk is low, it remains a potential risk. The mitigation strategy involves using a spare laptop, should the need arise, to ensure that research progress is not hindered.

Chapter 2

Literature Review

2.1 Introduction

The objective of this literature review is to explore the current research on appointment scheduling and no-show prediction in healthcare using a thematic approach. The review aims to comprehensively analyze the various traditional and advanced techniques employed in appointment scheduling optimization, including the use of machine learning algorithms to forecast patient no-shows. Through a critical examination of the strengths and limitations of each study, this review intends to identify gaps in the literature and highlight areas where further research can be conducted to improve the scheduling process in healthcare. Additionally, this review seeks to provide insight into the practical implications of these techniques in improving patient outcomes and enhancing the quality of healthcare services.

2.2 Background

In recent years, the demand for healthcare services has increased significantly due to several factors, including an ageing population, the prevalence of chronic diseases, and advancements in medical technology. As a result, healthcare providers face significant challenges in providing timely and efficient healthcare services to patients while maximizing clinic utilization and minimizing wait times. One of the key factors contributing to these challenges is the complexity of the appointment scheduling process.

Traditional appointment scheduling methods, such as the OLAS method and the MSM, have been widely used in healthcare settings for many years. The OLAS method involves scheduling multiple appointments for the same time slot, with the expectation that some patients will not show up, while the MSM uses historical data to predict patient arrivals and schedule appointments accordingly. While these methods are simple and easy to implement, they may not account for the unique needs and preferences of individual patients, leading to inefficiencies and longer wait times.

More recently, advanced methods such as machine learning and optimization techniques have gained popularity in healthcare settings due to their ability to improve the accuracy and efficiency of appointment scheduling systems. Machine learning algorithms can be trained on large datasets to predict patient arrivals and optimize appointment schedules based on various factors such as patient characteristics, physician availability, and clinic capacity. Optimization techniques can also be used to minimize wait times and improve clinic utilization by identifying the optimal appointment schedule given a set of constraints.

Despite the benefits of advanced methods, there are still limitations and challenges that need to be addressed. For example, machine learning algorithms require large amounts of data to train accurately, which may not be available in all healthcare settings. Additionally, these methods may require significant computational resources and expertise to implement and maintain, which may not be feasible for smaller clinics or hospitals with limited resources.

In conclusion, appointment scheduling in healthcare is a complex process that requires careful consideration of various factors, including patient preferences, physician availability, and clinic capacity. While traditional methods are still widely used due to their simplicity and ease of implementation, advanced methods such as machine learning and optimization techniques offer significant benefits in terms of accuracy and efficiency. However, further research is needed to address the limitations and challenges of these methods and to develop practical solutions that can be implemented in healthcare settings of different sizes and resource levels.

2.3 Methods

The research question for this literature review is "How can no-show appointment classification be used to improve appointment scheduling?"

A comprehensive search of the literature was conducted using Google Scholar, PubMed and Research Gate search engines. The following keywords and phrases were used: "Appointment Scheduling", "Appointment Scheduling AI", "Appointment Scheduling No-Show", and "Appointment attendance prediction". The search was limited to papers published within the past 5 years.

The inclusion criteria for this review were papers that focused on no-show appointment classification and its use in improving appointment scheduling. The papers were required to be published in English and have a clear abstract that addressed the research question.

The exclusion criteria for this review were papers that were not written in English or did not have a clear abstract that addressed the research question. Papers that were not published within the past 5 years were also excluded.

The abstract of each paper was screened to determine if it met the inclusion criteria. Papers that did not meet the inclusion criteria were excluded from the review.

The limitations of this review include the exclusion of papers published in languages other than English and the potential for bias in the selection and extraction of data. Additionally, the search was limited to papers published within the past 5 years, which may have excluded relevant literature.

2.4 Results

The articles reviewed provide a broad overview of various approaches to predicting and optimizing hospital attendance and scheduling, ranging from traditional methods such as the Overlapping Scheduling (OLAS) method and the Markovian Scheduling Method (MSM)[1], to more advanced machine learning algorithms such as Random Forest, AdaBoost[2], Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Artificial Neural Networks (ANNs), and Ant Colony Optimization (ACO)[3].

The studies suggest that machine learning algorithms can achieve high levels of accuracy in predicting hospital attendance, with Random Forest and AdaBoost models achieving 82% and 84% accuracy respectively in the study by Nelson et al. (2019)[2] and JRip and Hoeffding Trees models achieving 76.44% and 77.13% accuracy respectively in the study by AlMuhaideb et al. (2019)[4]. However, these studies do not address the challenge of class imbalance or the methodology used for splitting the dataset into training and test partitions.

The studies also demonstrate the effectiveness of optimization algorithms such as genetic algorithms in reducing both patients wait time and doctor idle time, as demonstrated in the study by Alizadeh et al. (2020)[5]. However, the assumption of a constant appointment duration and full patient attendance in the study dataset limits the realism of the results.

A study by Lindsay M. Guzek BS, Shelley D. Gentry ASM, Meredith R. Golomb MD indicateds that the monthly loss for a single pediatric neurology outpatient clinic ranged from \$15,652.33 to \$27,042.44 in October 2013 and January 2014[6]. With the total annual cost being \$257,724.57. Another study by Alex J. Mitchell and Thomas Selmes indicates that the rate of missed appointments is up to 50% in psychiatric services[7].

The review also identifies several areas where the literature falls short. The study by Nelson et al. (2019)[2] lacks a discussion of the process of feature engineering and feature selection, and the underlying dataset used in the study was not made available for review, precluding an assessment of potential biases. Similarly, the study by Harvey et al. (2018)[8] does not address the handling of the majority class of patients who show up for their appointments, potentially skewing the accuracy of the model.

Overall, the studies reviewed provide a valuable contribution to the field of hospital attendance and scheduling, highlighting the potential of machine learning algorithms and optimization techniques in improving the efficiency and effectiveness of healthcare services. However, further research is needed to address the limitations of the current studies and to develop more robust and accurate models for predicting and optimizing hospital attendance and scheduling.

During the review of existing literature a commonly used data-set was used for attendance classification was Medical Appointment No Shows[9]. This was used by Sara Alshaya, Andrew McCarren & Amal Al-Rasheed[10] and Alshammari, Abdulwahhab and Almalki, Raed and Alshammari, Riyad[11].

In the field of healthcare, appointment scheduling is an important task that can affect the efficiency and quality of care provided to patients. Researchers have developed various methods for creating appointment schedules, including using machine learning algorithms and optimization techniques. However, during the literature review, it was found that there was no existing work that used readily available datasets for appointment scheduling.

For example, Zacharias et al. proposed a joint panel sizing and appointment scheduling approach for outpatient care [12]. However, they used an artificially created dataset that was not based on real data. Similarly, Cayirli and Veral proposed a scheduling system for ambulatory care services [13], but they used private datasets that are not available to the general public.

The lack of readily available datasets for appointment scheduling can be a challenge for researchers who wish to develop and test new scheduling approaches. Without access to real-world data, it can be difficult to accurately simulate the complexities of appointment scheduling in a healthcare setting. However, efforts are being made to address this issue, such as the creation of publicly available datasets like the Medical Appointment No Shows dataset mentioned earlier [9].

The field of medical appointment scheduling has seen a growing interest in recent years, as evidenced by the increasing number of papers published on the topic. Figure 2.1 shows the trend in the number of papers matching the search term ”Al medical appointment attendance” over the past few years, while Figure 2.2 shows the trend in the number of papers matching the search term ”medical appointment scheduling AI”.

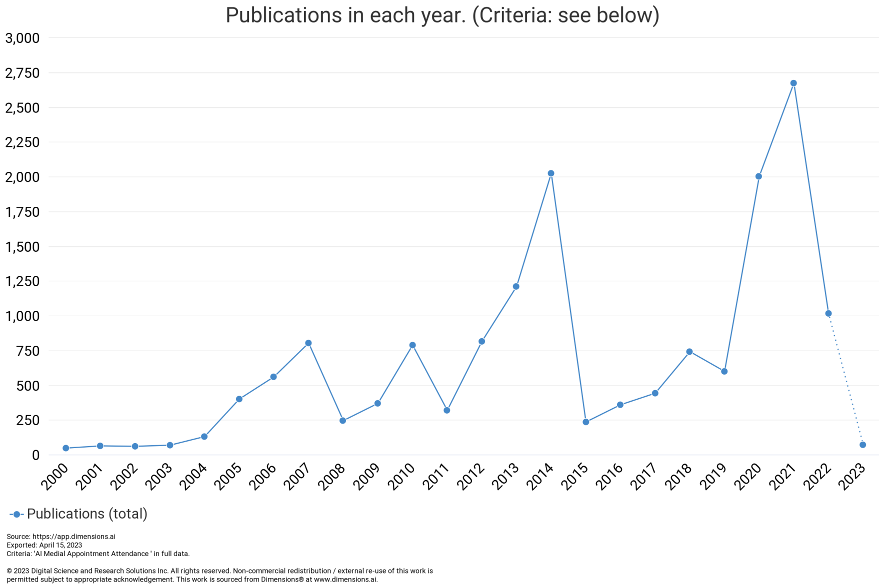

FIGURE 2.1: A number of papers matched the search term ”Al medical appointment attendance” over the past few years.

