# Improving MRI Operations through Data-Driven Methods and Predictive Modeling

Lun Li

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:

Christina Mastrangelo, Chair

John Y. Choe

Prashanth Rajivan

Program Authorized to Offer Degree:
Industrial & Systems Engineering

University of Washington

**Abstract**

Improving MRI Operations through Data-Driven Methods and Predictive Modeling

Lun Li

Chair of the Supervisory Committee:
Christina Mastrangelo
Industrial & Systems Engineering

Magnetic resonance imaging (MRI) is one of the most clinically valuable and operationally complex modalities in modern radiology departments. Its operations are characterized by significant exam duration variability, complex scheduling demands, and a heavy reliance on technologist expertise that is difficult to develop and assess systematically. Despite the availability of rich operational data from sources like electronic health records (EHRs) and modality log files (MLFs), MRI operations management has remained largely experience-based and reactive. This dissertation addresses this disconnect through five interconnected studies that apply data-driven methods to operational improvement in MRI services.

The first study established the feasibility of a virtual hub-and-spoke model for deploying MRI expertise across geographically distributed sites, demonstrating that key radiology functions can be performed remotely while improving patient turnaround time and wait time. The second analyzed modality log file data to identify actionable sources of workflow inefficiency that are invisible to EHR-based analysis alone. The third characterized the structure and distributional properties of EHR data underlying subsequent predictive work and demonstrated that clinical text embeddings derived from result narratives using BioClinicalBERT carry predictive signal for exam duration. The fourth introduced a concordance-based validation framework for integrating EHR and MLF data, raising the concordance correlation coefficient between systems from 0.33 to 0.87, and showed that a Random Forest model trained on the validated dataset outperformed template-based

scheduling for 11 of 12 procedure codes with mean absolute error reductions ranging from 2% to 57%. The fifth compared Gated Recurrent Unit (GRU) neural networks against XGBoost on three established benchmark datasets, finding that GRU models consistently outperformed XGBoost with average advantages ranging from 8.5% to 18.8%, particularly at longer temporal windows and with minimal feature engineering.

Collectively, these studies demonstrate that meaningful gains in MRI operational efficiency and patient access are achievable by employing data-driven methods, and that the choice of analytical approach can matter as much as the availability of data. The findings provide both practical tools for MRI operational improvement and a methodological foundation for future temporal modeling of dynamic technologist competency.

## TABLE OF CONTENTS

Page

List of Figures . . . . . iii

Chapter 1: Introduction . . . . . 1
1.1 Related Work . . . . . 2
1.2 Contributions and Organization . . . . . 6

Chapter 2: A Better Way to Deploy Expertise: Virtual Hub-and-Spoke Model for MRI Operations . . . . . 8
Chapter Summary . . . . . 8
2.1 Introduction . . . . . 9
2.2 Methods . . . . . 12
2.3 Results . . . . . 18
2.4 Discussion . . . . . 24

Chapter 3: A Better Way to Use Data: MRI Log File Analysis for Workflow Improvement . . . . . 26
Chapter Summary . . . . . 26
3.1 Introduction . . . . . 26
3.2 Analysis and Performance Metrics . . . . . 27
3.3 Analysis of Patient Characteristics . . . . . 35
3.4 Conclusion . . . . . 38

Chapter 4: EHR Data Characterization and Clinical Text Analysis for MRI Duration Prediction . . . . . 42
4.1 Introduction . . . . . 42
4.2 Dataset Overview . . . . . 43
4.3 Data Preprocessing . . . . . 46
4.4 Exam Duration Analysis . . . . . 49
4.5 Patient and Exam Characteristics . . . . . 52
4.6 Bivariate Analysis . . . . . 53

4.7 Clinical Notes Analysis . . . . . 56
4.8 Summary . . . . . 60

Chapter 5: A Better Way to Schedule: Data Reconciliation and Predictive Modeling for MRI Exam Duration . . . . . 62
Chapter Summary . . . . . 62
5.1 Introduction . . . . . 63
5.2 Methods . . . . . 64
5.3 Results . . . . . 68
5.4 Discussion . . . . . 76

Chapter 6: Toward Data-Driven Competency Modeling: Evaluating Learned Temporal Representations vs. Manual Feature Engineering . . . . . 80
6.1 Introduction . . . . . 80
6.2 Related Work . . . . . 82
6.3 Datasets . . . . . 86
6.4 Model Architectures . . . . . 95
6.5 Model Training . . . . . 104
6.6 Experimental Results . . . . . 114

Chapter 7: Conclusions . . . . . 144

Bibliography . . . . . 147

## LIST OF FIGURES


Figure Number — Page

2.1 SADT diagram of highest-level MR image acquisition functions. . . . . 10

2.2 SADT diagram of Prep Patient sub-functions. . . . . 11

2.3 SADT diagram of Scan Patient sub-functions. . . . . 11

2.4 Selected findings from the cognitive task analysis by MR workflow function. Color denotes high-level function: Review schedule (A01), Prep patient (A02), Scan patient (A03). FOV, field of view. . . . . 14

2.5 Baseline simulation model results. . . . . 19

2.6 TAT by patient attributes for baseline model: (A) habitus; (B) exam type; (C) allergy; (D) IV; (E) implant. . . . . 20

2.7 WT by patient attributes for baseline model: (A) habitus; (B) exam type; (C) allergy; (D) IV; (E) implant. . . . . 21

2.8 Pairwise comparison of mean TAT and mean WT of different models: (A) mean TAT (minutes); (B) mean WT (minutes). ****: $p < 0.001$. . . . . 23

3.1 Flowchart for a typical MRI exam, based on the log file information. The exam starts with the first table movement and ends after the last table movement. The log files capture data related to each one of these activities. . . . . 29

3.2 KPI by anatomy where 'n=' denotes the number of exams. Brain exams tend to have the smallest KPI variability while abdominal exams demonstrate larger variability. . . . . 32

3.3 Distribution of exam cards and the number of exams by anatomy. Top: Distribution of 1097 exam cards stored on scanners at the analysis site by body part. Middle: Number of exams by anatomy. Bottom: This plot shows the number of different exam cards from (a) that are used in the exams in (b). Each color represents one exam card. . . . . 36

3.4 Number of scans and average KPIs comparison per month. The months in brown occur after the exam card change and show a decreasing trend in exam duration. . . . . 37

3.5 Comparison between exam duration and schedule duration. The average scheduled time (denoted by the dot) is longer than the average duration of the MRI. . . . . 41

3.6 Comparison between exam duration and schedule duration (denoted by the dot) for prostate exams. Although the sequence changes lengthened exam durations, the schedule slots were not updated. . . . . 41

4.1 Missingness matrix of the EHR dataset sorted by exam start time. . . . . 48

4.2 Distribution of MRI exam durations across the EHR dataset (n = 33,566). The distribution exhibits strong positive skewness and a heavy right tail. . . 50

4.3 Distribution of MRI exam durations by procedure code (n = 33,566), ordered by median duration and color-coded by subspecialty. . . . . 51

4.4 Distribution of MRI exam durations by subspecialty group, ordered by median duration. . . . . 52

4.5 Distribution of patient and exam characteristics across the EHR dataset (n = 33,566). . . . . 55

4.6 Exam duration by patient and exam characteristics. . . . . 55

5.1 EHR durations vs. MLF durations for fuzzy-merged exams. . . . . 69

5.2 Distribution of exam duration differences ($\Delta$ = EHR duration − MLF duration) across all 30,275 fuzzy-matched exam pairs. . . . . 71

5.3 (a) Bland–Altman plot of the log-transformed data. (Note: The "banding" pattern results from discretization of exam durations.) (b) Bland–Altman plot on the original scale with back-transformed LoAs. . . . . 72

5.4 Schedule adherence represented by $\Delta_{schedule}$. Procedure codes ordered by descending exam count. . . . . 74

5.5 Comparison of template-based scheduling and prediction model-based scheduling. . . . . 77

6.1 Missingness matrix of air quality data. . . . . 88

6.2 Autocorrelation of the target variable CO(GT). . . . . 90

6.3 Autocorrelation of hourly rental count. . . . . 91

6.4 Autocorrelation of daily sales. . . . . 95

6.5 Air quality forecasting performance across historical context window sizes. (a) MAE and $R^2$ metrics. (b) RMSE and $R^2$ metrics. Solid lines show error metrics (left axis), dashed lines show $R^2$ (right axis). . . . . 117

6.6 Sample predictions for air quality dataset using 24-hour historical context window. . . . . 119

6.7 Bike sharing demand forecasting performance across historical context window sizes. (a) MAE and $R^2$ metrics. (b) RMSE and $R^2$ metrics. Solid lines show error metrics (left axis), dashed lines show $R^2$ (right axis). . . . . 121

6.8 Sample predictions for bike sharing dataset using 72-hour historical context window. . . . . 123

6.9 Predicted vs. actual bike demand for 72-hour historical context window. . . . 124
6.10 Rossmann sales forecasting performance across historical context window sizes. (a) MAE and $R^2$ metrics. (b) RMSPE and $R^2$ metrics. The 14-day window represents an anomaly where XGBoost achieves its highest $R^2$ despite moderate MAE. . . . . 128
6.11 Heatmap of learned 12-dimensional store embeddings for 100 Rossmann stores. Each row represents one store's complete embedding vector. . . . . 131
6.12 Cumulative variance explained by principal components of store embeddings. 132
6.13 Validation of learned store embeddings through sales pattern similarity. (a) Time series for stores with similar embeddings (Euclidean distance = 0.40). (b) Scatter plot confirms strong correlation (r = 0.960) between similar stores. (c) Stores with distant embeddings (distance = 1.74). (d) Scatter plot reveals negligible correlation (r = -0.062) between distant stores. . . . . 133

## ACKNOWLEDGMENTS

It feels almost unreal that this long journey, a major chapter of my life, has finally come to an end. It was a path I chose for myself, a chapter I penned in my own hand; yet it would not have been possible without the love and support of so many wonderful people.

I would first like to express my tremendous gratitude to my advisor, Dr. Christina Mastrangelo—thank you for your mentorship, your guidance, and the care and support you show your students. Thank you for opening the door to research for me; it was a pleasure and a great honor to have worked with you throughout this journey. I am also deeply grateful to my committee members, Dr. John Y. Choe, Dr. Prashanth Rajivan, and Dr. Andrea Stocco. I am grateful to have been your student and to have had the honor of working with you. Thank you for your time, your expertise, and your generous support throughout this process.

Next, I would like to thank the faculty and staff at UW ISE. Thank you for your dedication and hard work, which make it possible for students like me to learn and grow here. To my classmates-turned-friends—Peet, Anna, Jiaxin, Tianhao, Tianchen, Feng—thank you for enriching my journey with so much love and joy.

My gratitude also extends to my collaborators at UW Radiology and Philips Healthcare for their support and partnership in making this research possible—in particular, Dr. Mahmud Mossa-Basha, Dr. Shawn Stapleton, and Dr. Thusitha Mabotuwana. I am especially grateful to the MRI technologists at UW Medical Center—Mussie, Noah, Buu Buu, Yuka, Khanh, Charles, Kevin, and many others—thank you for welcoming me into your family and sharing your invaluable knowledge.

To my partner, Steven—thank you for being my rock. I know it has not been easy being a partner to a PhD student; I am grateful for your understanding, your patience, and your unwavering love. To the family and friends I got to know and love through you—thank you

for accepting me and making me feel loved thousands of miles from home.

To my dearest friends from home—Jialing, Yi, and Rui—thank you for your unconditional love and for always being there, no matter the distance.

I owe everything I have achieved to my family, especially my parents. Dad, thank you for instilling in me the courage to aspire and the perseverance to endure hardship. Mom, thank you for being my first teacher and nurturing in me the ability to love and to care. To the rest of my family, I am eternally grateful for your immense love and support.

## DEDICATION

to my parents, Daoyuan and Jirong

# Chapter 1

# INTRODUCTION

Magnetic resonance imaging (MRI) is among the most clinically valuable and operationally complex modalities in modern radiology due to its high contrast resolution, 3D capabilities, and ability to monitor anatomical detail, disease progression, and treatment response. However, the equipment, maintenance, staffing, and specialized training required to operate MRI scanners make it one of the most expensive imaging services a healthcare system provides. Meanwhile, the ever-increasing demand for MRI places constant pressure on radiology departments to improve throughput, reduce patient wait times, and deploy their workforce more effectively, while maintaining imaging quality and patient safety.

Despite the operational complexity of radiology departments, efforts to improve workflow have traditionally relied on the experience and often objective judgment of department leaders, supplemented by ad hoc manual observations and informal knowledge sharing among staff. While valuable and practical, these approaches are inherently limited in scope and scalability. MRI operations generate substantial amounts of heterogeneous data through scanner log files, electronic health records, scheduling systems, and other operational sources, yet the analytical capabilities required to extract actionable insights from these sources remain underdeveloped in most clinical organizations. Consequently, historical data that could reveal valuable patterns in exam duration, technologist competency, equipment utilization, and scheduling inefficiency have largely gone underutilized. This gap between available data and the required analytical capabilities constrains the ability of radiology departments to transition from the current experience-based, reactive management paradigm to an evidence-based, proactive approach to operational improvement.

This dissertation investigates data-driven approaches to MRI operational improvement utilizing a range of methods and data sources. The work begins by characterizing MRI workflow complexity and the role of technologist expertise, then identifies inefficiencies through

operational data analysis, and advances toward predictive modeling of exam duration to support more accurate and adaptive scheduling. A theme that echoes throughout this work is that operational data in clinical settings are rarely analysis-ready—data collected across multiple systems exhibit missingness and discrepancies that require careful reconciliation before any meaningful analysis can be performed. A further ambition of this work was to capture the dynamics of technologist competency development, examining whether competency evolution over time manifests as a detectable and learnable pattern in routinely collected longitudinal operational data. This question could not ultimately be answered with the available clinical data, but it motivated a rigorous methodological investigation into when and why temporal neural models provide meaningful advantages over tree-based approaches, a question with broad implications for future time-series forecast research.

As a whole, this dissertation advances the understanding of MRI operational improvement from multiple perspectives—operational, analytical, and methodological. It also demonstrates that data-driven approaches, when carefully designed and validated, can meaningfully support decision-making in complex clinical imaging environments.

## *1.1 Related Work*

This section provides background on several key areas that are integral to this dissertation. Because each of the subsequent research chapters is based on a published, submitted, or prospective manuscript, targeted literature reviews for the specific methods, as well as prior work that is most relevant to the study are provided within each chapter. For this reason, the related work presented here focuses on shared themes that span multiple chapters yet are not addressed in the manuscripts. The most important among them is the literature on competency assessment and modeling, as it motivates one of the original research questions of this dissertation. Additional context is also provided on temporal forecasting methods, including sequence modeling architectures and representation learning, situating the dissertation within the broader landscape of healthcare operations and time-series forecasting research.

### *1.1.1 Competency Assessment and Modeling in Healthcare*

Competency is a foundational concept in healthcare workforce management; however, its definition remains contested in the literature. An important distinction is between *competence*—referring to a professional's overall suitability for their role, encompassing knowledge, technical skill, clinical reasoning, and values [1]—and *competencies*, which are demonstrable, evaluable components of competence that can be assessed against accepted standards [2–4]. This distinction matters for operational research because competencies, unlike competence, are in principle measurable and therefore amenable to data-driven analysis. Being able to measure competence is considered essential for determining healthcare practitioners' readiness to provide quality care [5].

Given that competence is developmental and context-dependent [6, 7], major healthcare organizations have invested in generic competency frameworks intended to guide assessment across professional settings. Two of the most influential are the CanMEDS framework, developed by the Royal College of Physicians and Surgeons of Canada and recognized as the most widely applied healthcare competency framework in the world [8, 9], and the ACGME (Accreditation Council for Graduate Medical Education) framework of core competencies, which defines six domains required for a physician to enter autonomous practice [10]. Both frameworks comprehensively identify competency facets, but their exhaustive detail has proven difficult to apply directly to practical assessment contexts [11, 12]. This gap between framework and practice has motivated a parallel literature on competency assessment instruments.

Questionnaire-based instruments are the most widely used in this literature, including self-reported surveys assessing nurses' core competencies [13], evidence-based practice competency [14], and factors influencing registered nurse competency, such as years of experience and duty conditions [15]. Other instruments include simulations [16–19], global rating forms [20–22], and direct observation [23–25]. Despite their prevalence, these instruments face well-documented limitations. Quantitative and psychometric studies have repeatedly failed to develop reliable and valid measures even where competency domains are well-defined [12], and self-reported instruments are inherently subjective. Additionally,

many organizations do not assess competency routinely but only when performance problems emerge [5]—a reactive approach that produces sparse and irregular records poorly suited to modeling competency development over time. One proposed remedy is to link competencies to specific performance indicators as proxies for measuring competence [26], an approach that motivates the present work.

In the context of MRI operations, technologist competency is not captured by any standardized instrument and is not directly observable in routinely collected longitudinal data. However, exam duration, which is the time required to complete a scanning procedure, serves as a plausible behavioral proxy for competency development, as more experienced technologists are generally expected to complete exams more efficiently and consistently, and this trajectory should, in principle, leave a detectable pattern in routinely collected operational data over time. Framing competency development this way transforms an inherently qualitative concept into a measurable operational outcome, consistent with the broader recommendation to anchor competency assessment to observable performance indicators [26].

Modeling this trajectory temporally calls for architectures capable of capturing sequential dependencies in temporal data. Gated recurrent units (GRUs), a class of recurrent neural networks designed to learn patterns across variable-length sequences while mitigating vanishing gradients, are well-suited to this task [27]. Unlike tree-based methods that treat each observation independently, GRUs maintain hidden states across time steps and can represent how a technologist's performance at one point in time is conditioned on their prior history, which is the structure that competency development would exhibit if present in the data. This theoretical motivation drove the original design of the forecasting methodology in this dissertation. As described in Chapter 6, the available clinical data ultimately lacked sufficient longitudinal depth to validate this approach on competency outcomes directly; the GRU methodology was therefore evaluated on established benchmark datasets where temporal structure is known to exist, providing a rigorous foundation for its future application in time-series forecast research.

### *1.1.2 Temporal Forecasting Methods and Sequence Modeling*

Electronic health records (EHRs) contain rich but challenging data—noisy, sparse, high-dimensional, and irregular—that must be transformed before it can support downstream machine learning tasks [28, 29]. Representation learning addresses this by automating the extraction of meaningful features from raw data, reducing reliance on manual feature engineering and enabling models to discover patterns that human-designed features might miss [30, 31]. These properties make representation learning well-suited to operational healthcare data, where the relevant signal is often embedded in sequences of events recorded across time.

For temporal EHR data specifically, recurrent neural networks and their gated variants have emerged as the dominant approach. Doctor AI, one of the earliest applications of this paradigm, used GRU-based recurrent networks to predict future diagnoses and medication orders from longitudinal EHR sequences, outperforming non-sequential baselines and showing strong generalizability across patient populations [32]. Patient2Vec extended this approach by incorporating an attention mechanism into the RNN architecture, enabling the model to weight the relative importance of different time points when constructing patient representations—an approach that improved both predictive performance and interpretability [33]. These works establish GRU-based architectures with attention as a well-motivated and well-validated choice for learning from longitudinal clinical sequences.

The large majority of representation learning work with EHRs has focused on patients as the subject of analysis. Doctor2Vec represents a notable departure, proposing a framework that learns dynamic representations of physicians by embedding the patients they have seen as a memory structure, then applying attention to weigh the relevance of past experience to current tasks [34]. The framework outperformed logistic regression, random forest, and LSTM baselines on clinical trial recruitment prediction tasks. This work is directly relevant to the present dissertation, as it demonstrates the feasibility of learning provider-level representations from EHR data, a modeling paradigm that informs the original design of the forecasting architecture developed in Chapter 6, where the goal was to capture technologist competency trajectories rather than patient outcomes.

### *1.2 Contributions and Organization*

This dissertation makes contributions across three dimensions of MRI operational improvement. Operationally, it demonstrates the feasibility of virtual hub-and-spoke deployment of MRI expertise and identifies actionable sources of workflow inefficiency through scanner log file analysis. Analytically, it establishes a reproducible data reconciliation pipeline linking modality log files with electronic health records and develops an exam-level duration prediction model that substantially reduces scheduling error. Methodologically, it provides a rigorous empirical comparison of GRU and XGBoost architectures on established benchmark datasets, characterizing the conditions under which sequential neural models outperform tree-based approaches, a foundation for future temporal competency modeling in clinical operational research.

*Chapter 2* asks whether MRI expertise can be deployed more flexibly across a health system without sacrificing imaging quality or operational efficiency—a better way to deploy expertise. Using discrete event simulation, the chapter presents a feasibility study of a hub-and-spoke virtual operations model in which remote MRI technologists support satellite imaging sites. Originally conceived as a feasibility study, the model has since been piloted at the study site, lending practical credibility to the simulation findings.

*Chapter 3* asks whether MRI scanner log files, which contain data routinely collected by the scanner but historically underutilized, can reveal actionable insights about workflow efficiency—a better way to use data. The chapter presents a systematic analysis of modality log files across multiple imaging sites, applying lean metrics and generalized linear models to characterize sources of variability in scanner utilization and exam throughput.

*Chapter 4* provides the empirical foundation for the predictive modeling work that follows. It documents the data landscape available at the study site, including the structure and limitations of the data, and the challenges involved in reconciling discordant data across these two systems. Natural language processing (NLP) using BioClinicalBERT is explored as a tool for extracting structured information from clinical text, motivating future work on NLP-driven feature extraction and predictive modeling in MRI operational research.

*Chapter 5* asks whether MRI exam duration can be predicted accurately enough to sup-

port more adaptive and efficient scheduling—a better way to schedule MRI exams. Using a carefully constructed dataset that reconciles MLFs with EHRs, the chapter presents a Random Forest model that predicts exam duration at the exam level, achieving mean absolute error (MAE) reductions of up to 57% relative to current scheduling assumptions for 11 of 12 procedure codes.

*Chapter 6* asks whether temporal neural network architectures provide meaningful advantages over tree-based methods for learning from sequential operational data—a better way to model technologist development. Motivated by the original ambition to capture technologist competency trajectories as a learnable temporal pattern, and constrained by the limitations of available clinical data, the chapter evaluates GRU and XGBoost models using three established benchmark datasets, including Air Quality, Bike Sharing, and Rossmann Store Sales, where strong temporal patterns are known to exist. The results demonstrate consistent average GRU advantages ranging from 8.5% to 18.8% across datasets with minimal feature engineering, providing a rigorous methodological foundation for future competency modeling and time-series forecasting work.

*Chapter 7* synthesizes the dissertation's contributions, reflects on the limitations of the studies, and identifies directions for future research.

Chapter 2

# A BETTER WAY TO DEPLOY EXPERTISE: VIRTUAL HUB-AND-SPOKE MODEL FOR MRI OPERATIONS

*This chapter is based on: Li, L.*[1]*, Mastrangelo, C.*[1]*, Briller, N.*[2]*, Tesfaldet, M.*[2]*, Starobinets, O.*[3]*, & Stapleton, S.*[4] *(2022). Prioritizing magnetic resonance (MR) radiology functions for virtual operations: a feasibility study. Journal of Hospital Management and Health Policy, 6.*

## *Chapter Summary*[5]

Virtualization in radiology presents a growing opportunity to improve the efficiency and lower the cost of care delivery. Furthermore, it may provide a means to manage staffing shortages, reduce burnout, and encourage employee development. In radiology, the feasibility tof performing several non-patient-facing imaging procedure tasks remotely is explored. Virtualization of MRI exams is of specific interest as it typically involves a range of complex tasks, and inefficiencies can increase patient wait times (WTs) and reduce overall utilization.

To explore this, a computer simulation model is developed to approximate turn-around time (TAT) and patient WT while accounting for technologist expertise and workflow complexity. The model was validated against 1,618 inpatient/ED exams. Using cognitive task analysis (CTA), complex MR functions are identified from a technologist's perspective that may require a high level of expertise and thus influence workflow durations.

Results showed virtualizing complex, non-patient-facing functions may reduce MR workflow duration by up to 15% and patient in-exam WT by up to 75% when utilizing expert

[1]Industrial & Systems engineering, University of Washington, Seattle, WA, USA

[2]Department of Radiology, University of Washington Medical Center, Seattle, WA, USA

[3]Philips Research North America, Cambridge, MA, USA

[4]Philips Healthcare, Seattle, WA, USA

[5]Adapted from abstract of original article

technologists. For example, scans involving the abdomen and spine and addressing unreported implants benefit the most from experts and may be supported remotely.

In conclusion, modifying the workflow of MRI exams by segmenting complex,non-patient-facing functions to expert technologists within an organization or in a remote center is feasible, as it improves efficiencies and results in a better patient experience. In addition, administration now has guidance on how to effectively deploy a highly-trained workforce in a virtual setting.

### *2.1 Introduction*

Magnetic resonance imaging (MRI) is an important diagnostic scanning tool for the detection and monitoring of specific diseases and conditions. However, equipment, operations, maintenance, and personnel (e.g., MR technologists) make the diagnostic tool expensive, so improving workflow efficiency is valuable. Traditionally, telemedicine in radiology focuses on the electronic transmission of images and radiological reports for image interpretations [35, 36]. However, the processes in the MRI workflow can be improved as advances in computer and communications technology enable users with better tools for the comprehensive exchange of information in real time. This research studies improvements in MR imaging processes using expert technologists to alleviate workflow constrictions. In addition, image acquisition tasks that may be performed in a virtual environment are identified, and the efficiencies of acquiring MRI exams are evaluated. In this chapter, *virtual* indicates work that may be performed in a remote operating center by a group of highly skilled technologists.

The hierarchical structure of the MR imaging functions is shown at the highest level in Figure 2.1 and is depicted using an SADT (Structured Analysis and Design Technique) format [37]. The functions given in this section are the foundational elements of the modeling that is discussed in the next section. The first function, *Review Schedule*, occurs up to 3 weeks in advance of an appointment. The function reviews the exam parameters for exam code, exam protocol and sedation/anesthesia orders and then the patient's record for allergies, body habitus, mobility needs, implants and other special needs. The *Prep Patient* function (A02) is performed by both technologists and medical assistants. The function

begins with confirming arrival and patient identification in the electronic health record (EHR) then clearing implants that were not reported at the time the appointment was made, assigning lockers for patient changing, starting an IV or other port access, handling special needs (i.e., pacemaker, claustrophobia) and conducting final verification before walking the patient to the scanning room. The *Release Patient* function (A04) begins when the patient leaves the scanning room and ends after the patient changes, receives post-exam instructions and has any ports removed.

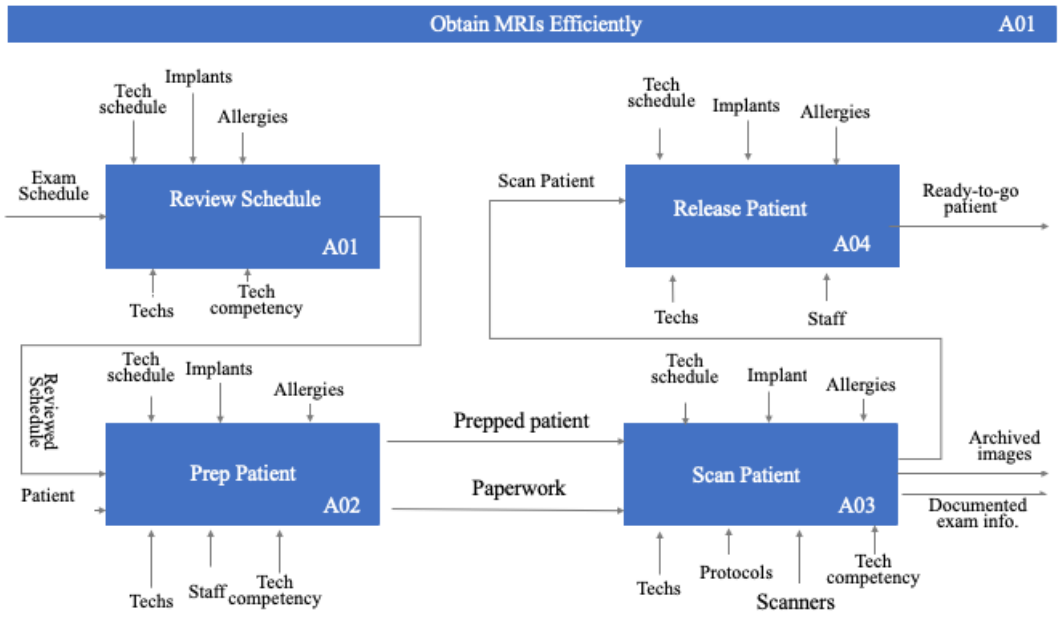


Figure 2.1: SADT diagram of highest-level MR image acquisition functions.

Figures 2.2 and 2.3 show the next level of functional decomposition which involves all activities that may affect image quality for the *Prep Patient* (A02) and the *Scan Patient* (A03) functions, respectively. Each top-level function, A01–A04, shown in Figure 2.1 was decomposed two additional levels to fully capture the MR workflow; however, *Prep Patient* (A02) and *Scan Patient* (A03) functions are only given here because they are most relevant to the discussion herein. The *Prep Patient* function starts with identifying and screening the patient, which involves asking about the patient's past medical treatments and procedures

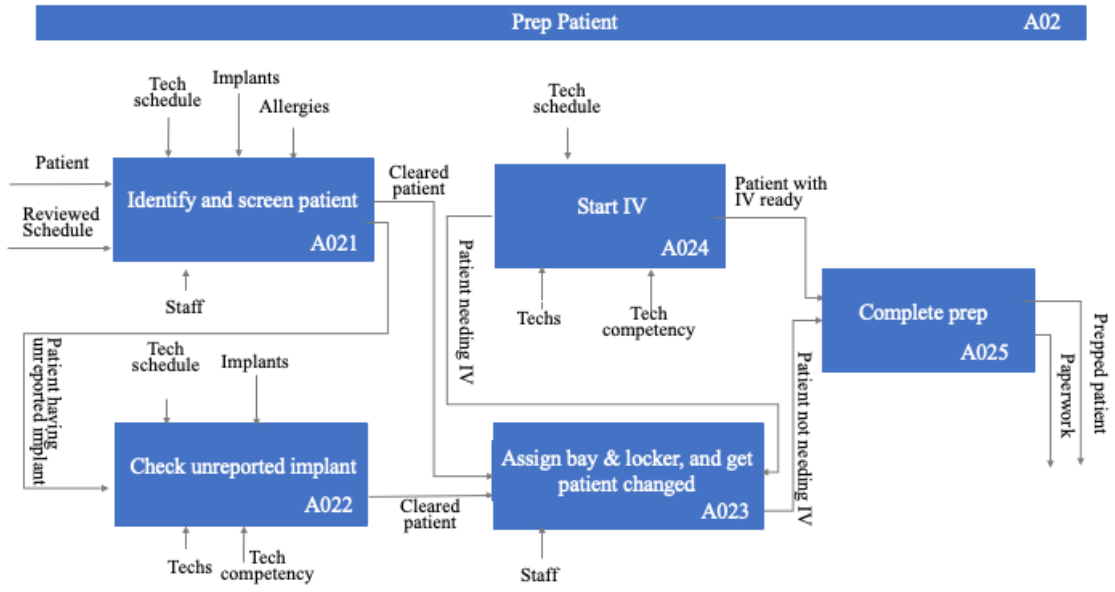


Figure 2.2: SADT diagram of Prep Patient sub-functions.

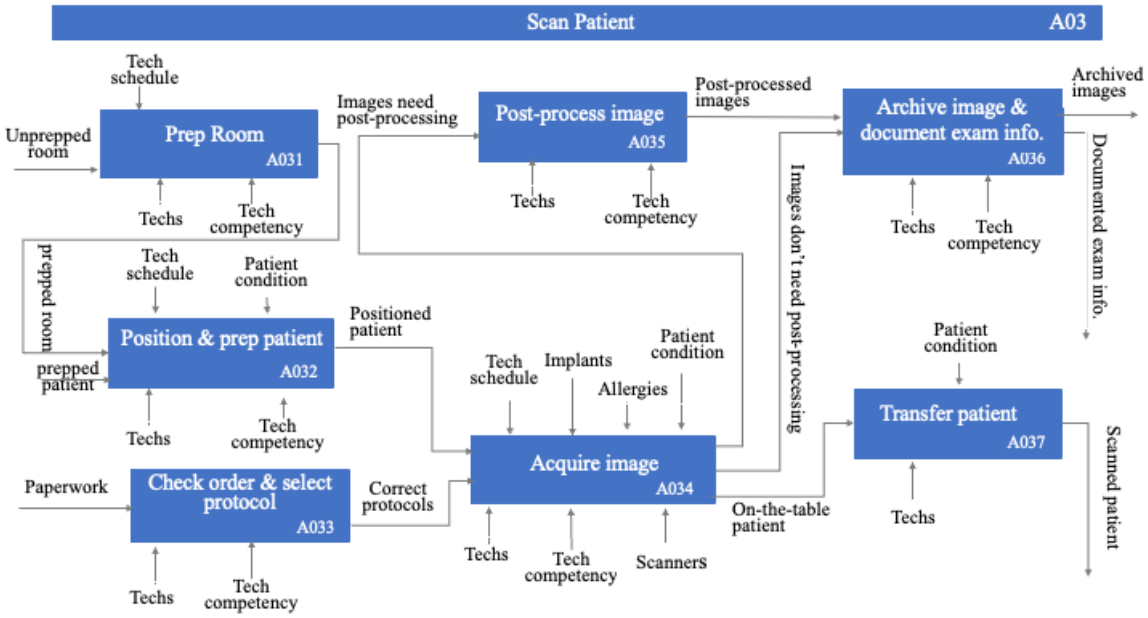


Figure 2.3: SADT diagram of Scan Patient sub-functions.

that may pose risks to patient safety during the MR exam (including but not limited to implants, surgical incisions, etc.). If an implant that is not reported in the patient's record surfaces in the screening process, the details about the implant will need to be obtained and addressed by a knowledgeable technologist to have it cleared; otherwise, the patient will be assigned a bay and a locker and change into the proper attire. If the patient needs an IV as suggested by the exam code, then after changing, an IV will be started. *Prep Room* includes preparing the coils and positioning devices as well as performing a last safety screening. Proper positioning of the patient ensures good images as well as patient safety and comfort. While the *Review Schedule* function has already occurred, there may be last-minute updates, so the order is reviewed and the protocol is selected from the MR scanner. The protocol may be updated with additional sequences at that point to ensure that the anatomy is fully scanned. As the protocol sequences are run in the *Acquire Image* function (A034), the technologist is continually monitoring the quality of the images. Some images, such as spectroscopy and angiograms, require post-processing before being sent to the picture archiving and communication system (PACS). Finally, exam details for billing are entered while the patient is escorted back to the changing bay and released. After a baseline simulation model that represents the traditional MR workflow is developed and validated, two expert-usage scenarios are analyzed. Turn-around time (TAT) and wait time (WT) associated with the two expert-usage scenarios decrease across all modeled MR functions and patient attributes when compared to the baseline model. These findings are extended to explore workflow virtualization for a remote operating center that will not only increase efficiency but has the potential to increase patient access, workforce development and care standardization.

## *2.2 Methods*

Opportunities for virtual operations reside with tasks that are most demanding in the MR workflow. A cognitive task analysis (CTA) is conducted to identify those demanding MR activities. Based on the findings of the CTA, a discrete-event simulation (DES) model is developed and used to explore different expert-usage scenarios.

### *2.2.1 Data Sources*

The data used in this study come from two sources: observations and scanner log files. Over 180 hours of observation were spent in the MR operations area conducting the CTA, characterizing the operations for functional decomposition, and gathering metric data for functional tasks. Log file data consisting of 68 variables on approximately 114,000 exams were utilized for scan time duration by anatomy, number of exams by exam type, and patient characteristics (i.e., age, weight, gender). This study focused on the seven anatomies that represent more than 95% of all outpatient diagnostic scans (Table 2.1). Scanner time mean and variability differ by anatomy, so the anatomy of the patient being scanned must be accounted for. For simplicity, only the top five anatomies, which account for more than 80% of all outpatient diagnostic scans, are modeled.

Table 2.1: Mean and standard deviation (SD) of scanner time by anatomy (minutes)

| | Brain | Head | Liver | Abdomen | Spine | Knee | Heart | Other |
|---|---|---|---|---|---|---|---|---|
| Mean | 39.1 | 34.81 | 37.4 | 40.05 | 44.72 | 35.59 | 75.20 | – |
| SD | 15.7 | 12.53 | 14.85 | 15.29 | 23.22 | 15.5 | 23.01 | – |
| % exams | 28% | 18% | 13% | 13% | 11% | 5.9% | 5.9% | 5.2% |

### *2.2.2 CTA: Identification of Demanding MR Activities*

Many of the functions described in the SADT diagrams are complex, and their durations will be impacted by the experience of the technologist. As such, the MR workflow first needs to be characterized from a cognitive perspective rather than a purely behavior-based task analysis. Applied cognitive task analysis (ACTA) is used to elicit and capture difficult cognitive elements, the rationale for their difficulty, and potential errors [38–40]. Figure 2.4 presents selected findings from the CTA that are considered "complicated" from a technologist's perspective; these findings identify the need for capturing and modeling expertise. Expert technologists are utilized in all functions except *Release Patient* (A04).

| Difficult cognitive element | Rationale | Potential errors |
|---|---|---|
| Knowing whether the exam codes and protocols are appropriate | Correct protocols need to be selected for the exam. Discrepancies arise in the comments from prescribing physicians, protocolling radiologists and schedulers and the protocols to be used | Missing Important information in the comment lines. Not discovering a conflict between the intent of the scan order and the scheduled exam |
| Obtaining implant & allergy information and scheduling the exam accordingly | Various implants exist and clearance is needed. The exam needs to be scheduled on the appropriate imaging machine | Failing to notice implant information may increase exam duration. Scheduling the exam on the wrong machine |
| Detecting that the patient has an unreported implant(s) | Patients fail to report an implant in the screening process Patients may lack knowledge of medical terminology | Failing to obtain information on unreported implants from patients may affect image quality or even damage the implants. Failing to clear implants may increase exam duration |
| Starting patient IV | Blood vessels may be hard to find | Failing to start an IV in two attempts (as per department policy) |
| Detecting mistakes or irrationalities in protocol and order | Extensive knowledge of human anatomy, pathology, and protocol, etc., is needed | Discovering the conflict during imaging or after post-processing increases exam duration |
| Planning images based on survey/previous scans to cover the pathology adequately | Parameters (e.g., FOV, # of slices, etc.) need to be set or adjusted to get the best view of the pathology on the images | Improperly setting the parameters to obtain the best images |
| Identifying the area(s) of interest for vague orders/protocols | Knowledge of human anatomy and disease pathology is needed to obtain images | Missing visibility of the pathology in the images |
| Identifying artifacts that cause poor image quality | Artifacts need to be detected and then controlled or eliminated | Not understanding the cause of the artifact |
| Modifying the exam card to obtain the best image (e.g., add/repeat sequences) | Technologist knowledge of imaging sequences is needed to modify the exam card | Repeating sequences or adding additional sequences increases scanning time |
| Positioning patients with special conditions/ needs | Technologists need to know how to use different positioning devices and switch coils when needed | Failing to position patients well may result in uncomfortable patients, images with motion or artifacts, unsafe patients |

Figure 2.4: Selected findings from the cognitive task analysis by MR workflow function. Color denotes high-level function: Review schedule (A01), Prep patient (A02), Scan patient (A03). FOV, field of view.

### *2.2.3 Modeling Expert Knowledge*

For a given task, a technologist's completion time reflects a relative experience level. As expertise accrues, the time technologists spend on MR operational tasks tends to decrease [41]. To model expert knowledge, task times are randomly sampled from designated probability distributions. Because expert task time is not documented for any tasks in the MR workflow, a triangular distribution is used as a subjective description of the population.