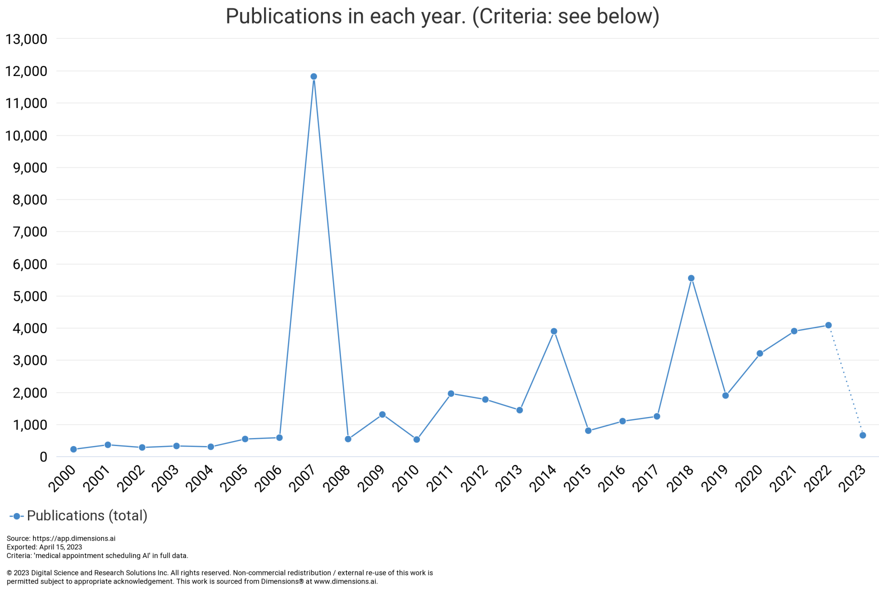

FIGURE 2.2: A number of papers matching the search term ”medical appointment scheduling AI” over the past few years.

The increasing trend in the number of papers on this topic is likely due to several factors. One possible factor is the growing use of electronic health records and other health information technologies, which have made it easier to collect and analyze data on medical appointment attendance. Additionally, the ageing population in many countries has led to an increase in demand for healthcare services, which has created a need for more efficient and effective appointment scheduling systems.

The use of readily available datasets such as the Medical Appointment No Shows dataset[9] has enabled researchers to develop and test new scheduling approaches more accurately. The use of AI and machine learning in appointment scheduling has also gained attention, as it holds the potential to improve scheduling accuracy and efficiency. This has further contributed to the growth in the number of papers published on medical appointment attendance and scheduling in recent years.

Overall, the increasing trend in the number of papers on medical appointment attendance and scheduling indicates a growing interest in this field and suggests that there is a need for further research and development in this area.

In conclusion, the literature review showed that there is a lack of readily available datasets for appointment scheduling in healthcare. While some researchers have used artificially created datasets or private datasets, these may not accurately represent the complexities of real-world appointment scheduling. The development of publicly available datasets can help address this issue and enable researchers to develop and test new scheduling approaches

2.5 Conclusion

In conclusion, the literature on appointment scheduling in healthcare systems and the use of artificial intelligence algorithms for predicting hospital attendance has grown significantly in recent years. The studies reviewed in this literature review have highlighted the potential benefits of utilizing machine learning techniques such as Random Forest, AdaBoost, and genetic algorithms for appointment scheduling and predicting no-shows in hospital appointments. However, many of these studies did not address challenges such as class imbalance, biased datasets, and unrealistic assumptions in the data, which can impact the generalizability of their findings.

Overall, this literature review emphasizes the need for further research to develop more sophisticated models and methodologies to address these challenges and to test these models on real-world data sets to evaluate their effectiveness. This could lead to improved patient outcomes and more efficient use of healthcare resources.

Paper	**Year**	**Accuracy/ Score**	**Dataset**	**Dataset Size**	**Algorithms**
[14]	2021	Precision 0.33	NHS Wales Informatics Service (NWIS)	1,011,897	LightGBM
[15]	2020	0.915	Medical Appointment No Shows[9]	110,527	SVM (Non-linear)
[16]	2021	"Various Sources"	0.99	Unkown	Boosting

TABLE 2.1: Table with paper information, year, accuracy, dataset name, dataset size, and algorithms used.

Chapter 3

Design

3.1 Problem Definition

The purpose of this investigation is to evaluate the feasibility of incorporating patient attendance probabilities into the appointment scheduling process. The goal is to improve the efficiency of patient processing by utilizing available appointment slots more effectively. The hypothesis is that by incorporating a prediction of patient attendance, it will be possible to schedule more patients in a given time period.

For instance, if there are ten patients scheduled for appointments, and two of them are unlikely to attend, these two appointments could be scheduled for the same time slot. This approach minimizes the possibility of wasting appointment slots, as even if one of the patients attends, the available time slot will still have been utilized.

By incorporating the prediction of patient attendance into the scheduling process, it is hoped that the efficiency of patient processing will be improved, leading to better utilization of resources and improved patient outcomes. The outcome of this investigation will provide valuable insights into the potential benefits of using attendance probabilities in appointment scheduling.

The proposed solution to the problem involves the use of a machine learning technique, specifically a classification model, to generate a new feature for each patient in the data set. This feature will represent the predicted probability of the patient showing up for their appointment. The scheduling component of the project is expected to be implemented using a Genetic Algorithm. The goal is to incorporate the predicted attendance probabilities into the fitness score calculation, so as to optimize the scheduling process and produce more efficient appointment schedules.

3.2 Objectives

The objectives of this project are to:

1. Develop a baseline scheduling model using the Overlapping Scheduling (OLAS) algorithm. This will be used to compare the results with the methods developed in this investigation.

2. Develop a machine learning model with a minimum accuracy of 70% in classifying patient attendance as "attended" or "missed". The model should be able to predict patient attendance based on relevant factors such as appointment time, patient demographics, and medical history.

3. Develop and implement a Genetic Algorithm for appointment scheduling that is capable of optimizing the use of equipment and doctors, leading to a more efficient use of resources. The algorithm should demonstrate improved performance over the baseline scheduler. The algorithm should also be able to generate schedules in a reasonable amount of time, with a maximum run time of 10 minutes for scheduling appointments for a single day.

4. Develop and implement a Genetic Algorithm for appointment scheduling that utilizes the classifier developed in Objective 3 to predict patient attendance. The algorithm should demonstrate improved performance over the baseline scheduler by achieving a 2% increase in equipment and doctor utilization on a daily basis. The algorithm should also be able to generate schedules in a reasonable amount of time, with a maximum run time of 10 minutes for scheduling appointments for a single day.

By accomplishing these objectives, the project aims to demonstrate the feasibility of incorporating patient attendance probabilities into appointment scheduling and to provide valuable insights into the potential benefits of this approach. The ultimate goal is to improve the efficiency and quality of patient care by effectively utilizing available appointment slots.

3.3 Requirements

1. The first requirement is to create a scheduling model that uses a traditional scheduling technique which has been widely used in the healthcare industry. This will serve as the baseline model and will be used to establish a benchmark against

which the other models will be compared. By fulfilling this requirement, objective 1 will be satisfied.

2. The second requirement is to create a model that predicts if a patient will attend their appointment or not. This model will use a classification technique to create a new feature for each patient, which will represent the likelihood of the patient showing up. This feature will be used to help schedule patients who are less likely to attend, thereby maximizing resource utilization and reducing wait times. Objective 2 will be fulfilled by meeting this requirement.

3. The third requirement is to create a scheduling model that aims to improve the traditional technique. This model will be designed to address the limitations of the traditional technique and will incorporate additional considerations to optimize the scheduling process. This model will be created based on the findings of the comprehensive review of existing scheduling methods. Meeting this requirement will fulfill objective 3.

4. The fourth requirement is to create a scheduling model that incorporates the prediction model outlined above. This model will incorporate the likelihood of patients showing up into the fitness score of the algorithm. The ultimate goal of this model is to create a schedule that is optimized based on the likelihood of patient attendance. The satisfaction of objective 4 will be achieved by fulfilling this requirement.

The scheduling process shall utilize the probability of patient attendance as a factor in the optimization of the schedule. This will result in a reduction of overall patient wait time and an improved schedule compared to the baseline model.

3.4 Design

3.4.1 Attendance Prediction Model

The Attendance Prediction Model will be developed using a combination of TensorFlow 2.11.0, Numpy 1.24.2, Pandas 1.5.3, and Imbalanced Learn 0.10.1 libraries.

The process of creating the model begins with importing and reading in the dataset, followed by creating a Pandas DataFrame to store the data. To handle class imbalance, Imbalanced Learn is employed. Next, the data is normalized and encoded using Numpy to ensure compatibility with the TensorFlow classification model.

Subsequently, multiple classification models are developed using TensorFlow, and these models are compared to determine the final model for deployment.