Two different classes of task time distributions are used: the *expert technologist task time distribution* ("expert distribution") and the *average technologist task time distribution* ("average distribution"). For any task, experts tend to have a smaller mean task time and a smaller variability. Historical data were not available for expert distributions; instead, the mode and the upper limit of the respective average distributions were decreased by a specific amount (assuming the lower limit stays the same). The percentage of decrease is determined by findings in [41] and verified by subject matter experts (SMEs).

### *2.2.4 Discrete-Event Simulation (DES) Model*

DES has long been used to model healthcare systems [42–46]. A DES model is developed and implemented using the SimPy package in Python (Version 3.8.3) [47, 48]. The key output metrics are TAT and WT. TAT is defined as the time from when a patient begins the identity confirmation and safety screening function (A021) to when the patient is returned to the changing area (A04). WT is the time that the patient is waiting for a resource such as a technologist, medical assistant, or scanner; WT is included in the TAT.

For each simulation run, a pool of 500 patients is generated, and each patient is assigned five attributes: (I) habitus (normal, obese); (II) exam type (brain, head, liver, spine, abdomen); (III) implant (no implant, unreported implant, non-interfering implant, conditional pacemaker, non-conditional pacemaker); (IV) IV (needed, not needed); and (V) allergy (yes, no). The frequency of occurrence of each attribute is determined by observation, log-file analysis, and SMEs. Table 2.2 shows how a simulated patient is represented and stored as a vector in the patient pool. The simulation model contains 12 functions in total decomposed from the highest-level functions in Figure 2.1, and patients arrive at a fixed interval of 15

minutes.

Table 2.2: Patient attribute distributions and example patient profile used in the simulation model.

| | **Habitus** | **Exam type** | **Implant** | **IV** | **Allergy** |
|---|---|---|---|---|---|
| | Normal 90% | Brain 34% | No implant 65% | Needed 80% | Yes 15% |
| | Obese 10% | Head 21% | Unreported implant 5% | Not needed 20% | No 85% |
| | | Liver 16% | Non-interfering implant 20% | | |
| | | Abdomen 16% | Conditional pacemaker 5% | | |
| | | Spine 13% | Non-conditional pacemaker 5% | | |
| **Example patient** | Normal | Head | No implant | Needed | No |

### *2.2.5 Baseline Model Validation*

The baseline simulation model is validated against MR inpatient/ED TAT data collected between March 2019 and February 2020 at the academic facility where the study was conducted. Because inpatients and ED patients are typically prepped before arriving at the MR unit, the data include only the time from when they enter the scanner room to when they leave; this process is captured by the *Scan Patient* (A03) function.

Data preparation required the elimination of duplicate records, infeasible exam records, and the concatenation of exam cards. Exams were grouped by exam code, and those with durations below the 10th percentile or above the 90th percentile within each category were removed. The top 14 most frequently occurring exam codes were retained, representing approximately 80% of total cases. Prior to cleaning, 2,931 exams comprised the dataset; after cleaning, 1,618 exams remained (446 excluded for duplication, 474 for exam codes outside the study scope, 393 for unreasonable durations).

The mean duration for the scan-patient function in the historical data is 41.36 minutes; the mean for the simulation model scan-patient function is 46.70 minutes. The $(\mu + 1\sigma)$ intervals are (28.91, 53.81) and (37.04, 56.36), respectively. The difference is likely caused by a small difference in exam types between inpatients and outpatients. SMEs concluded

that the simulation model output is in accordance with the historical data, and the model is validated.

### *2.2.6 Expert-Usage Scenarios*

Based on the validated baseline model and CTA findings, two expert-usage scenarios are examined. Weinger & Slagle [41] found that expert anesthesiologists may spend up to 50% less time on specific tasks than novice anesthesiologists. Since the MR technologists observed were not complete novices, a task time reduction well below 50% is assumed: the mode is decreased by 10% and the upper limit by 20% in the expert distribution compared to the average distribution. The resulting triangular distribution parameters for all expert-eligible functions are given in Table 2.3.

The **All-Expert-Onsite** scenario assigns all tasks that benefit from increased capability to expert personnel only. Such tasks include check unreported implant (A022), start IV (A024), prep room (A031), position and prep patient (A032), check order and select protocol (A033), acquire image (A034), and post-process image (A035). All task durations are drawn exclusively from expert distributions, representing an upper bound on system performance.

The **Remote-Technologist** scenario moves towards virtualization by offloading non-patient-facing tasks to a remote location staffed exclusively by experts. Only non-patient-facing tasks are considered because some functions requiring expert help—such as starting an IV—cannot be effectively assisted in a virtual environment. Functions suitable for virtualization include check unreported implant (A022), check order and select protocol (A033), acquire image (A034), and post-process image (A035). This scenario represents a situation where a local site is administered by medical assistants and/or nurses while a remote site supports multiple local sites.

### *2.2.7 Statistical Analysis*

All statistical analyses were performed in R (RStudio Version 1.1.456). Pairwise $t$-tests were used to determine whether statistically significant differences exist between the mean TAT and mean WT of all three models.

Table 2.3: Triangular distribution parameters for MR functions utilizing experts

| Function | Average Distribution | | | Expert Distribution | | |
|---|---|---|---|---|---|---|
| | Parameters (min) | Mean | SD | Parameters (min) | Mean | SD |
| Check unreported implant (A022) | (5, 7, 30) | 14.00 | 5.67 | (5, 6.3, 24) | 11.77 | 4.33 |
| Start IV (A024) | (3, 7, 20) | 10.00 | 3.63 | (3, 4.5, 16) | 7.83 | 2.90 |
| Prep room (A031) | (2, 3, 4.5) | 3.17 | 0.51 | (2, 2.7, 3.6) | 2.77 | 0.33 |
| Position & prep patient (A032) | (2.1, 3.9, 5.2) | 3.73 | 0.64 | (2.1, 3.5, 4.2) | 3.27 | 0.44 |
| Check order & select protocol (A033) | (0.5, 0.8, 1.3) | 0.87 | 0.16 | (0.5, 0.7, 1.0) | 0.73 | 0.10 |
| Post-process image (A035) | (4.5, 6, 8) | 6.17 | 0.72 | (4.5, 5.4, 6.4) | 5.43 | 0.39 |

Note: Acquire Image (A034) parameters vary by anatomy; see original manuscript for full values.

## *2.3 Results*

### *2.3.1 Baseline Model Results*

Figure 2.5 shows the results from one simulation run with 500 simulated patients. The baseline model is replicated 30 times, and the mean TAT and mean WT over the 30 simulation runs are shown in Table 2.4.

The baseline model provides insights into the patient attributes. Figure 2.6 shows that exam type and implant status tend to affect exam duration, while patient habitus, patient allergies, and IV requirements do not. Exam type and implant status are further explored. Figure 2.7 shows that patient characteristics do not significantly affect WTs. MR exam durations vary by anatomy, but utilizing experts may reduce longer scan times and reduce TAT. In addition, improving implant knowledge may also reduce TAT.

### *2.3.2 Expert-Usage Scenario Results*

Each of the expert models is replicated 30 times, and the results are shown in Table 2.4. Under all scenarios, the use of expert technologists reduces TAT and WT. By simply offloading the non-patient-facing functions to expert technologists in the remote-technologist model, the average TAT per exam is reduced up to 12.54 minutes and the average WT is reduced up to 6.44 minutes when compared to current operations where expert help is provided on

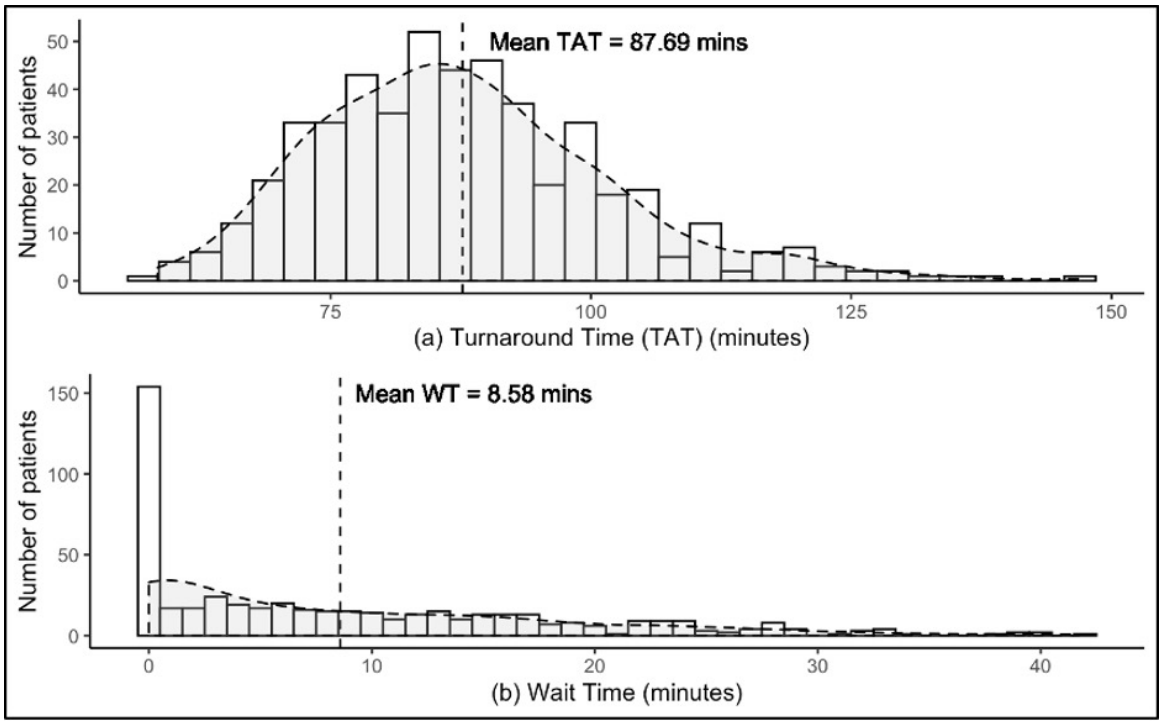


Figure 2.5: Baseline simulation model results.

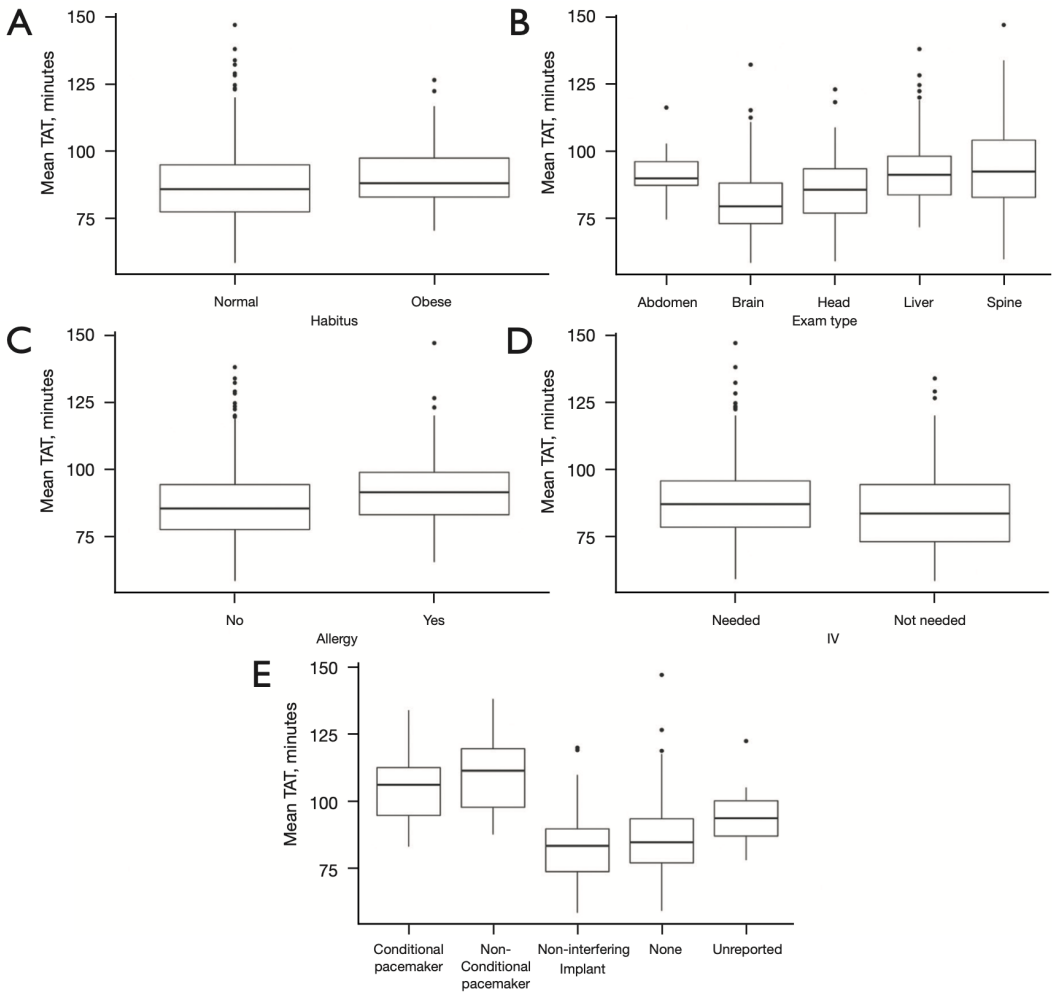


Figure 2.6: TAT by patient attributes for baseline model: (A) habitus; (B) exam type; (C) allergy; (D) IV; (E) implant.

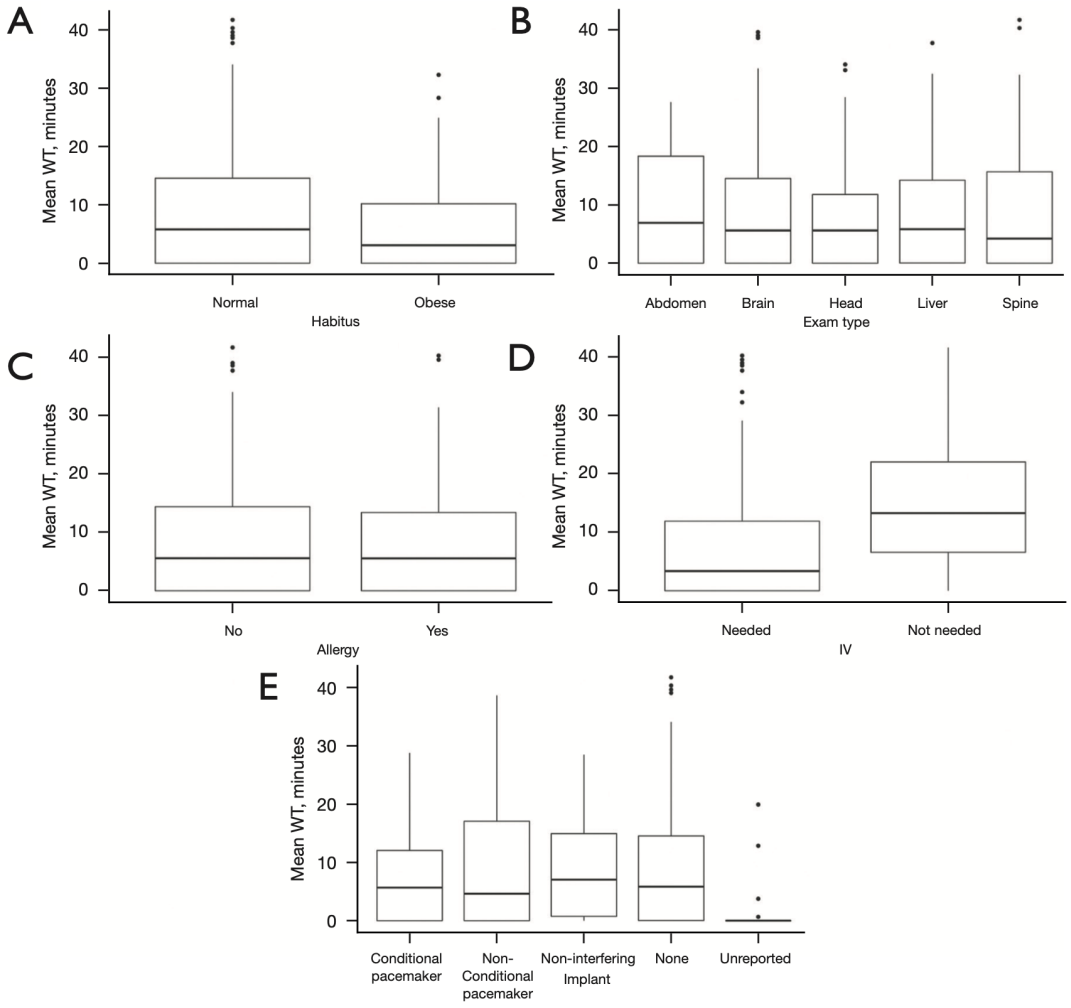


Figure 2.7: WT by patient attributes for baseline model: (A) habitus; (B) exam type; (C) allergy; (D) IV; (E) implant.

an informal basis. Note that the remote-technologist model has a comparatively longer TAT and WT because only the non- patient-facing functions are studied. In other words, this model only utilizes durations from four expert distributions, whereas the all-expert-onsite model uses durations drawn only from the expert distributions.

Table 2.4: Simulation results of baseline and expert models

| Model | Mean TAT (min) | Mean WT (min) |
|---|---|---|
| Baseline model | 87.69 (1.45) | 8.58 (1.27) |
| All-expert-onsite model | 71.59 (0.40) | 1.25 (0.20) |
| Remote-technologist model | 75.15 (0.46) | 2.14 (0.28) |
| Example (abdomen, spine, unreported implant) | 83.62 (0.81) | 5.54 (0.65) |

Note: Standard deviation given in parentheses.

### *2.3.3 An Illustrative Example*

While the all-expert-onsite model represents an idealized situation where all tasks are performed by experts, a more realistic situation is that experts are "tapped" on an informal basis when needed. Moreover, it is assumed in this example that the expert workload comprises both scheduled and on-demand components. The scheduled workload includes exams that are known to be difficult to perform for the less experienced technologists, and the on-demand workload includes impromptu requests for help. Note that abdomen and spine exams have longer image acquisition durations and/or variability, and are generally perceived as more challenging by MR technologists, Therefore, in this example, when these exams are on the schedule, they are routed to a remote operations center staffed by expert technologists. Since unreported implants require evaluation by an expert technologist, they are also be routed to the remote operations center; this represents an unscheduled or on-demand workload for the technologist. Table 2.4 (bottom row) shows the metrics for this example: offloading more complex exams and potentially risky implant situations reduces TAT and WT when compared to the current operations (i.e., baseline model).

### *2.3.4 Comparison*

Pairwise $t$-tests are used to determine if statistically significant differences exist between the mean TAT and mean WT of all three models: the baseline model and the two expert models. The box plots in Figure 2.8(A) represent the mean TAT for each model and the stars above the horizontal lines denote the level of significance for each of the 3 pairwise comparisons—all of which are significantly different. Figure 2.8(B) plots the same information for mean WT. Again, all pairwise comparisons are significant. The findings suggest that by having experts in the workflow, efficiency gains and reductions in variability can be achieved. What's more, the higher the expert involvement, the more efficiency gains are realized. While the remote-technologist model does have longer TAT and WTs than the all-expert- onsite model, the efficiency gains are substantial compared to the current workflow.

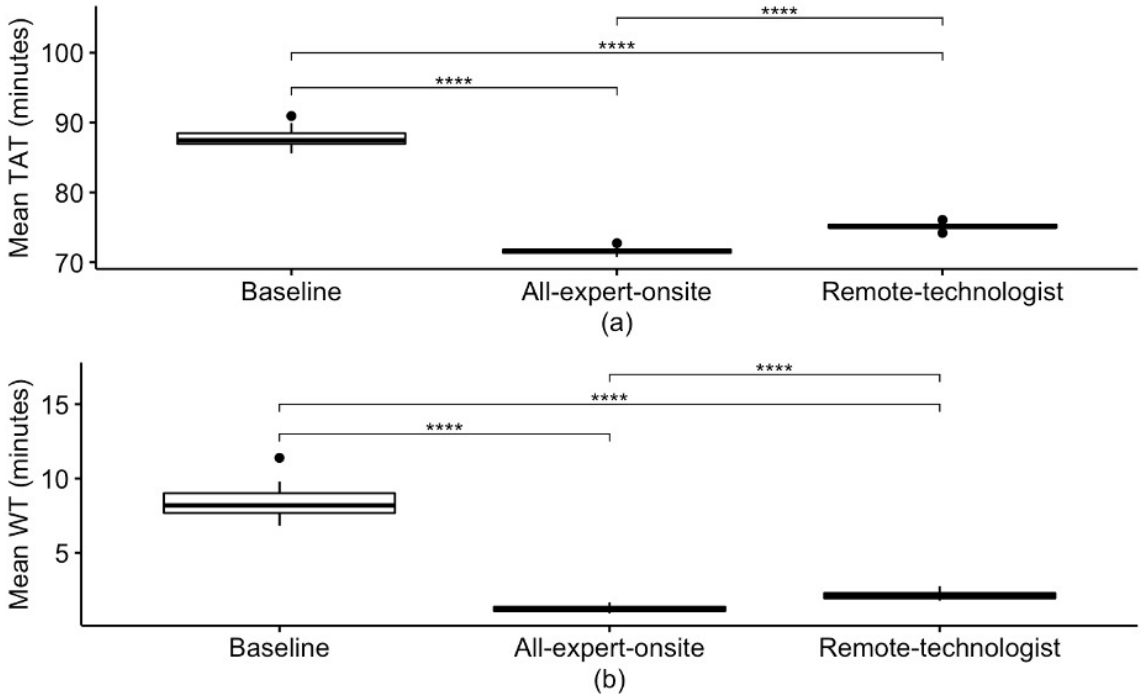


Figure 2.8: Pairwise comparison of mean TAT and mean WT of different models: (A) mean TAT (minutes); (B) mean WT (minutes). ****: $p < 0.001$.

### *2.4 Discussion*

Teleradiology is typically associated with the electronic transmission and processing of radiological images; however, advances in communication technology can support more efficient and reliable real-time exchange of information that is distinct from MR scanner technology. At the same time, healthcare is shifting toward a value-based care model. Virtualizing MR image acquisition can increase value by optimizing workflow, improving staff efficiency, and providing an opportunity for workforce development.

In this chapter, the activities of the MR workflow were analyzed from the competency perspective of a technologist, and functions with increased cognitive demand were identified. These functions represent opportunities for improved efficiency through the targeted use of expert technologists. Using DES, two expert models were explored. Results show that by offloading the functions of checking unreported implants and acquiring images to expert technologists, the average TAT per exam is reduced by up to 12.54 minutes and the average WT is reduced by up to 6.44 minutes compared to the traditional workflow in which expert help is provided on an informal basis.

Since these key functions do not require patient interaction, they are suitable for virtualization. Using experts in a remote capacity is feasible and significantly improves efficiency. In addition, using experts in a virtual environment offers further benefits:

1. **Increased scanner utilization.** Expert technologists could support multiple imaging rooms, including scanners at underutilized sites in rural areas.

2. **Improved patient experience.** Expert technologists function as shared assets in a network, reducing the need for patients to travel to receive high-quality care.

3. **Continued workforce development.** Formalizing expert roles provides a path to recognition, continued competency development, and increased job satisfaction.

4. **Increased care standardization.** A remote center staffed with expert technologists enables consistent image quality and diagnostic outcomes across both rural settings and academic medical centers.

The limitations of this work lie in what was not studied. Although the anatomies, exam codes, technologist and medical assistant functions, and workflow processes included represent more than 80% of the MR imaging acquisition workload, a bias may result from the exams not included. In addition, the study site is typical of large academic medical centers and may not reflect small radiology clinics or diagnostic imaging chains. However, the use of DES as the underlying modeling mechanism provides generalizability for application in not only MR image acquisition but other healthcare delivery workflows where expert knowledge affects process times and virtual operations are considered.

Advancements in virtual operations contribute to faster TATs and a better patient experience, and help increase the overall productivity of MR departments while providing greater operational sustainability. A remote site composed of expert technologists warrants further work in the areas of assessing and characterizing technologist competency, balancing image acquisition tasks with cognitive load in a virtual environment to maintain expert resiliency and capacity, and understanding the impact of task handoff between on-site and virtual locations on patient safety.

Chapter 3

# A BETTER WAY TO USE DATA: MRI LOG FILE ANALYSIS FOR WORKFLOW IMPROVEMENT

*This chapter is based on: Petroianu, L. P.[1], Li, L.[1], Mieloszyk, R. J.[2], Mastrangelo, C. M.[1], Stapleton, S.[3], & Hall, C.[3] (2024). MRI log file analysis for workflow improvement. Current Problems in Diagnostic Radiology, 53(2), 192-200.*

***Chapter Summary***[4]

Magnetic Resonance Imaging (MRI) is an important diagnostic scanning tool for the detection and monitoring of specific diseases and conditions. However, the equipment cost, maintenance, and specialty training of technologists make the examination expensive. Consequently, unnecessary scanner time caused by poor scheduling, repeated sequences, aborted sequences, scanner idleness, or capture of non-diagnostic or low-value sequences is an opportunity to reduce costs and increase efficiency. This chapter analyzes data collected from log files on 29 scanners over several years. 'Wasted' time is defined, and key performance indicators (KPIs) are identified. A decrease in exam duration results when actively modifying and monitoring the number of sequences that comprise the exam card for a protocol.

## *3.1 Introduction*

The increase in usage of diagnostic imaging—including Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and interventional X-ray (iXR)—has a projected annual growth rate of over 5% [49]. Cross-sectional imaging procedures are high-cost with the potential for large variability. Due to the high cost of MRI exams, improving efficiency in

[1]Industrial & Systems engineering, University of Washington, Seattle, WA, USA

[2]Microsoft, Raleigh, NC, USA

[3]Philips Healthcare, Department of Radiology, University of Washington, Seattle, WA, USA

[4]Adapted from abstract of original article

the imaging process is imperative and has been studied to improve access and reduce cycle time [50, 51], to identify bottlenecks [52], to optimize scheduling [53], to optimize image reading [54], and to study automation [54]. These studies rely on observational data collected either manually or electronically. Confined observation limits the time and volume of data analyzed, may make it difficult to guarantee data diversity, and introduces the risk that the presence of an observer modifies staff behavior.

An opportunity exists for utilizing MRI scanner log files to improve workflow efficiency because of the quantity, type, and automatic collection of data. Scanner log files track details of an MRI exam that are not easily accessible from other IT systems, including exam logistics (e.g., reliable start and end times), equipment status (e.g., table movement to infer patient positioning), exact protocoling, sequence repetition, faults (e.g., equipment failure or maintenance), and patient characteristics (e.g., age, weight). MRI log files have been studied in several ways: using MRI acquisition parameters to predict missing body regions in log files [55], using imaging and compressed imaging in parallel to improve exam efficiencies [56], and analyzing metadata to reduce process variability and optimize patient care [57]. In this study, data are taken from 29 MRI scanners located in different regions of the United States. Analysis of the log file data provides an opportunity to understand MRI scanner utilization and identify factors leading to long scan durations, unproductive sequences, and unnecessary sequence repetitions.

In this chapter, lean manufacturing principles [58] are used to facilitate the analysis of log files by identifying wastes and subsequently defining KPIs. The analysis reveals that an improvement opportunity exists in modifying and monitoring the number of sequences that comprise the exam card for a protocol. This results in a measurement methodology that may be used to improve MR imaging operations. Analytical models then identify relevant physiological factors that may influence the KPIs. While the analytical work focuses on MRI scanners at one site, the findings are replicable and extensible to other sites.

### *3.2 Analysis and Performance Metrics*

The initial step is to understand and preprocess the log file data. Figure 3.1 maps the MRI imaging steps based on the log file variables and summarizes the exam processes from ‘Exam

Start' or first table movement to 'Exam End' or last table movement. The data contains 60 variables on 373,711 exams that were completed from November 2013 to May 2018 on 29 scanners. The quantity of extracted data supports a detailed and rich analysis; however, the dataset contains outliers, system errors, and other factors that need to be reconciled. Since the focus of this effort is outpatient exams, start times outside the hours of 7am to 5pm were removed since the institution typically schedules inpatient exams in the evening. Exams with durations lasting more than two hours were removed as the institution has no protocols with a duration greater than 60 minutes. Exams with a gap greater than 4 hours were also removed; this occurs when one exam is the last exam of the day and the next log file entry on the scanner is the first exam of the next day. These exams are assumed to represent incomplete log files. Exam records were also removed when patient characteristics, such as age, gender, and weight, were missing. Finally, exams from ED patients were removed. After preprocessing, the dataset contains 114,214 outpatient-focused exams. Note that these data preprocessing filters were set by the MRI technologists and imaging specialists. The dataset contains 68 variables as 8 new variables were created to facilitate analysis, such as day of the week, time periods, and broader patient characteristics (i.e., weight and age categories). The data preprocessing and cleansing heuristic was validated by conducting a merge of the log file data with RIS data. This activity entailed matching exam date, gender, and age with approximate exam start time along with a fuzzy merge of exam description. This resulted in a log-file dataset comprising 11K exams, and this data is statistically equivalent, using confidence intervals on correlation coefficients between scanned body part and KPI (defined in next section), to ∼114K exam dataset which is used in the remainder of this chapter.

### *3.2.1 Lean Analysis and Key Performance Indicators*

A lean methodology, specifically the seven wastes concept, was used to identify wasted time during the exam. Previous studies have demonstrated the ability to improve performance in radiology through such an approach [51,53,59–62]. Table 3.1 identifies wastes from a log-file perspective. Strictly speaking, any activity not directly associated with successful image scanning and storage is considered a waste. However, it is not always practical or technically

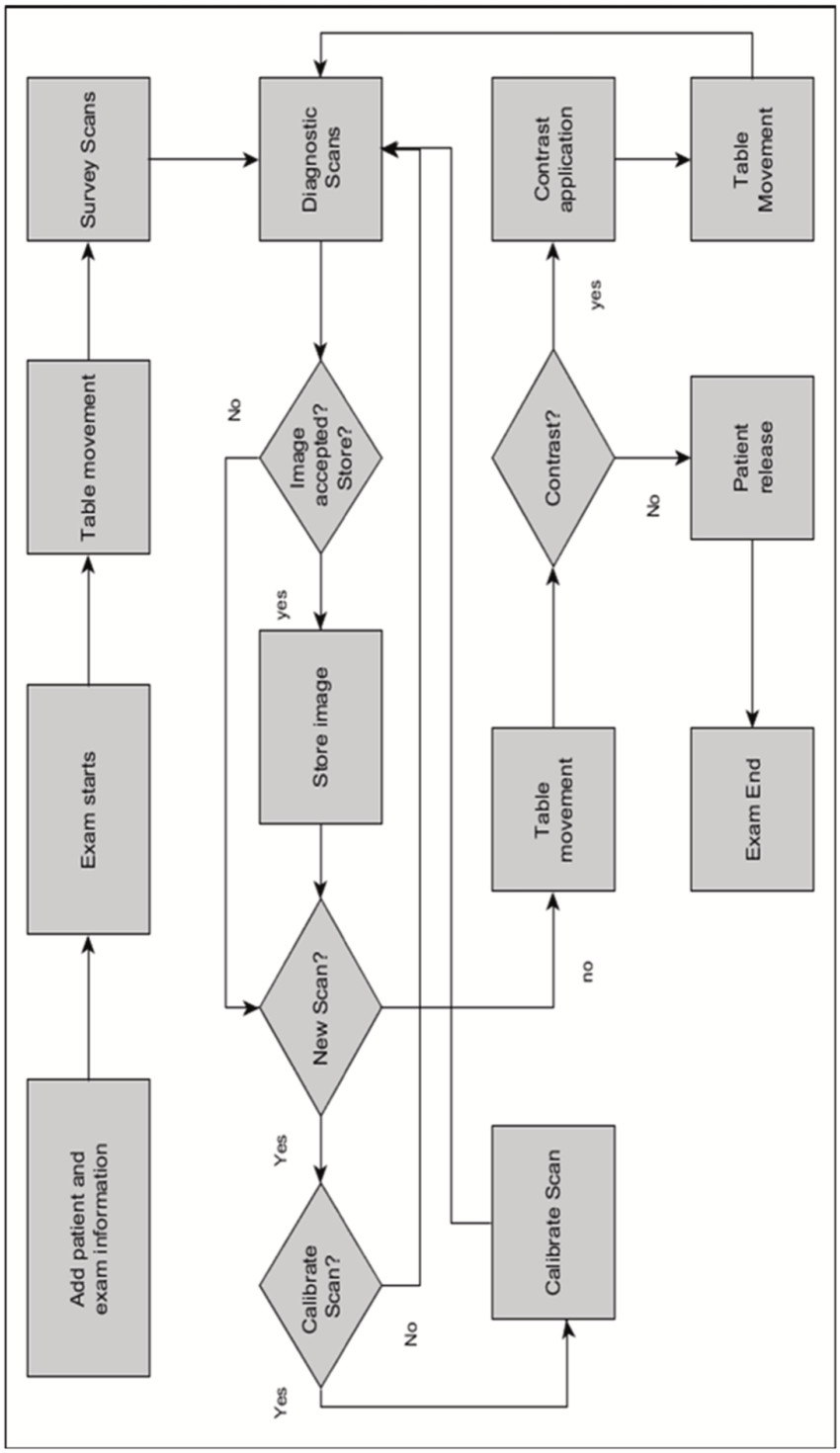


Figure 3.1: Flowchart for a typical MRI exam, based on the log file information. The exam starts with the first table movement and ends after the last table movement. The log files capture data related to each one of these activities.

feasible to eliminate a wasteful step—for example, the time associated with administering contrast, positioning the patient, or performing survey scans. On the other hand, scanning time may be considered a 'waste' if a scan is repeated or never reaches the picture archiving and communication system (PACS).

Table 3.1: The 7 lean 'wastes' interpreted for the MRI scanning process.

| **Waste** | **Brief Definition** | **MRI Waste Metrics** |
|---|---|---|
| Waiting | Waiting for the end of the previous process; waiting for material | Time between exams; total idle time; time between exam start and first table movement; time between scanner setup and first scan; idle time between scans |
| Defects | Rework due to product/process problems | Repeated sequences/scans; total time on repeated or aborted sequences/scans |
| Over-Processing | Processing more than is needed | Repeated sequences; repeated exams; excessive scanning time |
| Over-Production | Producing more than is needed | Unnecessary sequences/scans |
| Motion | Unnecessary movement within the process | Staff movements; excessive table movements |
| Inventory | Material or products not being processed; staff non-productive time | Not applicable to MRI log file variables |
| Transport | Unnecessary movement between processes; same staff working in different rooms | Not applicable to MRI log file variables |

Note: MRI Waste Metrics are included as representative examples.

Ten candidate KPIs that can be measured using the data in the MRI log files were identified. The measurable features were reviewed with subject matter experts for their potential as an indicator of MRI exam performance or as a driver of MRI variability. The

variability of each KPI by body part, by scanner, and as a whole was analyzed. With input from imaging specialists, three are considered actionable for improvement and the following KPIs were selected:

1. **Exam Duration:** The total duration of the exam taken from the start and end times in the log files. Note that this time includes all scan time (survey, calibration, and diagnostic), image acquisition time, contrast administration, and table movements.

2. **Total Idle Time:** The total time within the exam duration during which the MRI scanner was inactive.

3. **Fraction of Repeated Sequences:** This metric is calculated as the number of repeated sequences divided by the total number of sequences that occurred during the exam. A fraction is used as exam cards are comprised of a different number of sequences and converting to a fraction enables comparison across exam cards. For example, a primary brain tumor exam card may be comprised of 9 sequences and repeating one sequence results in a fraction of 0.11. Whereas a cardio myopathy exam card may have 24 sequences and repeating one sequence results in a fraction of 0.04.

The last two KPIs are related to defects/rework in the process and waiting; 'Total Idle Time' represents non-productive time in the process and 'Fraction of Repeated Sequences' (henceforth, called fraction- repeated) corresponds with errors that occurred and demand rework. Although 'Exam Duration' is not a waste as presented in Table 3.1 it directly corresponds to over-processing/production and excessive scanning time. It is also an important measure because it represents the total exam time to either reduce or stabilize.

Figure 3.2 demonstrates high variability within each KPI by anatomy. The six anatomies shown—abdomen, brain, head, liver, pelvis, and prostate—occur most frequently at the analysis site (comprising 4 MRI scanners), and the variability pattern is consistent across all 29 scanners. The fraction-repeated KPI exhibits particularly high variability across anatomies. Exams that occur more frequently, such as brain and head exams, exhibit less variability across all three KPIs.

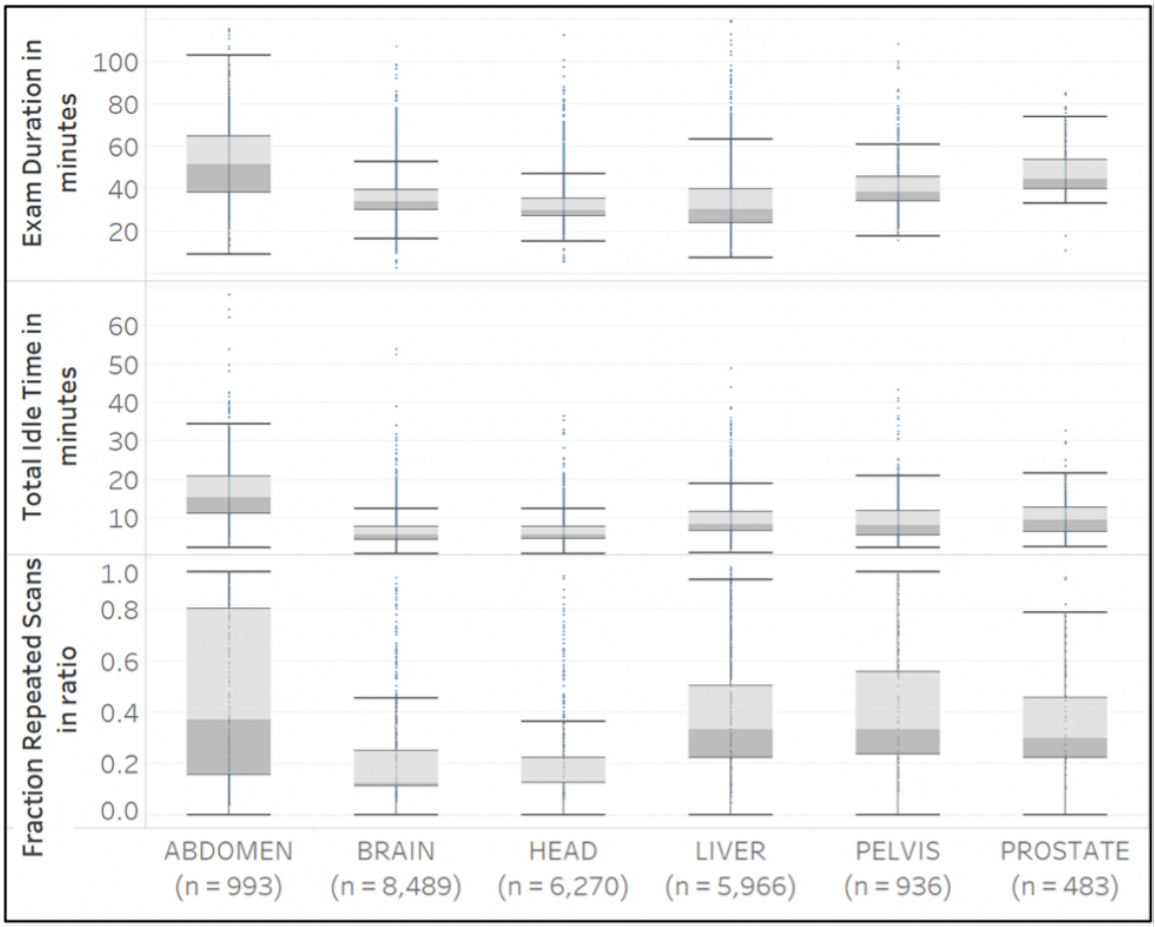


Figure 3.2: KPI by anatomy where 'n=' denotes the number of exams. Brain exams tend to have the smallest KPI variability while abdominal exams demonstrate larger variability.

### *3.2.2 KPI Site Comparison*

The data are consolidated to 5 health systems encompassing 24 MRI scanners. Table 3.2 shows the mean, standard deviation, and quantity of brain MRI exams by site. The smallest number of exams at any site is approximately 500. Mean exam duration and duration variability are statistically equivalent across the systems, irrespective of the number of exams. On average, total idle time is 7.4 minutes, or approximately 20% of exam duration, but with large variation (coefficient of variation ranging from 0.75 to 1.12). In other words, idle time ranges from 5% to 44% of exam duration across the systems. The fraction of repeated scans per exam and its variability also vary widely across systems.

Table 3.2: Comparison of KPIs across 5 systems (brain MRI exams). SD: standard deviation.

| Site | No. Exams | Exam Duration (min) | | Fraction Repeated | | Idle Time (min) | |
|---|---|---|---|---|---|---|---|
| | | Mean | SD | Mean | SD | Mean | SD |
| 1 | 616 | 37.15 | 20.29 | 0.20 | 0.21 | 8.01 | 6.22 |
| 2 | 4,501 | 37.90 | 13.15 | 0.27 | 0.31 | 8.04 | 8.84 |
| 3 | 1,177 | 39.57 | 17.86 | 0.21 | 0.28 | 7.12 | 6.31 |
| 4 | 1,497 | 36.78 | 17.22 | 0.17 | 0.25 | 6.77 | 6.03 |
| 5 | 3,380 | 39.09 | 15.70 | 0.14 | 0.34 | 7.63 | 7.19 |

Table 3.3 consolidates this information for all anatomies. System 3 has better KPI values overall; however, there is no statistically significant difference between the KPIs when comparing any two systems. System 1 serves as the analysis site for this work, and findings extend across all systems.

### *3.2.3 Exam Card Modification*

Exam duration is highly correlated with several variables. In particular, exam duration exhibits a strong positive correlation with total scanning time—defined as the sum of all scan times during the exam (including diagnostic, survey, calibration, and reference scans).

Table 3.3: Summary of average KPI values and their variance across 5 health systems.

| | Brain | | Head | | Liver | | Spine | | Abdomen | | Knee | | Heart | | Pelvis | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| System | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| **Exam Duration (minutes)** | | | | | | | | | | | | | | | | |
| 1 | 39.092 | 15.699 | 34.810 | 12.537 | 37.397 | 14.853 | 44.720 | 23.219 | 40.048 | 15.290 | 35.585 | 13.501 | 75.203 | 23.067 | 40.110 | 15.232 |
| 2 | 37.147 | 20.293 | 39.215 | 20.382 | 40.786 | 17.333 | 45.665 | 22.706 | 37.288 | 18.143 | 38.948 | 17.662 | 59.982 | 25.264 | 41.687 | 17.521 |
| 3 | 36.781 | 17.225 | 37.605 | 11.470 | 28.099 | 11.736 | 21.317 | 19.897 | 44.559 | 16.679 | 42.837 | 9.075 | 44.958 | 19.716 | 42.657 | 17.296 |
| 4 | 39.574 | 17.861 | 38.496 | 17.186 | 38.651 | 14.502 | 54.452 | 26.109 | 39.161 | 13.987 | 35.223 | 11.957 | | | 47.032 | 17.891 |
| 5 | 37.895 | 13.154 | 34.939 | 13.610 | 39.359 | 16.936 | 52.165 | 33.095 | 46.769 | 16.080 | 40.330 | 18.398 | 48.997 | 19.629 | 39.661 | 13.575 |
| **RANGE** | **2.79** | **7.14** | **4.40** | **8.91** | **12.69** | **5.60** | **33.14** | **13.20** | **9.48** | **4.16** | **7.61** | **9.32** | **30.24** | **5.63** | **7.37** | **4.32** |
| **Idle Time (minutes)** | | | | | | | | | | | | | | | | |
| 1 | 7.57 | 5.49 | 6.415 | 4.498 | 9.715 | 5.954 | 7.468 | 5.143 | 9.928 | 7.193 | 6.171 | 3.358 | 27.179 | 13.140 | 8.487 | 5.325 |
| 2 | 7.03 | 10.60 | 7.669 | 7.055 | 9.071 | 5.980 | 7.767 | 5.745 | 10.362 | 8.661 | 6.946 | 5.638 | 21.047 | 13.967 | 8.162 | 6.126 |
| 3 | 5.88 | 5.51 | 5.542 | 3.942 | 4.930 | 2.659 | 4.395 | 0.824 | 10.078 | 7.023 | 7.619 | 5.542 | 11.422 | 10.367 | 9.469 | 6.963 |
| 4 | 5.15 | 4.97 | 5.483 | 3.570 | 8.442 | 6.422 | 7.143 | 5.425 | 9.172 | 5.085 | 4.450 | 5.049 | | | 9.867 | 9.149 |
| 5 | 6.14 | 6.97 | 5.875 | 5.905 | 10.154 | 12.025 | 9.831 | 9.845 | 14.039 | 9.896 | 6.877 | 5.864 | 17.264 | 10.575 | 9.670 | 6.292 |
| **RANGE** | **2.43** | **5.63** | **2.19** | **3.48** | **5.22** | **9.37** | **5.44** | **9.02** | **4.87** | **4.81** | **3.17** | **2.51** | **15.76** | **3.60** | **1.70** | **3.82** |
| **Fraction Repeated** | | | | | | | | | | | | | | | | |
| 1 | 0.188 | 0.192 | 0.108 | 0.147 | 0.341 | 0.214 | 0.353 | 0.246 | 0.356 | 0.276 | 0.220 | 0.198 | 0.393 | 0.173 | 0.328 | 0.217 |
| 2 | 0.231 | 0.225 | 0.238 | 0.237 | 0.354 | 0.194 | 0.270 | 0.213 | 0.383 | 0.233 | 0.236 | 0.222 | 0.293 | 0.163 | 0.304 | 0.226 |
| 3 | 0.179 | 0.209 | 0.076 | 0.121 | 0.096 | 0.098 | 0.048 | 0.082 | 0.230 | 0.172 | 0.184 | 0.209 | 0.279 | 0.206 | 0.256 | 0.198 |
| 4 | 0.146 | 0.188 | 0.237 | 0.200 | 0.343 | 0.190 | 0.285 | 0.184 | 0.393 | 0.185 | 0.160 | 0.189 | | | 0.234 | 0.197 |
| 5 | 0.099 | 0.159 | 0.168 | 0.211 | 0.399 | 0.211 | 0.407 | 0.256 | 0.270 | 0.183 | 0.347 | 0.345 | 0.291 | 0.131 | 0.218 | 0.177 |
| **RANGE** | **0.13** | **0.07** | **0.16** | **0.12** | **0.30** | **0.12** | **0.36** | **0.17** | **0.16** | **0.10** | **0.19** | **0.16** | **0.11** | **0.08** | **0.11** | **0.05** |

Reducing the number of scans or sequences reduces total idle time, reduces the number of repeated scans, and ultimately reduces exam duration, thereby increasing operational efficiency. An exam card is a list of sequences that comprise a template exam and is usually stored on an MRI scanner. Exam cards are frequently modified by adding or subtracting sequences.

Figure 3.3 (top) shows that for a given protocol, there are redundant exam cards stored on a scanner that may contain unnecessary sequences. Figure 3.3 (middle) displays the most frequently occurring exams by anatomy: brain, liver, and head exams are the most prevalent at the analysis site. Figure 3.3 (bottom) merges the information from the above two subfigures and shows that, except for the head anatomy, there is a broad distribution of stored exam cards used in MRI exams, which can result in longer exam setup time and delayed starts.

A pilot study was conducted to examine the effect of reducing the number of sequences on exam duration for brain anatomy exams. Figure 3.4 shows a decrease in exam duration for a primary brain tumor protocol. Two exam cards were modified and combined in the middle of month 3 (x-axis) and used as the basic exam card for the next 2.5 months. Because the new exam card has fewer sequences, exam duration and idle time decrease from March to May. On average, the number of scans was reduced from 8.29 to 7.39 and exam duration by approximately 4 minutes ($\sim$12% of average exam duration). The fraction of repeated scans did not improve in the same period, indicating further room for improvement. This study demonstrates that exam duration may be reduced through active management of exam card composition.

### *3.3 Analysis of Patient Characteristics*

Supervised statistical learning is a framework for understanding data based on statistics and involves building a statistical model for predicting an output based on one or more inputs. In this section, Generalized Linear Models (GLMs) are used to analyze the association between patient characteristic variables and the KPIs (i.e., response variables). By developing an empirical model, understanding what variables affect the KPIs may be inferred, and their effect on the KPIs may be estimated.

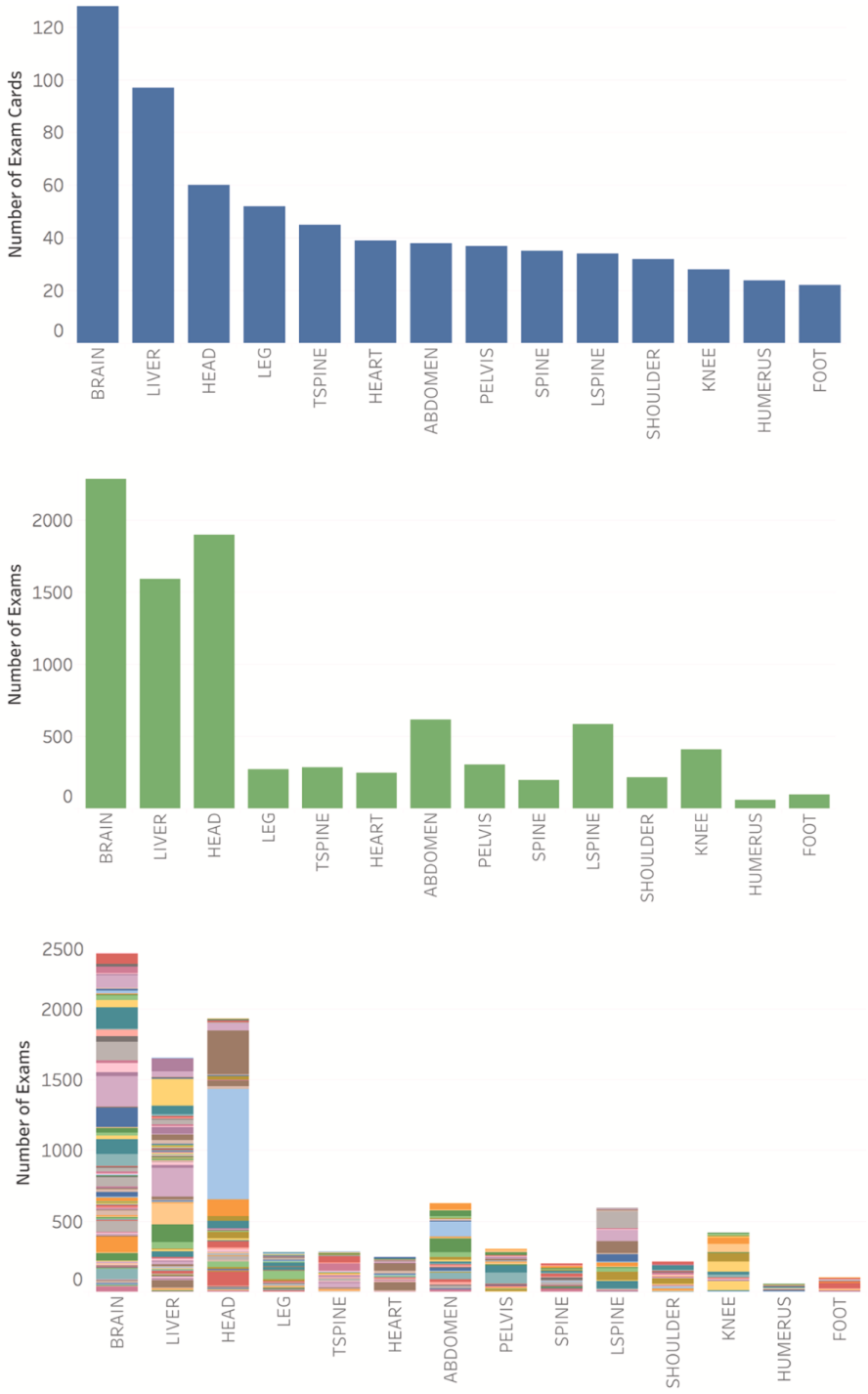


Figure 3.3: Distribution of exam cards and the number of exams by anatomy. Top: Distribution of 1097 exam cards stored on scanners at the analysis site by body part. Middle: Number of exams by anatomy. Bottom: This plot shows the number of different exam cards from (a) that are used in the exams in (b). Each color represents one exam card.


Figure 3.4: Number of scans and average KPIs comparison per month. The months in brown occur after the exam card change and show a decreasing trend in exam duration.

The GLM is an extension of the standard linear model that uses different distributions as a link function [63, 64]. GLMs are used here for several reasons: the interpretability of predictor variables, the ability to model both categorical and continuous predictors and response variables, and the non-normality and non-constant variability in the log file data. A model using 5-fold cross-validation was estimated for a single MRI site.

Two GLM models were created for each KPI[5]. The first model uses patient sex (binary: 0 = female, 1 = male), patient age (continuous), and patient weight (continuous). The second model uses the same variables with age and weight converted to nominal form: three age levels—child (<11 years), adult, and elderly (>69 years)—and two weight levels—overweight (>90 kg for women, >110 kg for men) and regular weight. These models were developed for the brain (8,489 records) and liver (5,966 records) anatomies, as they are the most frequently occurring in the data.

[5]Exam duration: Gaussian with identity link since exam duration is approximately normally distributed after filtering; Total idle time: Gamma with log link since idle time is strictly positive and right-skewed; Fraction repeated: Beta regression since the fraction is bounded between 0 and 1.

Table 3.4 shows the important variables for both models for each anatomy. Although the highly variable log-file data results in models with low $R^2$ values, the models have significant predictor variables that provide insights into what affects the KPIs.

For the *brain* anatomy:

- Young patients (<11 years) increase exam duration by approximately 15 minutes (∼878 s) and the fraction of repeated exams by 74% relative to the baseline adult female of normal weight.

- Young patients increase idle time by 72%, and overweight patients increase idle time by 11%, relative to the baseline.

For the *liver* anatomy:

- Overweight patients increase all three KPIs: exam duration by 6.5 minutes, idle time by 14% relative to the baseline, and fraction of repeated exams by 13% relative to the baseline.

Although potentially more complex predictive models would achieve higher prediction accuracy, these findings suggest that patient attributes can be used to predict exam duration, which may help improve exam scheduling by optimizing time slot allocation to increase throughput and see more patients per day.

### *3.4 Conclusion*

This chapter presents an opportunity for using MRI log files, which have traditionally been underutilized in radiology. Using a lean methodology, key performance metrics are identified and used to demonstrate that a quantitative analysis of MRI log files can identify problems in departmental processes and opportunities for improvement in radiology operations. The results presented here generalize well to different imaging sites.

Exam cards containing fewer sequences were developed and result in shorter exam durations. Exam cards with high sequence repetition by anatomy were analyzed and revisions

Table 3.4: Prediction of KPIs using generalized linear models. Exam duration: Gaussian GLM with identity link. Total idle time: Gamma GLM with log link. Fraction repeated: Beta regression.

| | Exam Duration | | | | Total Idle Time | | | | Fraction Repeated Scans | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Predictors | Estimate | Std Error | Pr(> \|t\|) | $R^2$ | Estimate | Std Error | Pr(> \|t\|) | $R^2$ | Estimate | Std Error | Pr(> \|t\|) | $R^2$ |
| **Brain** | | | | | | | | | | | | |
| *Effect of the number of scans* | | | | | | | | | | | | |
| (Intercept) | 545.26 | 15.46 | 0.00 | 0.57 | 117.56 | 8.05 | 0.00 | 0.11 | 0.09 | 0.01 | 0.00 | 0.032 |
| # of Scans | 175.39 | 1.66 | 0.00 | | 27.87 | 0.86 | 0.00 | | 0.01 | 0.00 | 0.00 | |
| *Effect of patient characteristics (continuous variable)* | | | | | | | | | | | | |
| (Intercept) | 2172 | 31.25 | 0.00 | 0.006 | 5.94 | 0.03 | 0.00 | 0.004 | -1.35 | 0.05 | 0.00 | 0.006 |
| Patient SexM | 50.17 | 13.32 | 0.00 | | -0.02 | 0.01 | 0.17 | | -0.04 | 0.02 | 0.07 | |
| Patient Weight | 0.52 | 0.29 | 0.08 | | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.48 | |
| Patient Age | -2.01 | 0.38 | 0.00 | | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.08 | |
| *Effect of patient characteristics (categorical variable)* | | | | | | | | | | | | |
| (Intercept) | 2085.6 | 9.34 | 0.00 | 0.014 | 5.89 | 0.01 | 0.00 | 0.011 | -1.43 | 0.02 | 0.00 | 0.003 |
| Patient SexM | 57.74 | 12.67 | 0.00 | | 0.00 | 0.01 | 0.94 | | -0.04 | 0.02 | 0.04 | |
| Overweight | 71.21 | 17.19 | 0.00 | | 0.1 | 0.02 | 0.00 | | 0.01 | 0.03 | 0.69 | |
| Age Group Child | 878.36 | 95.71 | 0.00 | | 0.54 | 0.09 | 0.00 | | 0.75 | 0.15 | 0.00 | |
| Age Group Elderly | -8.75 | 15.66 | 0.58 | | -0.01 | 0.02 | 0.59 | | -0.06 | 0.03 | 0.03 | |
| **Liver** | | | | | | | | | | | | |
| *Effect of the number of scans* | | | | | | | | | | | | |
| (Intercept) | 422.75 | 25.66 | 0.00 | 0.42 | 207.05 | 11.02 | 0.00 | 0.17 | 0.057 | 0.01 | 0.00 | 0.259 |
| # of Scans | 148.58 | 2.28 | 0.00 | | 34.23 | 0.98 | 0.00 | | 0.029 | 0.00 | 0.00 | |
| *Effect of patient characteristics (continuous variable)* | | | | | | | | | | | | |
| (Intercept) | 2132 | 59.79 | 0.00 | 0.084 | 6.34 | 0.04 | 0.00 | 0.018 | -0.94 | 0.07 | 0.00 | 0.008 |
| Patient SexM | -86.57 | 22.51 | 0.00 | | -0.09 | 0.01 | 0.00 | | -0.14 | 0.03 | 0.00 | |
| Patient Weight | 8.38 | 0.52 | 0.00 | | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.00 | |
| Patient Age | -12.73 | 0.76 | 0.00 | | 0.00 | 0.00 | 0.00 | | 0.00 | 0.00 | 0.02 | |
| *Effect of patient characteristics (categorical variable)* | | | | | | | | | | | | |
| (Intercept) | 1951 | 18.30 | 0.00 | 0.026 | 6.36 | 0.01 | 0.00 | 0.011 | -0.59 | 0.02 | 0.00 | 0.006 |
| Patient SexM | 25.69 | 21.69 | 0.24 | | -0.05 | 0.01 | 0.00 | | -0.06 | 0.03 | 0.02 | |
| Overweight | 390.85 | 31.27 | 0.00 | | 0.13 | 0.02 | 0.00 | | 0.19 | 0.04 | 0.00 | |
| Age Group Elderly | -46.43 | 27.96 | 0.1 | | 0.01 | 0.02 | 0.77 | | 0.14 | 0.03 | 0.00 | |

were identified. Simplification of the brain exam cards reduced exam duration by approximately seven percent in the first three months after revision. The reduction in exam duration can be estimated using a GLM with number of scans as the sole predictor variable: Table 3.4 shows that each additional scan increases exam duration by approximately 3 minutes. In another example not detailed here, a prostate sequence with a high fraction of repetition was identified and later replaced with a more robust sequence.

This work may also stimulate the redesign of the exam scheduling process. Understanding the variables that affect exam duration and the other KPIs may better define scheduling slots for different anatomies after accounting for exam and patient characteristics such as contrast application, gender, weight, and age. Figure 3.5 shows that the 75th percentile of historical exam duration generally lies within the standard scheduling slot. However, as changes to exams are made that lengthen the duration, the schedule should be revisited (Figure 3.6).

This study demonstrates that historically underutilized MRI log files contain rich exam information that may be used to facilitate operational improvements. The application of lean analysis, KPI development, exam card modification, and GLM-based patient characteristic modeling provides a replicable methodology for future researchers wishing to extract actionable insights from MRI log files at their own institutions.

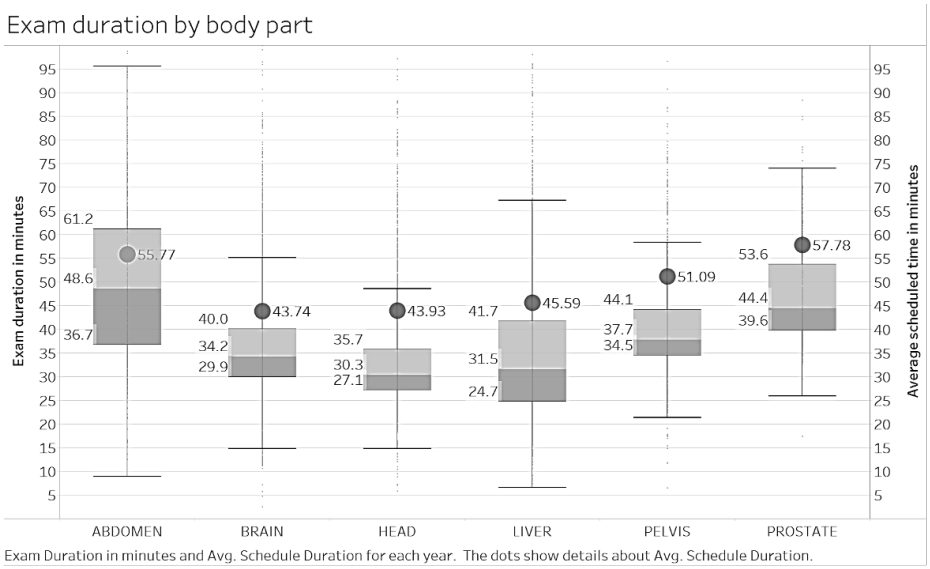


Figure 3.5: Comparison between exam duration and schedule duration. The average scheduled time (denoted by the dot) is longer than the average duration of the MRI.

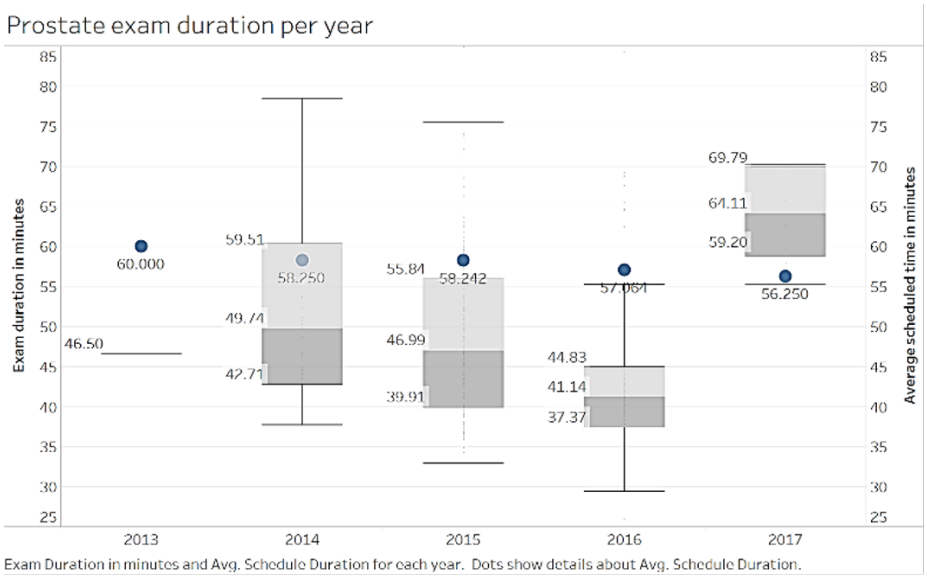


Figure 3.6: Comparison between exam duration and schedule duration (denoted by the dot) for prostate exams. Although the sequence changes lengthened exam durations, the schedule slots were not updated.

Chapter 4

# EHR DATA CHARACTERIZATION AND CLINICAL TEXT ANALYSIS FOR MRI DURATION PREDICTION

## *4.1 Introduction*

Data-informed workflow improvement in complex clinical environments depends critically on the accuracy, completeness, and interpretability of the underlying data. In high-throughput radiology departments where decisions such as exam scheduling, resource allocation, and staffing are made continuously under uncertainty, the ability to extract reliable insights from routinely-collected data is essential, as it provides the basis and rationale for improvement activities. However, in most cases, clinical data is collected without premeditated analytical purposes. Instead, it is often generated as a byproduct of operational and clinical workflows, across multiple systems with different data models, collection methodologies, and levels of granularity.

This chapter presents a comprehensive exploratory data analysis (EDA) of MRI exam data from two complementary sources: modality log files (MLFs) generated by MRI scanners and electronic health records (EHRs) maintained in the clinical information system. MLFs provide precise, machine-generated timestamps and exam-level operational details, whereas EHRs provide patient demographic and clinical information that contextualize each exam. The two sources together offer a richer basis for data-driven scheduling improvement than either alone.

The analysis serves three purposes. First, it characterizes the distributional properties of key variables, particularly exam duration, which exhibits substantial skewness and procedure-level variability with direct implications for modeling and statistical analysis. Second, it examines relationships between patient and exam-level characteristics and duration, providing empirical grounding for feature selection decisions in subsequent chapters. Third, it investigates the predictive content of unstructured clinical text through a BERT-based

regression experiment, assessing whether result narratives carry predictive signal beyond structured variables alone.

Several data quality challenges are identified through this analysis, including missingness, distributional irregularities, and discrepancies attributable to differences in how each system records exam duration. These observations motivate the concordance-based data quality assessment and cleaning methodology presented in the following chapter. All analyses in this chapter were performed on data extracted across two campuses of the University of Washington Medical Center, a large academic medical center located in the Pacific Northwest, USA.

## *4.2 Dataset Overview*

The following subsections describe the study setting and the two data sources from which these data were obtained.

### *4.2.1 Study Setting*

Data for this study were obtained from two campuses of the University of Washington Medical Center (UWMC): UWMC-Montlake, a quaternary care academic medical center equipped with four MRI scanners, and UWMC-Northwest, a community hospital equipped with two MRI scanners. The two sites differ in case mix, patient acuity, and operational workflows, thereby introducing heterogeneity that strengthens the generalizability of the findings.

### *4.2.2 Data Sources*

*Electronic Health Records*

A total of 56,578 EHRs were extracted from Epic[1], the clinical information system used at both UWMC campuses, covering the period from February 2022 through February 2024. In contrast to MLFs, EHRs are populated in part through manual entry by clinical staff, introducing potential for data entry variability and inaccuracy. However, EHRs provide

[1] `https://www.epic.com/`

patient-specific information that is absent from MLFs, including demographic characteristics, clinical context, and procedure-level details. The key EHR variables used in this study are summarized in Table 4.1.

Table 4.1: Key variables in the EHR data

| **Category** | **Variables** |
|---|---|
| Patient demographics | Age, gender, race, height, weight |
| Exam information | Exam duration, procedure code, radiology subspecialty group, patient status, order priority |
| Clinical notes | Result narrative (clinical indication and findings) |

*Modality Log Files*

MLFs are records automatically generated by MRI scanners that capture events, actions, and operational details during imaging acquisition. In this study, MLFs are accessed through Philips OneSpace[2], newly deployed at the time of study, an integrated operational dashboard that extracts and consolidates scanner-generated data across the enterprise. While the underlying log data are generated automatically by scanner hardware, Philips OneSpace introduces an additional layer of extraction and processing that, as a newly deployed system, may introduce artifacts or inconsistencies not present in the raw scanner output. Consequently, MLF-derived duration measures are not treated as unambiguous ground truth in this study; rather, their reliability is assessed through concordance analysis with EHR-recorded durations, as described in the following chapter.

MLF data spanning January 2022 through July 2025 were extracted from both campuses, resulting in a total of 90,747 records. Each record corresponds to a single exam and includes the variables summarized in Table 4.2. Notably, the MLFs decompose total exam duration into three constituent components: preparation time, scan series duration, and finalization time, enabling a more detailed characterization of the exam workflow than total duration

[2] `https://www.usa.philips.com/healthcare/service/onespace-insights`

alone. Despite their automated origin, MLFs lack patient-specific information due to patient privacy compliance requirements and the separation of medical information systems.

Table 4.2: Variables in the MLF data

| **Variable** | **Description** |
|---|---|
| Hospital_Asset_ID | Campus identifier (UWMC-Montlake or UWMC-Northwest) |
| EquipmentNumber | Unique scanner identifier |
| StudyID | Exam-level identifier |
| StudyType | Anatomical focus of the exam |
| StudyStartDateTime | Exam start timestamp |
| StudyEndDateTime | Exam end timestamp |
| ProtocolName | Imaging protocol used |
| StudyDuration_Seconds | Total exam duration |
| PreparationDuration_Seconds | Time elapsed during exam preparation |
| ScanSeriesDuration_Seconds | Time elapsed during active scanning |
| FinalizationDuration_Seconds | Time elapsed during exam finalization |

### *4.2.3 Study Scope*

The two data sources are complementary by design—MLFs provide objective, machine-generated operational data with fine temporal granularity, whereas EHRs provide clinical and patient context essential for understanding the factors that drive variability in exam duration. Neither source alone is sufficient for a comprehensive analysis: MLFs lack patient-level information, and EHR duration data is subject to manual-entry inconsistencies. Integrating the two sources, therefore, requires careful attention to data quality, a challenge addressed in detail in the following chapter.

In this study, EHR data were obtained first and served as the primary basis for the exploratory analysis presented in this chapter. MLF data were obtained subsequently and cover a broader time window (January 2022–July 2025); however, the data integration and

predictive modeling presented in the following chapter are restricted to the overlapping period of February 2022 through February 2024, during which both data sources are available. This ensures that the merged dataset used for concordance assessment and duration prediction reflects a consistent observation window across both sources. 1

## *4.3 Data Preprocessing*

### *4.3.1 Exclusion Criteria*

As discussed in previous sections, both data sources are likely to include records that reflect data entry errors, workflow anomalies, or insufficient sample sizes required for meaningful analysis. Therefore, these records were excluded in our analysis.

For both datasets, exams with durations of 0 minutes or greater than 300 minutes were removed, as such values are likely attributable to data entry errors, workflow nuances (e.g., scout scans and unclosed studies), or implausible exam lengths. In the MLF data, study types with fewer than 20 occurrences were additionally excluded to ensure sufficient sample size and generalizability. In the EHR data, procedure codes with fewer than 400 records over the study period were excluded from downstream modeling analyses to ensure stable parameter estimation.

As the EHR data were obtained first and contain richer patient and clinical information, the exploratory analyses presented in the remainder of this chapter focus primarily on the EHR dataset. The MLF data are incorporated in the next chapter, where the two sources are integrated for concordance assessment and predictive modeling.

### *4.3.2 Linked Exam Identification and Removal*

An additional data quality issue specific to the EHR data is the presence of linked exams, which are instances where two or more exams with different procedure codes for the same patient share identical exam start times and often identical or near-identical end times. This occurs when technologists initiate and close multiple exams for the same patient simultaneously, recording identical or extremely close ($\pm 2$ minutes) exam durations across all linked procedures rather than logging each exam independently. While this practice helps

technologists streamline administrative activities and focus on imaging acquisition, linked exams do not reflect independent duration measurements and would introduce bias into any analysis that treats each record as a distinct exam. All records within an identified linked exam group were excluded from the dataset, including the record with the longest duration, as it likely reflects the combined duration of multiple procedures rather than a single exam. Linked exams were identified by matching on patient identifier and exam start time.

Table 4.3 summarizes the filtering steps and resulting record counts for the EHR dataset. Corresponding results for the MLF dataset are reported in Chapter 5, where the two sources are integrated for concordance assessment and predictive modeling.

Table 4.3: EHR record counts at each filtering step

| **Filtering Step** | **Records Remaining** |
|---|---|
| Raw extraction | 56,578 |
| Remove duration = 0 or > 300 min | 53,475 |
| Remove linked exams | 43,946 |
| Remove procedure codes with < 400 records | 33,566 |

### *4.3.3 Missing Data*

Missing data in the EHR dataset were limited to three variables. Figure 6.1 presents the missingness matrix for the EHR dataset, sorted by exam start time, where white streaks indicate missing values and dark cells indicate observed values.

Height and Weight each had a missingness rate of 17.05%, with both variables missing for the same records, suggesting that such data were not recorded for a subset of patients. SupportStaff exhibited a substantially higher missingness rate of 67.84%, resulting from the fact that support staff are not required for most exams. Given this high rate of missingness, SupportStaff was excluded from predictive modeling analyses. All remaining variables were fully observed, with no missing values. Missing Height and Weight values were imputed using the median prior to modeling.

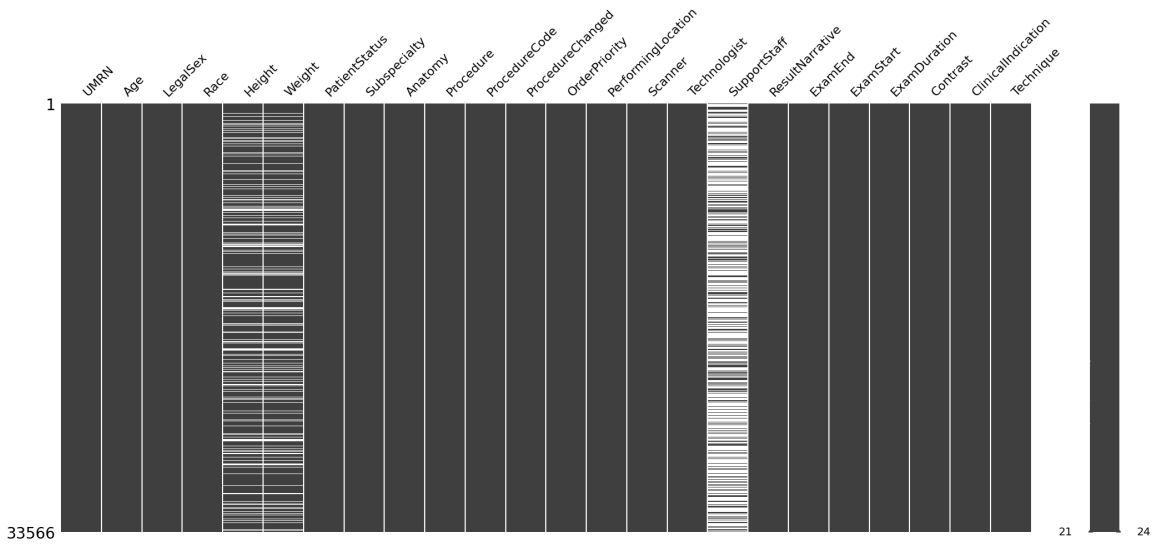


Figure 4.1: Missingness matrix of the EHR dataset sorted by exam start time.

### *4.3.4 Derived Variables*

Two variables, namely contrast use and time of day, were derived from existing fields to capture factors potentially associated with exam duration variability.

**Contrast Use**: Contrast use was derived primarily from the Procedure field, where procedure names were parsed to extract contrast-related nomenclature, resulting in four categories: without contrast (WO), with and without contrast (WW), with contrast (W), and Other. For records where the Procedure field did not yield a clear contrast classification (i.e., categorized as "Other"), the ResultNarrative field was searched for contrast-related keywords to assign a category. The classification rules applied to the result narrative were as follows:

- **WO**: narrative contains "without contrast" but no mention of "with contrast", "post-contrast", or "post contrast"

- **WW**: narrative contains "without contrast" AND at least one of "with contrast", "post-contrast", or "post contrast"

- **W**: narrative contains "with contrast", "post-contrast", or "post contrast", but no mention of "without contrast"

- **ARTH**: narrative contains "arthrogram", indicating intra-articular contrast administration without intravenous contrast

Records for which neither the Procedure field nor the ResultNarrative field yielded a contrast classification were retained as "Other". This two-step approach ensured that contrast classification was as complete as possible while leveraging the more structured Procedure field as the primary source of information.

**Time of Day**: A time of day variable was derived from the ExamStart timestamp. Exams were classified into three categories based on the hour of the exam start time: AM (6:00–12:00), PM (12:00–18:00), and NIGHT (before 6:00 or after 18:00). This variable was included to capture potential workflow-related variation in exam duration across different periods of the scheduling day.

## *4.4 Exam Duration Analysis*

Exam duration is the primary outcome variable of interest in this study; its distributional properties and sources of variability are examined in the following subsections.

### *4.4.1 Overall Duration Distribution*

Figure 4.2 shows the distribution of MRI exam durations across the entire EHR dataset after preprocessing. Exam durations ranged from 1 to 295 minutes, with a mean of 40.4 minutes (SD = 24.5) and a median of 34 minutes. The distribution shows strong positive skewness (skewness = 2.23) and high kurtosis (kurtosis = 10.10), suggesting a leptokurtic distribution with a heavy right tail. As seen in Figure 4.2, the majority of exams were completed within 50 minutes, while a smaller proportion of exams extended beyond 100 minutes, which were likely due to complex multi-sequence protocols or procedures requiring additional preparation time. The non-normality of exam duration data has direct implications for downstream statistical analysis, as methods assuming normality are not appropriate without transformation.

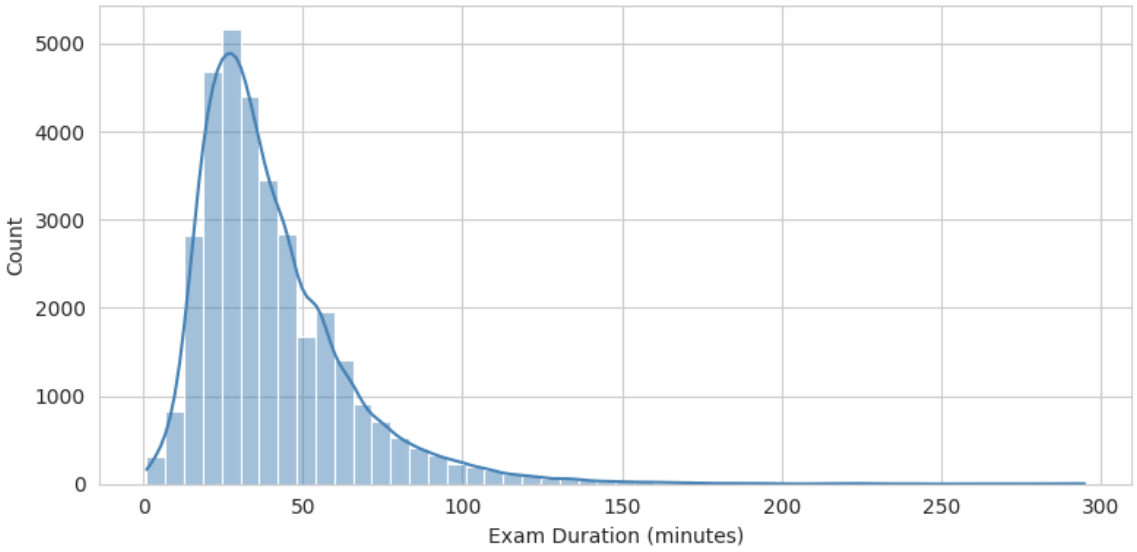


Figure 4.2: Distribution of MRI exam durations across the EHR dataset (n = 33,566). The distribution exhibits strong positive skewness and a heavy right tail.