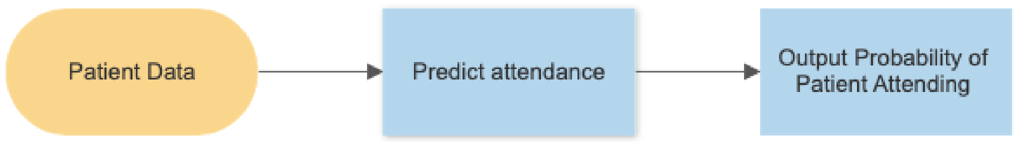

FIGURE 3.1: Appointment Attendance Prediction Model Diagram

3.4.2 Scheduling Model

The proposed scheduling model comprises three key components. The first component involves the creation of an "Individual," which is essentially a chromosome that represents a single schedule. Each Individual will have multiple genes, with each gene representing a patient's scheduled appointment time. A fitness score will be generated for each individual using a function that calculates the doctor's idle time and patient wait time, with the aim of minimizing these parameters.

The second component of the scheduling model is the genetic algorithm, which consists of a population of N Individuals. The genetic algorithm contains generation algorithms, breeding algorithms, and selection algorithms. The objective of the genetic algorithm is to identify the Individual with the best fitness score.

Overall, the scheduling model is designed to optimize patient scheduling and reduce doctor idle time and patient wait time. The genetic algorithm framework is employed to generate and select the best possible schedules for patients, with the aim of enhancing the overall efficiency of the scheduling process.

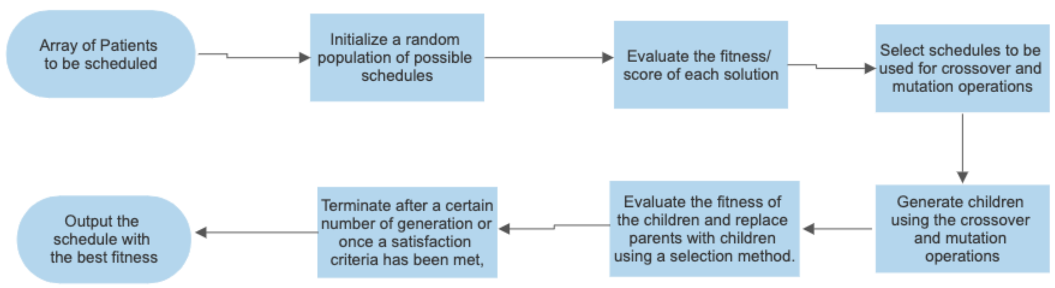

FIGURE 3.2: Diagram of the proposed Genetic Algorithm for Scheduling

3.4.3 Model Integration

In order to consolidate the two models explicated previously, a dedicated method will be developed for each gene, which employs the classification model to probabilistically assign the gene's attendance at its scheduled appointment. This predictive capability will be incorporated into the fitness function of the genetic algorithm, potentially facilitating the scheduling of multiple genes with coinciding or overlapping appointment times.

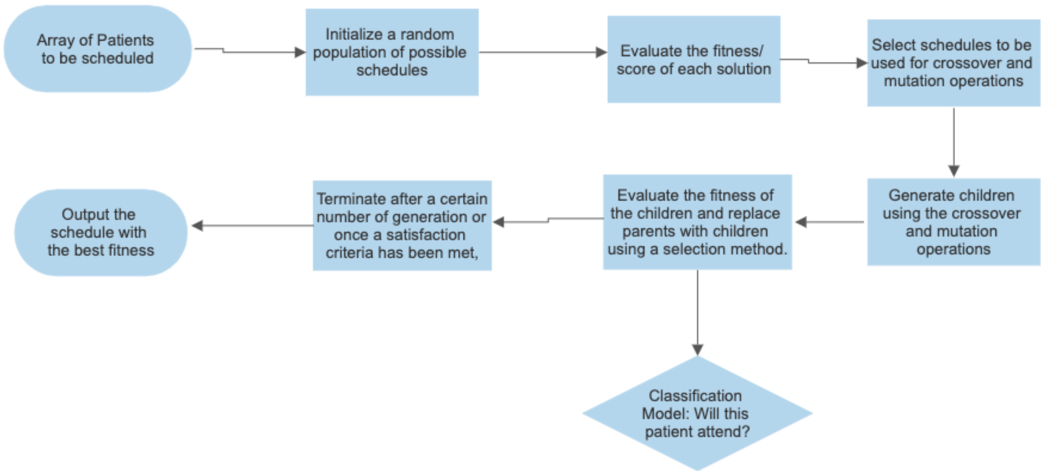

FIGURE 3.3: Diagram For Integrating Classification Model with GA Scheduler

Chapter 4

Implementation

Firstly, a novel method was developed for creating a dataset on appointment duration, enabling more accurate scheduling algorithms. Secondly, a baseline scheduler was proposed that allocates the same appointment duration to each patient and schedules them consecutively. Thirdly, an innovative overlapping scheduler was introduced that overlaps the start time of patients' appointments with the expected end time of the previous patients, reducing idle time and improving resource utilization. Fourthly, an appointment attendance classifier was designed to predict the likelihood of patients attending their appointments, enabling the scheduler to make necessary adjustments. Finally, a generic algorithm scheduler was developed that incorporates the appointment attendance classifier and can be adapted to various scheduling scenarios. These contributions demonstrate the potential of this approach to enhance appointment scheduling and optimize the utilization of healthcare resources.

4.1 Appointment Duration Dataset

In the real world, the actual appointment duration varies from patient to patient. It is unrealistic to assume that each patient will take exactly X minutes as this would make the results from scheduling algorithms unrealistic.

The process of gathering and generating this dataset was difficult and required extensive research. Eventually, the NHS was contacted and they provided an appointment duration dataset[17] that was gathered during the month of August 2022. This dataset contains 178620 examples.

Obtaining the dataset from the NHS did not affect the original design of the project as the requirement for a dataset on appointment duration was known from the beginning.

However, the dataset provided by the NHS was not particularly granular, with a wide range of appointment durations recorded. To make the dataset more manageable, the appointment durations were grouped into the following categories: 1-5 minutes, 6-10 minutes, 11-15 minutes, 16-20 minutes, 21-30 minutes, and 31-60 minutes.

The process of obtaining a suitable dataset on appointment duration represented a high-risk factor for the overall project. Without a suitable dataset, the project may not have been possible. To address this risk, extensive research was conducted to find a suitable dataset, including searching Kaggle and various government websites such as https://data.gov.ie/dataset and GOV.AU. After these efforts proved unsuccessful, the decision was made to reach out to the NHS for assistance, which ultimately resulted in obtaining the required dataset. Despite the challenges faced during this process, the dataset was successfully obtained within two weeks, ensuring the project could proceed as planned.

To create an accurate dataset on appointment duration, it was not sufficient to simply assign each patient in the dataset a range of appointment duration. Instead, a weighted distribution was used to assign each appointment duration range based on the frequency of occurrence in the dataset. This was achieved by counting the number of appointments with each duration range and using this count as the weight for that range. Each patient was then randomly assigned an appointment duration based on a value generated within the selected range. This approach resulted in a dataset with integer values representing the actual appointment durations and more accurately reflecting the real-world variation in appointment times.

To assign weights to each of the duration ranges, the count of each duration was counted using the following code. Figure 4.1 shows the appointment duration probabilities from August 2022.

```
fin = open("Actual_Duration_CSV_Aug_22.csv", "r")

for index, row in enumerate(fin):
  row = row.split(",")
  if index == 0:
    continue
  duration = row[-2]
  duration = duration.replace("m", "Minutes")
  if "Unknown" in duration:
    continue
  count = row[-1].strip()
  if count.isnumeric() is False:
    continue
  count = int(count)
  if duration not in duration_counts:
    duration_counts[duration] = 0
  duration_counts[duration] += count
```

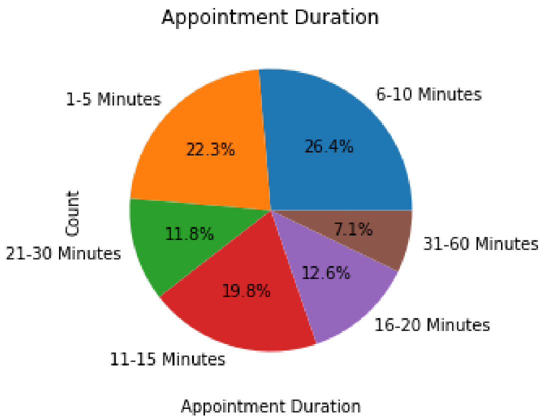

FIGURE 4.1: Appointment Duration Probabilities

4.2 Baseline Scheduler

To evaluate the effectiveness of the proposed scheduling algorithms, a baseline scheduler was developed. This scheduler assumes that each patient's appointment has the same duration and schedules patients consecutively, one after the other. This method is simplistic and does not take into account variations in actual appointment durations, leading to potentially unrealistic scheduling outcomes.