### *4.4.2 Duration Variability by Procedure Code*

Figure 4.3 presents the distribution of exam durations for the 21 procedure codes, ordered by median duration and color-coded by subspecialty. Median durations ranged from 22 minutes (MBRASTROKE) to 77 minutes (MMSBRAIN), highlighting the substantial variation in protocol complexity across procedure types. Within-procedure variability also differed considerably: MMSBRAIN exhibited the highest variability (SD = 39.4 minutes), followed by MSPCTLWW (SD = 30.5 minutes), while knee procedures (MKNEWOL, MKNEWOR) showed comparatively tighter distributions (SD $\approx$ 14 minutes). Outliers were present across nearly all procedure codes, with MBRAWW exhibiting a particularly large number due to its high record count (n = 8,524). These findings suggest that procedure code alone is insufficient to fully characterize exam duration variability, and that additional patient and exam-level factors may contribute meaningfully to prediction.

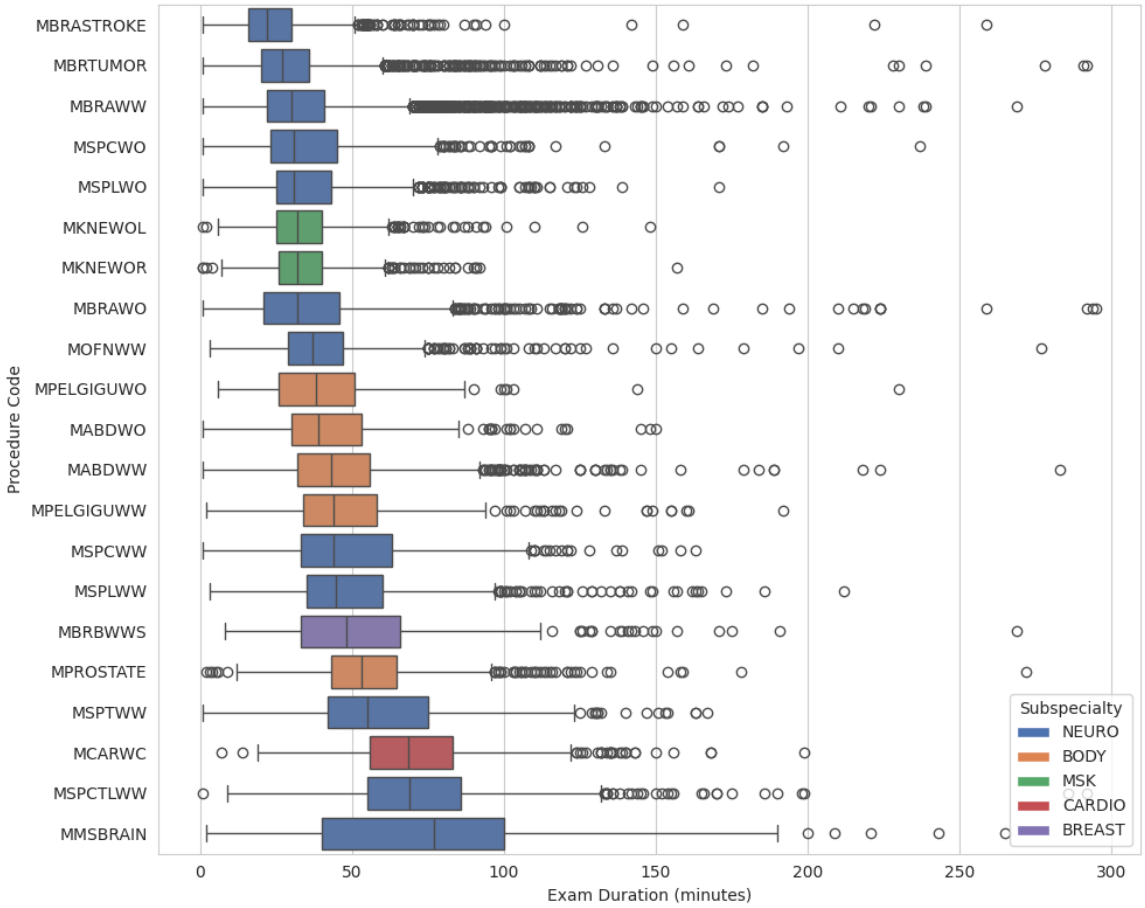


Figure 4.3: Distribution of MRI exam durations by procedure code (n = 33,566), ordered by median duration and color-coded by subspecialty.

### *4.4.3 Duration by Subspecialty and Site*

Figure 4.4 presents exam duration distributions aggregated by subspecialty group. CARDIO exhibited the highest median duration (median = 68.5 minutes), consistent with the complexity of cardiac MRI protocols, while NEURO and MSK showed the lowest median durations (median = 30 and 32 minutes, respectively). NEURO accounted for the largest proportion of exams (n = 24,660; 73.5% of the dataset), and consequently exhibited the greatest number of outliers. BODY and BREAST showed intermediate durations with moderate variability.

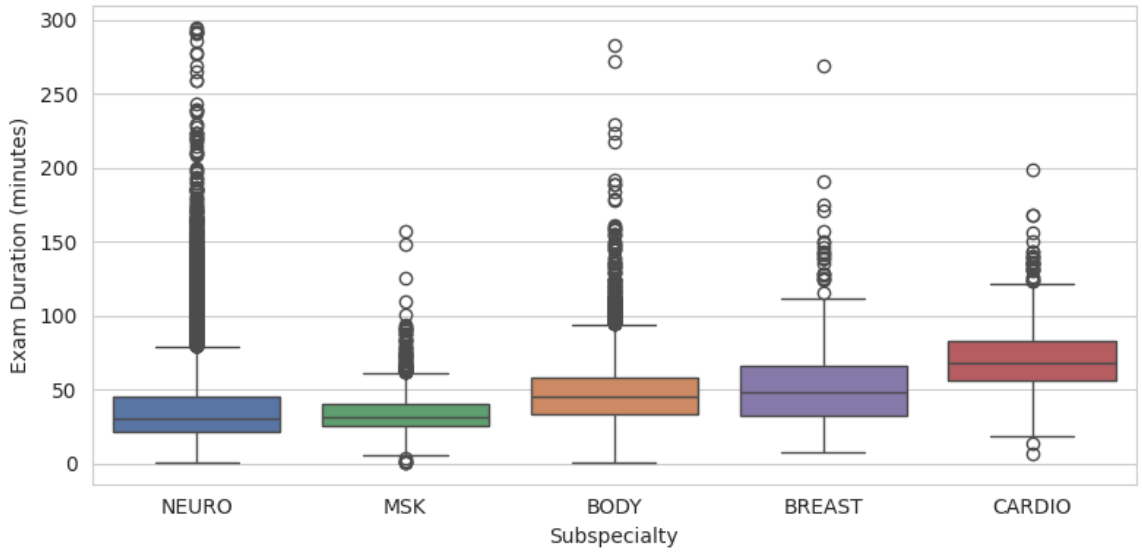


Figure 4.4: Distribution of MRI exam durations by subspecialty group, ordered by median duration.

Regarding site differences, UWMC-Montlake accounted for 65.9% of records (n = 22,110) and UWMC-Northwest for 34.1% (n = 11,456), consistent with the difference in scanner capacity between the two campuses. Differences in case mix between sites are expected given their distinct clinical roles: UWMC-Montlake as a quaternary care center and UWMC-Northwest as a community hospital. This is further explored in the bivariate analysis presented in Section 4.6.

## *4.5 Patient and Exam Characteristics*

The following subsections present the patient demographic and exam-level characteristics of the EHR dataset. A summary of the characteristics is shown in Table 4.4 and Figure 4.5.

### *4.5.1 Patient Demographics*

The study population had a mean age of 55.6 years (SD = 17.7), with a median of 58 years (IQR: 41–70), showing the predominantly adult patient population at both campuses. Ages ranged from 0 to 103 years, with the minimum value likely representing a neonate or a

data entry anomaly. The sex distribution was approximately balanced, with 52.6% female and 46.2% male patients; 1.2% of records had unknown sex. With respect to race, White patients comprised the majority of the dataset (75.6%), followed by Other (10.0%), Asian (9.1%), and Black or African-American (5.3%).

### *4.5.2 Exam-Level Variables*

The majority of exams were outpatient (76.2%), with inpatient and emergency exams comprising 15.6% and 8.3% of records, respectively. Most exams were classified as routine priority (76.4%), with STAT and urgent orders accounting for 16.2% and 7.4% of the records, respectively. The parallel distributions of patient status and order priority suggest a strong association between the two variables—most routine orders correspond to outpatient visits, and most STAT orders to inpatient or emergency settings.

Regarding contrast use, the majority of exams included both pre- and post-contrast imaging (WW: 60.8%), followed by exams without contrast (WO: 32.7%). Exams with contrast only (W) and arthrogram procedures (ARTH) were rare, comprising 2.9% and less than 0.1% of records in the preprocessed data, respectively. Records for which contrast classification could not be determined from either the Procedure field or the ResultNarrative accounted for 3.7% of the dataset and were retained as "Other".

Regarding time of day, approximately half of all exams were scheduled in the AM period (49.4%), with PM exams accounting for 37.4% and night exams for 13.2%, which reflects the natural variation in staffing levels across different periods of the day.

## *4.6 Bivariate Analysis*

Figure 4.6 presents the relationship between exam duration and key patient and exam-level variables.

**Patient Status and Order Priority** Emergency exams had the shortest median duration (26.5 minutes), followed by inpatient (33.0 minutes) and outpatient (36.0 minutes). A similar pattern was also observed for order priority, where STAT orders had the shortest median duration (29.5 minutes), compared to urgent (34.0 minutes) and routine orders

Table 4.4: Summary of patient and exam characteristics (n = 33,566)

| **Variable** | **Category** | **n** | **%** |
|---|---|---|---|
| Age (years) | Mean (SD) | 55.6 (17.7) | |
| | Median [IQR] | 58 [41–70] | |
| | Range | 0–103 | |
| Legal Sex | Female | 17,645 | 52.6 |
| | Male | 15,509 | 46.2 |
| | Unknown | 412 | 1.2 |
| Race | White | 25,385 | 75.6 |
| | Other | 3,345 | 10.0 |
| | Asian | 3,053 | 9.1 |
| | Black or African-American | 1,783 | 5.3 |
| Patient Status | Outpatient | 25,560 | 76.2 |
| | Inpatient | 5,222 | 15.6 |
| | Emergency | 2,784 | 8.3 |
| Order Priority | Routine | 25,642 | 76.4 |
| | STAT | 5,434 | 16.2 |
| | Urgent | 2,490 | 7.4 |
| Contrast Use | WW | 20,411 | 60.8 |
| | WO | 10,959 | 32.7 |
| | Other | 1,236 | 3.7 |
| | W | 959 | 2.9 |
| | ARTH | 1 | <0.1 |
| Time of Day | AM (06:00–12:00) | 16,573 | 49.4 |
| | PM (12:00–18:00) | 12,555 | 37.4 |
| | Night (<06:00 or >18:00) | 4,438 | 13.2 |

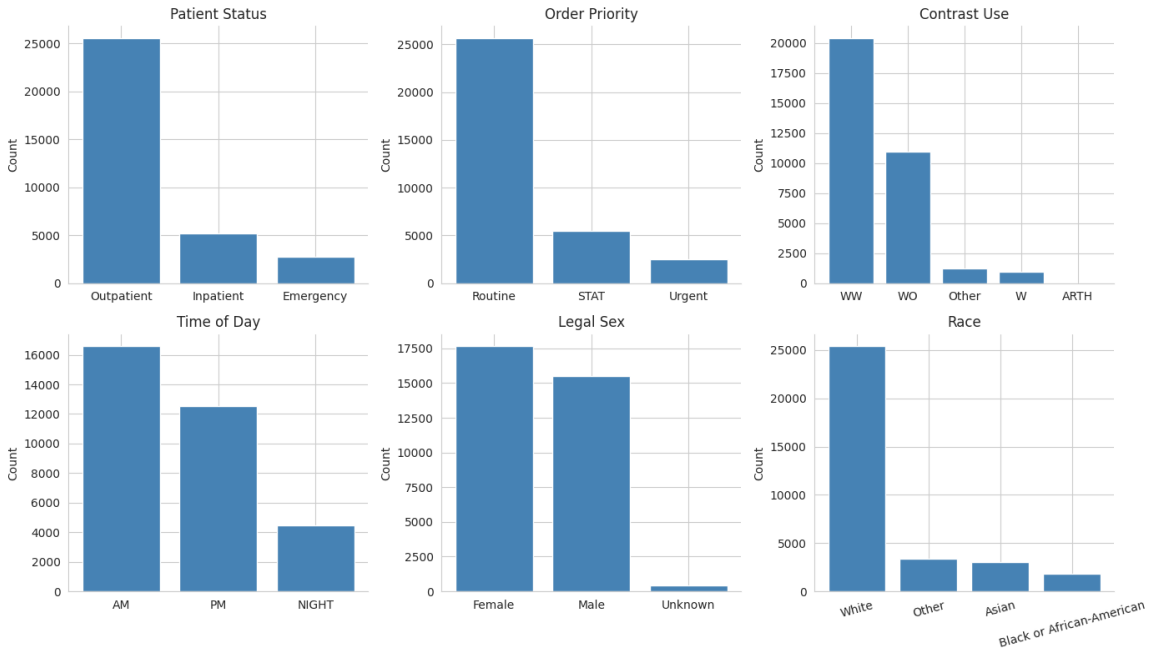


Figure 4.5: Distribution of patient and exam characteristics across the EHR dataset (n = 33,566).

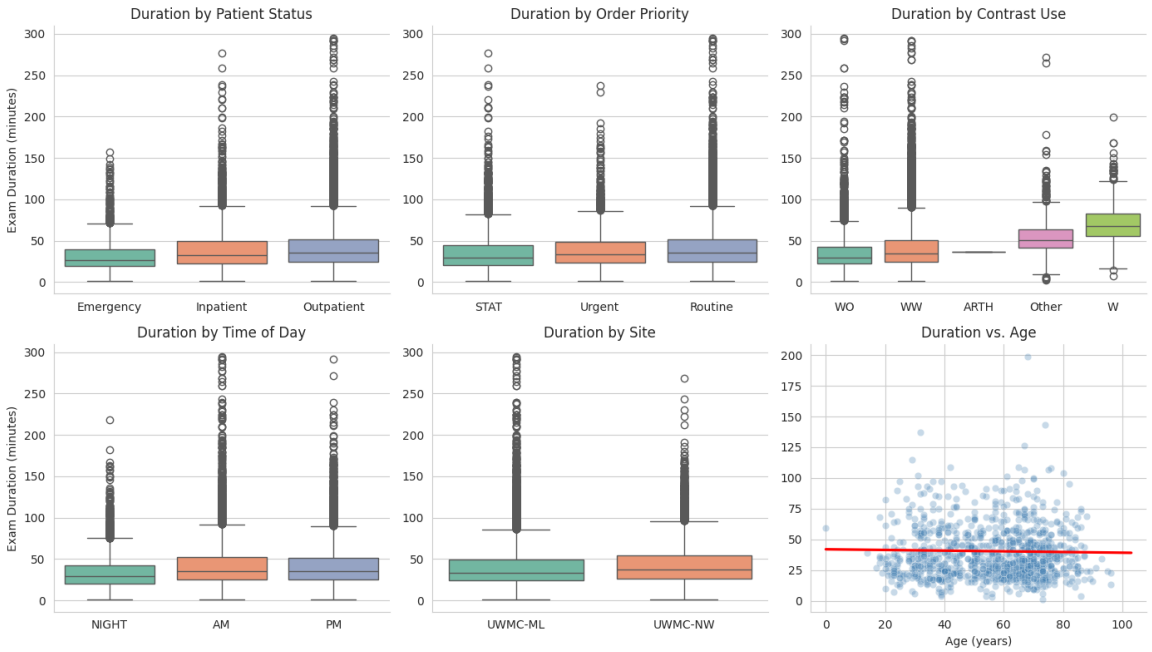


Figure 4.6: Exam duration by patient and exam characteristics.

(36.0 minutes). These findings are consistent with the focused nature of urgent and emergency imaging, where exams are typically limited to specific clinical questions rather than comprehensive multi-sequence protocols.

**Contrast Use** Contrast use showed a strong association with exam duration. Exams with contrast only (W) had the highest median duration (68.0 minutes), followed by Other (51.0 minutes), WW (35.0 minutes), ARTH (n=1, excluded from comparison), and WO (30.0 minutes). The substantially longer duration of W exams likely reflects the additional time required for intravenous contrast administration and post-contrast imaging sequences. These findings suggest that contrast use is an important predictor of exam duration and should be included in predictive modeling.

**Time of Day** Night exams had the shortest median duration (29.0 minutes) compared to AM and PM exams (both 35.0 minutes). One possible explanation for this is that during overnight hours, the case mix shifts toward emergency and urgent exams, which tend to be shorter and more focused than routine outpatient exams.

**Site** UWMC-Northwest exhibited a slightly higher median duration (37.0 minutes) compared to UWMC-Montlake (33.0 minutes). While the difference is modest, it may reflect differences in case mix and technologist expertise level between the quaternary care and community hospital settings, with UWMC-Northwest serving a higher proportion of procedures with longer median durations.

**Age** Age showed no meaningful association with exam duration ($r = -0.02$), suggesting that patient age is not a useful predictor of MRI exam duration in this dataset.

### *4.7 Clinical Notes Analysis*

Beyond structured variables, the EHR dataset contains unstructured clinical text in the form of radiology result narratives. These reports are generated by radiologists following each exam and typically consist of five sections: Examination, Clinical Indication, Technique, Comparison, and Findings. While most sections are produced after the exam is

completed, the Clinical Indication section is derived from the referring clinician's order and is therefore available prior to the exam. This distinction is important for predictive modeling: only information available at scheduling time can be used without introducing data leakage. Accordingly, the predictive analysis presented in Section 4.7.2 uses Clinical Indication exclusively, as it captures the clinical context and complexity of the requested exam without introducing data leakage from post-exam information.

### *4.7.1 Characterization of Result Narratives*

An example of a typical result narrative is presented below.

> **EXAMINATION:** MRI BRAIN W/O CONTRAST
>
> **CLINICAL INDICATION:** Significant cognitive decline in the past 6 months, concern for dementia, metastatic prostate cancer
>
> **TECHNIQUE:** MRI Head without contrast: Memory E (B 1t). Non-contrast: Sagittal T1. Axial T1, T2, FLAIR, DWI, T2*, pcASL. Coronal 3D T1. Volumetric hippocampal and ventricular volume calculations were performed on an independent workstation.
>
> **COMPARISON:** MR brain with and without contrast [*prior date*]. CTA head and neck [*prior date*].
>
> **FINDINGS:** MASS EFFECT & VENTRICLES: No significant midline shift. There is parenchymal volume loss with commensurate expansion of the ventricles and extra-axial/subarachnoid spaces.*[remaining findings omitted for brevity]*

The Clinical Indication in this example—"significant cognitive decline in the past 6 months, concern for dementia, metastatic prostate cancer"—illustrates the kind of clinical complexity cues that may not be fully captured by the procedure code alone. A patient presenting with multiple comorbidities and concern for dementia may require a more complex protocol than a straightforward brain MRI, potentially resulting in a longer exam duration.

### *4.7.2 BERT-Based Duration Prediction*

To assess whether unstructured clinical text contains predictive information for exam duration, a regression model was developed using the Clinical Indication field in the EHR data as the sole input. As noted above, Clinical Indication is available prior to the exam, making it a legitimate input for exam duration prediction without introducing data leakage from post-exam information.

#### *Clinical Indication Embedding*

Clinical Indication text was encoded using BioClinicalBERT [65], a domain-specific BERT model [66], which was pre-trained on clinical notes from MIMIC-III. Tokenization was performed using the corresponding tokenizer, with sequences truncated or padded to a maximum length of 512 tokens, as imposed by the BioClinicalBERT architecture. The maximum observed token length in the dataset was 828 tokens; truncation to 512 tokens may have resulted in the loss of some clinical information for longer notes, which is acknowledged as a limitation of this analysis. Sentence-level embeddings were generated by averaging the second-to-last hidden layer across all tokens, yielding a 768-dimensional embedding vector for each exam [66].

#### *Predictive Model*

A multilayer perceptron (MLP) regressor was trained on the BioClinicalBERT embeddings to predict exam duration. An MLP regressor was chosen for its ability to capture nonlinear relationships between the high-dimensional BioClinicalBERT embeddings and exam duration, a task for which linear models are likely insufficient given the complexity of clinical language representations. As a preliminary experiment, the analysis was restricted to exams with durations between the 15th and 85th percentiles of the duration distribution, focusing on the typical range of exam lengths and reducing the influence of extreme values. The dataset was split into training (80%) and test (20%) sets. Hyperparameters were tuned using 5-fold cross-validation with grid search, yielding an optimal architecture of a single hidden layer with 64 units, tanh activation, Adam optimizer, and a regularization parameter

($\alpha$) of 0.003.

*Results*

Table 4.5 presents the model performance on the training and test sets, and Table 4.6 presents the average RMSE by procedure code on the test set.

Table 4.5: BioClinicalBERT + MLP model performance

| **Set** | **MAE (min)** | **RMSE (min)** | **$R^2$** |
|---|---|---|---|
| Train | 6.34 | 8.20 | 0.621 |
| Test | 8.12 | 11.03 | 0.331 |

The model achieved a test $R^2$ of 0.331, indicating that Clinical Indication text alone explains approximately one-third of the variance in exam duration within the typical duration range. The gap between training ($R^2 = 0.621$) and test performance ($R^2 = 0.331$) suggests moderate overfitting, likely attributable to the limited discriminative capacity of short clinical indication phrases relative to the high-dimensional embedding space.

Procedure-level RMSE varied considerably, with MSPCTLWW exhibiting substantially higher error (31.4 minutes) than all other procedure codes, consistent with the inherently high variability of this procedure (SD = 30.5 minutes, as noted in Section 4.4.2). Procedures with lower intrinsic variability, such as MKNEWOL and MKNEWOR (SD $\approx$ 14 minutes), showed correspondingly lower RMSE values (5.5–5.6 minutes).

These findings suggest that unstructured clinical indication text carries meaningful but limited predictive signal for exam duration. The modest $R^2$ of 0.331 is nonetheless noteworthy given that the model relies solely on free-text clinical context, without access to any structured variables. A combined model incorporating both structured features and text-derived embeddings would potentially improve predictive performance and represents a promising direction for future work.

Table 4.6: Average RMSE by procedure code (test set)

| **Procedure Code** | **RMSE (minutes)** |
|---|---|
| MSPCTLWW | 31.40 |
| MCARWC | 12.12 |
| MSPCWO | 9.57 |
| MBRAWO | 8.99 |
| MPELGIGUWW | 8.69 |
| MABDWW | 8.41 |
| MSPLWO | 8.26 |
| MBRASTROKE | 7.82 |
| MOFNWW | 7.04 |
| MBRAWW | 7.02 |
| MBRTUMOR | 6.28 |
| MPROSTATE | 6.24 |
| MKNEWOL | 5.57 |
| MKNEWOR | 5.49 |

### *4.8 Summary*

This chapter presented an exploratory data analysis of MRI exam data obtained from two complementary sources—modality log files and electronic health records—across two campuses of the University of Washington Medical Center. Several data quality issues were identified during preprocessing, including the presence of linked exams, in which multiple procedure codes were recorded with identical start times for the same patient, and moderate missingness in anthropometric variables (Height and Weight: 17.1%) and support staff information (SupportStaff: 67.8%). These issues were addressed through targeted exclusion criteria and median imputation prior to analysis.

Exam duration exhibited strong positive skewness (skewness = 2.23) and excess kurtosis (kurtosis = 10.10), with a median of 34 minutes and considerable variability across

procedure codes (median range: 22–77 minutes). Procedure code emerged as the strongest determinant of exam duration, with subspecialty group providing additional stratification. Among patient and exam-level characteristics, contrast use showed the most meaningful association with duration, with contrast-only exams (W) having a median duration more than twice that of exams without contrast (WO). Emergency and STAT exams were shorter than outpatient and routine exams, which likely results from the focused nature of urgent imaging protocols. Surprisingly, age showed no meaningful association with exam duration ($r = -0.02$).

A preliminary investigation of the predictive content of unstructured clinical indication text using BioClinicalBERT embeddings and an MLP regressor yielded a test $R^2$ of 0.331, demonstrating that free-text clinical context encodes meaningful information about clinical context and complexity that complements the structured variables examined in this chapter. The train/test performance gap ($R^2 = 0.621$ vs. 0.331) suggests that further refinement of the text-based modeling approach would be beneficial, and a combined model incorporating both structured features and text-derived embeddings represents a promising direction for future work.

Together, the distributional characteristics and data quality challenges identified in this chapter provide the necessary context for the concordance-based data integration and predictive modeling methodology presented in the following chapter.

Chapter 5

# A BETTER WAY TO SCHEDULE: DATA RECONCILIATION AND PREDICTIVE MODELING FOR MRI EXAM DURATION

*This chapter is based on: Li, L.*[1]*, Mastrangelo, C.*[1]*, Mossa-Basha, M.*[2,3]*, Mabotuwana, T.*[4]*, "Concordance-based validation of electronic health records and modality log files to improve MRI exam duration prediction and scheduling performance," Journal of Imaging Informatics in Medicine, provisionally accepted pending revision.*

***Chapter Summary***[5]

The objectives of this chapter are to evaluate inter-system concordance between MRI exam duration data derived from modality log files (MLFs) and electronic health records (EHRs), develop a concordance-based data quality assessment framework, and assess the impact of this framework on the performance of predictive models for MRI exam duration.

Exam duration data from February 2022 through February 2024 were extracted from MLFs and EHRs. After fuzzy merging, concordance was assessed using Bland–Altman analysis and the concordance correlation coefficient (CCC). A concordance-based outlier removal threshold was established to improve inter-system agreement. A Random Forest (RF) regression model was then trained on the cleaned data and evaluated against template-based scheduling.

We extracted 52,112 records from MLFs and 46,570 from EHRs. After exclusions and fuzzy merging, 30,275 records were retained; restricting to procedure codes with $\geq$400 records yielded 22,737 records, and 16,297 remained after concordance-based outlier re-

[1]Industrial & Systems Engineering, University of Washington, Seattle, USA

[2]Department of Radiology, University of Washington, Seattle, USA

[3]Department of Radiology, University of Alabama at Birmingham, Birmingham, USA

[4]Philips Healthcare, Bothell, USA

[5]Adapted from abstract of original article

moval. Bland–Altman analysis revealed discordance between MLF and EHR duration measurements, with systematic differences attributable to data collection methodology in each system. Concordance-based filtering improved data quality for downstream analysis. The Random Forest model outperformed template-based scheduling for 11 of 12 procedure codes, with mean absolute error reductions ranging from 2.0% to 57.0%. For high-variability procedures, the proportion of exams completed within ±10 minutes of the scheduled duration increased from as low as 29% to over 79%.

In conclusion, concordance-based validation is critical when integrating multi-source healthcare data. Machine learning models trained on validated data substantially improved MRI scheduling accuracy, particularly for procedures with high intrinsic variability, providing a scalable framework for data-driven scheduling optimization.

### *5.1 Introduction*

MRI is one of the most frequently used among core imaging modalities in the radiology department due to its high contrast resolution, 3D capabilities, and ability to monitor anatomical detail, disease progression, and treatment response. The duration of an MRI exam can be impacted by factors such as the protocol(s) used, patient attributes, institutional workflows, staffing, and performing technologist [67]. Historically, recorded MRI exam durations have exhibited significant variability. In contrast to exam duration variability, scheduled exam slots are more deterministic. At the study site, exam slot lengths are primarily determined by a scheduling template that is maintained and periodically updated, with default durations assigned based on procedure codes. Here, deterministic means that once established, the template remains relatively stable; however, templates may vary considerably across institutions and patient settings (e.g., inpatient, outpatient, or emergency). Schedule adherence refers to the extent to which the actual exam durations align with the scheduled time slots. High schedule adherence is desirable because it minimizes delays and idle time, thereby improving workflow efficiency, increasing patient throughput, reducing wait times, and improving patient experience.

Variability in MRI exam duration is often accommodated by lengthening exam slots during scheduling, leading to lower resource utilization and higher exam costs [67]. Typically,

MRI exams are manually assigned to fixed-length time slots by experienced technologists, a process that requires considerable time and effort [53]. Given the high variability in exam durations, a deterministic scheduling approach can lead to low scanner utilization [68], reduced patient satisfaction [69], and reduced patient access for timely imaging. A substantial body of literature has sought to improve MRI scheduling through operations research-based optimization models, discrete-event simulation, or a combination of both [70–77], and more recently, digital twins [78]. These methods aim to address key sources of uncertainty, including demand, patient arrivals, and exam duration. In most cases, duration is modeled using probability distributions [76, 78]. However, since optimization is often applied at the block-scheduling level [79,80], it limits flexibility and adapts poorly to variability in individual exams. This limitation has spurred another line of research focused on predicting the duration of individual exams, aiming to improve scheduling adherence at the exam level.

Using log file data to predict exam duration has been studied with a variety of analytical methods: linear regression [67], generalized linear models [81], feedforward neural network [82], and random forest [83], among others. However, most prior studies relied on single-source data and did not systematically assess agreement across clinical information systems. In operational radiology settings, exam duration may be derived from both EHR documentation and modality-generated log files, yet discrepancies between these sources are rarely evaluated before predictive modeling. This study addresses this gap by introducing a concordance-based validation framework for multi-source MRI duration data and examining its impact on downstream machine learning performance.

## *5.2 Methods*

### *5.2.1 Study Setting and Data Sources*

This retrospective study was approved by the institutional review board (IRB), and the requirement for written informed consent was waived. The study site is a large academic research hospital with multiple campuses, two of which were included in this analysis: one equipped with 4 MRI scanners (quaternary care medical center) and the other with 2 MRI scanners (community hospital). Data were obtained from two sources: EHRs [84] and

MLFs [85]. Records from February 2022 through February 2024 were extracted from MLFs and EHRs at both sites.

MLFs are records automatically generated by MRI scanners that capture events, actions, and operational details during imaging exams. In the system used at our institution, the extracted MLFs include the key variable of *exam duration*, recorded with exact start and end timestamps for each sequence, as well as supplementary variables, including *scanner ID*, *study type* (i.e., anatomical focus), and *protocol name*. Per the MLF system's User Guide, exam duration is defined as the time elapsed between the first evidence that the patient is in the scanner room and the last evidence of patient presence. Evidence is derived from a combination of events, including patient selection or console closure, table movements, and other modality-specific actions. While our MLF implementation contains detailed machine-focused exam information, including executed sequences and event timestamps, it lacks patient-specific information to meet patient privacy compliance requirements and to preserve the architectural boundary between clinical and operational data systems. Table 5.1 presents key variables from the EHR data used in this study.

Table 5.1: Key variables in EHR data.

| **Category** | **Variables** |
|---|---|
| Patient demographics | Age, Gender, Race, Height, Weight |
| Exam information | Exam duration, Procedure code, Radiology subspecialty group, Patient status, Order priority |
| Clinical notes | Result narrative (including clinical indication and findings) |

Exam duration in the EHR is defined similarly to the MLF definition. Both systems theoretically capture the same interval—the time between a patient entering and exiting the scanner room. However, whereas MLF durations are recorded automatically via scanner-

generated events, EHR start and end times are entered manually by the performing technologist and are therefore subject to human error, including delayed entry, estimation, and inconsistent documentation practices across technologists [86]. Therefore, it is necessary to evaluate the exam duration data from both systems relative to one another to ensure consistency and reliability. To do this, the data from both sources must be merged.

### *5.2.2 Fuzzy Merge*

A typical data merge joins data from different sources based on one or more shared variables, commonly referred to as keys. Ideally, for this study, the data would be merged using a unique patient or exam identifier; however, as discussed earlier, such variables were unavailable. Consequently, the data were merged using fuzzy merge. Fuzzy merge identifies records that are *likely* to correspond to the same entity based on existing or user-defined similarity measures. By tolerating certain levels of discrepancy between data sources rather than pursuing a perfect, deterministic match, fuzzy merge allows the integration of heterogeneous data with a relatively high degree of accuracy.

The fuzzy merge methodology was developed through close examination of variables in both datasets and input from subject matter experts. Records were matched based on scanner and subspecialty group, with temporal overlap used to validate matches. The underlying rationale is that the likelihood of two distinct exams occurring simultaneously on the same scanner within the same subspecialty group is sufficiently low; therefore, acceptable temporal overlap combined with concordant scanner and subspecialty information provides reasonable evidence that records belong to the same exam. When multiple records matched on scanner and subspecialty group, the record with the largest temporal overlap was selected.

Some necessary data cleaning and preprocessing were performed prior to the fuzzy merge. Study type in the MLF data and subspecialty group in the EHR data both represent the anatomical focus of the exam. However, subspecialty group provides a broader classification, whereas study type is more specific. Accordingly, each study type was mapped to a corresponding subspecialty group. There were 63 different study types in the MLF data. To ensure sufficient sample size and generalizability, study types with fewer than 20 occur-

rences were excluded. Exams with durations of 0 minutes or greater than 300 minutes were excluded, as such values are likely due to data entry errors, workflow nuances (e.g., scout scans and unclosed studies), or implausible exam lengths.

### *5.2.3 Data Quality Assessment*

For generalizability and sufficient sample size, only procedure codes with more than 400 records over the study period were included in subsequent analyses.

Agreement between MLF and EHR exam durations was assessed using Bland–Altman analysis [87] and the concordance correlation coefficient (CCC) [88–90]. The CCC quantifies agreement by combining measures of correlation and mean bias. The Bland–Altman method requires differences between measurements to be normally distributed [91]; because the original data followed a leptokurtic distribution, a logarithmic transformation was applied [92]. Note that, on a log scale, the difference between two values corresponds to their ratio. Limits of agreement (LoA) were calculated as $\bar{d} \pm 1.96s$, where $\bar{d}$ is the mean difference and $s$ is the standard deviation of the differences. The LoAs were back-transformed to the original scale for clinical interpretability [93].

Outlier removal was based on inter-source concordance rather than absolute thresholds. Elimination thresholds were defined as multiples of the mean exam duration, with the final threshold selected to balance data retention with acceptable concordance levels.

### *5.2.4 Predictive Modeling*

Currently, exam slot lengths are primarily determined by a scheduling template that assigns default durations based on procedure codes. These slots may be adjusted to accommodate patient-specific needs, such as mobility limitations or the requirement for anesthesia. To evaluate whether a data-driven approach could improve upon template-based scheduling, a Random Forest regression model [6] was developed to predict exam duration. Procedure codes without a default duration in the scheduling template were excluded from the mod-

[6] Random Forest was selected because it handles mixed feature types and non-linear interactions naturally, requires minimal distributional assumptions, and has demonstrated strong performance for tabular healthcare data in prior work.

eling dataset, as the primary performance metric requires a template-based baseline for comparison.

Data were split into training (80%) and validation (20%) sets. Numerical variables were imputed using the median and then standardized, while categorical variables were one-hot encoded. Procedure-specific outliers were removed using the interquartile range (IQR) method, excluding durations below Q1 − 1.5 × IQR or above Q3 + 1.5 × IQR, a standard preprocessing measure to prevent extreme MLF duration values from disproportionately influencing model training. This preprocessing step is distinct from the concordance-based outlier removal described in Section 5.2.3, which addresses inter-system data quality prior to any modeling decisions, rather than procedure-level extreme values within the modeling pipeline. Additional variables, including contrast use and time of day, were derived to capture factors potentially influencing exam duration. Contrast use was derived as a categorical variable from the procedure name and result narrative, with four categories: without contrast (WO), with contrast (W), without and with contrast (WW), and arthrography (ARTH). Time of day was derived as a categorical variable from the exam start time, with three categories: AM (06:00–12:00), PM (12:00–18:00), and night (all other hours). Model hyperparameters were tuned using randomized search with 5-fold cross-validation over 50 randomly sampled combinations, optimizing for mean absolute error. The final model used the following hyperparameters: n_estimators = 500, max_depth = 15, min_samples_split = 15, min_samples_leaf = 2, and max_features = None. Model performance was evaluated against the current template-based scheduling approach using mean absolute error (MAE) and the proportion of predictions within ±5 and ±10 minutes of actual duration. Predicted durations were rounded to the nearest 5- or 10-minute interval for scheduling practicality. All analyses were performed using Python (version 3.12) with scikit-learn (version 1.6.1).

## *5.3 Results*

### *5.3.1 Fuzzy Merge Results*

A total of 52,112 records were extracted from MLFs and 46,570 from EHRs. After applying exclusion criteria, 51,902 and 43,937 records remained in the MLF and EHR datasets, re-

spectively. Using the fuzzy merge methodology, 30,275 exam records were identified in the merged dataset, achieving a match rate of 69% with respect to the EHR data and 58% with respect to the MLF data. Among the 30,275 matched pairs, the median temporal overlap was 0.659 (mean 0.608), with 69.0% of matches achieving an overlap greater than 0.50 and 36.1% exceeding 0.75. Samples of both matched and unmatched records were manually inspected to verify the validity of the matching criteria. Matched records with lower temporal overlap values were found to be consistent with genuine matches based on concordant scanner identity, procedure code, and general time window, while unmatched records most likely resulted from operational factors such as scout scans, research or quality assurance studies, and mid-workflow scanner reassignments. Table 5.2 summarizes the number of records retained at each stage of the data processing pipeline, along with the rationale for each exclusion. Figure 5.1 plots exam durations recorded in the EHR against those in the MLF for all 30,275 fuzzy-merged exam pairs. Figure 5.2 presents the distribution of exam duration differences ($\Delta$ = EHR duration − MLF duration) across all fuzzy-merged pairs, illustrating the inter-system discordance that motivated the concordance-based filtering framework described in Section 5.2.3.

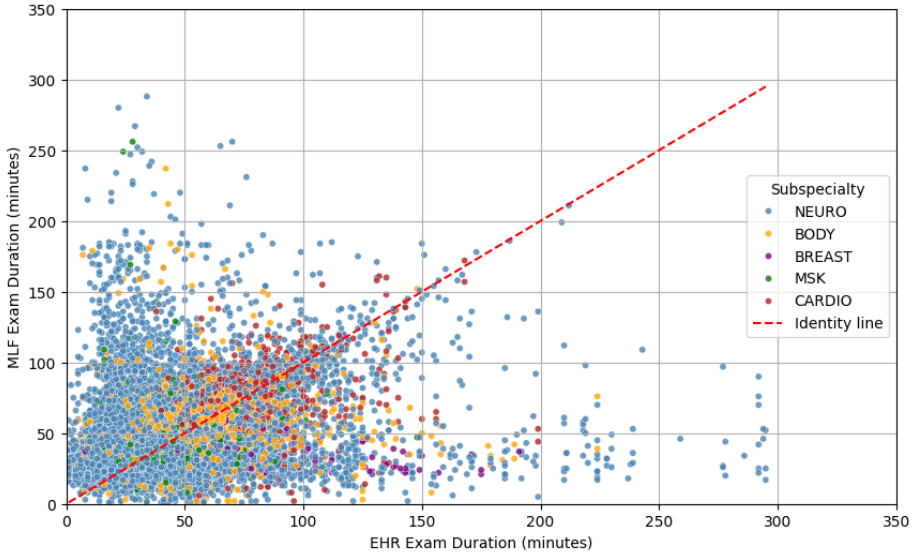


Figure 5.1: EHR durations vs. MLF durations for fuzzy-merged exams.

Table 5.2: Data attrition across processing stages

| **Stage** | **MLF Records** | **EHR Records** | **Reason for Exclusion** |
|---|---|---|---|
| Initial extraction | 52,112 | 46,570 | — |
| After exclusion criteria | 51,902 | 43,937 | Durations of 0 or >300 min; study types with <20 occurrences |
| After fuzzy merge | 30,275 | 30,275 | Unmatched records (no concordant scanner/subspecialty/temporal overlap) |
| After procedure code frequency filter | 22,737 | 22,737 | Procedure codes with <400 records over study period |
| After concordance-based outlier removal | 16,297 | 16,297 | Inter-system duration discordance exceeding thresholds |
| After template exclusion (for modeling) | 14,159 | — | Procedure codes without a default duration in the scheduling template excluded from modeling dataset |
| After IQR-based outlier removal (for modeling) | 13,503 | — | Procedure-level MLF duration outliers removed prior to RF model training |

*Note:* Record counts from the fuzzy merge stage onward reflect the merged dataset. The "—" in the EHR Records column from the template exclusion stage onward indicates that record counts are tracked by MLF records only, as MLF exam duration serves as the model target variable; EHR-derived variables continue to serve as model features. IQR-based outlier removal was applied to MLF exam durations within the training set only as part of the modeling pipeline and did not affect the concordance analysis reported in Section 5.3.2.

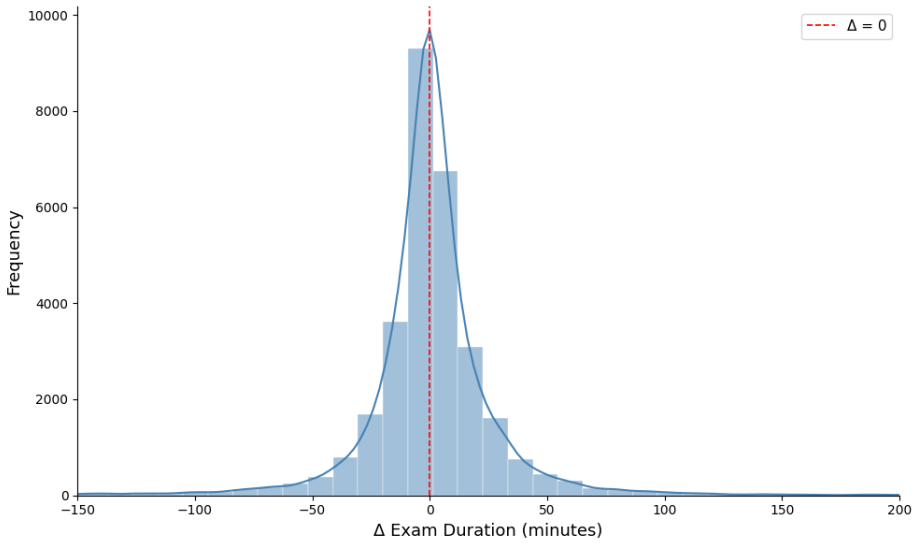


Figure 5.2: Distribution of exam duration differences ($\Delta$ = EHR duration − MLF duration) across all 30,275 fuzzy-matched exam pairs.

### *5.3.2 Data Quality and Agreement Analysis*

Of the 30,275 merged records, 22,737 exams remained after limiting analysis to procedure codes with more than 400 records. Figure 5.3(a) shows the differences between exam durations in MLFs and EHRs on the log scale plotted against the mean exam duration. The LoAs were back-transformed to the original scale for clinical interpretability [93], as shown in Figure 5.3(b).

The Bland–Altman analysis demonstrated magnitude-dependent disagreement between systems. Differences increased proportionally with exam duration, resulting in heteroscedastic dispersion and wide limits of agreement on the original scale. This pattern suggests systematic measurement differences related to exam duration rather than purely random error. The absolute differences tended to increase with longer mean durations; however, the differences appear roughly proportional to the mean. Points beyond LoAs appeared more densely clustered around shorter mean durations, whereas relatively fewer such points are observed for longer mean durations. This partly supports the notion that, although prolonged exams (>120 mins) are uncommon, both the MLF and EHR capture them with

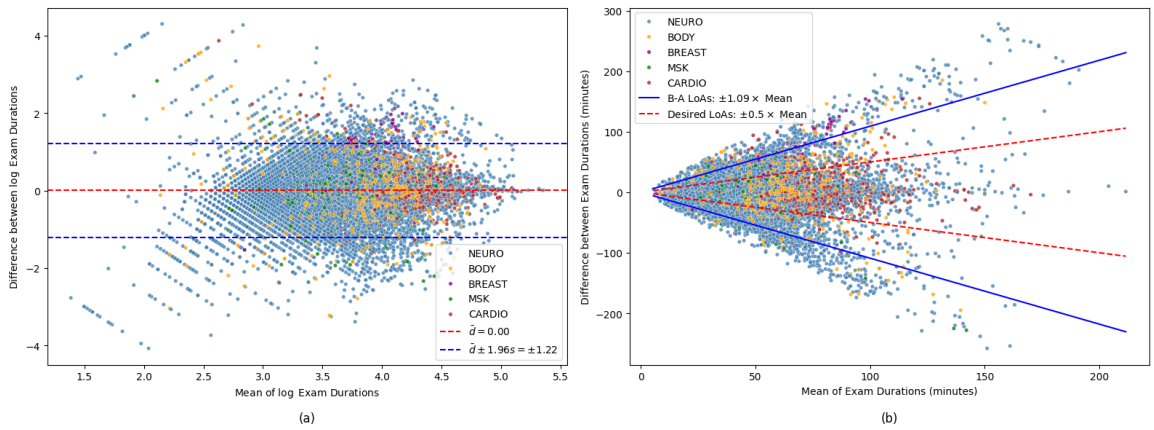


Figure 5.3: (a) Bland–Altman plot of the log-transformed data. (Note: The "banding" pattern results from discretization of exam durations.) (b) Bland–Altman plot on the original scale with back-transformed LoAs.

fair agreement when they do occur.

The CCC of exam durations in both datasets was 0.3321 (95% CI: 0.3205–0.3436). Applying the concordance-based elimination thresholds (Figure 5.3(b)) retained 16,297 records (72%) and improved the CCC to 0.8681 (95% CI: 0.8642–0.8718), exceeding commonly cited thresholds for strong agreement (>0.80) [94–96]. Table 5.3 presents the CCC before and after concordance-based outlier removal for each of the 17 procedure codes meeting the frequency threshold. This substantial improvement supports the use of inter-system concordance as a defensible criterion for data validation prior to predictive modeling.

### *5.3.3 Template-Based Scheduling Performance*

Figure 5.4 presents the difference between actual and scheduled exam durations by procedure code ($\Delta_{schedule}$ = actual duration − scheduled duration). A positive value of $\Delta_{schedule}$ indicates that the actual exam duration exceeded the scheduled slot, whereas a negative value indicates that the exam was completed in less time than scheduled. The red vertical line at zero indicates perfect schedule adherence.

Figure 5.4 shows that using template-based scheduling, schedule adherence is suboptimal. For certain procedure codes (e.g., MSPLWW, MPROSTATE, MMSBRAIN), at least

Table 5.3: Per-procedure concordance correlation coefficient (CCC) before and after concordance-based outlier removal

| **Procedure Code** | **N (pre)** | **CCC (pre)** | **95% CI (pre)** | **N (post)** | **CCC (post)** | **95% CI (post)** |
|---|---|---|---|---|---|---|
| MBRAWW | 5,859 | 0.22 | 0.19–0.24 | 4,044 | 0.84 | 0.83–0.85 |
| MBRAWO | 2,333 | 0.18 | 0.15–0.22 | 1,612 | 0.85 | 0.84–0.87 |
| MBRTUMOR | 2,310 | 0.16 | 0.12–0.20 | 1,578 | 0.83 | 0.82–0.85 |
| MSPLWO | 1,951 | 0.16 | 0.12–0.21 | 1,452 | 0.77 | 0.75–0.79 |
| MABDWW | 1,684 | 0.24 | 0.20–0.29 | 1,302 | 0.73 | 0.71–0.76 |
| MBRASTROKE | 1,335 | 0.06 | 0.00–0.11 | 850 | 0.75 | 0.72–0.78 |
| MSPCWO | 1,172 | 0.17 | 0.12–0.23 | 800 | 0.81 | 0.78–0.83 |
| MCARWC | 916 | 0.29 | 0.23–0.34 | 815 | 0.65 | 0.60–0.68 |
| MPROSTATE | 795 | 0.12 | 0.05–0.19 | 670 | 0.55 | 0.50–0.60 |
| MSPLWW | 754 | 0.35 | 0.29–0.41 | 564 | 0.82 | 0.80–0.85 |
| MOFNWW | 717 | 0.26 | 0.19–0.33 | 527 | 0.81 | 0.77–0.83 |
| MKNEWOR | 511 | 0.07 | −0.01–0.16 | 410 | 0.66 | 0.60–0.71 |
| MMSBRAIN | 505 | 0.45 | 0.38–0.52 | 369 | 0.90 | 0.88–0.92 |
| MKNEWOL | 484 | 0.21 | 0.12–0.29 | 385 | 0.67 | 0.61–0.72 |
| MBRBWWS | 523 | −0.01 | −0.10–0.07 | 310 | 0.61 | 0.53–0.67 |
| MSPCTLWW | 474 | 0.08 | −0.01–0.17 | 307 | 0.75 | 0.70–0.80 |
| MABDWO | 414 | 0.32 | 0.23–0.40 | 302 | 0.80 | 0.75–0.84 |

*Note:* Pre = before concordance-based outlier removal ($n$ = 22,737). Post = after concordance-based outlier removal ($n$ = 16,279). A negative CCC indicates negligible or inverse agreement; the confidence interval for MBRBWWS crosses zero, indicating no statistically meaningful agreement prior to filtering. CI = confidence interval.

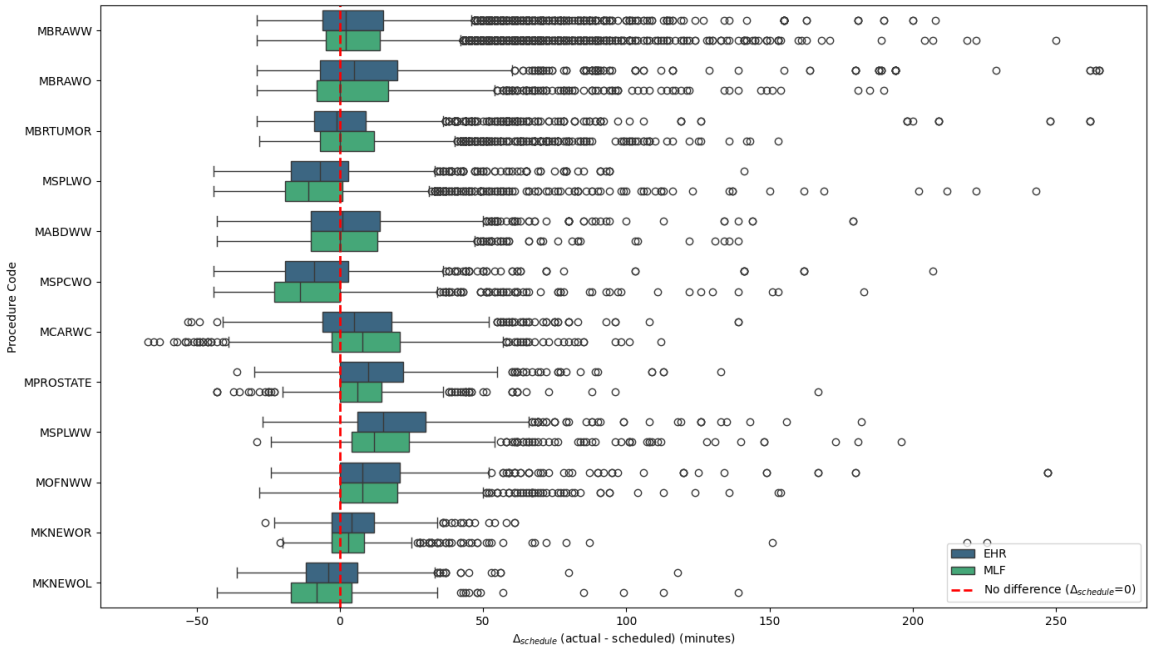


Figure 5.4: Schedule adherence represented by $\Delta_{schedule}$. Procedure codes ordered by descending exam count.

75% of exam durations exceeded their scheduled time, whereas for others (e.g., MSPCWO, MSPLWO), the majority of exams finished earlier than their scheduled time. This indicates that the current scheduling template fails to adequately account for exam duration variability, leading to under- or overestimated exam slots.

### *5.3.4 Predictive Model Performance*

Table 5.4 presents model performance compared to template-based scheduling across all 12 procedure codes.

The RF model outperformed template-based scheduling for 11 of 12 procedure codes, with mean absolute error (MAE) reductions ranging from 2.0% to 57.0%. The largest improvements were observed for procedures with high baseline variability: MSPCWO (56.96% reduction, from 15.79 to 6.80 minutes), MKNEWOL (51.77% reduction), and MSPLWO (48.40% reduction). Only MBRTUMOR showed slightly worse performance (−1.54%),

Table 5.4: Comparison of template-based and model-based scheduling approaches across MRI procedure codes. MAE = mean absolute error. Practical accuracy metrics (columns 7–10) represent the percentage of exams within ±5 minutes and ±10 minutes of actual exam duration under each scheduling approach.

| Procedure Code | Mean | Std | Template MAE (min) | Model MAE (min) | MAE Reduction (%) | Template ±5min (%) | Model ±5min (%) | Template ±10min (%) | Model ±10min (%) |
|---|---|---|---|---|---|---|---|---|---|
| MBRAWW | 32.94 | 10.71 | 8.97 | 8.79 | 2.04 | 38.91 | 36.19 | 69.00 | 67.83 |
| MBRAWO | 34.42 | 14.72 | 11.18 | 9.59 | 14.26 | 31.19 | 34.73 | 57.23 | 62.70 |
| MBRTUMOR | 30.06 | 9.34 | 7.75 | 7.87 | -1.54 | 39.08 | 37.68 | 74.30 | 73.24 |
| MSPLWO | 31.31 | 7.77 | 12.25 | 6.32 | 48.40 | 23.85 | 50.38 | 43.46 | 81.92 |
| MABDWW | 47.48 | 14.31 | 11.59 | 11.14 | 3.93 | 23.35 | 26.46 | 53.31 | 53.70 |
| MSPCWO | 29.17 | 9.22 | 15.79 | 6.80 | 56.96 | 16.28 | 43.02 | 29.07 | 79.07 |
| MCARWC | 73.85 | 15.13 | 14.77 | 11.80 | 20.13 | 25.00 | 26.32 | 43.42 | 47.37 |
| MPROSTATE | 52.80 | 9.90 | 10.72 | 8.42 | 21.44 | 30.66 | 40.15 | 53.28 | 70.07 |
| MSPLWW | 44.40 | 12.62 | 13.80 | 10.30 | 25.36 | 27.68 | 30.36 | 51.79 | 58.93 |
| MOFNWW | 39.28 | 11.21 | 11.71 | 9.74 | 16.89 | 31.87 | 38.46 | 59.34 | 59.34 |
| MKNEWOL | 31.67 | 7.35 | 11.49 | 5.54 | 51.77 | 23.46 | 59.26 | 46.91 | 82.72 |
| MKNEWOR | 32.58 | 7.01 | 5.73 | 5.29 | 7.66 | 49.32 | 56.16 | 90.41 | 91.78 |

*Note:* Procedure code expansions: MBRAWW, MRI brain without and with contrast; MBRAWO, MRI brain without contrast; MBRTUMOR, MRI brain tumor protocol; MSPLWO, MRI lumbar spine without contrast; MABDWW, MRI abdomen without and with contrast; MSPCWO, MRI cervical spine without contrast; MCARWC, MRI cardiac with contrast; MPROSTATE, MRI prostate; MSPLWW, MRI lumbar spine without and with contrast; MOFNWW, MRI orbit/face/neck without and with contrast; MKNEWOL, MRI left knee without contrast; MKNEWOR, MRI right knee without contrast.

though this difference was negligible. The model achieved higher rates of ±10-minute accuracy for 10 of 12 procedures, with particularly notable gains for MSPCWO (29.07% to 79.07%) and MSPLWO (43.46% to 81.92%). Notably, for some procedures, such as MBRAWW, MAE improved, while the ±5 and ±10-minute accuracy rates declined, creating a seeming contradiction. This is because while MAE reflects the average magnitude of all errors, the ±5 and ±10-minute metrics are essentially binary indicators. For procedures like MBRAWW, the model reduced large prediction errors (improving MAE) but increased the number of predictions falling just outside the threshold ranges, resulting in lower ±5 and ±10-minute accuracy rates despite overall MAE improvement. This reflects different error distributions between the two approaches rather than contradictory performance.

Figure 5.5 compares the two scheduling approaches (i.e., template-based and model-based) in terms of schedule adherence. For most procedure codes, the model-based scheduling approach improved schedule adherence compared with the template-based approach by (1) shifting the distributions closer to the vertical line denoting perfect adherence, and (2) reducing the variability of the differences. This indicates that with the proposed model-based scheduling approach, a greater proportion of exams would fall within their scheduled time slots, with reduced variability in the differences between actual and scheduled durations.

### *5.4 Discussion*

This study demonstrated that concordance-based data quality assessment can substantially improve multi-source healthcare data for predictive modeling. After fuzzy merging MLF and EHR data and applying concordance-based outlier removal, the CCC improved from 0.33 to 0.87. A Random Forest model trained on this cleaned dataset outperformed template-based scheduling for 11 of 12 procedure codes, with MAE reductions ranging from 2% to 57%.

MRI log files contain comprehensive exam-level information that can be utilized to improve operational efficiency. However, like most medical data, they can be noisy, incomplete, and potentially erroneous. Therefore, it is essential to evaluate data quality. Results showed that shorter exams tended to exhibit greater variability and lower agreement between data sources. This is partly attributable to the case mix at the study site, where the majority of MRI exams are relatively short (median duration $<$ 30 minutes). Some subspecialties and

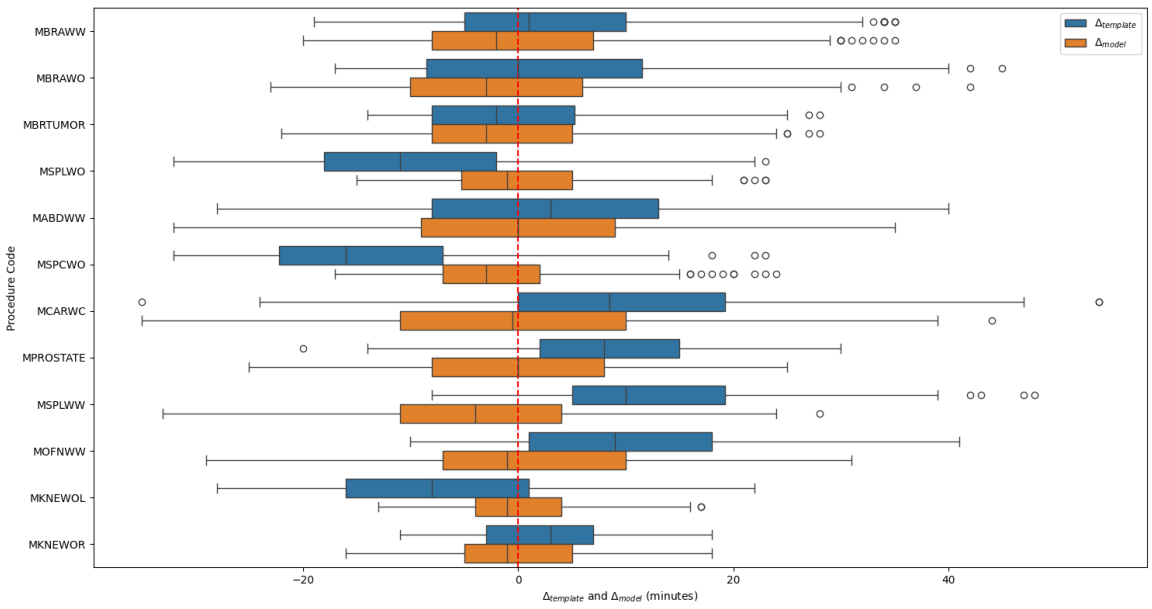


Figure 5.5: Comparison of template-based scheduling and prediction model-based scheduling.

procedure codes also exhibit greater inherent variability. For example, neuro exams often show higher variability due to factors such as protocol complexity, patient-related factors, and case-specific clinical requirements.

Data cleaning is essential in predictive modeling; however, conventional outlier removal using absolute or percentile-based thresholds may inadvertently eliminate clinically meaningful variability (e.g., removing exam durations above the 90th percentile or below the 10th percentile might eliminate valid data points indicative of interesting underlying factors or causalities). In multi-source environments, inter-system concordance provides an alternative validation criterion that preserves atypical but plausible cases while excluding discordant measurements likely attributable to documentation or system error.

The current template-based scheduling approach performs adequately for certain procedures and, with expert input, can accommodate some variability. However, there is opportunity to better address the variability observed in other procedures. For some procedures, such as MSPCWO (cervical spine without contrast) and MSPLWO (lumbar spine without

contrast), not only did the majority of exams fall outside their scheduled slots, but the deviations suggested systematic patterns—the current scheduling template systematically underestimates or overestimates the time required to complete these exams. On days when such exams predominate, they can considerably disrupt the entire schedule. The predictive model in this study offers an alternative to the conventional template-based scheduling approach. Trained using variables from both EHRs (predominantly patient characteristics and procedure codes) and MLFs (predominantly exam durations), the model improves schedule adherence for almost all procedure codes studied by allowing more exams to fall within their scheduled time slots and reducing variability in the differences between actual and scheduled durations. In practice, improved schedule adherence may reduce downstream cascade delays, decrease technologist rescheduling workload, and increase effective scanner utilization. For high-volume procedures demonstrating systematic under- or overestimation, even modest per-exam improvements in duration prediction may translate into meaningful aggregate gains in scanner availability over time. Although prospective validation is required, these findings suggest potential operational benefits beyond prediction accuracy alone. These findings are consistent with prior studies demonstrating the utility of machine learning for MRI duration prediction. Avey et al. [67] identified patient and protocol factors influencing exam duration variability, while Wang et al. [83] showed that Random Forest models could reduce scheduling discrepancies. However, most prior work relied on single-source data. This study extends the literature by addressing data quality challenges inherent in multi-source data integration and demonstrating that concordance-based cleaning can substantially improve predictive model performance.

This study has several limitations. First, it was conducted at a single academic health system using a specific scanner vendor and scheduling infrastructure; results may differ in community settings, sites with different case mixes, or institutions using alternative documentation workflows. Second, analyses were limited to high-volume procedure codes to ensure statistical stability, potentially limiting generalizability to lower-frequency exams. Third, fuzzy matching without a unique identifier introduces potential misclassification, although concordance-based filtering was designed to mitigate this risk.

Future research could examine unusual data points in both datasets to identify their

underlying causes, which may inform workflow standardization and protocol development. Additionally, prediction accuracy for high-volume, high-variability procedures such as MAB-DWW and MBRAWW—where only marginal improvements were observed (3.93% and 2.04% MAE reduction, respectively)—may benefit from additional features such as clinical indication, specific imaging sequences, or more granular procedure classifications. Future studies could extend this work through prospective validation and implementation studies. Specifically, integrating prediction outputs into scheduling systems at the time of order entry would allow dynamic adjustment of exam slot lengths based on patient- and procedure-level features. Such studies should evaluate real-world effects on scanner utilization, technologist workload, patient wait times, and schedule stability. Demonstrating operational impact in live clinical environments will be essential to translate retrospective predictive improvements into measurable health system value.

Chapter 6

# TOWARD DATA-DRIVEN COMPETENCY MODELING: EVALUATING LEARNED TEMPORAL REPRESENTATIONS VS. MANUAL FEATURE ENGINEERING

## *6.1 Introduction*

As discussed in 1.1.1, systematic competency assessment in healthcare has long relied on subjective instruments including direct observation, global rating forms, and self-reported evaluations, which are resource-intensive, prone to rater bias, and typically administered at discrete time points rather than continuously throughout a practitioner's career. For MRI technologists specifically, competency manifests across a range of technical and cognitive demands: selecting and executing appropriate imaging protocols, managing patient-specific challenges such as implants and claustrophobia, detecting and correcting image artifacts, and adapting sequences in real time to clinical findings. These competencies are currently assessed through periodic supervisory review and informal peer feedback, neither of which provides the granular, longitudinal signal needed to track development trajectories or identify early indicators of performance gaps. The availability of routinely collected operational data, including exam durations, protocol selections, and patient encounter records, presents an opportunity to move beyond episodic assessment toward continuous, data-driven competency monitoring. Additionally, performing technologist has been identified as a significant factor influencing MRI exam duration variability [67], suggesting that competency-related differences in procedural efficiency are detectable in routinely collected operational records—and potentially learnable from them.

The architecture proposed to address this opportunity was a GRU-based "competency portfolio" model that dynamically learns a representation of each technologist's competency state from their longitudinal exam history. At each prediction step, the model would receive a sequence of the technologist's most recent exams—each characterized by procedure type, patient attributes, and contextual variables—and learn to weight historical encounters by

their relevance to predicting future performance. The GRU's recurrent hidden state would serve as a continuously updated competency representation, evolving as new exam experiences accumulated. An attention mechanism over the competency window would allow the model to selectively focus on the most informative historical encounters, analogous to how an experienced technologist might weight recent performance more heavily than distant history while still accounting for formative early experiences.

Preliminary analysis of the available technologist data, however, revealed three fundamental challenges that precluded direct evaluation of this architecture on the available clinical data. The first was a data quantity problem at the individual level: while the aggregate dataset contained a substantial number of exams, the distribution of exams across technologists and procedure codes was highly uneven, with many technologists contributing relatively few records. The longitudinal depth required for a recurrent model to learn meaningful competency trajectories was not available for a sufficient number of technologists to support robust model training and evaluation. The second challenge was more fundamental: exam duration as a proxy for competency proved too noisy to isolate a meaningful temporal signal. Unlike domains where output quality is directly and objectively measurable, MRI exam duration is jointly determined by technologist efficiency and a collection of factors entirely outside the technologist's control, including patient characteristics, protocol complexity, contrast requirements, equipment performance, and case mix. Disentangling the technologist's contribution from these confounds would require either experimental control over case assignment, which is impractical in clinical settings, or sufficiently rich covariate data to statistically isolate the technologist effect, which the available dataset did not support. The third challenge was specific to the study site: the MRI technologists at the study site are highly skilled and experienced, and as a result, exam durations exhibited minimal temporal variation at the individual level. In other words, the learning curves and performance evolution trajectories the architecture was designed to detect were largely absent.

To rigorously evaluate the proposed model, we pivoted to three established time series forecasting datasets with rich and diverse temporal dynamics: UCI Air Quality monitoring, Capital Bikeshare demand, and Rossmann Store sales. This enabled us to test whether the

competency portfolio mechanism, which is fundamentally an architecture for learning which historical contexts are relevant for predicting the future, generalizes beyond healthcare to other sequential prediction tasks.

Traditional machine learning approaches, particularly gradient boosting methods like XGBoost (eXtreme Gradient Boosting) [97], have achieved strong performance in forecasting tasks through careful and extensive feature engineering of temporal patterns. However, these methods require domain expertise to design effective features and struggle to capture complex, long-range temporal dependencies. Recent advances in deep learning, including recurrent architectures [27, 98], attention mechanisms [99], and transformer-based foundation models [100], have demonstrated superior ability to automatically learn temporal representations [101]. In this work, we compare our GRU-based approach against XGBoost as a representative classical baseline, with the aim of determining whether learned temporal embeddings outperform manually engineered features under controlled conditions. By comparing these fundamentally different approaches under controlled conditions with minimal domain-specific feature engineering, we address two research objectives. First, we evaluate the performance and generalizability of the proposed competency portfolio architecture across diverse sequential prediction tasks. Second, we contribute to the broader methodological question of when recurrent learning offers advantages over tree-based models for time series forecasting. By maintaining consistent preprocessing, identical input features, and comparable model complexity across both approaches, we isolate a fundamental question for practitioners: when feature engineering effort is limited whether due to time constraints, limited domain expertise, or application to new domains, does automatic representation learning through recurrent architectures provide advantages over the manual lag specification and extensive feature engineering typically required for strong tree-based model performance?

### *6.2 Related Work*

This section reviews prior work on time series forecasting methods, positioning our comparative study within the broader landscape of machine learning and deep learning approaches for sequential prediction.

### *6.2.1 Tree-Based Methods for Time Series Forecasting*

Gradient boosting methods, particularly XGBoost [97], LightGBM [102], and CatBoost [103], have established themselves as dominant approaches for tabular prediction tasks, including time series forecasting. These methods construct ensembles of decision trees sequentially, with each tree correcting the residual errors of its predecessors. Their success stems from several practical advantages: robustness to feature scaling, native handling of missing values, built-in regularization mechanisms, and computational efficiency.

For time series applications, tree-based methods require explicit feature engineering to capture temporal dependencies. The standard approach constructs lag features representing past values at fixed offsets, rolling statistics (moving averages, standard deviations), and calendar features (day-of-week, month, holiday indicators). This feature engineering process requires domain expertise to select appropriate lag windows and aggregations; when done well, it produces highly competitive models. The effectiveness of this approach is evidenced by its dominance in various data forecasting competitions. In the M5 Forecasting Competition [104], which focused on retail sales prediction using Walmart data, gradient boosting methods dominated the leaderboard. The winning solutions used LightGBM with extensive feature engineering, including hierarchical aggregations, price elasticity features, and carefully tuned lag structures. Similarly, XGBoost-based solutions achieved top rankings in the original Rossmann Store Sales Kaggle competition, demonstrating the strength of tree-based approaches for retail demand forecasting when combined with domain-specific feature engineering.

However, this reliance on manual feature engineering presents limitations: practitioners must decide which lags to include, risking either leaving out important temporal relationships or including irrelevant features that increase dimensionality and risk of overfitting. As sequence length increases, the feature space grows linearly, potentially triggering the curse of dimensionality. Furthermore, tree-based models treat each lag as an independent feature, lacking an inherent understanding of the relationships between temporally adjacent observations.

### *6.2.2 Deep Learning Approaches for Time Series Forecasting*

Recurrent neural networks (RNNs) offer an alternative approach that processes sequences through learned temporal dynamics rather than explicit feature engineering. Long Short-Term Memory (LSTM) networks [98] introduced gating mechanisms to address the vanishing gradient problem in standard RNNs, enabling learning over longer sequences. Gated Recurrent Units (GRUs) [27] simplified this architecture while maintaining comparable performance, using update and reset gates to control information flow through the hidden state. These recurrent architectures process sequences timestep-by-timestep, maintaining a hidden state that evolves based on each new observation. This design naturally preserves temporal ordering and enables the network to learn which past information to retain or discard. Unlike lag-based features, the model's capacity remains constant regardless of sequence length, since it uses the same parameters for processing both short and long histories.

Attention mechanisms [99] further enhanced sequential modeling by allowing direct access to historical context at any position in the input sequence, bypassing the information bottleneck of recurrent state propagation. For time series forecasting, attention enables models to focus on relevant historical periods (e.g., the same hour yesterday) without carrying that information through intermediate timesteps. The Transformer architecture, built entirely on attention without recurrence, demonstrated remarkable success in natural language processing and has since been adapted for time series applications. The Temporal Fusion Transformer (TFT) [100] represents a prominent example of transformer-based forecasting, combining recurrent layers for local processing with attention for capturing long-range dependencies. TFT also incorporates variable selection networks and interpretable attention weights, addressing the "black box" criticism of neural approaches. Other notable architectures include Informer [105], which introduces sparse attention for computational efficiency on long sequences, and N-BEATS [106], which uses residual stacks without recurrence. DeepAR [107] demonstrated the effectiveness of autoregressive RNNs for probabilistic forecasting across many related time series (e.g., demand across thousands of products). By training a global model that shares parameters across entities while conditioning on entity-specific features, DeepAR showed that deep learning can leverage cross-series patterns that

entity-specific models cannot capture.

A comprehensive survey [101] reviews these and other deep learning approaches for time series forecasting, noting that neural methods excel when (1) large training datasets are available, (2) complex nonlinear patterns exist, and (3) manual feature engineering is impractical. However, the survey also acknowledges that simpler methods often suffice for well-understood domains with established feature engineering practices.

### *6.2.3 Comparative Studies*

Despite the architectural sophistication of deep learning approaches, their superiority over traditional methods is not guaranteed. The M4 Competition [108] evaluated 100,000 time series across multiple domains, and found that pure machine learning methods, including deep learning, generally underperformed classical statistical approaches like exponential smoothing and ARIMA. The winning method was a hybrid combining exponential smoothing with neural networks, suggesting that leveraging statistical structure remains valuable. This finding sparked significant debate about when deep learning provides genuine advantages. [108] argued that deep learning's strength in domains like computer vision and natural language processing does not automatically transfer to time series, where patterns are often simpler and datasets are smaller. Later research has clarified when deep learning offers advantages: forecasting multiple horizons simultaneously, capturing dependencies across variables, leveraging patterns across related time series, and situations where practitioners lack the expertise or time for extensive feature engineering.

The M5 Competition [104] provided a contrasting data point. Unlike M4's diverse short series, M5 featured hierarchical retail data with rich exogenous variables (prices, promotions, calendar events). In this competition, gradient boosting with extensive feature engineering dominated, but deep learning approaches showed competitive performance, particularly in capturing hierarchical relationships. The key insight was that data characteristics, including volume, complexity, and available covariates, determine which approach achieves better overall performance.

Several studies have directly compared tree-based and neural approaches. Elsayed et

al. [109] found that gradient boosting with careful feature engineering, including window-based input transformation and output concatenation, outperformed eight state-of-the-art deep learning models across nine datasets. This suggests that tree-based models require substantial feature engineering to achieve their full potential, while neural architectures can learn effective representations automatically. Our work examines the complementary question: when both approaches receive minimal, identical preprocessing, do learned temporal representations provide an advantage? We test this across three distinct forecasting domains, providing concrete empirical evidence for answering this question.

### *6.3 Datasets*

Table 6.1 shows a summary of the three datasets used for the model comparison in this chapter. The datasets are chosen because of the various domains they represent and the diverse temporal patterns they embody. Details of each dataset and preprocessing activities are given in the following subsections.

Table 6.1: Overview of datasets used for model comparison.

| **Dataset** | **Domain / Type** | **Freq.** | **Missing Data** | **Target ($y$)** | **Input Features** | **Notes** |
|---|---|---|---|---|---|---|
| **Air Quality (UCI)** | Environmental sensors (CO concentration) | Hourly | Yes (sensor dropouts) | CO(GT) (carbon monoxide, ppm) | Gas sensors (PT08.S1–S5), temperature (T), humidity (RH, AH), cyclical time encodings | Strong diurnal signal. |
| **Bike Sharing (UCI)** | Human behavioral demand | Hourly | No | cnt (total rentals) | Weather (temp, atemp, hum, windspeed), categorical dummies (season, weather, holiday, workingday), cyclical time encodings | Strong daily + weekly patterns. |
| **Rossmann Store Sales (Kaggle)** | Retail transactions | Daily (per store) | Yes (store closures) | Sales (per-store revenue) | Store-level features (Promo, Open, Competition, etc.), categorical encodings (StoreType, Assortment, StateHoliday, SchoolHoliday), cyclical time encodings | Store-dependent. Some have strong weekly, holiday, and promotional effects. |

### *6.3.1 Air Quality Dataset (UCI)*

The Air Quality dataset [110] contains 9,358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device, located in a significantly polluted area at road level within an Italian city. Data were recorded from March 2004 to February 2005, representing the longest freely available recordings of field-deployed air quality chemical sensor responses at the time of publication. Ground truth hourly averaged concentrations for carbon monoxide (CO), nonmetanic hydrocarbons, benzene, total nitrogen oxides (NOx), and nitrogen dioxide (NO2) were provided by a co-located reference certified analyzer. Exogenous variables, such as temperature and relative humidity, are also included. Missing values, typically caused by sensor failures, are tagged with a -200 value. Given the temporal patterns and complex dependencies between variables in the dataset, it is frequently used for multivariate time series forecasting and missing data imputation studies. We use ground truth CO concentration (CO(GT)) as the target variable due to its direct health impact, strong temporal patterns, and relatively complete measurements compared to other pollutants in the dataset. Additionally, CO prediction enables direct comparison with prior studies using the same benchmark.

#### *Data Preprocessing*

Figure 6.1 shows where the missing values for each variable are located in the dataset. Note that only the independent variables (sensor readings PT08.S1-S5 and exogenous variables) and the dependent variable (CO(GT)) that are actually used in the models are included in the visualization. Other ground truth measurements are excluded due to their extensive missingness and potential data leakage issues.

Similar to the data preprocessing procedures in [111], the temperature variable (T) was linearly interpolated based on time, and a 24-hour rolling window mean was applied to relative humidity (RH) to capture recent diurnal patterns. However, unlike their approach of imputing the dependent variable using the median, we first applied linear interpolation to fill gaps of up to 6 consecutive missing values; the remaining missing values were then

imputed with the median CO concentration for the corresponding hour of the day. This approach preserved temporal continuity while retaining hourly variability. The rest of the numerical variables (i.e., sensor readings) were imputed with the corresponding median values. The outlier mitigation strategy was similar to that in [111], where extreme values were clipped to the closest boundaries calculated by $Q1 - 1.5 \times IQR$ (interquartile range) and $Q3 + 1.5 \times IQR$, respectively. This standard, non-parametric approach preserves the size of the dataset while limiting the impact of data anomalies.

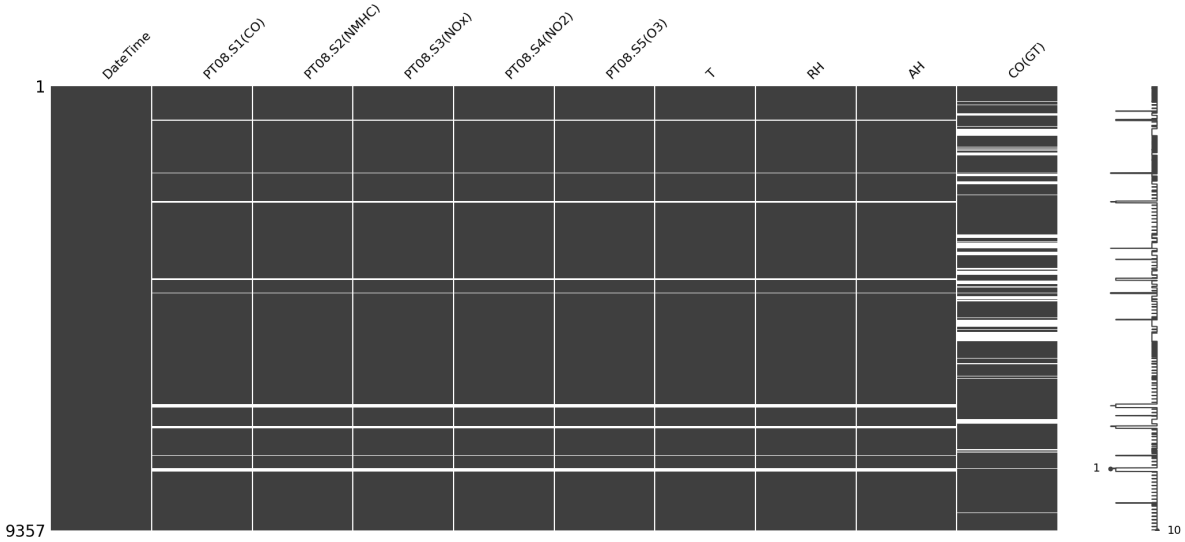


Figure 6.1: Missingness matrix of air quality data.

***Cyclical Time Feature Encoding*** Time variables, including month, day of the week, and hour of the day were extracted from the *DateTime* variable, and subsequently transformed using the sine and cosine functions to represent their cyclical nature [112] [113]. The formulas for transforming the variables are given below:

$$var\_sin = sin(2\pi \times \frac{var}{N}) \tag{6.1}$$

$$var\_cos = cos(2\pi \times \frac{var}{N}) \tag{6.2}$$

where *var* is the feature being transformed, and $N$ is the total number of categories in the feature (e.g., $N$=24 for time of day).