Designing the baseline scheduler did not present significant challenges. The class representing this scheduling method inherits all the relevant aspects of the Overlap scheduling class, with the exception that the overlap percentage is set to 0. The resulting scheduling approach can be used as a point of comparison for evaluating the effectiveness of the more sophisticated scheduling algorithms developed in this project. The implementation fulfils the **requirement 1** of creating a baseline scheduler.

```
class UniformScheduler(OLASScheduler):
    def init(self, patients, appointment_duration):
        super().init(patients, appointment_duration, 0)
```

4.3 Overlapping Scheduler

The Overlapping Scheduler was created to represent the OLAS algorithm that is widely employed in contemporary industry. This enabled a comparative analysis between the newly formulated scheduling algorithm and the prevalent industry standards.

The development of the algorithm posed a minimal challenge, with the notable exception of devising a mathematical expression to determine appointment start time based on the conclusion of the preceding appointment. Consequently, the following formula was created:

$$(appointment\_slot\_duration - (appointment\_slot\_duration \times overlap\_percentage)) + previous\_appointment\_scheduled\_time \tag{4.1}$$

This formula was then represented using the following Python3 code:

```
def __get_next_start_time(self, previous_patient: Patient, current_patient: Patient):
    start_time = previous_patient.start_time + (self.__appointment_duration \
        * (1 - self.__overlap_percentage))
    current_patient.start_time = start_time
```

The complete implementation of the Overlapping Scheduler is presented below.

```
class OLASScheduler:
    def __init__(self, patients, appointment_duration, overlap_percentage):
        self.__patients = patients
        self.__appointment_duration = appointment_duration
        self.__overlap_percentage = overlap_percentage

    def get_schedule(self):
        schedule = []
        previous_patient = None
        for patient in self.__patients:
            if previous_patient is None:
                schedule.append(patient)
            else:
                self.__get_next_start_time(previous_patient, patient)
                schedule.append(patient)
            previous_patient = patient
        return schedule
```

The implementation fulfils the **requirement 2** of creating an OLAS scheduler.

4.4 Fitness/ Scoring Function

The implementation of the fitness function used to evaluate the performance of the scheduling methods is presented below. This function takes as input an output schedule and computes the percentage of utilized time versus wasted time, with a score closer to 1 indicating superior performance and a score closer to 0 indicating inferior performance.

One of the challenges encountered in developing this function was the need to accurately track the current time throughout the simulation. To address this, the current time is updated after each patient appointment by taking the maximum of the previous current time and the sum of the patient start time and the actual appointment duration.

```
def get_fitness(self, schedule: List[Patient]):
"""
Method to get the fitness score of a scheduled list of patients.
"""

    for i in range(len(schedule) - 1):
        patient = schedule[i]
        if patient.attends:
            self.used_time += patient.appointment_duration
            self.wasted_time += max(0, patient.start_time - self.current_time)
            self.current_time = max(self.current_time, patient.start_time) +\
                patient.appointment_durations
    return self.used_time / (self.wasted_time + self.used_time)
```

4.5 Attendance Classifier

The development of an attendance classifier is a critical task that aims to predict whether a patient will attend a scheduled appointment. This binary classifier produces output values of 1 or 0, indicating the likelihood of attendance[18]. Building an accurate and reliable attendance classifier requires several essential stages, including data preprocessing, handling class imbalance, feature engineering, and classifier implementation[**?**]. This section will outline each of these stages' implementation and highlight their key learnings, providing a comprehensive overview of the development process. By understanding each stage's implementation and its significance, healthcare professionals can leverage attendance classifiers to better manage appointments, improving patient outcomes and overall healthcare efficiency.

4.5.1 Preprocessing

Data preprocessing is a crucial step in creating a classifier, as it involves removing missing or invalid data. For instance, in the dataset used for this project, the minimum age was -1 year old and the maximum age was 115, which are likely invalid entries. To address this issue, ages over 100 and under 0 were filtered out.

Some features, such as the day and hour of the appointment, are difficult to identify as it is unclear from the dataset when appointment scheduling was accepted. To identify invalid data in these features, Seaborn's strip plot method was used to plot the frequency of each value and identify outliers. Figure 4.2 shows the output of this method for the appointment day feature, indicating that day 5 (Saturday) is clearly an outlier as it has few value counts compared to the other days. This process of identifying and filtering outliers was carried out for all features in the dataset.

Overall, by carefully preprocessing the data, we can improve the quality of the input data and make the subsequent modelling steps more accurate and effective.

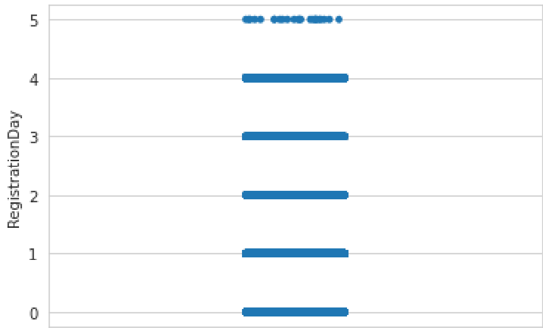

FIGURE 4.2: Frequency of appointment day

```
sns.stripplot(data = df, y = 'RegistrationDay', jitter = True)
plt.show()
```

4.5.2 Class Imbalance

One of the most important aspects of machine learning that is often overlooked is handling class imbalance. In the dataset used for this project, 70% of the patients attended their appointment, which means a naive classifier that always predicts attendance would achieve 70% accuracy without actually learning anything.

There are two main approaches to handling class imbalance: undersampling and oversampling. Undersampling randomly deletes examples from the majority class until the two classes have the same number of examples, which can greatly reduce the number of examples available for training. On the other hand, oversampling involves creating new examples of the minority class until both classes have the same number of examples, which allows for more data to be available for the model.

For this project, the oversampling method Synthetic Minority Over-sampling Technique (SMOTE) was used. SMOTE selects N random examples from the minority class for each example in the minority class, calculates the Minkowski distance between the original example's feature space and the random examples, takes the average of these values, and creates a new example at this point[19].

The imbalanced learn module[20] provides the SMOTE method out of the box, so there was little implementation needed in the project.

4.5.3 Feature Engineering

Feature engineering is a critical process in machine learning, involving the creation of additional features from existing ones[21]. In the current dataset, two features were used for feature engineering: the appointment booking timestamp and the scheduled appointment date. Four new features were derived from these two: the day of the week on which the appointment was booked, the hour of the appointment booking, the day of the week when the appointment was scheduled, and the time duration between the booking and the scheduled appointment date.

The implementation of these new features was relatively straightforward, using the functions below. Incorporating these new features into the dataset has the potential to improve the accuracy and effectiveness of the machine learning model, enabling it to better identify patterns and make more accurate predictions.

```
def date_to_day_of_week(date):
  if type(date) != str:
    return -1
  date = datetime.strptime(date, '%Y-%m-%dT%H:%M:%SZ')
  return date.weekday()

def date_to_hour(date):
  if type(date) != str:
    return -1
  date = datetime.strptime(date, '%Y-%m-%dT%H:%M:%SZ')
  return date.hour

def minutes_between_dates(date1, date2):
  date1 = datetime.strptime(date1, '%Y-%m-%dT%H:%M:%SZ')
  date2 = datetime.strptime(date2, '%Y-%m-%dT%H:%M:%SZ')
  return (date2 - date1).total_seconds() // 60
```

4.5.4 Model Creation

The machine learning model developed in this investigation was trained to classify patient attendance using relevant factors such as appointment time, patient demographics, and medical history. Initially, KNN, RandomForrest and NaiveBayes classifiers[22] were tried, but they did not perform as well as the neural network model on the validation data. Table 4.1 shows the results of the classifiers on the training and validation data.The final model was a sequential neural network consisting of 5 layers: an input layer, 3 hidden layers, and an output layer, with the number of neurons in each layer being 16, 32, 32, 16 and 1.. The architecture for the network can be seen in Figure4.3.

The rationale behind selecting a 5-layer model was that a 4-layer model was experiencing underfitting, while a 6-layer model was encountering overfitting. Although additional

Table 4.1: Accuracy of Machine Learning Algorithms

Algorithm	Training Accuracy	Validation Accuracy
KNN	0.57	0.38
Gaussian Naive Bayes	0.56	0.41
Decision Tree	0.78	0.65
Support Vector	0.49	0.71
Neural Network	0.69	0.68

layers, such as dropout[23] layers, could have been incorporated to expand the model size while averting overfitting, it was deemed unnecessary since the 5-layer model yielded satisfactory outcomes.

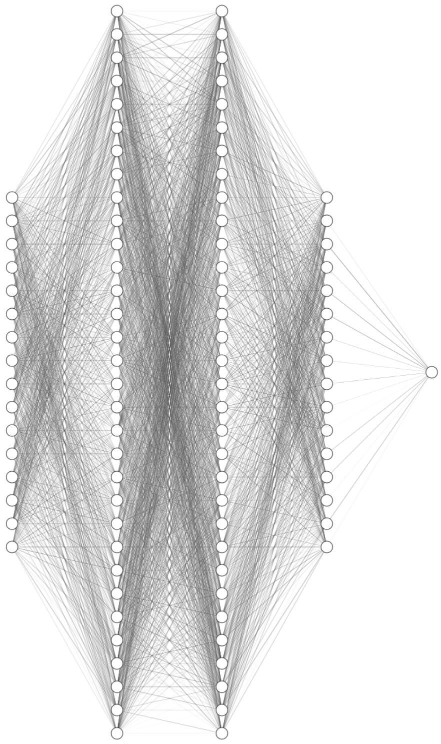

Figure 4.3: Neural Network containing 1 input layer, 3 hidden layers and 1 output layer

With regrades the activation functions for each neuron, in the input and hidden layers the activation function was defined as ReLu[24]. More importantly, the activation function in the output layer was set to Sigmoid[25] to all the probability of a patient not attending to be between 0 and 1.

While training the model each epoch was run on a batch of training data and at the end of the epoch it was evaluated on a batch of validation data. This allows the model to get close to the performance expected in the real world while keeping unseen data for evaluation at a later stage. To prevent the model from over-fitting the validation loss was monitored and the weights of the model were restored to the best state seen during training.

The creation of the model can be seen in the code snippet below.

```
from keras.callbacks import EarlyStopping
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

def build_nn():
    network = Sequential()
    network.add(Dense(16, activation="relu", input_shape=(train_X_copy.shape[1],)))
    network.add(Dense(32, activation="relu"))
    network.add(Dense(32, activation="relu"))
    network.add(Dense(16, activation="relu"))
    network.add(Dense(1, activation="sigmoid"))
    network.compile(optimizer=SGD(lr=0.005), loss="binary_crossentropy", metrics=["accuracy"])
    return network

model = build_nn()

history = model.fit(
    train_X_copy,
    train_y_copy,
    epochs=100,
    validation_data=(validation_X, validation_y),
    callbacks=[early_stopping]
    )
```

The implementation fulfils the **requirement 3** of creating the attendance classifier.

4.6 Genetic Algorithm

4.6.1 Mutation

In a genetic algorithm, mutation operators are used to introducing diversity into the population by making random changes to individuals. Several mutation operators are available in genetic algorithms, each with its advantages and disadvantages.

Swap mutation[26] is a commonly used mutation operator that involves selecting two random indices from a parent and swapping them. In the case of scheduling, a parent or schedule is selected, and the two indices are selected at random, representing two patients. Within the schedule, both of these patients' appointments are swapped.

```
def __swap_mutation(self, individual):
    left_index = random.randint(0, len(individual) - 1)
    right_index = random.randint(0, len(individual) - 1)
    individual[left_index], individual[right_index] = \
        individual[right_index], individual[left_index]
    return individual
```

Scramble mutation[26] is another mutation operator that involves selecting two indices within the parent and randomly shuffling the sub-array between these two points. In the case of a schedule of patients, a random selection of patients is made, and the ordering of these patients is rearranged or shuffled.

```
def __scramble_mutation(self, individual):
    left_index = random.randint(0, len(individual) - 2)
    right_index = random.randint(left_index, len(individual) - 1)
    left_copy = individual[0: left_index]
    right_copy = individual[right_index:]
    middle = individual[left_index: right_index]
    random.shuffle(middle)
    return left_copy + middle + right_copy
```

Inversion mutation[26] is a mutation operator similar to scramble mutation, but instead of randomly reordering the sub-array, the ordering of the sub-array is simply reversed.

```
def __inversion_mutation(self, individual):
    left_index = random.randint(0, len(individual) - 2)
    right_index = random.randint(left_index, len(individual) - 1)
    left_copy = individual[0: left_index]
    right_copy = individual[right_index:]
    middle = individual[left_index: right_index][::-1]
    return left_copy + middle + right_copy
```

4.6.2 Selection Operations

In the implementation of a genetic algorithm, the selection operation is a crucial step in choosing the best individuals from the current population to be used as parents for the next generation. Several selection methods are available in genetic algorithms, each with its advantages and disadvantages.

One common selection method is tournament selection, which involves selecting individuals at random and then choosing the fittest among them to be a parent for the next generation. This process is repeated until the desired number of parents is selected.

```
def __tournament_selection(self, population, num_parents):
    tournament_size = int(len(population) * 0.25)
    parents = []

    # repeat tournament process to select desired number of parents
    while len(parents) < num_parents:
        # randomly select a subset of the population
        tournament = random.sample(population, tournament_size)

        # select the fittest individual in the tournament as a parent
        parent = max(tournament, key=lambda individual: individual.fitness)
        if parent not in parents:
            parents.append(parent)
```

```
        return parents
```

Another popular selection method is roulette wheel selection, also known as fitness proportionate selection. This method assigns a probability of selection to each individual in the population based on its fitness score. The probability of an individual being selected is proportional to its fitness score, with fitter individuals having a higher probability of being chosen.

```
def __tournament_selection(self, population, num_parents):
    tournament_size = int(len(population) * 0.25)
    parents = []
    # repeat tournament process to select desired number of parents
    while len(parents) < num_parents:
        # randomly select a subset of the population
        tournament = random.sample(population, tournament_size)
        # select the fittest individual in the tournament as a parent
        parent = max(tournament, key=lambda individual: individual.fitness)
        if parent not in parents:
            parents.append(parent)
    return parents
```

Rank-based selection is another commonly used selection method that assigns a probability of selection to each individual based on their rank in the population, rather than their absolute fitness score. This method helps to reduce the influence of outliers and can be useful in maintaining diversity in the population.

```
def __rank_based_selection(self, population, num_parents):
    sorted_pop = sorted(population, key=lambda x: x[1], reverse=True)
    ranks = [i + 1 for i in range(len(sorted_pop))]
    rank_sum = sum(ranks)
    probabilities = [rank / rank_sum for rank in ranks]
    return random.choices(sorted_pop, probabilities)[:num_parents]
```

4.6.3 Genetic Algorithm

The initialization step in a genetic algorithm involves generating a random population of a given size. In the context of scheduling, the population is an array of schedules, where each schedule consists of patients arranged in random order. The use of random ordering is crucial in enabling the genetic algorithm to efficiently explore a vast space of possible schedules. Without random ordering, the GA would require significantly more generations to discover a wide range of possibilities.

Random initialization also helps to ensure diversity in the population, which is essential in preventing premature convergence. If the initial population was biased towards a particular subset of schedules, the GA would be less likely to explore the full range of possibilities and may converge prematurely to suboptimal solutions.

In summary, the initialization step of a genetic algorithm is critical in ensuring diversity in the population and enabling efficient exploration of the solution space. In scheduling, random initialization of schedules is essential in enabling the GA to discover a broad range of possibilities and prevent premature convergence.

```
for i in range(self.population_size):
    self.population.append(
        Schedule(
            self.create_individual()
        )
    )

.........................................

def create_individual(self):
    random.shuffle(self.patients)
    return self.patients
```

Subsequently, every individual schedule or candidate is ascribed a degree of fitness, which is derived from the fitness function. It is important to note that the fitness function utilized in this approach deviates from conventional genetic algorithms as it relies on the classifier previously established and described, aimed at prognosticating whether a patient will be present for their scheduled appointment. The fitness metric, in this case, is contingent upon the proportion of time that has been effectively utilized, versus the amount of time that has been rendered futile.

```
def fitness(self, schedule: List[Patient]):
    used_time = 0
    wasted_time = 0
    current_time = 0

    for i in range(len(schedule) - 1):
        patient = schedule[i]
        if self.classifier.attends(patient):
            used_time += patient.appointment_duration
            wasted_time += max(0, patient.start_time - current_time)
            current_time = max(current_time, patient.start_time) +\
                patient.appointment_duration

    return used_time / (wasted_time + used_time)
```

The next step involves selecting a set of parents using one of the selection methods mentioned earlier. Subsequently, two parents are randomly chosen to create two offspring through the crossover. The resulting offspring undergo mutation using one of the previously mentioned methods. The fitness of each child is then calculated, and both children are added to the next generation. Once the next generation is populated, it will become the population used for the next iteration.

```
while len(offspring) < self.population_size:
   parent1, parent2 = self.select_two_parents(parents)
   child1, child2 = self.cross_over(parent1, parent2)
   child1 = self.mutation_operator.mutate(child1)
   child2 = self.mutation_operator.mutate(child2)
   child1 = Schedule(child1)
   child1.set_fitness(self.fitness(child1.schedule))
   child2 = Schedule(child2)
   child2.set_fitness(self.fitness(child2.schedule))
   offspring.append(child1)
   offspring.append(child2)
```

At the end of all the iterations are completed the schedule with the best fitness is returned. The implementation fulfills **requirement 4** of creating a genetic algorithm scheduler that utilizes the attendance classifier.

Chapter 5

Testing and Evaluation

5.1 Attendance Classifier

In order to assess the efficacy of the attendance classifier, a sub-sample of the original data set was segregated for the purpose of testing, following the completion of the model's training. This approach was taken to mitigate the risk of information leakage and to ensure that the model was not manipulated during the training phase to achieve optimal test scores. By testing the model on previously unseen data, it is possible to evaluate its performance in a manner that closely resembles the real-world scenario of making predictions on new data in a production environment.