As mentioned before, the goal is to compare the different sequential prediction models under controlled conditions with minimal domain-specific feature engineering. Therefore, data preprocessing and feature engineering efforts were deliberately kept minimal and consistent. To ensure a fair comparison, we added lag features to the XGBoost model to provide it with the same temporal information available to the GRU model, which inherently captures sequential patterns through its recurrent architecture. The extent of the lag features was kept consistent with the competency window in the GRU model, ensuring both models had access to the same historical context for prediction. Additionally, we applied log-transformation to the target variable to handle its skewed distribution [114] and used StandardScaler to normalize the features to ensure stable training and convergence across models [115].

*Temporal Pattern Overview*

Figure 6.2 shows the autocorrelation function (ACF) plot of the target variable, CO(GT). The ACF visualizes the correlation of the time series with itself at different time lags. The plot reveals a strong 24-hour periodicity with slow decay over multiple weeks, which is consistent with air quality patterns driven by the daily and weekly cycles of human activities such as traffic and industrial operations. In addition, although the amplitude of the peaks are slowly decreasing, the autocorrelation remains statistically significant (well above zero) even at lag 600 hours ( 25 days), indicating strong long-term dependencies in the data. This characteristic makes the air quality dataset particularly well-suited for comparing sequential models, as it presents a non-trivial temporal pattern where the ability of model architectures to identify and leverage this pattern in predicting the future can be evaluated and compared.

### *6.3.2 Bike Sharing Dataset (UCI)*

Bike sharing systems have become prevalent in large cities around the world, making it easy for users to rent a bike from a particular location, go on a ride, and return it at another location. The Bike Sharing dataset [116] contains hourly bike rental data from the Capital Bikeshare system in Washington D.C., spanning 2011-2012. The dataset includes

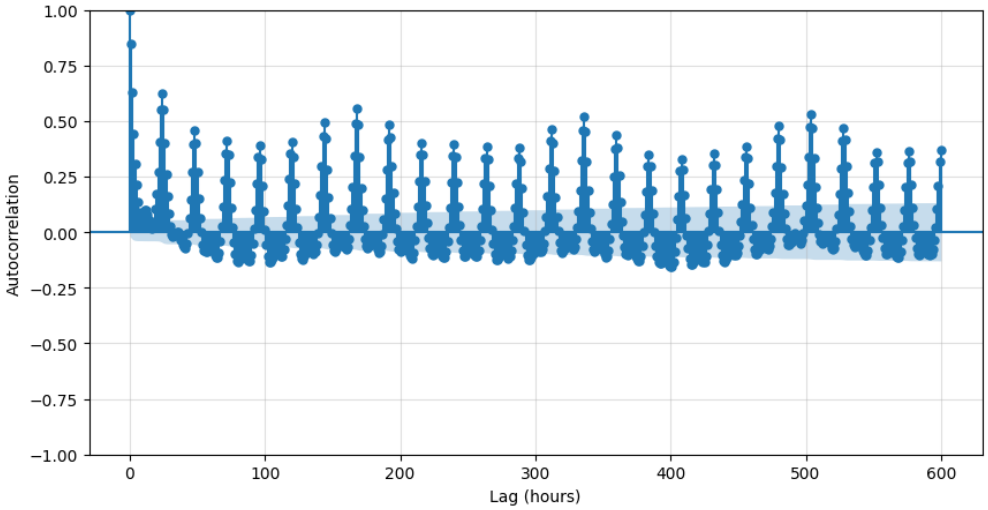


Figure 6.2: Autocorrelation of the target variable CO(GT).

17,379 hourly observations with weather variables (temperature, humidity, wind speed) pre-normalized to [0,1] by the dataset creators, along with temporal features (hour, day, month) and categorical variables (season, weather situation, holiday status).

*Data Preprocessing*

Unlike the Air Quality data in 6.3.1, there are no missing values in the Bike Sharing dataset. Therefore, the preprocessing did not include data imputation. However, although weather variables were provided pre-normalized, we reapplied StandardScaler using training set statistics to ensure proper train/test separation and consistency across datasets. Categorical variables were one-hot encoded. The target variable (hourly rental count) was log-transformed to stabilize variance. Same cyclical time feature encoding technique as in 6.3.1 was applied to the time variables. Lag features were created for the XGBoost model to ensure both models had access to the same historical context for prediction.

*Temporal Pattern Overview*

Figure 6.3 shows the ACF plot of the target variable, hourly rental count, in the dataset. Similar to the carbon monoxide concentration in the Air Quality dataset, there is a strong

diurnal pattern with weekly cycles; however, the decay appears to be even slower, with peaks remaining above or near 0.5 at lag 600 hours.

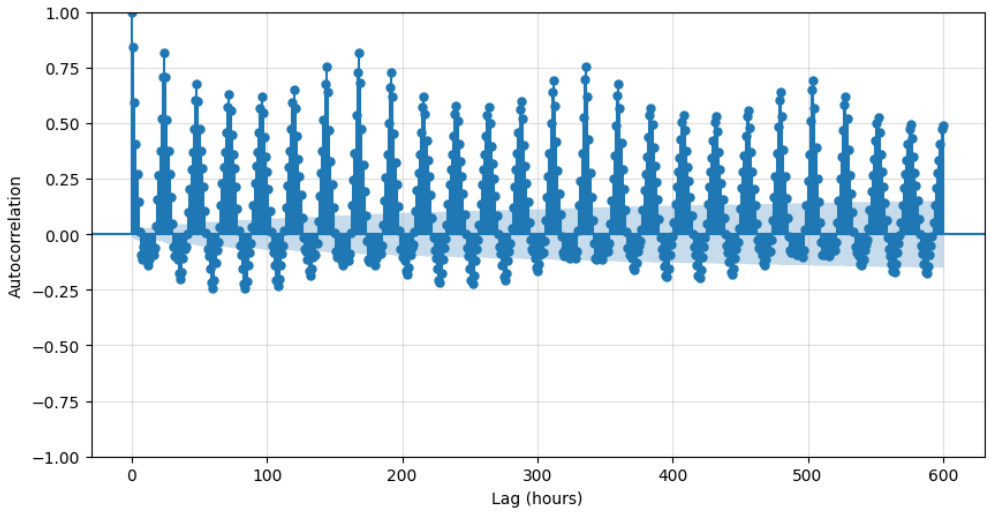


Figure 6.3: Autocorrelation of hourly rental count.

### *6.3.3 Rossmann Store Sales Dataset (Kaggle)*

The Rossmann dataset [117] contains daily sales records from 1,115 Rossmann stores across Germany, spanning from January 1, 2013 to July 31, 2015 (942 days). For the purpose of this study, stores with more than 15% missing sales records were excluded to ensure reliable temporal sequence modeling with recurrent architectures. From the remaining stores (note that the data may still include closed days with zero sales), we randomly sampled 100 to balance computational feasibility with sufficient diversity in sales patterns across store types, locations, and assortment categories. The Rossmann dataset presents several forecasting challenges relevant to our research objectives:

1. *Promotional effects:* Sales exhibit sharp increases during promotional periods, requiring models to capture both baseline demand and promotion-driven spikes.

2. *Store heterogeneity:* Different stores exhibit distinct sales patterns based on location,

size, assortment type, and competitive environment. This heterogeneity motivates our use of learned store embeddings.

3. *Seasonal patterns:* Strong weekly cycles (weekday vs. weekend), monthly patterns (beginning/end of month), and holiday effects require models to capture multiple temporal scales.

4. *Store Closure days:* Stores closed on Sundays and holidays, resulting in zero sales. This requires careful handling during preprocessing and evaluation.

These challenges have also proved the Rossmann dataset particularly suitable for evaluating our competency portfolio architecture due to the following reasons:

- Store-level heterogeneity enables validation of learned embeddings capturing behavioral patterns

- Multi-scale temporal patterns (daily, weekly, monthly, seasonal) test the model's ability to leverage varying lookback windows

- Complex relationships between variables provide a good test case for automatic feature learning versus manual engineering

*Data Preprocessing*

The preprocessing began with merging the sales data file with store metadata. After this, missing values were handled. In the Rossmann dataset, there were two types of missingness in the target variable:

- *Store Closure Days*: These were days when stores were closed (e.g., Sundays, holidays) and legitimately had zero sales. These days were retained as valid entries.

- *Data Recording Gaps*: To ensure temporal continuity for recurrent architectures while maintaining data quality, we first re-indexed each store to the complete date range [2013-01-01, 2015-07-31], setting store open indicator variable to zero for new dates

and forward/backward-filling static store features. Target variables (sales) were set to zero for newly created dates. This is consistent with the interpretation that missing sales correspond to either store closures or zero-sales days. Then, we selected 100 stores with ≥85% data completeness (proportion of open days with valid sales).

After filling the gaps in the records, all 100 stores had complete 942-day observations with no gaps in the temporal index, ensuring reliable sequence modeling for the recurrent neural network architecture while avoiding target imputation that could introduce artificial patterns. It should be noted that although closed days were retained in the data, they were excluded as prediction targets for all models. In addition, closed days may appear in input sequences to the models, as historical closure patterns provide context for forecasting (e.g., post-holiday recovery patterns). This approach ensures that evaluation focuses on operationally relevant predictions while preserving the temporal context that closed days provide.

Table 6.2 summarizes the features used. To maintain a fair comparison between the models, by design, we applied only minimal domain-agnostic feature engineering. Rather than extensive feature engineering typical of gradient boosting approaches [118] [119], we focused on basic temporal encodings and readily available metadata. Lastly, similar to the previous datasets, the target variable (daily sales) was log-transformed to address skewness and heteroscedasticity. Numerical features (CompetitionDistance, CompetitionOpen, Promo2Open, DayOfMonth, WeekOfYear) were standardized to zero mean and unit variance using StandardScaler.

*Temporal Pattern Overview*

Figure 6.4 shows the autocorrelation of daily sales for four example stores. We can see that different stores exhibit diverse temporal sales patterns. Store 262 exhibits strong weekly seasonality with autocorrelation peaks (∼0.5) at 7-day intervals persisting across the full 120-day window. Store 602 shows similar weekly periodicity but with negative autocorrelations between peaks, suggesting pronounced within-week variation. Store 736 displays weaker alternating positive and negative autocorrelations with weekly periodicity. Store 837

Table 6.2: Features for Rossmann dataset

| **Category** | **Feature** | **Description** |
|---|---|---|
| **Competition** | CompetitionOpen | Months since competitor opened |
| | CompetitionDistance | Distance to nearest competitor (m) |
| | HasCompetition | Binary: competition exists |
| **Promotion** | Promo | Binary: current promotion active |
| | Promo2Open | Weeks participating in Promo2 |
| | IsPromo2Month | Binary: current month in Promo2 cycle |
| **Temporal** | dow_sin, dow_cos | Cyclical day of week: $\sin(2\pi d/7)$, $\cos(2\pi d/7)$ |
| | month_sin, month_cos | Cyclical month: $\sin(2\pi m/12)$, $\cos(2\pi m/12)$ |
| | DayOfMonth | Day within month (1-31) |
| | WeekOfYear | Week number (1-53) |
| | IsMonthStart | Binary: days 1-3 of month |
| | IsMonthEnd | Binary: days 28-31 of month |
| | IsWeekend | Binary: Saturday or Sunday |
| **Categorical** | StoreType | One-hot encoded |
| | Assortment | One-hot encoded |
| | StateHoliday | One-hot encoded |
| **Sequential** | Lag features (XGBoost) | Previous $L$ days of all base features |
| | Store embeddings (GRU) | Learned $d$-dimensional vectors ($d$=4-16, optimized during training) |

demonstrates persistently positive autocorrelation across all lags with superimposed weekly cycles—even at 120 days, autocorrelation remains around 0.25—indicating both trend-like persistence and seasonality. This heterogeneity in temporal dynamics across stores motivates our multi-entity modeling approach (detailed in Section 6.4.3).

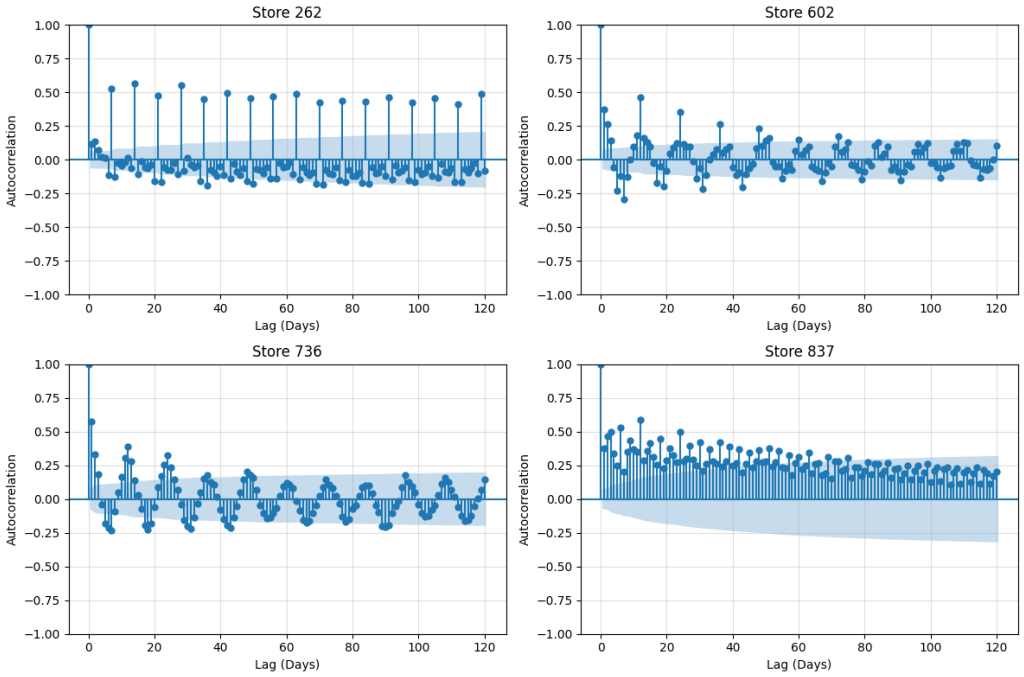


Figure 6.4: Autocorrelation of daily sales.

## *6.4 Model Architectures*

This section describes the architectural design of the two models evaluated in this study: Gated Recurrent Unit (GRU) networks for sequential processing and XGBoost for gradient-boosted decision trees. We provide detailed specifications of network structures, key components, and the fundamental differences in how each architecture represents and processes temporal information. The training procedures for these models are described subsequently in Section 6.5.

### *6.4.1 GRU Architecture*

*Network Structure*

The GRU model implements a deep architecture combining dense input transformation, a two-layer GRU stack with attention, and dense prediction layers to process sequential temporal data. We employ two stacked GRU layers in this study, which is a depth consistent with established time series forecasting literature and sufficient for the temporal complexity present in our evaluation datasets. The modular architecture readily extends to deeper configurations when problem complexity warrants additional representational capacity. The network accepts three inputs:

1. **Competency window:**[1] A sequence of L historical timesteps, each containing all features (temporal encodings, domain-specific variables). Shape: $(batch, L, n_{features})$

2. **Prediction features:** Features at timestep L+1 (temporal encodings, exogenous variables such as weather, if applicable). Shape: $(batch, n_{pred\_features})$

3. **Store index** (Rossmann only): Integer identifier for store-specific embeddings. Shape: $(batch, 1)$

The core architecture comprises two functional modules: one with sequential processing layers and one with prediction layers.

**Sequential Processing Layers:**

- **Input transformation:** Dense layers (256 $\rightarrow$ 128 units, ReLU activation) process the competency window features before recurrent processing

- **First GRU layer:** 64-128 units (dataset/window-dependent) with dropout, return_sequences=True to output hidden state at each timestep

[1]This terminology was inherited from the architecture's original application to MRI technologist competency modeling, where historical procedure performance informs predictions of future capability.

- **Second GRU layer:** Compressed by a ratio 0.25-0.75 of the first layer size, with dropout, return_sequences=True to provide timestep-wise hidden states for the attention mechanism

- **Attention mechanism:** Learns a weighted combination of all timesteps in sequence (detailed in Section 6.4.1)

**Prediction Layers:**

- **Context integration:** Concatenates attention-weighted sequence representation, prediction features, and store embedding (if applicable)

- **Dense layer 1:** 256 units, ReLU activation, L2 regularization

- **Dense layer 2:** 128 units, ReLU activation, L2 regularization

- **Output layer:** Single linear unit producing scalar prediction

The model also incorporates several regularization techniques to mitigate overfitting:

- Dropout (5-30%, tuned via hyperparameter optimization) after each GRU layer

- L2 weight regularization ($\lambda = 10^{-4}$) on GRU kernels and dense layers

- Optional Gaussian noise layers (stddev=0.0 in final models, retained for architectural flexibility)

This architecture balances representational capacity with computational efficiency. The two-layer design with moderate unit counts (64-128 in the first layer) prevents overfitting on the moderately-sized datasets (1,800-16,000 test samples) while maintaining sufficient capacity to learn complex temporal dependencies. The compression ratio between layers encourages the network to learn increasingly abstract representations, with the second layer capturing higher-level temporal patterns distilled from the first layer's more detailed sequence processing.

*Attention Mechanism*

To enable selective focus on relevant historical periods within the L-timestep competency window, we implement a temperature-scaled attention mechanism over the second GRU layer's timestep-wise outputs. This mechanism addresses a key limitation of standard recurrent models: later timesteps in the sequence can dominate the hidden state, potentially obscuring important patterns from earlier periods. The attention computation proceeds as follows:

1. **Attention scoring:** Apply a dense layer with tanh activation to each timestep's GRU output, producing scalar attention logits: $e_t = \tanh(W_a h_t + b_a)$, where $h_t$ is the GRU hidden state at timestep $t$

2. **Temperature scaling:** Divide logits by temperature parameter $\tau = 0.6$ to control attention sharpness: $e'_t = e_t/\tau$. Lower temperature ($\tau < 1$) sharpens the distribution, encouraging the model to focus strongly on the most relevant timesteps rather than distributing attention uniformly

3. **Normalization:** Apply softmax across the sequence dimension to obtain attention weights: $\alpha_t = \frac{\exp(e'_t)}{\sum_{i=1}^{L} \exp(e'_i)}$

4. **Weighted aggregation:** Compute context vector as the weighted sum of GRU outputs: $c = \sum_{t=1}^{L} \alpha_t h_t$

The temperature parameter $\tau = 0.6$ was selected based on preliminary experiments showing this value balances between overly uniform attention ($\tau \to \infty$, equal weights on all timesteps) and overly concentrated attention ($\tau \to 0$, attending only to single timestep). This intermediate value allows the model to flexibly distribute attention across multiple relevant periods while still prioritizing the most informative timesteps.

For time series forecasting, attention proves particularly valuable at longer sequence lengths. In a 96-hour window, for example, certain historical timesteps (such as yesterday's corresponding hour) are likely more predictive than others (such as 72 hours ago). The

attention mechanism can learn to weight relevant timesteps more heavily, assigning high $\alpha_t$ to informative lags without requiring manual specification of which lags to emphasize.

*Store Embeddings*

For the multi-store Rossmann dataset, we incorporate learned store embeddings to efficiently model store-specific characteristics. Rather than using one-hot encoding (which would require 100 binary features for 100 stores), the model learns a dense embedding matrix of shape $(n_{stores}, d)$. The embedding dimensionality $d$ represents a trade-off: while higher dimensions provide more representational capacity to capture store-specific nuances, they also risk overfitting with limited data per store. Therefore, it is tuned via hyperparameter optimization during model training (Section 6.5.2). Each store's embedding vector is retrieved via lookup using the store index and concatenated with features at two points in the architecture:

1. **Sequential processing:** Concatenated with each timestep in the competency window before GRU processing, allowing store-specific temporal pattern learning

2. **Prediction step:** Concatenated with the attention-weighted context and prediction features before final dense layers

This dual integration mechanism ensures the model has store-specific information available throughout processing: the GRU layers can learn how sales typically evolve over time for a store, while the prediction layers can utilize this information to predict sales for that specific store. Compared to one-hot encoding, embeddings provide parameter efficiency and enable knowledge sharing between stores with similar sales patterns, facilitating model generalization.

### *6.4.2 XGBoost Architecture*

XGBoost implements an ensemble learning approach based on gradient-boosted decision trees [97]. The algorithm builds an additive ensemble of decision trees sequentially:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{6.3}$$

where $K$ is the number of trees, $f_k$ represents the $k$-th tree function, and $x_i$ is the feature vector for sample $i$. Each new tree $f_k$ is trained to minimize the loss function with respect to the current predictions:

$$\mathcal{L}^{(k)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(k-1)} + f_k(x_i)) + \Omega(f_k) \tag{6.4}$$

where $l(\cdot)$ is the loss function (squared error in our case), $\hat{y}_i^{(k-1)}$ is the prediction from the first $k-1$ trees, and $\Omega(f_k)$ is a regularization term penalizing tree complexity. We employ histogram-based tree construction (tree_method="hist"), which bins continuous features into discrete buckets for computational efficiency and GPU acceleration.

Unlike GRU's sequential processing of temporal data, XGBoost processes each sample independently without sequential context; therefore, temporal information must be explicitly encoded as features rather than learned through recurrent state. Input features for the XGBoost model include:

- **Lag features:** For window size L, we create L separate features representing target values at timesteps $t-L, t-L+1, \ldots, t-1, t$ (e.g., 48 lag features for a 48-hour window)

- **Temporal encodings:** Cyclical sine/cosine transformations of hour, day-of-week, month

- **Domain features:** Weather conditions (if applicable), promotional indicators (Rossmann), store characteristics

- **Store identity:** One-hot encoded store IDs (Rossmann only)

Hyperparameter configuration and tuning procedures for the XGBoost model are detailed in Section 6.5.

### *6.4.3 Architectural Comparison*

The fundamental distinction between GRU and XGBoost architectures lies in how they represent and process temporal information, leading to different strengths and limitations for time series forecasting.

*Temporal Representation*

GRU maintains a hidden state vector that evolves as it processes the sequence timestep-by-timestep. At each step, the model updates its internal representation based on the current input and previous hidden state:

$$h_t = \text{GRU}(x_t, h_{t-1}) \tag{6.5}$$

This recurrent processing naturally preserves temporal ordering: the model understands that observation at $t = 5$ occurs between $t = 4$ and $t = 6$, even without explicit encoding. The gating mechanisms (update gate, reset gate) learn which information to retain from the past and which to discard, enabling selective memory over long sequences. Attention further refines this by allowing the model to directly access relevant earlier timesteps without routing all information through the recurrent bottleneck.

XGBoost receives temporal context as a flat feature vector where each lag is an independent column. For a 24-hour forecast window:

$$x = [\text{lag}_1, \text{lag}_2, \ldots, \text{lag}_{24}, \text{hour\_sin}, \text{hour\_cos}, \ldots] \tag{6.6}$$

The model has no inherent understanding of temporal ordering; therefore, temporal patterns must be learned through tree splits that discover predictive lag values. While the model can learn feature interactions (e.g.,"if $\text{lag}_{12}$ is high AND weather is fair, predict high"), capturing complex sequential dependencies requires many trees with specific split combinations.

*Model Scalability with Sequence Length*

For GRU, computational and memory requirements scale linearly with sequence length $L$. Processing a 96-hour sequence takes approximately twice as long as a 48-hour sequence, as the recurrent layers must iterate through more timesteps. However, the number of parameters remains constant regardless of $L$: the same GRU weights process every timestep in the sequence.

For XGBoost, feature dimensionality scales linearly with $L$: a 96-hour window requires 96 lag features per variable, doubling the feature count compared to 48 hours. This creates several challenges:

- **Search space explosion:** More features mean more potential splits to consider during tree construction
- **Overfitting risk:** High-dimensional feature spaces can lead to spurious patterns, especially with limited training data
- **Dilution of signal:** Many lags may be irrelevant (e.g., 83 hours ago may provide little information for hourly CO prediction), adding noise that regularization must filter out

Our results (Section 6.6.3) demonstrate this limitation: XGBoost performance catastrophically degrades at the 120-day Rossmann window where the feature space becomes excessively large, while GRU maintains stable performance across all window sizes.

*Multi-Entity Modeling*

For datasets involving multiple entities (e.g., Rossmann's 100 stores), the architectures employ different strategies:

**GRU:** Learns compact entity embeddings (4-16 dimensions per store) that are concatenated with temporal features. The same GRU weights process all stores' sequences, with embeddings providing store-specific adjustments. This parameter-efficient approach enables

knowledge sharing: patterns learned from Store A's sales dynamics can inform predictions for Store B if their embeddings are similar.

**XGBoost:** Uses one-hot encoding (100 binary features for 100 stores) concatenated with lag features. The trees must learn separate decision paths for each store through splits on these binary indicators. This approach treats stores as categorically distinct without built-in similarity structure, requiring more data per store to learn robust patterns. Each store effectively needs its own subtree structure within the ensemble.

*Model Interpretability*

The two architectures differ greatly in interpretability, presenting a trade-off between predictive flexibility and explainability.

**GRU:** Neural networks are often considered as black boxes:

- Hidden states are high-dimensional distributed representations without clear semantic meaning
- Predictions result from complex non-linear transformations across multiple layers
- Attention weights provide some interpretability, but not full prediction explanations

However, techniques like attention visualization, gradient-based saliency, and embedding analysis (Section 6.6.3) can provide partial insights into model behavior.

**XGBoost:** Compared to neural networks, tree-based models offer more direct interpretability through:

- Feature importance scores: Measure how much each feature contributes to splits across all trees
- Decision paths: Can trace exactly which conditions led to a specific prediction
- Partial dependence plots: Visualize how predictions change with individual features

This transparency is particularly valuable in production systems where stakeholders require explainability.

## *6.5 Model Training*

This section describes the computational infrastructure, hyperparameter optimization strategies, and the overall training procedures used to develop the GRU and XGBoost models for all three datasets. To ensure fair architectural comparison, both models undergo systematic hyperparameter optimization using identical data splits, optimization frameworks, and evaluation metrics, with search spaces tailored to each architecture's specific parameters. We prioritize reproducibility through fixed random seeds, documented hyperparameter search spaces, and validation-based early stopping to ensure consistent model selection. The training methodology balances comprehensive hyperparameter exploration with computational efficiency, enabling evaluation across multiple window sizes for each dataset while maintaining rigorous model selection procedures.

### *6.5.1 Training Infrastructure*

All experiments were conducted on Google Colab[2] utilizing NVIDIA T4 or A100 GPU(s) for accelerated training. The GPU environment provides sufficient memory (16GB T4, 40GB A100) for batch processing and dynamic memory allocation during training. TensorFlow's data pipeline with prefetching enabled efficient GPU utilization during GRU training, while XGBoost's histogram-based tree construction (tree_method="hist") leveraged GPU acceleration for rapid tree building.

To ensure reproducibility, we set global random seeds (seed=42) across all random number generators: Python's native random module, NumPy, TensorFlow, and XGBoost. All experiments used TensorFlow 2.x and XGBoost 2.x, with specific version information documented in the code repository. These measures ensure consistent results when rerunning experiments with identical configurations.

---

[2] `https://colab.research.google.com`

### *6.5.2 Hyperparameter Optimization*

*Optimization Framework*

We employed Optuna [120], a hyperparameter optimization framework using Tree-structured Parzen Estimator (TPE) sampling for efficient search space exploration. The TPE sampler models the relationship between hyperparameters and objective values, allocating the computational budget to promising regions of the search space. We configured Optuna with a MedianPruner to terminate unpromising trials early based on intermediate validation performance, substantially reducing total optimization time without sacrificing final model quality.

For each dataset and window size combination, we conducted 15-20 independent trials (15 for Air Quality and Bike Sharing, and 20 for Rossmann), where each trial trains a model with a specific hyperparameter configuration sampled from the search space. The objective function minimizes mean absolute error (MAE) on the validation set, computed on inverse-transformed predictions to ensure optimization occurs in the original target scale rather than log-transformed space. After completing all trials, we select the configuration achieving the lowest validation MAE and retrain the final model using combined training and validation data (described in Section 6.5.5).

*GRU Search Space*

The GRU hyperparameter search space balances architectural flexibility with computational constraints. Network architecture parameters include:

- **gru_units_1**: First GRU layer size, discrete choice from {64, 96, 128}

- **ratio_gru2**: Second layer size as ratio of first layer, discrete choice from {0.25, 0.33, 0.5, 0.67, 0.75}. For example, gru_units_1=96 with ratio_gru2=0.5 yields a second layer of 48 units.

- **dropout**: Dropout rate applied after each GRU layer, continuous uniform [0.05, 0.3]

- **store_emb_dim** (Rossmann only): Store embedding dimensionality, discrete uniform integer [4, 16]

- **learning_rate**: AdamW learning rate, log-uniform [1e-5, 3e-3]

This parameterization allows the optimization to discover both compact models (64 units, high compression ratio, low dropout) and larger capacity models (128 units, low compression ratio, high dropout) while maintaining reasonable trial durations. Each trial trains for 15-25 epochs on the training set with validation evaluation after each epoch, selecting the configuration with the lowest validation MAE.

The search space intentionally constrains certain parameters to maintain training stability and computational efficiency. We fix the batch size at 32, weight decay at 5e-4, and use a two-layer GRU architecture rather than optimizing depth, as preliminary experiments showed minimal benefit from three or more layers for our sequence lengths. The attention mechanism uses fixed temperature scaling ($\tau$=0.6) based on prior work demonstrating this value balances focus and smoothness for time series attention in our study.

*XGBoost Search Space*

The XGBoost search space encompasses tree structure, learning dynamics, and regularization parameters. We adapt certain ranges based on dataset characteristics:

**Tree Structure:**

- **max_depth**: Maximum tree depth, discrete [3, 8] for Rossmann; [3, 10] for Air Quality and Bike Sharing

- **min_child_weight**: Minimum sum of instance weights in child nodes, discrete [1, 10]

**Learning:**

- **learning_rate**: Step size shrinkage, log-uniform [0.01, 0.3] for Rossmann; [0.001, 0.1] for Air Quality and Bike Sharing

**Regularization:**

- **subsample**: Fraction of training instances per tree, continuous uniform [0.6, 1.0]
- **colsample_bytree**: Fraction of features per tree, continuous uniform [0.6, 1.0]
- **gamma**: Minimum loss reduction for split, continuous uniform [0.0, 5.0]
- **reg_lambda**: L2 regularization, log-uniform [0.001, 10.0]
- **reg_alpha**: L1 regularization, log-uniform [0.001, 10.0]

**Fixed Parameters:**

- **n_estimators**: Maximum of 2000-3000 trees with early stopping (dataset-dependent)
- **tree_method**: "hist" for GPU-accelerated histogram-based construction
- **random_state**: 42 for reproducibility

The dataset-specific adjustments reflect differing sample sizes and complexity. Rossmann's larger dataset (16,433 test samples across 100 stores) benefits from shallower trees (max_depth capped at 8) to prevent overfitting, while the constrained learning rate range [0.01, 0.3] avoids excessively slow convergence given the substantial training data. Air Quality and Bike Sharing, with approximately 1,800 test samples each, benefit from deeper trees (up to depth 10) for fine-grained pattern learning and lower learning rates for gradual refinement.

### *6.5.3 GRU Training Procedure*

*Loss Function and Optimizer*

GRU models optimize mean absolute error (MAE) as the training objective, chosen for its robustness to outliers and direct interpretability in the original target units. We employ the AdamW optimizer [121], which decouples weight decay from gradient-based updates, providing more effective regularization than standard Adam. The weight decay coefficient is fixed at 5e-4 across all experiments, with learning rates tuned via hyperparameter optimization as described in Section 6.5.2.

*Batch Construction and Data Flow*

Training data for GRU models consists of sliding window sequences with stride=1, generating overlapping temporal contexts. Each training sample comprises: (1) a competency window of L historical timesteps providing sequential context, (2) prediction features at timestep L+1 (time encodings, weather conditions if applicable), and (3) the target value at L+1. For a dataset with T total timesteps and window size L, this produces approximately T-L training samples, though the exact count depends on dataset-specific constraints (e.g., per-store processing for Rossmann).

Batches of size 32 are constructed by randomly sampling these sequences, with full dataset shuffling occurring at the start of each epoch. This stochastic sampling prevents the model from learning spurious patterns based on sample ordering while maintaining temporal integrity within each sequence. For the Rossmann dataset, we implement additional batch filtering to skip samples where the store is closed (store_open=0), as predicting zero sales provides no useful gradient signal for learning temporal dynamics.

*Training Loop*

The final GRU model training follows a multi-phase procedure designed to maximize performance while preventing overfitting:

**Phase 1 - Combined Training:** The model trains on the combined training and validation sets (previously split for hyperparameter optimization) to utilize all available data for final model fitting. Training proceeds for up to 100 epochs with batch shuffling at each epoch start.

**Phase 2 - Validation Monitoring:** Despite training on combined data, we maintain a small held-out validation subset (approximately 5% of training data) to monitor generalization and guide early stopping. After each epoch, we compute MAE on this subset and track the best validation loss encountered.

**Phase 3 - Early Stopping:** Training terminates if validation MAE fails to improve by at least 1e-4 for 20 consecutive epochs (30 epochs for Rossmann due to its larger scale and slower convergence). Upon early stopping, we restore the model weights from the epoch

with best validation performance rather than using the final epoch's weights.

**Phase 4 - Learning Rate Reduction:** If validation MAE plateaus for 5 consecutive epochs without triggering early stopping, we reduce the learning rate by a factor of 0.8 to enable finer-grained optimization near local minima. This continues until the learning rate falls below 1e-6 or training is terminated by early stopping.

*Rossmann-Specific Adaptations*

The multi-store structure of the Rossmann dataset requires architectural and procedural modifications. Store embeddings are integrated by concatenating the learned embedding vector (dimensionality tuned via hyperparameter optimization, typically 4-16) with both the competency window features and prediction features. This provides the model with store identity information throughout the sequential processing of historical data and at final prediction.

Batch construction implements per-store windowing to prevent temporal contamination across stores. Sequences never span multiple stores; each L-timestep competency window comes from a single store's chronological history. Additionally, we skip batches where the store is closed, as these samples contribute zero gradient (predicting zero when the target is zero) and dilute the learning signal from open-store periods.

### *6.5.4 XGBoost Training Procedure*

*Loss Function and Training Strategy*

XGBoost models use squared error as the objective for tree construction, whereas our evaluation focuses on MAE. The squared error loss provides smooth gradients that facilitate effective tree splitting, whereas direct MAE optimization can lead to unstable splits due to the loss function's non-differentiability at zero error. During hyperparameter optimization, we evaluate candidate configurations using the validation MAE to maintain consistency with our evaluation metrics as well as the GRU optimization objective.

Training employs an internal 80/20 split of the training data for early stopping: 80% builds trees while 20% evaluates generalization after each boosting round. We train up

to 2,000-3,000 trees (dataset-dependent) but employ early stopping with patience of 100-150 rounds. If validation error fails to improve for this many consecutive trees, training terminates, and the iteration with the best validation performance is selected. The tree_method="hist" setting enables GPU-accelerated histogram-based tree construction, substantially reducing training time while automatically handling feature interactions through tree structure.

*Feature Construction*

XGBoost requires explicit feature engineering to represent temporal context, contrasting with GRU's implicit sequential processing. For each prediction at timestep t+1, we construct:

**Lagged Features:** For each continuous variable (target variable, weather conditions if applicable), we create L explicit lag features representing values at timesteps [t-L+1, ..., t-1, t]. For a 96-hour window, this generates 96 lagged values per continuous variable.

**Temporal Features:** Cyclical time encodings (sine/cosine of hour-of-day, day-of-week, month-of-year) evaluated at the prediction timestep t+1.

**Store Features (Rossmann only):** One-hot encoded store identities, creating 100 binary features indicating which store the prediction concerns.

All features undergo standardization using StandardScaler fit exclusively on training data to prevent data leakage. The scaler state (mean and standard deviation per feature) is saved for consistent preprocessing during evaluation.

*Rossmann-Specific Adaptations*

The Rossmann dataset's multi-store structure necessitates careful lag feature construction to prevent cross-store contamination. Lagged features are generated independently per store. This per-store processing means the first L days of each store's chronological data must be dropped, as they lack sufficient history to construct complete lag feature vectors.

### *6.5.5 Training Data Management*

Both models use identical train/test splits to ensure fair comparison, with training data further subdivided for hyperparameter optimization. The data management strategy differs slightly between single-entity (Air Quality, Bike Sharing) and multi-entity (Rossmann) datasets.

*Air Quality and Bike Sharing*

For single-entity time series, we perform chronological splitting to preserve temporal ordering:

1. Split complete dataset at 80% temporal position → Training set, Testing set
2. Reserve final 10% of Training set → Validation set
3. Hyperparameter optimization uses Training-Validation split (train on 72%, validate on 8% of total data)
4. Final model trains on combined Training+Validation (90% of total data)
5. Evaluation occurs exclusively on Testing set (final 20% of data)

This procedure ensures that: (1) models never observe test data during training or hyperparameter optimization, (2) temporal ordering is strictly maintained (no information leakage from future to past), and (3) both GRU and XGBoost have access to identical information at each stage.

*Rossmann*

For the multi-store dataset, we first reindex each store to a complete global date range spanning the entire dataset timeline to ensure temporal consistency. After this reindexing, all stores share the same chronological date range. We then apply a single 80/20 train/test split at the same calendar date for all stores:

1. Reindex all stores to complete global date range (filling missing dates)
2. Split at 80% of the timeline → All stores' Training sets end at the same date
3. Reserve final 10% of Training period → All stores' Validation periods align temporally
4. Hyperparameter optimization uses all stores' Training sets, validates on all stores' Validation sets
5. Final model trains on all stores' combined Training+Validation sets
6. Evaluation uses all stores' Testing sets (final 20% of timeline, same dates for all stores)

This approach ensures temporal ordering is strictly maintained and all stores are evaluated on the same test period calendar dates, enabling meaningful aggregation of predictions across stores for system-level performance metrics.

### *6.5.6 Multi-Window Training Strategy*

To evaluate how historical context window size affects forecasting performance, we train independent models for each window size tested, with each model undergoing its own hyperparameter optimization.

Window sizes tested per dataset reflect the natural temporal scales of each domain:

- **Air Quality:** 6, 12, 24, 48, 72, 96 hours (6 windows)
- **Bike Sharing:** 3, 6, 12, 24, 48, 72, 96 hours (7 windows)
- **Rossmann:** 7, 14, 28, 56, 84, 91, 120 days (7 windows)

For each window size, we:

1. Conduct hyperparameter optimization (15-20 trials) using train/validation splits
2. Train final model with best hyperparameters on combined train+validation data

3. Evaluate on test set and save predictions, metrics, and model artifacts

4. Proceed to next window size with completely independent optimization

This independent training ensures that each window size is fairly evaluated without compromises, despite the fact that it requires more computation than sharing a single model across all windows.

### *6.5.7 Training Monitoring and Convergence*

*Tracked Metrics*

During training, we monitor several metrics to ensure convergence and diagnose potential issues:

**GRU Metrics:**

- Training loss (MAE) per batch and per epoch
- Validation loss (MAE) evaluated after each epoch
- Learning rate across epochs
- Epoch at which early stopping triggers

**XGBoost Metrics:**

- Training error (squared error) per boosting iteration
- Validation error (squared error) per iteration for early stopping
- Number of trees at convergence
- Best iteration (selected by early stopping)

*Convergence Criteria*

Models are considered converged when:

**GRU:** (1) Validation MAE fails to improve by at least 1e-4 for 20 consecutive epochs (30 for Rossmann), triggering early stopping, or (2) training reaches 100 epochs maximum, or (3) learning rate decays below 1e-6. The model weights from the epoch with the best validation MAE are restored as the final model.

**XGBoost:** (1) Validation error fails to improve for 100-150 consecutive trees (dataset-dependent patience), triggering early stopping, or (2) training reaches 2,000-3,000 maximum trees. The iteration wthe ith best validation error is automatically selected as the final model. This automatic selection of best checkpoints rather than final iterations proved essential for avoiding overfitting, particularly for XGBoost where validation error often showed some degradation in late training despite early stopping patience.