To evaluate the classifier on the unseen data, the following metrics were used Accuracy, Precision, Recall and F1 score.

5.1.1 Model Generalization: Over-fitting and Under-fitting

It is important that a model performs well on new unseen data or in other words that the model generalizes well to new data. To ensure this the model must fit well, that is that it is not over or under-fit during the training process. Figure5.1 shows the history of the model during the training. At epoch 2(indicated by a dashed cyan line) the model is underfitting quite a lot. This can be seen by the large difference between the training metrics and the validation metrics. The green dashed line at epoch 12 indicates the point of time in training that both the training and validation metrics are closely aligned. Finally, the pink dashed line at epoch 20 indicates a point in time where the model is over-fitting. As the model monitored validation loss while training and restored its weights to the optimal point. The model would have set its weights to epoch 12 where there was no under or overfitting.

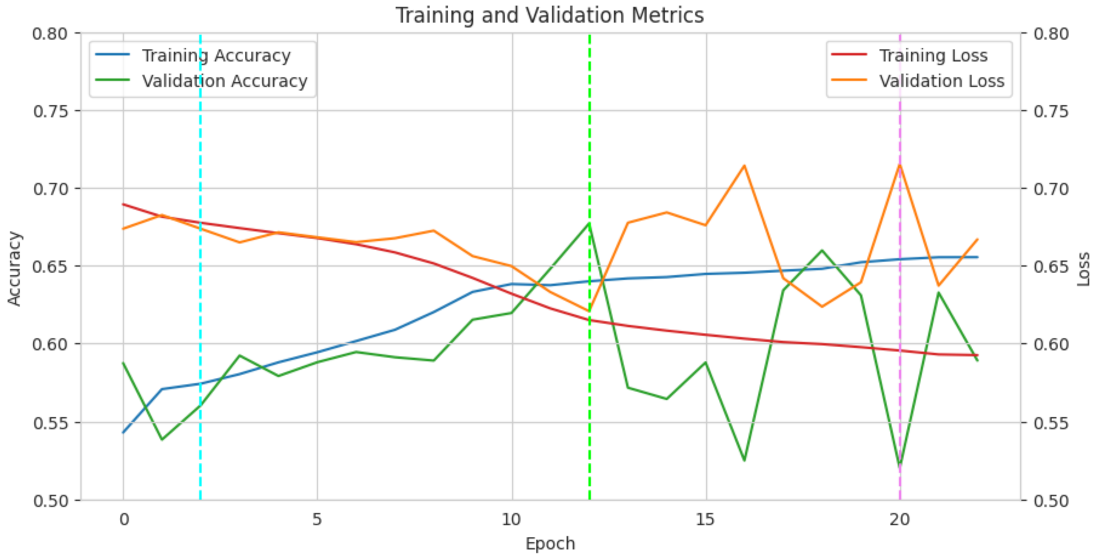

FIGURE 5.1: Model Training History

5.1.2 Real World Accuracy

The accuracy metric refers to the proportion of examples that are classified correctly out of the total number of examples. This metric is displayed as a percentage. For example, if there are 10 examples and 8 are classified correctly then the accuracy is 0.8 or 80%.

The accuracy of this model on unseen test data was 71.4%. This aligns closely with the accuracy achieved in Deep learning for predicting non-attendance in hospital outpatient appointments[27] achieving 71% and Predictors of outpatients' no-shows: big data analytics using apache spark[28] achieving 79%. However, this model is likely performing better than the ones previously outlined as the training data that the model was trained on had been balanced using SMOTE. This meant that the model was unable to become a majority class, classifier.

5.2 Scheduler

To evaluate the efficacy of the genetic algorithm scheduler, a comparative analysis was conducted against two well-established scheduling methods. The first method, referred to as the Naive Scheduler, simply allocates each patient a fixed appointment time exactly when the previous patient's appointment finishes. Specifically, if the appointment duration is 20 minutes, then each patient is allotted an appointment start time at 20-minute intervals, with the first patient starting at time 0, the second patient at 20 minutes, and so on.

The second scheduling method, known as Overlapping Scheduling (OLAS), operates by allocating a patient's appointment time prior to the conclusion of the previous patient's appointment. The extent of the overlap is governed by an overlap percentage, which indicates the amount of overlap to be incorporated into the scheduling. For instance, if the appointment duration is 20 minutes and the overlap percentage is 10%, then the first patient would be scheduled for minute 0, the second patient would be scheduled for minute 18, and the third patient would be scheduled for minute 26.

5.2.1 Scoring Function

In order to assess the efficacy of various scheduling methods, it is necessary to employ a scoring function. The scoring function utilized in this context computes the proportion of utilized time relative to the total time, expressed as $\frac{used_t ime}{used_t ime+unused_t ime}$. This scoring function is based on the "Doctor idle time" as described in "Scheduling doctors' appointments: optimal and empirically-based heuristic policies"[29], which ranks the scoring of a schedule as the time the doctor starts examining the final patient (Sn), minus the total time that he or she has spent examining the first n 1 patients.

For example, if a given scheduling algorithm produces a score of 0.8, then it indicates that the equipment or medical professional is occupied for 80% of the total time and idle for the remaining 20%. A score closer to 1 signifies a more optimal schedule.

5.2.2 Scoring of Non-Classification Algorithms

Due to the time-intensive nature of running genetic algorithms, the permutations of each selection and mutation method were executed without the presence of a classifier. This approach assumes that each patient will attend their appointment, similar to the baseline and OLAS methods. Subsequently, the optimal combination was identified and integrated with the attendance classifier. The performance of this composite model will be evaluated in the forthcoming chapter.

For this experimental assessment, a sample of 50 patients was generated, each with a duration of actual appointment time generated according to the methodology outlined in Chapter 4. Additionally, each patient was assigned an 80% likelihood of attending their appointment, based on the average probability observed across the utilized dataset for classification purposes.

Figure 5.2 demonstrates that the genetic algorithm using swap mutation and tournament selection yielded the highest score, followed by the OLAS algorithm. Conversely, the baseline algorithm exhibited poor performance across the 50 patients tested.

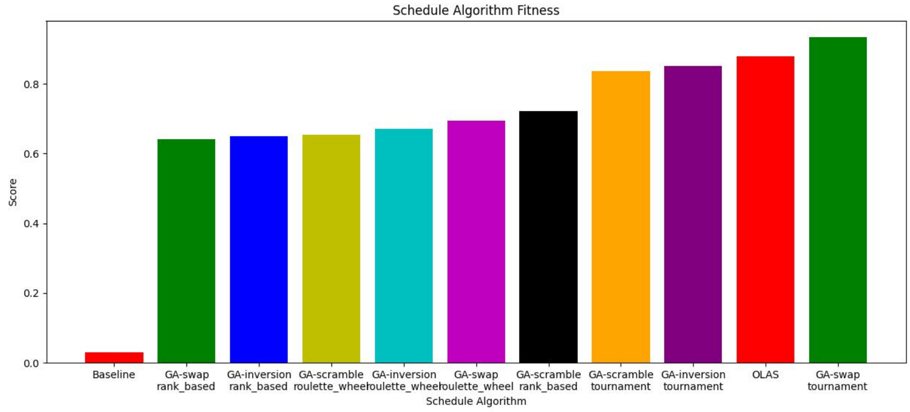

FIGURE 5.2: Fitness Scores Without Classifier

5.2.3 Schedule Creation Timing

In addition to algorithmic performance, it is critical to consider the time required to generate a schedule. This is of utmost importance as generating a schedule that is delayed beyond the time of the first appointment is impractical. Although this is an unlikely scenario, it is essential to assess the algorithmic performance under such conditions.

Figure 5.3 indicates that the Genetic Algorithm using swap mutation and tournament selection had the longest scheduling time taking 91.88s. The baseline and OLAS had the quickest processing time of 0.0000259 and 0.000019 seconds respectively.

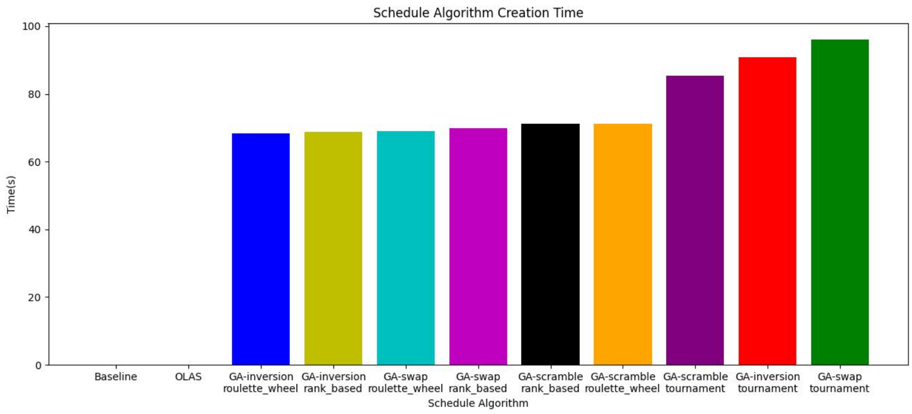

FIGURE 5.3: Fitness Timing Without Classifier

5.3 Scheduler With Classifier

To evaluate the performance of the scheduler using the classifier the optimal GA scheduler from the Section5.2 was brought forward and integrated with the classifier. This was then compared using the same scoring function used for the schedulers outlined in the previous section.