*Post-Training Adjustments*

After training completes, we apply minimal post-processing to ensure predictions satisfy domain constraints:

**Non-negativity Clipping:** For all datasets, predictions are clipped to ensure non-negative values: $\hat{y} = \max(0, \hat{y}_{model})$. This simple adjustment handles rare cases where models predict small negative values, particularly near the lower boundary of the target distribution.

These adjustments are minimal and domain-agnostic, avoiding dataset-specific tuning that might advantage one architecture over another. The emphasis throughout training and post-processing remains on fair comparison: both models receive identical data, similar computational budgets for hyperparameter search, and minimal post-hoc adjustments.

### *6.6 Experimental Results*

This section presents a comprehensive evaluation of GRU and XGBoost model performance across the three datasets described in Section 6.3. Our experimental design evaluates both architectures under identical conditions—same train/test splits, same feature sets,

and same hyperparameter optimization procedures—to provide a fair assessment of their relative strengths for temporal prediction tasks with minimal feature engineering.

For each dataset, we first present quantitative performance metrics. Following the metric presentation, we a provide detailed analysis examining model behaviors, window size effects, and comparative insights. Section 6.6.4 synthesizes findings across all three datasets to identify generalizable patterns in architectural advantages for time series forecasting.

### *6.6.1 Air Quality Dataset*

*Performance Metrics*

We evaluated GRU and XGBoost models for hourly carbon monoxide (CO) concentration prediction using the UCI Air Quality dataset with varying amounts of historical context (6 to 96 hours). Table 6.3 presents the complete performance evaluation across all tested window sizes. All models perform one-step-ahead prediction (predicting hour t+1 given hours t-L+1 to t).

Table 6.3: Air quality forecasting performance across historical context window sizes

| **Window** | $n_{test}$ | **GRU** | | | **XGBoost** | | | **Improvement** |
|---|---|---|---|---|---|---|---|---|
| (hours) | | MAE | RMSE | $R^2$ | MAE | RMSE | $R^2$ | (%) |
| 6 | 1,866 | 0.463 | 0.696 | 0.718 | 0.527 | 0.756 | 0.669 | 12.1 |
| 12 | 1,860 | 0.484 | 0.723 | 0.696 | 0.534 | 0.752 | 0.673 | 9.4 |
| 24 | 1,848 | **0.457** | **0.695** | **0.720** | 0.529 | 0.752 | 0.673 | **13.6** |
| 48 | 1,824 | 0.510 | 0.725 | 0.695 | 0.551 | 0.769 | 0.651 | 7.4 |
| 72 | 1,800 | 0.493 | 0.724 | 0.696 | 0.557 | 0.772 | 0.643 | 11.5 |
| 96 | 1,776 | 0.507 | 0.742 | 0.680 | 0.543 | 0.755 | 0.659 | 6.6 |
| **Mean** | **1,829** | **0.486** | **0.717** | **0.701** | **0.540** | **0.759** | **0.661** | **10.1** |

*Note:* MAE and RMSE reported in $mg/m^3$ (original scale after inverse log transformation). Best GRU performance highlighted in bold.

GRU consistently outperformed XGBoost across all tested window sizes, achieving MAE values ranging from 0.457 to 0.510 mg/m$^3$ compared to XGBoost's 0.527 to 0.557 mg/m$^3$. The relative improvement ranged from 6.6% at the 96-hour window to 13.6% at the 24-hour window, with an average advantage of 10.1% across all windows. Both models achieved moderate explanatory power with $R^2$ scores between 0.64 and 0.72, reflecting the inherent difficulty of CO prediction given the minimal feature engineering approach in our study.

The optimal performance for GRU is achieved at the 24-hour window (MAE: 0.457, RMSE: 0.695, $R^2$: 0.720), while XGBoost performed best at the shortest 6-hour window (MAE: 0.527, RMSE: 0.756, $R^2$: 0.669). Test set sizes decreased slightly with longer windows due to edge effects in batch construction, ranging from 1,866 samples at 6 hours to 1,776 samples at 96 hours. All reported metrics are computed on aligned test sets to ensure fair comparison between models.

*Analysis*

**Window Size Effects** Figure 6.5 shows the model performance across historical context window sizes. The 24-hour window emerged as optimal for GRU, representing a 13.6% improvement over XGBoost at this window and achieving the model's best overall performance. This window size aligns naturally with atmospheric pollutant dynamics, capturing one complete diurnal cycle of CO accumulation and dispersion driven by traffic patterns, temperature inversions, and photochemical processes. Performance degraded modestly at longer windows, with the 96-hour window showing 10.9% higher MAE (0.507 vs. 0.457) relative to the 24-hour optimum. Despite this degradation, GRU maintained relatively stable performance across the tested range, with MAE values confined to a narrow band of 0.457–0.510 mg/m$^3$ (11.6% spread).

XGBoost exhibited a different sensitivity pattern, achieving its best performance at the shortest 6-hour window before gradually degrading as window sizes increased. The model's MAE increased from 0.527 at 6 hours to 0.557 at 72 hours (5.7% degradation), with modest recovery to 0.543 at 96 hours. This pattern suggests that XGBoost's lag-based feature representation becomes less effective as the number of lagged features grows, potentially

introducing noise despite model regularization.

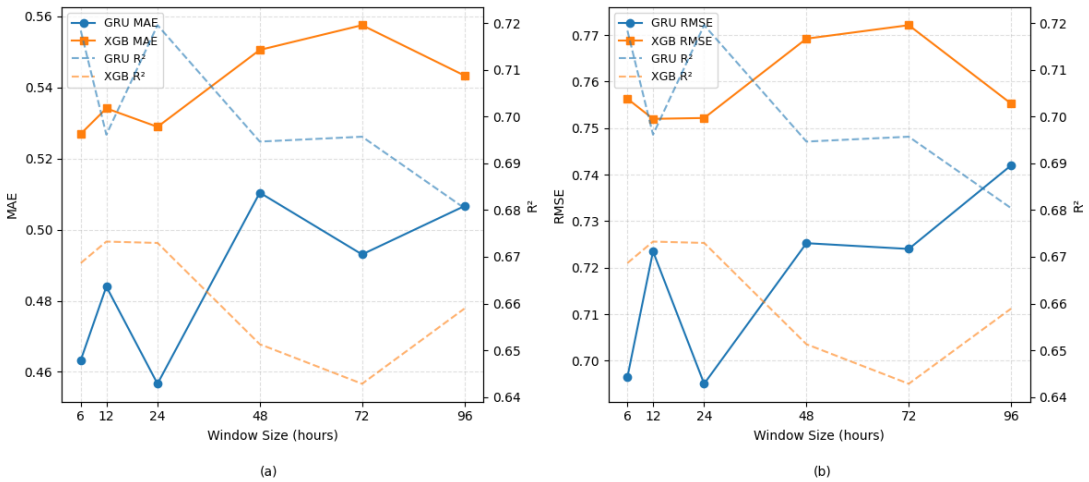


Figure 6.5: Air quality forecasting performance across historical context window sizes. (a) MAE and $R^2$ metrics. (b) RMSE and $R^2$ metrics. Solid lines show error metrics (left axis), dashed lines show $R^2$ (right axis).

**Sequential vs. Tabular Processing** The consistent GRU advantage across all windows (6.6%–13.6% improvement) demonstrates the value of sequential processing for atmospheric time series. GRU's recurrent architecture with attention mechanisms appears to selectively leverage relevant historical context while discarding irrelevant older information. This capability becomes particularly valuable at the 24-hour window, where the model can attend to the previous day's corresponding hour, a known strong predictor for air quality given repeating daily human activity patterns.

In contrast, XGBoost treats each lagged feature independently, losing the temporal ordering information inherent in the sequence. While the model's tree structure can learn interactions between specific lags, it cannot naturally capture the notion that lag-23h is "one hour before the daily cycle" in the same way that lag-25h is "one hour after the daily cycle." This fundamental architectural difference likely explains XGBoost's preference for shorter windows (fewer independent lag features) and GRU's superior performance at the 24-hour natural period.

**Prediction Quality and Temporal Dynamics** Figure 6.6 illustrates representative prediction quality over a 300-hour test period using the optimal 24-hour historical context window. Both models successfully capture the strong diurnal periodicity characteristic of urban CO concentrations, with regular oscillations between low overnight values ( 0.5 mg/m$^3$) and daytime peaks (4–6 mg/m$^3$) driven by traffic patterns and atmospheric mixing dynamics.

However, closer examination reveals systematic differences in prediction behavior. GRU predictions (orange) track ground truth values (blue) more faithfully, particularly during peak concentration periods and rapid transitions. For example, between hours 50–75, GRU captures both the magnitude and timing of concentration spikes, while XGBoost predictions (green) show slight lag and dampened peak responses. This pattern repeats throughout the time series, with XGBoost exhibiting greater smoothing—a characteristic behavior of tree-based models that average predictions across leaf nodes rather than producing continuous functional mappings.

The close alignment of all three curves confirms that both architectures have learned the fundamental temporal structure of CO dynamics. The 10.1% average MAE advantage for GRU (Table 6.3) manifests visually as small but consistent improvements in tracking rapid concentration changes and peak magnitudes. These incremental gains, while modest in absolute terms (0.05 mg/m$^3$), compound over extended forecast periods and could prove meaningful for applications requiring threshold-exceedance detection.

**Practical Implications** For operational air quality monitoring systems, these results provide several actionable insights. First, using 24 hours of historical context optimizes one-hour-ahead CO prediction accuracy (MAE: 0.457 mg/m$^3$, 13.6% improvement over XGBoost), capturing one complete diurnal cycle that provides essential context for short-term forecasting and real-time alert systems. Second, GRU's consistent 6.6–13.6% advantage across all window sizes (6–96 hours) demonstrates robust performance that reduces the need for application-specific window tuning, simplifying operational deployment. Third, the moderate $R^2$ values (0.64–0.72) achieved by both models indicate that substantial further improvements would likely require richer feature sets, such as incorporating meteorological

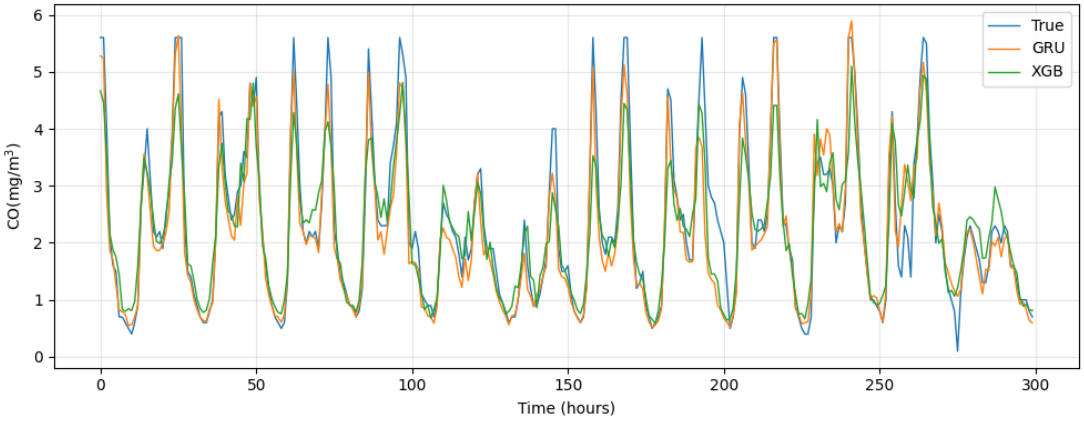


Figure 6.6: Sample predictions for air quality dataset using 24-hour historical context window.

conditions, traffic patterns, or spatial data from nearby monitoring stations, rather than purely architectural refinements.

### *6.6.2 Bike Sharing Dataset*

*Performance Metrics*

We evaluated GRU and XGBoost models for hourly bike sharing demand prediction using the Capital Bikeshare dataset with varying amounts of historical context (3 to 96 hours). Table 6.4 presents the complete performance evaluation across all tested window sizes. All models perform one-step-ahead prediction (predicting hour t+1 given hours t-L+1 to t). We report Root Mean Squared Log Error (RMSLE) in addition to standard metrics, as it is the conventional evaluation metric for bike sharing demand forecasting benchmarks.

GRU consistently outperformed XGBoost across all tested window sizes, achieving MAE values ranging from 85.22 to 92.65 bikes compared to XGBoost's 95.40 to 99.56 bikes. The relative improvement ranged from 5.4% at the 96-hour window to 11.8% at the 72-hour window, with an average advantage of 8.5% across all windows. This represents a more modest advantage compared to the air quality domain (10.1% average), suggesting that bike sharing's strong periodic patterns are more amenable to tabular representations.

Both models achieved moderate explanatory power with $R^2$ scores between 0.58 and 0.68, reflecting the challenging nature of capturing the high variability in urban bike demand.

The optimal performance for GRU occurred at the 72-hour window (MAE: 85.22, RMSE: 123.65, $R^2$: 0.684), while XGBoost performed best at the longest 96-hour window (MAE: 95.40, RMSE: 136.45, $R^2$: 0.615). Unlike the air quality dataset where performance degraded at longer windows, both models here benefited from extended historical context, with GRU improving 8.0% from the 3-hour to 72-hour window and XGBoost improving 2.8% from the 3-hour to 96-hour window.

Table 6.4: Bike sharing demand forecasting performance across historical context window sizes

| **Window** | **GRU** | | | | **XGBoost** | | | | **Improv.** |
|---|---|---|---|---|---|---|---|---|---|
| (hours) | MAE | RMSE | RMSLE | $R^2$ | MAE | RMSE | RMSLE | $R^2$ | (%) |
| 3 | 92.65 | 132.18 | 0.533 | 0.641 | 98.15 | 139.82 | 0.559 | 0.598 | 5.6 |
| 6 | 89.15 | 126.35 | 0.525 | 0.671 | 98.55 | 140.56 | 0.556 | 0.593 | 9.5 |
| 12 | 90.84 | 129.98 | 0.560 | 0.652 | 99.56 | 142.11 | 0.573 | 0.584 | 8.8 |
| 24 | 91.14 | 131.07 | 0.579 | 0.646 | 98.04 | 139.81 | 0.556 | 0.597 | 7.0 |
| 48 | 85.91 | 124.56 | 0.527 | 0.679 | 96.58 | 137.86 | 0.559 | 0.607 | 11.1 |
| 72 | **85.22** | **123.65** | **0.542** | **0.684** | 96.61 | 138.46 | 0.578 | 0.603 | **11.8** |
| 96 | 90.29 | 127.47 | 0.570 | 0.664 | 95.40 | 136.45 | 0.572 | 0.615 | 5.4 |
| **Mean** | **89.32** | **127.89** | **0.548** | **0.662** | **97.55** | **139.29** | **0.565** | **0.600** | **8.5** |

*Note:* MAE and RMSE reported in bike counts (original scale after inverse log transformation). RMSLE (Root Mean Squared Log Error) included as standard metric for bike sharing benchmarks. Improvement calculated as (XGB_MAE - GRU_MAE) / XGB_MAE × 100%. Best GRU performance highlighted in bold.

*Analysis*

**Window Size Effects** Figure 6.7 shows model performance across historical context window sizes. Both models demonstrated a clear preference for longer lookback windows,

with performance generally improving from 3 to 72-96 hours. GRU achieved its optimal performance at the 72-hour window, capturing three complete daily cycles of bike demand patterns. This contrasts with the air quality dataset, where the 24-hour (single-cycle) window was optimal, suggesting that bike sharing demand benefits from observing multiple days of usage history to capture weekly patterns and day-to-day variations in commuter behavior.

The improvement trajectory reveals interesting differences between architectures. GRU's MAE decreased consistently from 92.65 bikes at 3 hours to 85.22 bikes at 72 hours (8.0% improvement), before slightly degrading to 90.29 bikes at 96 hours. XGBoost showed more gradual improvement, declining from 98.15 bikes at 3 hours to 95.40 bikes at 96 hours (2.8% improvement), with relatively flat performance across the 48-96 hour range. This pattern indicates that GRU leverages extended temporal context more effectively, while XGBoost experiences diminishing returns as the lag feature space expands beyond approximately 48 hours.

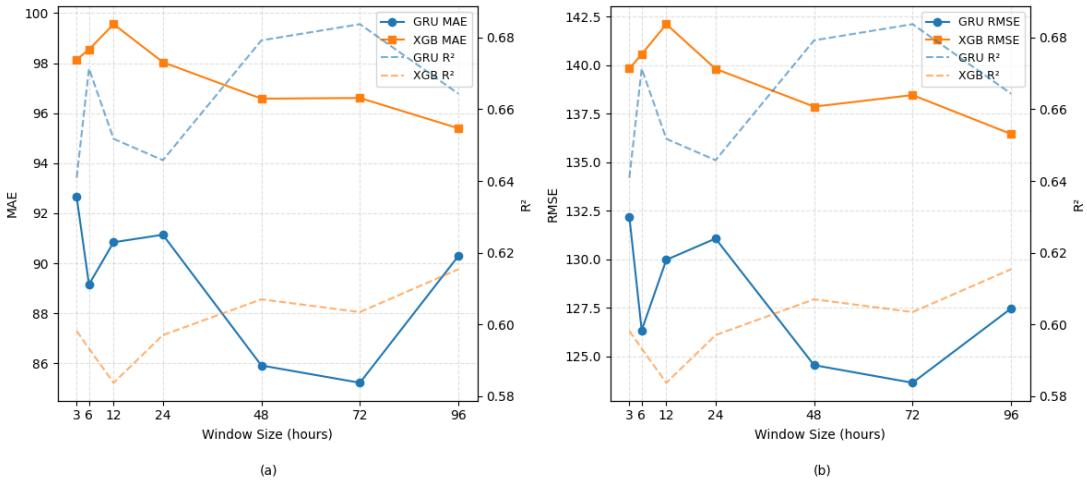


Figure 6.7: Bike sharing demand forecasting performance across historical context window sizes. (a) MAE and $R^2$ metrics. (b) RMSE and $R^2$ metrics. Solid lines show error metrics (left axis), dashed lines show $R^2$ (right axis).

**Sequential vs. Tabular Processing** The 8.5% average GRU advantage, while consistent across all windows, is smaller than the 10.1% advantage observed in air quality forecasting. This narrower gap reflects the fundamental characteristics of bike sharing demand: highly regular diurnal and weekly patterns that tree-based models can effectively capture through lag-based features. The strong 24-hour and 168-hour (weekly) periodicities in urban bike usage create clear structural patterns that XGBoost's tree partitioning can identify without requiring sequential processing.

However, GRU's advantage becomes more apparent at intermediate-to-long windows (48-72 hours), achieving 11.1-11.8% improvements. At these scales, the sequential model can integrate information across multiple daily cycles, learning how demand evolves from weekday to weekend patterns or how weather trends over 2-3 days affect usage. XGBoost, treating each lag independently, must rely on explicit interactions learned through tree splits, which becomes increasingly difficult as the feature space grows with window size.

The $R^2$ scores reveal an interesting pattern: XGBoost achieves lower variance-explained (0.58-0.62) than GRU (0.64-0.68) despite having access to identical information. This suggests GRU's recurrent processing captures more of the systematic variation in bike demand, particularly the complex interactions between time-of-day, day-of-week, and recent usage trends that define urban mobility patterns.

**Peak Underprediction Pattern** An obvious limitation shared by both models is the systematic underprediction of peak demand periods. Figure 6.8 illustrates this behavior over a representative 500-hour test period using the optimal 72-hour window. While both models successfully capture the regular diurnal pattern with clear morning and evening commute peaks and overnight valleys, they consistently underestimate the magnitude of high-demand periods. True demand peaks frequently reach 800-900 bikes, yet both GRU and XGBoost predictions plateau around 400-600 bikes, representing underprediction of 30-50% during peak hours.

This pattern persists throughout the time series and appears consistent across different demand levels and time periods. Notably, both models perform well during low-demand periods (overnight hours with 0-100 bikes), accurately tracking the valleys in the diurnal

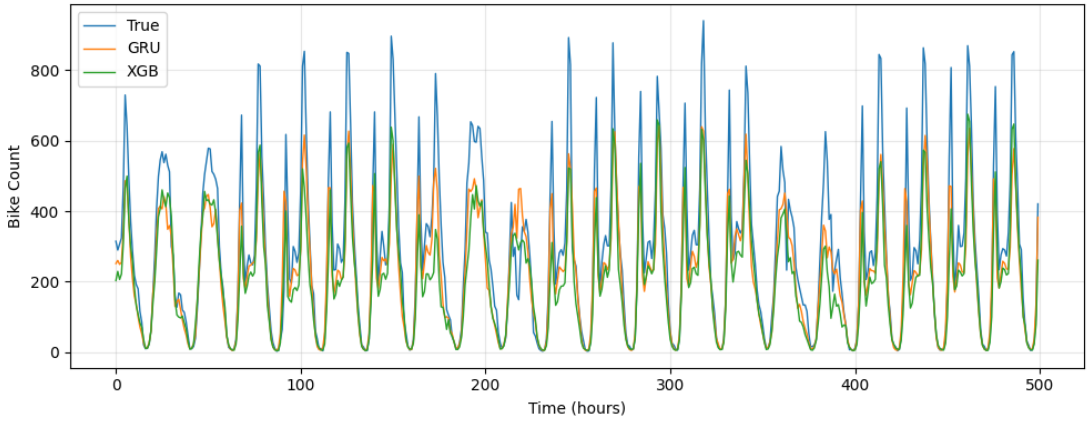


Figure 6.8: Sample predictions for bike sharing dataset using 72-hour historical context window.

cycle. The asymmetric error distribution—accurate low predictions but conservative high predictions—suggests both architectures learn to avoid large errors by regressing toward the mean, particularly during the volatile peak periods where prediction uncertainty is highest.

Several factors contribute to this systematic bias. First, the extreme peak-to-valley ratio (16-18x difference between overnight lows and rush-hour peaks) creates a challenging prediction space where models trained on MSE-based losses favor conservative predictions that minimize squared errors. Second, despite including weather conditions (temperature, humidity, weather situation), peak demand remains difficult to predict, suggesting that either (a) the categorical weather encoding lacks sufficient granularity to capture the nuanced conditions that drive extreme peaks, or (b) other unobserved factors such as special events (holidays, sporting events, festivals, public transit disruptions) introduce irreducible uncertainty that models handle by dampening peak predictions. Third, the sharp transitions between low and high demand (demand can triple within 1-2 hours during morning rush) are inherently difficult to predict, especially when constrained to produce non-negative outputs.

**Prediction Quality and Error Structure** Figure 6.9 provides additional insight into model behavior through a scatter plot of predicted versus actual demand. The systematic deviation from the y=x diagonal (perfect prediction line) becomes increasingly pronounced

above 400 bikes, confirming that peak underprediction is not isolated to specific time periods but represents a fundamental model characteristic. Both GRU (orange) and XGBoost (blue) fitted lines show slopes substantially less than unity, indicating compression of the prediction range relative to true demand variability.

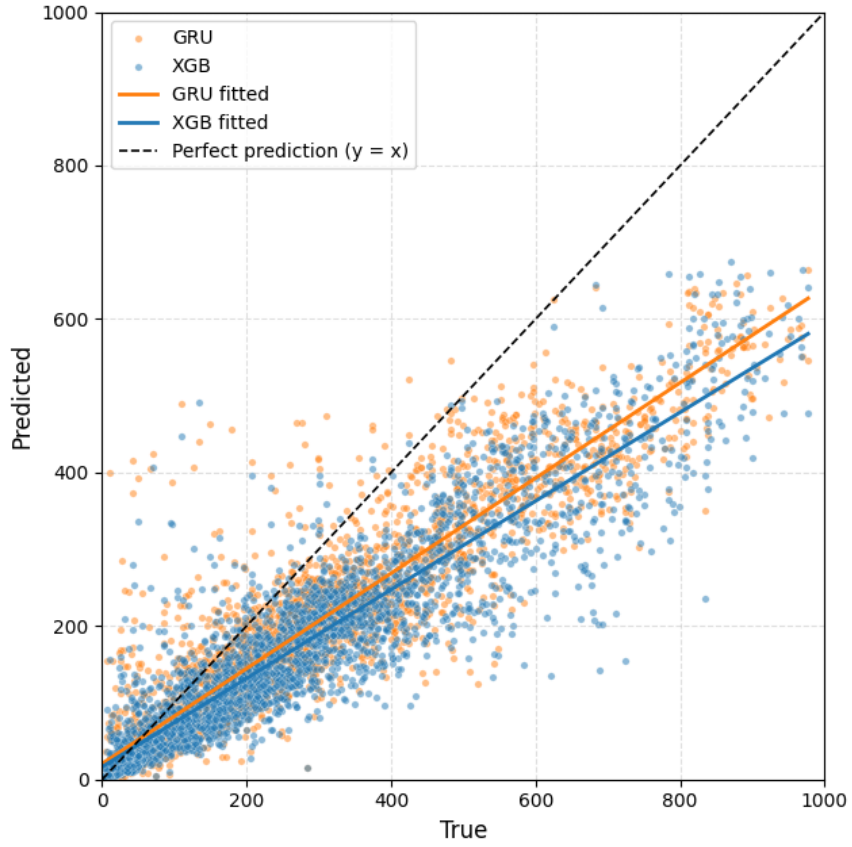


Figure 6.9: Predicted vs. actual bike demand for 72-hour historical context window.

Importantly, GRU's fitted line lies closer to the ideal y=x diagonal than XGBoost's, quantifying the 8.5% MAE advantage observed in Table 6.4. While both models exhibit similar qualitative behavior—accurate at low/moderate demand, conservative at peaks—GRU produces predictions that extend slightly higher into the peak range, capturing more of the true demand variability. This difference, though modest, represents a meaningful improvement for operational planning where underestimating peak demand by 30% versus 40%

translates directly to service quality through bike availability.

The scatter plot also reveals heteroscedasticity in prediction errors: variance increases substantially with true demand levels. At low demand (<200 bikes), predictions cluster tightly around actual values with errors typically under 50 bikes. At high demand (>600 bikes), prediction errors can exceed 200-300 bikes. This pattern suggests that while both models have learned the general demand structure, they lack the information or representational capacity to confidently predict the specific magnitude of peak periods.

**Practical Implications** For operational bike sharing systems, these results provide several important insights. First, incorporating 72 hours (three full days) of historical demand data optimizes prediction accuracy for one-hour-ahead forecasts, achieving 11.8% better performance than tree-based alternatives. This multi-day window captures both daily patterns and day-to-day variations (weekday vs. weekend, weather trends), providing essential context for short-term demand forecasting used in rebalancing operations.

Second, the systematic peak underprediction revealed in Figures 6.8 and 6.9 has direct operational implications. Rebalancing systems that rely solely on predicted demand will systematically underestimate bike requirements during rush hours, potentially leading to stock-outs at popular stations. Operators should consider applying empirically-derived correction factors (e.g., multiplying predicted peak demand by 1.3-1.5x) or implementing conservative safety buffers that increase with predicted demand levels.

Third, the moderate $R^2$ values (0.58-0.68) and substantial peak prediction errors indicate limitations beyond architectural choice. Despite incorporating weather conditions (temperature, humidity, weather situation categories) and temporal features (hour, day, season, holidays), both models struggle with peak periods. Potential improvements could include: (1) more granular weather information (e.g., real-time precipitation intensity, wind gusts, "feels-like" temperature that better captures cycling comfort), (2) special event indicators (sporting events, concerts, festivals that drive non-routine demand spikes), (3) spatial features capturing station-specific characteristics and neighborhood effects, and (4) external signals such as public transit disruptions. Our results establish that GRU provides consistent advantages over tree-based approaches even with standard weather features, but

both architectures face fundamental challenges in predicting volatile peak demand driven by factors beyond routine patterns.

### *6.6.3 Rossmann Store Sales Dataset*

*Performance Metrics*

We evaluated GRU and XGBoost models for daily sales prediction across 100 Rossmann drugstores with varying amounts of historical context (7 to 120 days). Table 6.5 presents the complete performance evaluation across all tested window sizes. All models perform one-step-ahead prediction (predicting day t+1 given days t-L+1 to t). We include a naive baseline using Last Week Same Day (LWSD) persistence forecasting, predicting that tomorrow's sales will equal the same day last week, to contextualize model performance relative to simple heuristics commonly used in retail operations.

GRU substantially outperformed both XGBoost and the baseline across all tested window sizes, achieving MAE values ranging from 984 to 1,160 EUR compared to XGBoost's 1,166 to 1,726 EUR and the baseline's consistent 1,998 EUR. Relative to XGBoost, GRU's advantage ranged from 1.5% at the 14-day window to 36.2% at the 120-day window, with an average improvement of 18.8%. Both neural and tree-based models delivered substantial improvements over the naive baseline, reducing MAE by 40-51% (GRU) and 14-42% (XGBoost), demonstrating that both architectures successfully learned temporal sales patterns beyond simple weekly periodicity.

The optimal performance for GRU occurred at the 91-day window (MAE: 984, RMSPE: 18.5%, $R^2$: 0.759), capturing approximately three months of historical sales patterns. This window provides sufficient context to learn both short-term promotional effects and longer-term seasonal trends while avoiding the performance degradation observed at very long windows. XGBoost achieved its best MAE at the 91-day window as well (1,166 EUR), though notably, its highest $R^2$ occurred at the 14-day window (0.793), suggesting different window sizes optimize for different aspects of prediction quality. The substantial test set size (16,433 daily sales observations across 100 stores) provides strong statistical evidence for the reliability of these performance differences. The GRU's optimal RMSPE of 18.5%

represents substantial improvement over the 39.3% baseline, achieving less than half the naive forecasting error and demonstrating the effectiveness of sequential architectures for retail sales prediction.

Table 6.5: Rossmann store sales forecasting performance across historical context window sizes

| **Window** | $n_{test}$ | **GRU** | | | **XGBoost** | | | **Baseline (LWSD)** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| (days) | | MAE | RMSPE | $R^2$ | MAE | RMSPE | $R^2$ | MAE | RMSPE | $R^2$ |
| 7 | 16,433 | 1,072 | 19.5% | 0.736 | 1,436 | 25.4% | 0.703 | 1,998 | 39.3% | 0.472 |
| 14 | 16,433 | 1,160 | 24.5% | 0.738 | 1,177 | 22.1% | 0.793 | 1,998 | 39.3% | 0.472 |
| 28 | 16,433 | 1,048 | 19.7% | 0.746 | 1,292 | 23.4% | 0.757 | 1,998 | 39.3% | 0.472 |
| 56 | 16,433 | 1,063 | 19.0% | 0.740 | 1,297 | 23.4% | 0.753 | 1,998 | 39.3% | 0.472 |
| 84 | 16,433 | 1,119 | 19.4% | 0.716 | 1,197 | 22.3% | 0.782 | 1,998 | 39.3% | 0.472 |
| 91 | 16,433 | **984** | **18.5%** | **0.759** | 1,166 | 21.8% | 0.792 | 1,998 | 39.3% | 0.472 |
| 120 | 16,433 | 1,102 | 20.7% | 0.724 | 1,726 | 30.3% | 0.588 | 1,998 | 39.3% | 0.472 |
| **Mean** | **16,433** | **1,078** | **20.2%** | **0.737** | **1,327** | **24.1%** | **0.724** | **1,998** | **39.3%** | **0.472** |

*Note:* MAE reported in EUR (original scale after inverse log transformation). RMSPE: Root Mean Squared Percentage Error. All models evaluated on 100 stores with ≥85% data completeness. Best GRU performance and XGBoost's best $R^2$ highlighted in bold.

*Analysis*

**Window Size Effects** Figure 6.10 illustrates model performance across historical context window sizes. Both GRU and XGBoost demonstrate complex, non-monotonic relationships between window size and prediction accuracy, contrasting sharply with the simpler patterns observed in air quality and bike sharing datasets. GRU performance improves from the 7-day window (MAE: 1,072) to reach its optimum at 91 days (MAE: 984), representing an 8.2% improvement. However, extending the window further to 120 days results in modest degradation to 1,102 EUR, suggesting diminishing returns from excessively long historical

context.

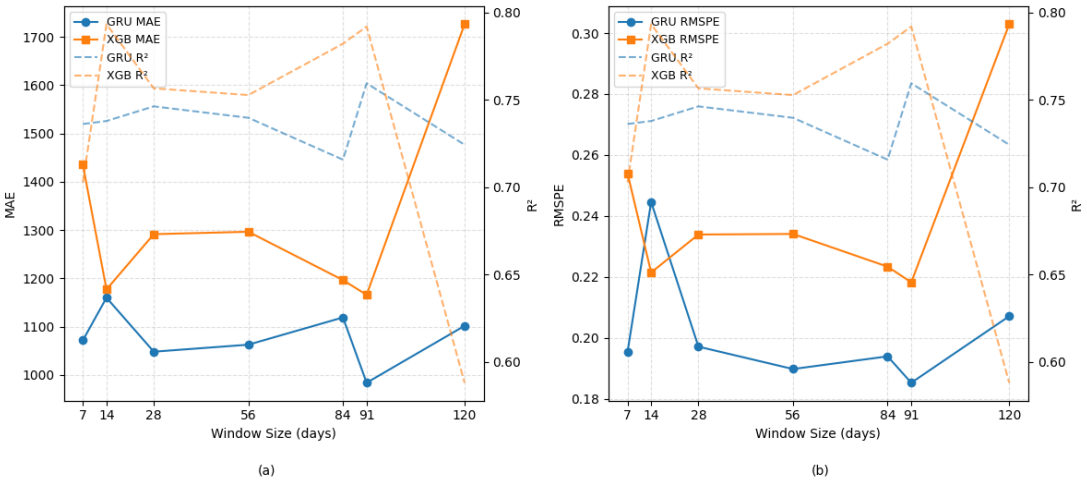


Figure 6.10: Rossmann sales forecasting performance across historical context window sizes. (a) MAE and $R^2$ metrics. (b) RMSPE and $R^2$ metrics. The 14-day window represents an anomaly where XGBoost achieves its highest $R^2$ despite moderate MAE.

XGBoost exhibits even greater window size sensitivity. Performance varies substantially across the 7-91 day range, with MAE fluctuating between 1,166 and 1,436 EUR. The model achieves its lowest MAE at 91 days but paradoxically attains its highest $R^2$ (0.793) at the much shorter 14-day window, a pattern suggesting that variance explanation and absolute error minimization optimize at different scales for tree-based models. Most notably, XGBoost experiences dramatic performance collapse at the 120-day window, with MAE jumping 48% to 1,726 EUR and $R^2$ plummeting from 0.792 (at 91 days) to 0.588. This failure mode demonstrates a fundamental limitation of lag-based feature representations when the feature space becomes excessively large.

The 91-day optimal window for GRU aligns well with retail forecasting requirements. Three months of historical data captures quarterly seasonal patterns (e.g., summer vs. winter sales trends), promotional campaign cycles, and sufficient examples of each day-of-week under varying conditions. This window is substantially longer than the 24-72 hour optima observed in air quality and bike sharing, reflecting the different temporal scales of retail sales dynamics where weekly and monthly patterns dominate rather than purely

diurnal cycles.

**Sequential vs. Tabular Processing** The 18.8% average GRU advantage over XGBoost represents the largest improvement observed across our three datasets, exceeding both air quality (10.1%) and bike sharing (8.5%) domains. This superior performance occurs despite retail sales being a domain where tree-based models traditionally excel, suggesting that the per-store temporal modeling approach employed here plays particularly well to GRU's strengths. The recurrent architecture's ability to maintain hidden state across sequences proves valuable for tracking store-specific trends, promotional effects, and seasonal progressions that unfold over weeks to months.

However, the advantage varies dramatically with window size. At the 14-day window, GRU and XGBoost perform nearly identically in terms of MAE (1,160 vs. 1,177 EUR, a mere 1.5% difference), with XGBoost actually achieving superior variance explanation ($R^2$: 0.793 vs. 0.738). This suggests that for relatively short forecasting contexts, tree-based partitioning of lagged features captures the relevant patterns adequately. The GRU advantage emerges most strongly at longer windows: 25.3% at 7 days, 18.8% at 28 days, and a massive 36.2% at 120 days. This pattern indicates that sequential processing becomes increasingly valuable as the temporal context extends, allowing GRU to selectively attend to relevant historical periods while XGBoost struggles with the expanding lag feature space.

The incorporation of store embeddings provides GRU with a critical architectural advantage for this multi-store forecasting task. Rather than relying on one-hot encoded store identities or hand-crafted store features, the model learns compact representations (4-16 dimensions, tuned via hyperparameter optimization) that capture store-specific characteristics through end-to-end training. These embeddings allow the model to rapidly adapt its predictions based on which store is being forecast, effectively learning a shared temporal model with store-specific adjustments. XGBoost, using one-hot encoded store identities, must learn separate decision rules for each store through tree splits, a less efficient approach that becomes increasingly difficult as the number of stores grows.

**The 120-Day Collapse** XGBoost's dramatic performance degradation at the 120-day window warrants detailed examination. The MAE increases from 1,166 EUR (91 days) to 1,726 EUR (120 days)—a 48% degradation that brings performance closer to the naive baseline (1,998 EUR) than to the model's own performance at adjacent window sizes. Simultaneously, $R^2$ collapses from 0.792 to 0.588, indicating the model explains barely more than half the variance in sales, down from nearly 80% at shorter windows. This catastrophic failure reveals a systematic limitation of lag-based tree models rather than a dataset-specific anomaly.

The mechanism underlying this collapse likely involves feature space explosion combined with pattern dilution. At 120 days, XGBoost must process 120 lagged values for each continuous feature (sales, temperature, humidity, etc.), resulting in a feature space with hundreds of dimensions. Tree-based models split on individual features independently, meaning the model must discover which specific lags (or combinations thereof) are predictive through recursive partitioning. With 120 lags, many are likely irrelevant or redundant, injecting noise that degrades tree quality. Additionally, the optimal trees grown on training data may overfit to spurious patterns in the extended lag space, failing to generalize to test data. The regularization hyperparameters (tuned via Optuna) apparently cannot prevent this degradation, as evidenced by strong performance at 91 days collapsing just 29 days later.

In contrast, GRU's performance at 120 days (MAE: 1,102, $R^2$: 0.724) remains reasonable, representing only 12% degradation from its 91-day optimum. The recurrent architecture processes temporal sequences through hidden state updates rather than independent lag features, allowing it to learn which historical information to retain and which to discard through gating mechanisms. The attention mechanism over the competency window provides additional capability to focus on relevant time periods, preventing the dilution effect that plagues fixed-lag representations. This architectural robustness to extended context represents a key advantage for scenarios where the optimal temporal window is unknown or varies across different prediction contexts.

**Store Embedding Analysis** To understand what the GRU learned through its store embeddings, we analyzed the 12-dimensional embedding space from the optimal 91-day window

model. Figure 6.11 displays the embedding values across all 100 stores. The heatmap reveals clear structural patterns rather than random representations: certain dimensions (notably D3, D11, and D12) show particularly strong positive or negative activations for specific store groups, suggesting they encode interpretable characteristics such as sales volume tier, volatility, or promotional sensitivity.