5.3.1 Fitness Score

Figure 5.5 displays the results of scheduling 50 patients. Again we can see that both the genetic algorithms and OLAS outperformed the naive baseline algorithm. It can be seen that introducing the classifier increased the fitness score of the generated schedule by a large amount achieving a utilization percentage of 90.4%.

However, from this test run it can be seen that the genetic algorithm achieved a worse fitness score than OLAS which contradicts the results in the previous section. This is likely due to the genetic algorithm having an unlucky start which meant it did not enough iterations to achieve an optimal solution. To solve this, the next section will run the tests over multiple restarts and calculate the average.

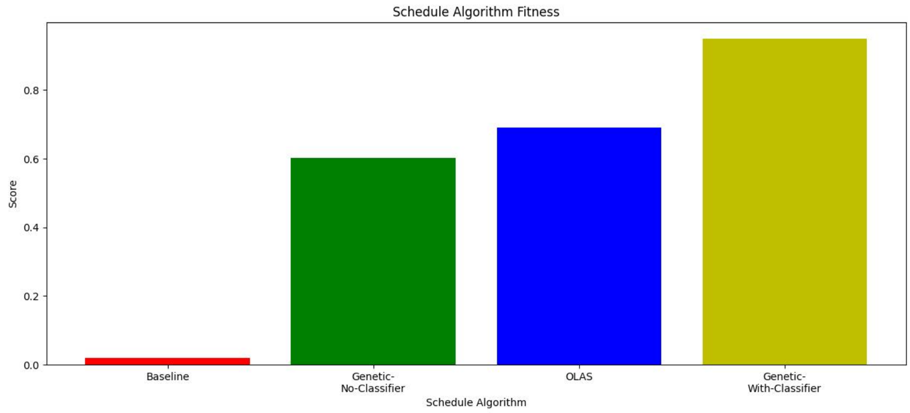

FIGURE 5.4: Fitness Score With Classifier

5.3.2 Schedule Creation Time

Looking at Figure5.5 it can be seen that including the classifier does not increase the time of creating the schedule over the GA without the classifier. This is due to the implementation of a cache that stores the classification of a patient so that each patient is only classified once. It is surprising however to see that this actually reduced the schedule-creating time. The non-classification GA averaged 11.01 iterations per second and the classification GA averaged 16.75 iterations per second. An experiment was run to view the impact of the cache in the timing but this took too long on 50 patients averaging 36 seconds per iteration, with an estimated run time of 10 hours and 45 minutes.

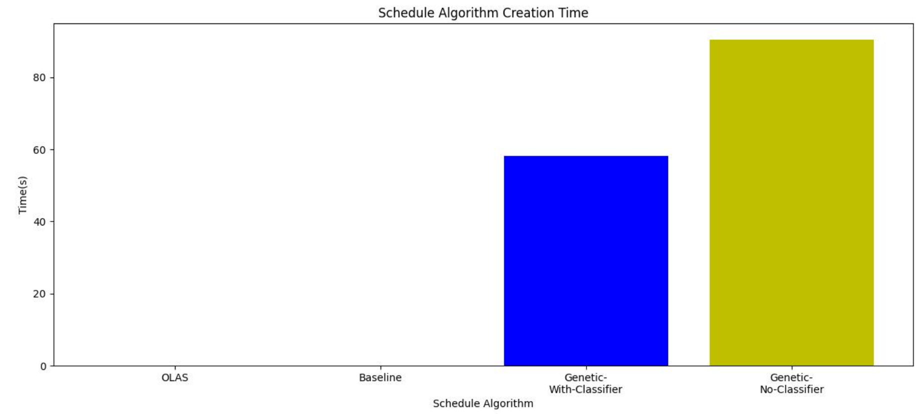

FIGURE 5.5: Fitness Timing With Classifier

In order to visualise the impact of the cache, the experiments were run on 4 patients. The results of this can be seen in Figure 5.6. Without the cache, the scheduled time takes much longer with a time of 1551 seconds. This is compared to the 7.24 seconds with the cache.

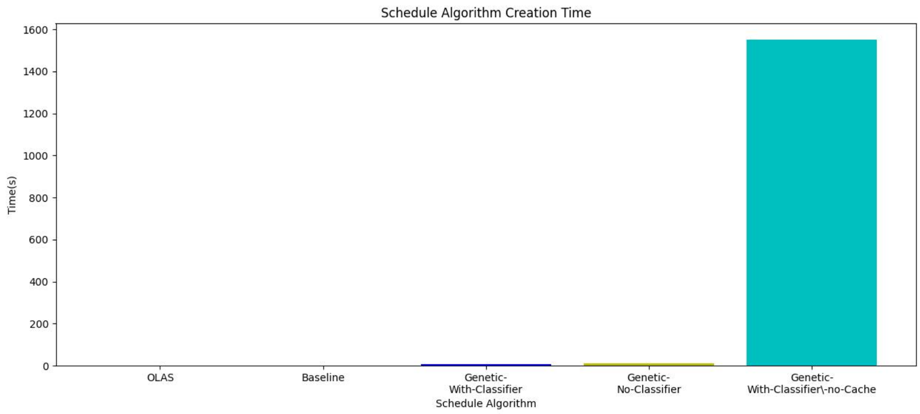

FIGURE 5.6: Fitness Timing With Classifier

Chapter 6

Discussion and Conclusions

The present study was motivated by the research question of whether predicting patient attendance can enhance scheduling efficiency. In order to address this research question, a set of objectives was formulated, which are restated below:

1. Investigate existing literature on patient no-show predictions and optimal scheduling in healthcare.
2. Develop a predictive model for patient no-shows using machine learning techniques.
3. Evaluate the performance of the predictive model and compare it to existing methods.
4. Develop an optimal scheduling algorithm that takes into account patient no-show predictions and minimizes wait times for patients and family members.
5. Test the optimal scheduling algorithm on real-world patient data and evaluate its effectiveness in reducing wait times.
6. Discuss the practical implications of the research and suggest potential areas for future work.

This study successfully achieved its stated objectives and effectively addressed the research question. The integration of a Genetic Algorithm scheduler with an attendance classifier proved effective in reducing idle time for healthcare providers and equipment by approximately 6%.

However, the research encountered certain challenges, notably the absence of a unified source of data encompassing both attendance and appointment duration information. To generate a comprehensive dataset, the study relied on merging two distinct data sources derived from real patient data. Nonetheless, it is important to recognize that such a merger could potentially introduce discrepancies in the performance of the developed scheduler, thus necessitating further evaluation in real-world settings.

6.1 Discussion

This study effectively conducted research on the effectiveness of implementing an attendance classifier to enhance healthcare scheduling by reducing doctor and equipment idle time. As demonstrated in the evaluation section of this report, idle time decreased by approximately 6% across a sample of 50 patients. Nonetheless, the integration of a genetic algorithm, particularly one with a classifier, may significantly increase the time required to generate the schedule. While this may not present a significant challenge in most applications, it could render this approach impractical in situations where schedules require frequent updates, such as in an accident and emergency department.

Several obstacles were encountered during the study, primarily related to dataset selection. Despite searching various platforms such as Kaggle, NHS, Gov.au, and the Google Dataset search engine, no single dataset contained all the necessary information. This was partially overcome by acquiring an appointment duration dataset from the NHS and a patient attendance dataset from Kaggle.

The implementation of genetic algorithms also presented challenges as they were designed to operate on a single thread, making it time-consuming to evaluate and make modifications to the algorithm. This limitation was addressed by redesigning the algorithm to allow for parallel mutation and crossover operations, thereby reducing the processing time.

Given the project's time constraints, it would have been advantageous to dedicate more time to contacting healthcare organizations for dataset creation. However, this was not feasible under the project's limitations.

6.2 Conclusion

The main conclusion to draw from this work is that implementing a classifier to predict patient attendance and utilizing this when creating a schedule can reduce equipment and doctor idle time.

Based on the findings and analysis presented in this study, it can be concluded that implementing a classifier to predict patient attendance and utilizing this when creating a schedule can reduce equipment and doctor idle time. The results of this study support the notion that predicting patient attendance for scheduling and highlight the significance of reducing doctor and equipment idle time. This study contributes to the existing literature by investigating the effectiveness of predicting patient attendance in healthcare scheduling and has important implications for reducing doctor and equipment idle time and therefore reducing the overall financial burden of non-attendance on the healthcare system.

However, there are several limitations to this study that should be acknowledged. First, is the generalizability of the scheduler, it will have to be retrained and fine-tuned for every application and location to see the best results. Second, is the unknown biases introduced as part of the classification model, it is unfair to penalize those who attend their appointments because a model predicts that they will not attend. These limitations suggest the need for future research into the generalizability of this application and into bias detection and mitigation.

This investigation successfully achieved all the stated objectives. A baseline scheduling model using the Overlapping Scheduling (OLAS) algorithm was developed and used as a benchmark to compare the performance of the newly developed methods (Objective 1). A machine learning model was developed that accurately classified patient attendance, achieving a minimum accuracy of 70% (Objective 2). Furthermore, a Genetic Algorithm was developed and implemented for appointment scheduling, leading to a more efficient use of resources by optimizing the use of equipment and doctors. This algorithm demonstrated improved performance over the baseline scheduler and was able to generate schedules in a reasonable amount of time (Objective 3). Finally, the Genetic Algorithm was extended to utilize the classifier developed in Objective 2, resulting in a 2% increase in equipment and doctor utilization on a daily basis. The algorithm was also able to generate schedules in a reasonable amount of time (Objective 4). Overall, this investigation presents effective and practical solutions to optimize appointment scheduling in healthcare facilities.

Throughout the course of this study, various competencies were acquired and applied, such as devising machine learning models, assessing the associated hazards linked with each model, managing class imbalance, and appraising classification models. Apart from the technical proficiencies honed, several non-technical proficiencies were obtained, which encompassed identifying relevant papers, datasets and articles, as well as the creation of a LaTeX-based report.

Overall, this study provides valuable insights into the effectiveness of predicting patient attendance for scheduling in healthcare and has important implications for patients and healthcare administrators. The findings of this study can inform that predicting patient attendance can be used to reduce wasted resources, and can be used to guide healthcare administrators to create an optimal schedule.

6.3 Future Work

Future research in this area can build upon the findings of this study and extend it in several directions.

Firstly, the study used a limited dataset due to the difficulties in acquiring appropriate data. Future research can expand on this by acquiring larger and more diverse datasets to improve the generalizability of the classifier model. Additionally, different types of data sources can be considered, such as electronic medical records or data collected from wearable devices, to provide a more comprehensive view of patient behaviour and factors that affect attendance.

Secondly, the study utilized a genetic algorithm to optimize the scheduling process. Alternative optimization techniques, such as machine learning or deep learning models, can be explored to determine the most effective approach for predicting patient attendance and optimizing scheduling.

Thirdly, the study focused on reducing the idle time of doctors and equipment. Future research can explore other potential benefits of predicting patient attendance, such as reducing patient waiting times or improving patient outcomes.

Fourthly, the study did not address the issue of bias in the classifier model. Future research can investigate bias detection and mitigation techniques to ensure fair and equitable scheduling practices.

Fifthly, the study did not consider the impact of external factors, such as unexpected emergencies or staffing shortages, on schedule. Future research can explore how to incorporate such contingencies into the scheduling process to ensure efficient and effective use of resources.

Finally, the possibility of creating and publishing a paper from the work carried out during this research will be investigated and if possible a paper will be produced.

In conclusion, while this study provides valuable insights into the effectiveness of predicting patient attendance for scheduling in healthcare, there is still much to be learned in this area. Future research can expand upon the limitations of this study and investigate alternative optimization techniques, consider other potential benefits of predicting patient attendance, address issues of bias and contingency planning, and explore a wider range of data sources to improve the generalizability of the classifier model.

Bibliography

[1] A. Ala and F. Chen, “Appointment scheduling problem in complexity systems of the healthcare services: a comprehensive review,” *Journal of Healthcare Engineering*, vol. 2022, 2022.

[2] A. Nelson, D. Herron, G. Rees, and P. Nachev, “Predicting scheduled hospital attendance with artificial intelligence,” *NPJ digital medicine*, vol. 2, no. 1, p. 26, 2019.

[3] L. H. Salazar, A. Fernandes, R. Dazzi, N. Garcia, and V. R. Leithardt, “Using different models of machine learning to predict attendance at medical appointments,” *Journal of Information Systems Engineering and Management*, vol. 5, no. 4, p. em0122, 2020.

[4] S. AlMuhaideb, O. Alswailem, N. Alsubaie, I. Ferwana, and A. Alnajem, “Prediction of hospital no-show appointments through artificial intelligence algorithms,” *Annals of Saudi medicine*, vol. 39, no. 6, pp. 373–381, 2019.

[5] R. Alizadeh, J. Rezaeian, M. Abedi, and R. Chiong, “A modified genetic algorithm for non-emergency outpatient appointment scheduling with highly demanded medical services considering patient priorities,” *Computers & Industrial Engineering*, vol. 139, p. 106106, 2020.

[6] L. M. Guzek, S. D. Gentry, and M. R. Golomb, “The estimated cost of “no-shows” in an academic pediatric neurology clinic,” *Pediatric Neurology*, vol. 52, no. 2, pp. 198–201, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0887899414006511

[7] A. J. Mitchell and T. Selmes, “Why don’t patients attend their appointments? maintaining engagement with psychiatric services,” *Advances in Psychiatric Treatment*, vol. 13, no. 6, p. 423–434, 2007.

[8] H. B. Harvey, C. Liu, J. Ai, C. Jaworsky, C. E. Guerrier, E. Flores, and O. Pianykh, “Predicting no-shows in radiology using regression modeling of

data available in the electronic medical record," *Journal of the American College of Radiology*, vol. 14, no. 10, pp. 1303–1309, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1546144017305835

[9] "Medical appointment no shows - kaggle," https://www.kaggle.com/datasets/joniarroba/noshowappointmentsm.

[10] S. Alshaya, A. McCarren, and A. Al-Rasheed, "Predicting no-show medical appointments using machine learning," in *Advances in Data Science, Cyber Security and IT Applications*, A. Alfaries, H. Mengash, A. Yasar, and E. Shakshuki, Eds. Cham: Springer International Publishing, 2019, pp. 211–223.

[11] A. Alshammari, R. Almalki, and R. Alshammari, "Developing a predictive model of predicting appointment no-show by using machine learning algorithms [j]," *Journal of Advances in Information Technology Vol*, vol. 12, no. 3, 2021.

[12] C. Zacharias and M. Armony, "Joint panel sizing and appointment scheduling in outpatient care," *Management Science*, vol. 63, no. 11, pp. 3978–3997, 2017.

[13] T. Cayirli, E. Veral, and H. Rosen, "Designing appointment scheduling systems for ambulatory care services," *Health care management science*, vol. 9, pp. 47–58, 2006.

[14] E. Incze, P. Holborn, G. Higgs, and A. Ware, "Using machine learning tools to investigate factors associated with trends in 'no-shows' in outpatient appointments," *Health Place*, vol. 67, p. 102496, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1353829220318906

[15] T. Batool, M. Abuelnoor, O. El Boutari, F. Aloul, and A. Sagahyroon, "Predicting hospital no-shows using machine learning," in *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*. IEEE, 2021, pp. 142–148.

[16] G. Fan, Z. Deng, Q. Ye, and B. Wang, "Machine learning-based prediction models for patients no-show in online outpatient appointments," *Data Science and Management*, vol. 2, pp. 45–52, 2021.

[17] N. H. Service, "Appointments in general practice, august 2022." [Online]. Available: https://digital.nhs.uk/data-and-information/publications/statistical/appointments-in-general-practice/august-2022

[18] B. Kolo, *Binary and Multiclass Classification*. Weatherford Press, 2011. [Online]. Available: https://books.google.ie/books?id=XTuTAgAAQBAJ

[19] X. Zheng, *SMOTE Variants for Imbalanced Binary Classification: Heart Disease Prediction*. University of California, Los Angeles, 2020. [Online]. Available: https://books.google.ie/books?id=0SQTzgEACAAJ

[20] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[21] F. Chollet, *Deep Learning with Python.* Manning Publications Company, 2017. [Online]. Available: https://books.google.ie/books?id=Yo3CAQAACAAJ

[22] G. Bonaccorso, *Machine Learning Algorithms.* Packt Publishing, 2017. [Online]. Available: https://books.google.ie/books?id=_-ZDDwAAQBAJ

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with relu activation function and linear spline-type methods," *Neural Networks*, vol. 110, pp. 232–242, 2019.

[25] S. Narayan, "The generalized sigmoid activation function: Competitive supervised learning," *Information sciences*, vol. 99, no. 1-2, pp. 69–82, 1997.

[26] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[27] M. Dashtban and W. Li, "Deep learning for predicting non-attendance in hospital outpatient appointments," 2019.

[28] T. Daghistani, H. AlGhamdi, R. Alshammari, and R. H. AlHazme, "Predictors of outpatients' no-show: Big data analytics using apache spark," *Journal of Big Data*, vol. 7, pp. 1–15, 2020.

[29] L. W. ROBINSON and R. R. CHEN, "Scheduling doctors' appointments: optimal and empirically-based heuristic policies," *IIE Transactions*, vol. 35, no. 3, pp. 295–307, 2003. [Online]. Available: https://doi.org/10.1080/07408170304367

s