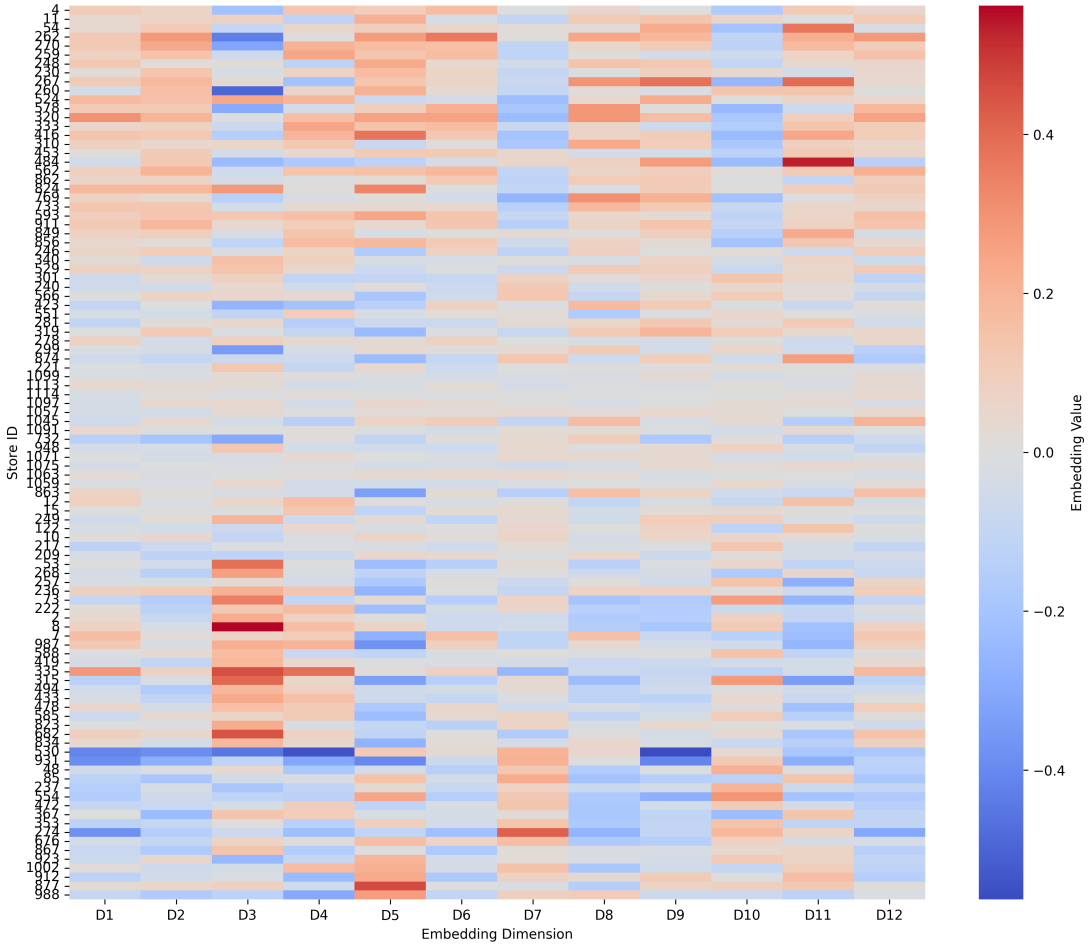


Figure 6.11: Heatmap of learned 12-dimensional store embeddings for 100 Rossmann stores. Each row represents one store's complete embedding vector.

Principal component analysis (Figure 6.12) reveals that despite the 12-dimensional embedding space, the effective dimensionality is substantially lower: four components capture approximately 90% of variance, and six components approach 95%. This dimensional redun-

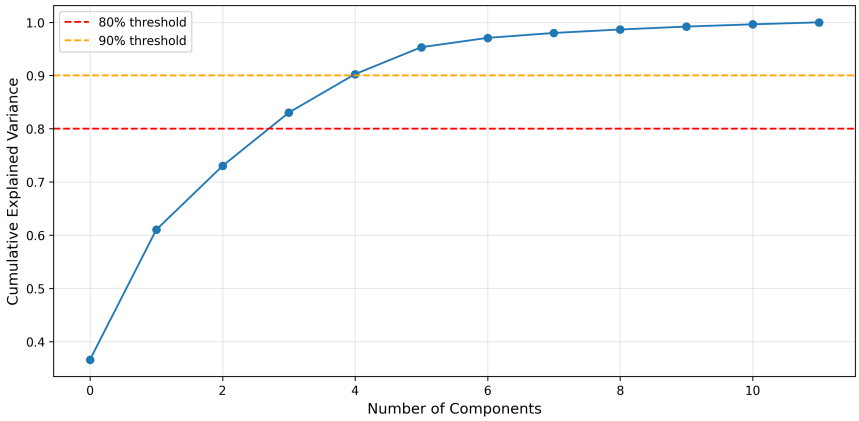


Figure 6.12: Cumulative variance explained by principal components of store embeddings.

dancy indicates the embeddings represent stores along a compact manifold rather than utilizing the full 12-dimensional space. The combination of structured patterns in the heatmap and low intrinsic dimensionality provides strong evidence that the model discovered meaningful store-level features during training rather than simply memorizing individual store identities.

To validate that embedding similarity corresponds to actual sales pattern similarity, we examined specific store pairs from opposite ends of the distance spectrum (Figure 6.13). Stores 676 and 948, with small Euclidean embedding distance (0.40), exhibit highly correlated sales patterns (r = 0.960). Their time series track nearly identically, with synchronized weekly cycles, comparable sales volumes (8,000-14,000 EUR range), and coordinated responses to promotional periods. The scatter plot of Store 948 sales versus Store 676 sales clusters tightly around the identity line, confirming their behavioral similarity.

Conversely, stores 335 and 530, with a large embedding distance (1.74), show fundamentally different characteristics. Store 335 operates at high volume with substantial volatility (15,000-25,000 EUR daily sales), displaying large amplitude weekly cycles and pronounced promotional spikes. Store 530 maintains consistently low, stable sales (3,000-8,000 EUR), with much smaller absolute variation and muted response to weekly patterns. Their scatter plot shows no coherent relationship (r = -0.062), with Store 530's sales remaining flat

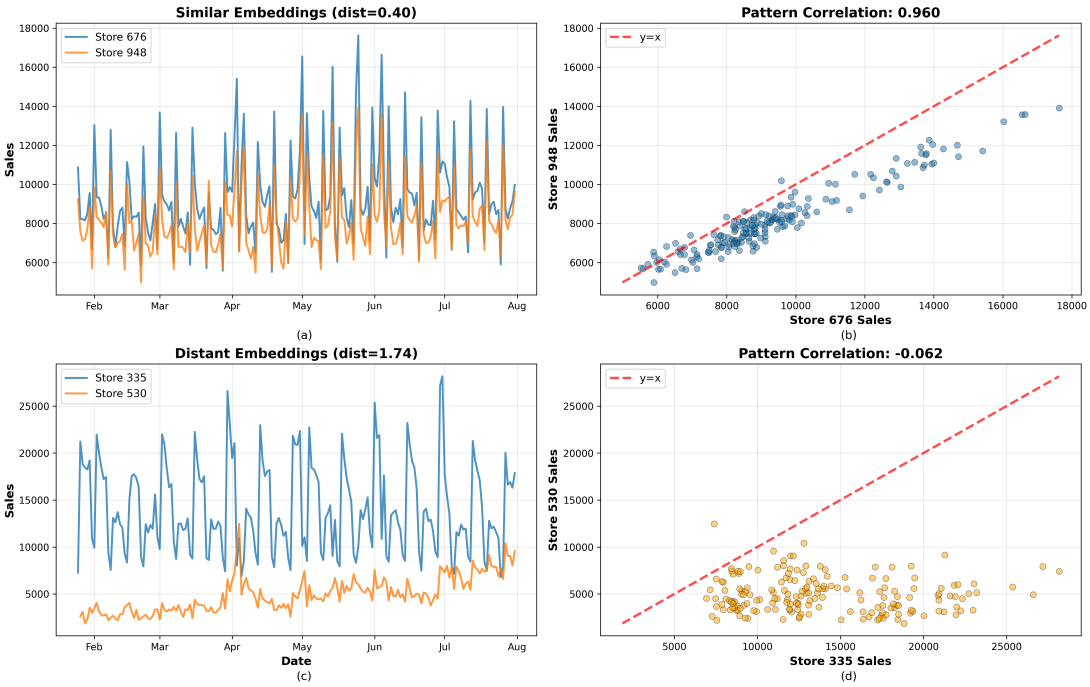


Figure 6.13: Validation of learned store embeddings through sales pattern similarity. (a) Time series for stores with similar embeddings (Euclidean distance = 0.40). (b) Scatter plot confirms strong correlation (r = 0.960) between similar stores. (c) Stores with distant embeddings (distance = 1.74). (d) Scatter plot reveals negligible correlation (r = -0.062) between distant stores.

across Store 335's full range. This demonstrates that Euclidean distance in embedding space meaningfully corresponds to similarity in actual sales behavior, validating the embeddings' utility for capturing store-level characteristics that inform forecasting.

**Practical Implications** For operational retail forecasting systems, these results provide several valuable insights. First, incorporating 91 days (approximately three months) of historical sales data optimizes one-day-ahead prediction accuracy, achieving 50.8% improvement over naive baseline and 15.6% improvement over tree-based alternatives. This window captures quarterly seasonal patterns and sufficient examples of promotional effects while avoiding the performance degradation that arises with excessively long contexts. Operational systems should prioritize this medium-term historical window rather than attempting to incorporate all available data, which our results show can actively harm tree-based model performance.

Second, the learned store embeddings provide a valuable byproduct of GRU training that could inform business operations beyond forecasting. Stores with similar embeddings (small Euclidean distance) exhibit comparable sales patterns and may benefit from coordinated inventory management, shared promotional strategies, or similar staffing models. The compact 4-5 dimensional structure underlying the embeddings (as revealed by PCA) suggests a small number of fundamental store characteristics drive most sales variation, potentially corresponding to factors such as location type (urban vs. suburban), customer demographics, competitive environment, or store size. Understanding these latent factors could guide strategic decisions about store clustering, regional management structure, or targeted interventions for underperforming locations.

Third, the dramatic failure of tree-based models at 120 days (48% performance degradation) highlights risks in operational deployment. Systems that automatically adjust their lookback windows or aggregate features across varying time scales could inadvertently trigger catastrophic failures if lag counts exceed model capacity. This argues for either: (1) constraining tree-based models to shorter, validated windows ($\leq$91 days for this dataset), or (2) adopting recurrent architectures that demonstrate robustness across extended temporal contexts. The GRU's slight degradation at 120 days (only 12% worse than optimal)

compared to XGBoost's collapse suggests sequential models provide safer operational characteristics for production systems where context window requirements may evolve.

Finally, both models' substantial improvements over the 39.3% RMSPE baseline demonstrate that even minimal feature engineering (temporal encodings, competition information, promotional indicators) can improve sales forecasting when paired with appropriate architectures. Organizations currently relying on naive heuristics like "last week same day" could realize 40-50% error reductions through model-based forecasting, translating directly to improved inventory positioning, labor scheduling, and customer service levels. The consistency of both models' performance across 16,433 test predictions provides strong evidence that these improvements could be attainable in operational deployment.

### *6.6.4 Cross-Dataset Synthesis*

The preceding analyses demonstrated that GRU consistently outperforms XGBoost across three diverse forecasting domains: environmental monitoring (Air Quality), transportation demand (Bike Sharing), and retail sales (Rossmann). However, the magnitude of this advantage varied substantially (from 8.5% in Bike Sharing to 18.8% in Rossmann), and manifested differently depending on dataset characteristics and window size choices. This section synthesizes findings across all three datasets to identify the underlying principles governing when and why sequential architectures provide meaningful improvements over tree-based approaches for time series forecasting with minimal feature engineering. Table 6.6 summarizes key performance characteristics and dataset properties, revealing systematic relationships between domain characteristics, optimal window sizes, and relative model performance.

**Consistency of GRU Superiority** Across all three domains, temporal resolutions, and window sizes tested, GRU demonstrated superior MAE across all 20 window configurations (6 air quality + 7 bike sharing + 7 Rossmann)—though the 14-day Rossmann window showed the smallest margin (1.5%), with XGBoost achieving higher $R^2$ at that window despite higher absolute error. This consistency is remarkable given the substantial differences in domain characteristics: air quality involves smooth atmospheric pollutant dynamics with moderate diurnal variation; bike sharing exhibits extreme demand volatility with very

Table 6.6: Cross-dataset performance summary and characteristics

| **Characteristic** | **Air Quality** | **Bike Sharing** | **Rossmann** |
|---|---|---|---|
| **Domain** | Environmental | Transportation | Retail |
| **Temporal Resolution** | Hourly | Hourly | Daily |
| **Test Samples** | 1,848 | ∼1,800 | 16,433 |
| **GRU Best Window** | 24 hours | 72 hours | 91 days |
| **GRU Best MAE** | 0.457 mg/m³ | 85.22 bikes | 984 EUR |
| **GRU Best $R^2$** | 0.720 | 0.684 | 0.759 |
| **XGB Best Window** | 6 hours | 96 hours | 91 days |
| **XGB Best MAE** | 0.527 mg/m³ | 95.40 bikes | 1,166 EUR |
| **XGB Best $R^2$** | 0.673 | 0.615 | 0.793* |
| **Avg. GRU Advantage** | 10.1% | 8.5% | 18.8% |
| **Advantage Range** | 6.6–13.6% | 5.4–11.8% | 1.5–36.2% |
| **Signal Smoothness** | High | Medium | Medium |
| **Peak:Valley Ratio** | ∼8x | 16–18x | Varies (3–8x) |
| **Periodicity Strength** | Strong (24h) | Very Strong (24h + 168h) | Medium (7d + seasonal) |
| **Key Finding** | Stable across windows | Peak under-prediction | XGB collapse at 120d |

Note: MAE units vary by dataset (mg/m³, bike count, EUR). Advantage calculated as mean improvement across all windows. *XGB's best $R^2$ occurs at different window (14d) than best MAE, indicating optimization trade-offs. Peak:valley ratios represent typical cyclical variation ranges.

strong periodic patterns; and retail sales combine multi-store heterogeneity with promotional effects and seasonal trends.

The average performance advantage ranges from 8.5% (Bike Sharing) to 18.8% (Rossmann), with air quality falling in between at 10.1%. These differences are not merely statistical artifacts—the large test sets (1,800–16,433 samples) and consistent patterns across multiple window sizes provide strong evidence that domain characteristics systematically influence the relative effectiveness of sequential versus tree-based architectures. Importantly, even the smallest advantage (8.5%) represents meaningful improvement: for a bike sharing system with 1,000 daily predictions, this translates to 85 fewer erroneous forecasts, potentially preventing stock-outs at dozens of stations during peak demand periods.

The universality of GRU's advantage suggests that sequential processing provides fundamental benefits for temporal prediction that tree-based approaches cannot fully replicate through lag-based feature representations. Even in bike sharing, where strong 24-hour and weekly periodicities might favor XGBoost's ability to learn fixed lag patterns, the recurrent architecture still demonstrates consistent superiority by better capturing the interactions between different temporal scales and contextual factors. This finding has important implications for practitioners: when choosing between architectures for time series forecasting with minimal feature engineering, sequential models should be the default choice unless specific constraints (interpretability requirements, computational limits, extremely short sequences) favor tree-based alternatives.

**Dataset Characteristics and Performance Patterns** The magnitude of GRU's advantage correlates systematically with specific dataset characteristics, revealing which temporal prediction scenarios benefit most from sequential processing. The largest advantage appears in the Rossmann retail dataset (18.8% average), which combines three key characteristics: (1) long temporal dependencies spanning weeks to months rather than hours to days, (2) multi-entity structure requiring the model to learn store-specific patterns while sharing knowledge across stores, and (3) complex interactions between promotions, holidays, and seasonal trends that unfold over extended time periods. GRU's architecture, with learned store embeddings, attention mechanisms over long sequences, and recurrent

hidden states, aligns naturally with these requirements.

The intermediate advantage in air quality forecasting (10.1%) corresponds to a domain with smooth, continuous signals where temporal ordering matters but periodicities are relatively simple. Atmospheric CO concentrations evolve gradually throughout the day, driven by accumulation during traffic peaks and dispersion during low-activity periods. This smooth temporal structure favors sequential processing, which can track the continuous evolution of concentrations, over lag-based features that treat time-t-24 and time-t-25 as independent variables despite their adjacent positions in the temporal flow. The absence of extreme volatility (peak:valley ratios around 8x) also allows both models to achieve reasonable absolute performance ($R^2$: 0.66–0.72), but GRU's sequential nature provides consistent advantage in tracking the gradual diurnal progression.

The smallest advantage emerges in bike sharing (8.5%), despite this dataset having the largest test set and the clearest periodic patterns. The very strong 24-hour and 168-hour periodicities, with bike demand showing near-identical patterns on each weekday and systematically different weekend patterns, create a scenario where XGBoost's lag-based features can effectively capture much of the temporal structure. A tree model with lags at hours 24, 25, 48, 168, and 169 can learn"if it's Monday morning and yesterday's Monday morning was high demand, predict high" without requiring sequential processing to understand temporal progression. However, even in this favorable scenario for tree-based models, GRU still outperforms by better handling the interactions between time-of-day effects, recent weather trends, and deviation from typical patterns—nuances that require sequential integration of context rather than fixed lag patterns.

An interesting counter-pattern emerges in the bike sharing data: both models systematically underpredict peak demand by 30–50%, regardless of architecture. This shared failure mode indicates some challenges transcending architectural choice and depending instead on fundamental task characteristics: in this case, extreme peak:valley ratios (16–18x), sharp transitions (demand tripling within 1–2 hours), and missing information about special events driving non-routine peaks. The presence of such shared limitations provides important calibration for interpreting the GRU advantages observed elsewhere: sequential processing improves handling of temporal dependencies but cannot overcome fundamental information

deficits or inherent predictability limits of the domain.

**Window Size Selection and Model Robustness** The optimal historical context window varies systematically with the natural temporal scales of each domain, but importantly, both architectures benefit from matching these scales, although they have different sensitivity to window choice. Air quality achieves optimal performance at 24 hours, one complete diurnal cycle that allows models to leverage yesterday's corresponding hour as a strong predictor while incorporating recent trends. Bike sharing prefers longer 72–96 hour windows that capture multiple daily cycles and the beginning of weekly patterns, enabling the model to distinguish weekday versus weekend behavior. Rossmann requires substantially longer 91-day windows (approximately three months) to capture quarterly seasonal patterns, promotional cycles, and sufficient examples of each day-of-week under varying conditions.

However, the two architectures respond very differently to deviation from optimal windows. GRU demonstrates graceful degradation, maintaining reasonable performance even when windows are suboptimal. In air quality, performance varies only 11.6% across the 6–96 hour range; in bike sharing, the range is 8.6% across 3–96 hours. Even at the problematic 120-day window for Rossmann, GRU degrades only 12% from its 91-day optimum. This robustness has significant practical value: operational systems can tolerate some imprecision in window selection without catastrophic consequences, and a single model configuration may suffice for multiple prediction contexts with different natural temporal scales.

XGBoost exhibits much greater window sensitivity, particularly at extremes. Performance fluctuates more across the tested range, and critically, the model experiences catastrophic failure when windows become excessively long. The Rossmann 120-day collapse, where MAE jumps 48% and $R^2$ plummets from 0.79 to 0.59, provides clear evidence of a fundamental limitation in lag-based feature representations. With 120 days of history, the model must process 120 lagged values per feature, resulting in hundreds of dimensions where most lags are likely irrelevant or redundant. Tree-based partitioning cannot efficiently navigate this high-dimensional space, leading to overfitting on training data that fails to generalize. This failure mode represents a critical operational risk: systems that adaptively adjust window sizes or aggregate multi-scale features could inadvertently trigger dramatic

performance degradation.

The window size patterns also reveal architectural trade-offs in computational efficiency versus adaptation capability. XGBoost often performs reasonably at short windows (Air Quality 6 hours, Rossmann 14 days), suggesting that when minimal historical context suffices, tree-based models can achieve competitive accuracy with faster training and inference. Conversely, GRU's advantages emerge most strongly at longer windows where sequential processing can selectively attend to relevant historical periods across extended time series. This suggests a practical decision heuristic: for real-time systems requiring predictions every few seconds based on recent history (e.g., sub-second sensor data), tree-based models may suffice; for applications requiring hourly or daily predictions integrating weeks to months of history, sequential architectures justify their additional computational cost.

**Architectural Advantages and Failure Modes** The consistent GRU superiority across datasets stems from several architectural advantages inherent to recurrent processing of temporal sequences. First, sequential models naturally preserve temporal ordering information that lag-based representations discard. When XGBoost sees features representing time t-23, t-24, and t-25, it treats these as independent dimensions in the feature space; the model can learn that t-24 (yesterday same hour) is predictive, but it cannot inherently understand that t-23 and t-25 bracket t-24 in a continuous temporal flow. GRU processes these sequentially through hidden state updates, maintaining awareness that observations form an ordered progression rather than unordered feature values.

Second, attention mechanisms and gating structures allow selective focus on the most relevant historical context. In the air quality 96-hour window, not all 96 past observations contribute equally to predicting the next hour—yesterday's corresponding hour matters far more than 83 hours ago. GRU's attention weights (with temperature-scaled softmax, $\tau = 0.6$ in our implementation) learn to emphasize relevant positions while discounting irrelevant history. XGBoost achieves similar results only if regularization successfully prunes splits on non-informative lag features, a less direct mechanism that struggles as dimensionality increases. This explains why GRU maintains stable performance at 96-hour windows while XGBoost degrades: the recurrent model explicitly learns what to remember and forget,

whereas the tree model implicitly hopes regularization prevents overfitting to noise.

Third, learned embeddings in multi-entity scenarios (Rossmann stores) provide efficient parameterization of entity-specific patterns. Rather than creating 100 one-hot encoded store features, GRU learns compact 4–16 dimensional embeddings that capture store similarities in latent space. Our analysis showed that stores with similar embeddings exhibit correlated sales patterns (r=0.96), while distant embeddings correspond to fundamentally different stores (r=-0.06). This demonstrates genuine feature learning rather than memorization. XGBoost must learn store-specific patterns through tree splits on one-hot features, essentially building separate subtrees per store—a parameter-inefficient approach that becomes increasingly difficult as the number of entities grows. The store embedding advantage likely explains much of GRU's 18.8% superiority in Rossmann: efficient multi-entity modeling at scale.

However, XGBoost offers counter-balancing advantages in specific scenarios. Tree-based models provide direct interpretability through feature importance scores and decision paths, valuable for domains where understanding why predictions deviate matters as much as accuracy itself (e.g., explaining sales forecast changes to business stakeholders). Training and inference speeds favor XGBoost substantially: our models typically trained 3–5x faster than GRU and produced predictions nearly instantaneously versus GRU's sequential inference that scales with sequence length. For applications requiring real-time predictions on resource-constrained devices (e.g., embedded sensors), these computational differences may dominate accuracy considerations. Additionally, XGBoost naturally handles missing data and mixed feature types (categorical, continuous, ordinal) without preprocessing, whereas neural networks require careful handling of missingness and heterogeneity.

The failure modes observed also inform architectural selection. XGBoost's catastrophic collapse at very long windows represents a hard constraint: these models cannot safely process sequences beyond certain lengths without risking dramatic degradation. GRU avoids this failure but exhibits different weaknesses: the models require careful hyperparameter tuning (learning rates, dropout, architecture size), show sensitivity to initialization and training dynamics, and can be difficult to debug when training dynamics are unstable or predictions are sensitive to initialization. The systematic peak underprediction in bike

sharing affected both architectures equally, suggesting that some limitations stem from loss function design (MAE/MSE encourage conservative predictions) or fundamental task difficulty rather than architectural choice.

**Practical Implications** These findings support several concrete principles for practitioners selecting between sequential and tree-based architectures for time series forecasting. Choose GRU or similar recurrent architectures when: (1) temporal dependencies extend beyond 50–100 timesteps, where lag-based features become unwieldy; (2) the prediction task involves multiple entities (stores, sensors, users) requiring efficient parameterization of entity-specific patterns; (3) signal characteristics involve smooth continuous evolution where temporal ordering matters beyond fixed periodicities; (4) extensive feature engineering is impractical, time-consuming, or domain expertise is limited, as sequential models achieve strong performance with minimal features through automatic temporal feature learning; (5) robustness to varying window sizes is valuable, as sequential models degrade gracefully rather than catastrophically; or (6) the application can tolerate longer training times and sequential inference in exchange for improved accuracy.

Choose XGBoost or similar tree-based models when: (1) temporal dependencies are short (fewer than 50 timesteps), where lag features remain manageable; (2) interpretability is critical, as feature importance and decision paths provide direct insight into prediction logic; (3) computational constraints favor fast training and inference over marginal accuracy gains; (4) the data contains strong periodic patterns with weak inter-period dependencies, where fixed lag features can capture most relevant structure; or (5) operational considerations like handling missing data and mixed feature types favor tree-based robustness. Additionally, tree-based models serve as strong baselines even when ultimately deploying sequential models in production: their fast training enables rapid iteration during exploratory analysis.

For scenarios falling between these extremes, hybrid approaches may be considered: XGBoost for short-term predictions (next 1–3 timesteps) based on recent context, with GRU handling longer-horizon forecasts requiring extended historical integration. Alternatively, XGBoost could generate lag-based features that GRU processes sequentially, combining the feature extraction efficiency of trees with the temporal modeling capability of recur-

rent networks. Our results suggest such hybrid approaches would likely improve upon pure tree-based methods in all three domains studied, though they would add architectural complexity.

Importantly, the consistent 8.5–18.8% improvements demonstrate that architectural choice matters substantially even with minimal feature engineering. Practitioners investing effort in extensive feature creation, domain-specific preprocessing, or ensemble methods should first ensure they have selected an appropriate base architecture. Applying sophisticated feature engineering on top of XGBoost for long-sequence retail forecasting (as in our Rossmann example) may yield smaller gains than simply adopting a sequential architecture with basic features—architecture and feature engineering are complementary but not substitutable. Conversely, the fact that both architectures achieved 40–50% improvements over naive baselines across all domains confirms that either approach, properly tuned, delivers substantial value compared to simple heuristics.

# Chapter 7

# CONCLUSIONS

This dissertation addresses a persistent disconnect between experience-based MRI operations management and the data-driven tools available to extract actionable insights from MRI operational data, through a series of interconnected studies spanning workflow analysis, data integration, predictive modeling, and methodological validation. The chapters, taken together, demonstrate that meaningful gains in MRI operational efficiency and patient access are achievable through careful application of data-driven methods, and that the choice of analytical approach can matter as much as the availability of data.

*Chapter 2* establishes the feasibility of a hub-and-spoke model for deploying MRI expertise across geographically distributed sites. By demonstrating that key radiology functions can be performed remotely by experienced technologists while improving performance metrics such as patient turnaround time and wait time, the study provides an empirical foundation for a care delivery model that has since moved from feasibility study to pilot implementation.

*Chapter 3* extends this operational perspective by examining what scanner log files reveal about workflow inefficiency. Through systematic analysis of modality log file variables, the study identifies actionable sources of idle time and exam duration variability that are invisible to EHR-based analyses alone, highlighting the value of machine-generated operational data as a complement to clinical records.

*Chapter 4* documents the structure, quality, and distributional properties of the EHR data underlying the subsequent predictive work, characterizing exam-level duration variability across anatomy types and patient characteristics. Notably, clinical text embeddings derived from result narratives using BioClinicalBERT are found to carry predictive signal for exam duration, suggesting that unstructured EHR data contains operationally relevant information beyond structured variables alone.

*Chapter 5* introduces a concordance-based validation framework for integrating EHR and modality log file data. By combining fuzzy merging with Bland–Altman analysis and concordance correlation coefficients, the framework raises the concordance correlation coefficient from 0.34 to 0.87, substantially improving data quality for downstream modeling. A Random Forest model trained on the validated dataset outperforms template-based scheduling for 11 of 12 procedure codes, with mean absolute error reductions ranging from 2% to 57%, with the largest gains observed for high-variability procedures that most frequently disrupt clinical schedules.

*Chapter 6* addresses a methodological question with implications beyond MRI operations: under what conditions do sequential neural architectures outperform tree-based models for temporal forecasting, particularly when feature engineering effort is constrained? Evaluated across three established benchmark datasets—Air Quality, Bike Sharing, and Rossmann Store Sales—GRU models consistently outperform XGBoost, with average advantages ranging from 8.5% on the Bike Sharing dataset to 18.8% on Rossmann. These results support the conclusion that learned temporal representations can effectively substitute for manual feature engineering in forecasting tasks characterized by sequential dependencies, and that tree-based methods, while powerful with extensive feature engineering, degrade more sharply at longer temporal windows. This finding provides a principled methodological foundation for future temporal modeling of MRI technologist competency, one of the original research questions motivating this dissertation, when sufficient longitudinal data become available.

This research has several limitations that warrant acknowledgment. First, all clinical studies were conducted at a single academic healthcare system, which may limit generalizability to community settings, alternative scanner vendors, or institutions with different documentation workflows. Second, the competency modeling question that motivated Chapter 6 could not be directly addressed with the available data which lacked the longitudinal depth required to detect meaningful temporal signal; the benchmark validation represents a necessary but not entirely sufficient step toward that goal. Additionally, the scheduling model developed in Chapter 5 has not yet been validated in a live clinical environment, and translating retrospective predictive improvements into measurable operational impact will

require dedicated implementation studies.

Future work could pursue three directions. First, prospective validation and deployment of the scheduling model developed in Chapter 5, potentially integrated at the time of order entry, would allow real-world assessment of effects on scanner utilization, technologist workload, and patient wait times. Second, prediction accuracy for high-variability procedure codes may benefit from finer-grained data collection, particularly for those that currently serve as umbrella terms aggregating clinically distinct exam types; disaggregating these codes or enriching the feature set with additional sources of variability such as clinical indication and imaging sequences could meaningfully improve model performance. Third, the NLP infrastructure developed in Chapter 4 could be extended toward automated protocoling using clinical text from order narratives to recommend or assign imaging protocols, which is a task that currently depends heavily on radiologist time and expertise and represents a natural next application of the text embedding work.

MRI operations sit at the intersection of clinical care, resource management, and patient experience. This dissertation contributes a suite of analytical tools and empirical findings that move that intersection toward greater efficiency, not through any single innovation, but through the cumulative effect of better data, better models, and better understanding of when each approach is warranted.

# BIBLIOGRAPHY

[1] Ronald M Epstein and Edward M Hundert. Defining and assessing professional competence. *Jama*, 287(2):226–235, 2002.

[2] Nadine J Kaslow, Catherine L Grus, Linda F Campbell, Nadya A Fouad, Robert L Hatcher, and Emil R Rodolfa. Competency assessment toolkit for professional psychology. *Training and Education in Professional Psychology*, 3(4S):S27, 2009.

[3] Nadine J Kaslow, Kathi A Borden, Frank L Collins Jr, Linda Forrest, Joyce Illfelder-Kaye, Paul D Nelson, Joseph S Rallo, Melba JT Vasquez, and Mary E Willmuth. Competencies conference: Future directions in education and credentialing in professional psychology. *Journal of Clinical psychology*, 60(7):699–712, 2004.

[4] Anne F Marrelli, Janis Tondora, and Michael A Hoge. Strategies for developing competency models. *Administration and Policy in Mental Health and Mental Health Services Research*, 32(5-6):533–561, 2005.

[5] Neeraj Kak, Bart Burkhalter, and Merri-Ann Cooper. Measuring the competence of healthcare providers. *Operations Research Issue Paper*, 2(1):1–28, 2001.

[6] Roger Watson, Anne Stimpson, Annie Topping, and Davina Porock. Clinical competence assessment in nursing: a systematic review of the literature. *Journal of advanced nursing*, 39(5):421–431, 2002.

[7] Natasha Franklin and Paula Melville. Competency assessment tools: An exploration of the pedagogical issues facing competency assessment for nurses in the clinical environment. *Collegian*, 22(1):25–31, 2015.

[8] J Frank. A history of canmeds-chapter from royal college of physicians of canada 75th anniversary history. *Royal College of Physicians and Surgeons of Canada*, 2004.

[9] Jason R Frank, Linda Snell, Jonathan Sherbino, and Andrée Boucher. Canmeds 2015. *Physician competency framework series I*, 2015.

[10] Accreditation Council for Graduate Medical Education. ACGME common program requirements. Accreditation Council for Graduate Medical Education, 2019.

[11] Irene W Leigh, I Leon Smith, Muriel J Bebeau, James W Lichtenberg, Paul D Nelson, Sanford Portnoy, Nancy J Rubin, and Nadine J Kaslow. Competency assessment models. *Professional Psychology: Research and Practice*, 38(5):463, 2007.

[12] Stephen J Lurie, Christopher J Mooney, and Jeffrey M Lyness. Measurement of the general competencies of the accreditation council for graduate medical education: a systematic review. *Academic Medicine*, 84(3):301–309, 2009.

[13] Lisa J Sundean. Healthcare board competency survey for nurses: Assessing board readiness. *Nursing Economics*, 35(6):295–303, 2017.

[14] Bernadette Mazurek Melnyk, Lynn Gallagher-Ford, Cindy Zellefrow, Sharon Tucker, Bindu Thomas, Loraine T Sinnott, and Alai Tan. The first us study on nurses' evidence-based practice competencies indicates major deficits that threaten healthcare quality, safety, and patient outcomes. *Worldviews on Evidence-Based Nursing*, 15(1):16–25, 2018.

[15] Evelyn E Feliciano, Alfredo Z Feliciano, Jestoni D Maniago, Ferdinand Gonzales, Adelina M Santos, Abdulrhman Albougami, Mehrunnisha Ahmad, and Hadeel Al-Olah. Nurses' competency in saudi arabian healthcare context: A cross-sectional correlational study. *Nursing Open*, 8(5):2773–2783, 2021.

[16] Ross J Scalese, Vivian T Obeso, and S Barry Issenberg. Simulation technology for skills training and competency assessment in medical education. *Journal of general internal medicine*, 23:46–49, 2008.

[17] Florence H Sheehan and R Eugene Zierler. Simulation for competency assessment in vascular and cardiac ultrasound. *Vascular Medicine*, 23(2):172–180, 2018.

[18] Amanda S Keddington and Jill Moore. Simulation as a method of competency assessment among health care providers: a systematic review. *Nursing Education Perspectives*, 40(2):91–94, 2019.

[19] Ana K Stankovic, Barbara J Blond, Suzanne N Coulter, Thomas Long, and Paul F Lindholm. Preanalytic competency assessment: A q-probes study involving 46 health care institutions, 447 blood collectors/phlebotomists, and 2212 individual assessments. *Archives of Pathology & Laboratory Medicine*, 147(3):304–312, 2023.

[20] Catherine J Robbins, Elizabeth H Bradley, and Maryanne Spicer. Developing leadership in healthcare administration: A competency assessment tool. *Journal of Healthcare Management*, 46(3):188–202, 2001.

[21] Earl J Reisdorff, Oliver W Hayes, Brian Reynolds, Keith C Wilkinson, David T Overton, Mary Jo Wagner, Terry Kowalenko, David Portelli, Gregory Walker, and Dale Carlson. General competencies are intrinsic to emergency medicine training: a multicenter study. *Academic emergency medicine*, 10(10):1049–1053, 2003.

[22] Karen J Brasel, Dawn Bragg, Deborah E Simpson, and John A Weigelt. Meeting the accreditation council for graduate medical education competencies using established residency training program assessment tools. *The American journal of surgery*, 188(1):9–12, 2004.

[23] Carey D Chisholm, Laura F Whenmouth, Elizabeth A Daly, William H Cordell, Beverly K Giles, and Edward J Brizendine. An evaluation of emergency medicine resident interaction time with faculty in different teaching venues. *Academic emergency medicine*, 11(2):149–155, 2004.

[24] Philip Shayne, Fiona Gallahue, Stephan Rinnert, Craig L Anderson, Gene Hern, and Eric Katz. Reliability of a core competency checklist assessment in the emergency department: the standardized direct observation assessment tool. *Academic Emergency Medicine*, 13(7):727–732, 2006.

[25] Bruce E Landon, Sharon-Lise T Normand, David Blumenthal, and Jennifer Daley. Physician clinical performance assessment: prospects and barriers. *Jama*, 290(9):1183–1189, 2003.

[26] Dorothy S Lane. Defining competencies and performance indicators for physicians in medical management. *American journal of preventive medicine*, 14(3):229–236, 1998.

[27] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[28] Feng Xie, Han Yuan, Yilin Ning, Marcus Eng Hock Ong, Mengling Feng, Wynne Hsu, Bibhas Chakraborty, and Nan Liu. Deep learning for temporal data representation in electronic health records: A systematic review of challenges and methodologies. *Journal of biomedical informatics*, 126:103980, 2022.

[29] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE journal of biomedical and health informatics*, 22(5):1589–1604, 2017.

[30] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[31] Wei-Hung Weng and Peter Szolovits. Representation learning for electronic health records. *arXiv preprint arXiv:1909.09248*, 2019.

[32] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, pages 301–318. PMLR, 2016.

[33] Jinghe Zhang, Kamran Kowsari, James H Harrison, Jennifer M Lobo, and Laura E Barnes. Patient2vec: A personalized interpretable deep representation of the longitudinal electronic health record. *IEEE Access*, 6:65333–65346, 2018.

[34] Siddharth Biswal, Cao Xiao, Lucas M Glass, Elizabeth Milkovits, and Jimeng Sun. Doctor2vec: Dynamic doctor representation learning for clinical trial recruitment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 557–564, 2020.

[35] David B Larson, Yasmin S Cypel, Howard P Forman, and Jonathan H Sunshine. A comprehensive portrait of teleradiology in radiology practices: results from the american college of radiology's 1999 survey. *American Journal of Roentgenology*, 185(1):24–35, 2005.

[36] TB Hunter. Teleradiology: A brief overview. 2014 march 03 [cited 2020 oct 02]. *The Arizona Telemedicine Program Blog [Internet]. Tucson: University of Arizona Health Sciences [about 1 screen]. Available online: https://telemedicine. arizona. edu/blog/teleradiology-brief-overview*.

[37] MN Lakhoua and F Khanchel. Overview of the methods of modeling and analyzing for the medical framework. *Scientific Research and Essays*, 6(19):3942–3948, 2011.

[38] Laura G Militello, Robert JB Hutton, Rebecca M Pliske, Betsy J Knight, and Gary Klein. Applied cognitive task analysis (acta) methodology. Technical report, 1997.

[39] Richard E Clark, David F Feldon, Jeroen JG Van Merrienboer, Kenneth A Yates, and Sean Early. Cognitive task analysis. In *Handbook of research on educational communications and technology*, pages 577–593. Routledge, 2008.

[40] Aviv Shachak, Michal Hadas-Dayagi, Amitai Ziv, and Shmuel Reis. Primary care physicians' use of an electronic medical record system: a cognitive task analysis. *Journal of general internal medicine*, 24(3):341–348, 2009.

[41] Matthew B Weinger and Jason M Slagle. Task and workload analysis of the clinical performance of anesthesia residents with different levels of experience. *Anesthesiology*, 95:A1145, 2001.

[42] David Fone, Sandra Hollinghurst, Mark Temple, Alison Round, Nathan Lester, Alison Weightman, Katherine Roberts, Edward Coyle, Gwyn Bevan, and Stephen Palmer. Systematic review of the use and value of computer simulation modelling in population health and health care delivery. *Journal of Public Health*, 25(4):325–335, 2003.

[43] ANNA Sciomachen, ELENA Tanfani, ANGELA Testi, et al. Simulation models for optimal schedules of operating theatres. *International Journal of Simulation*, 6(12-13):26–34, 2005.

[44] Jan MH Vissers, Ivo JBF Adan, and Nico P Dellaert. Developing a platform for comparison of hospital admission systems: An illustration. *European Journal of Operational Research*, 180(3):1290–1301, 2007.

[45] Murat M Günal and Michael Pidd. Discrete event simulation for performance modelling in health care: a review of the literature. *Journal of simulation*, 4(1):42–51, 2010.

[46] Renzo Akkerman and Marrig Knip. Reallocation of beds to reduce waiting time for cardiac surgery. *Health care management science*, 7(2):119–126, 2004.

[47] Norm Matloff. Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, 2(2009):1–33, 2008.

[48] Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017.

[49] R B Patil, M A Patil, V Ravi, et al. Predictive modeling for corrective maintenance of imaging devices from machine logs. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pages 1676–1679. IEEE, 2017.

[50] G Tokur, K Lederle, D D Terris, et al. Process analysis to reduce MRI access time at a German university hospital. *International Journal of Quality in Health Care*, 24(1):95–99, 2012.

[51] J J O'Brien et al. Optimizing MRI logistics: focused process improvements can increase throughput in an academic radiology department. *American Journal of Roentgenology*, 208(2):W38–W44, 2017.

[52] S B Gay, A H Sobel, L Q Young, et al. Processes involved in reading imaging studies: workflow analysis and implications for workstation development. *Journal of Digital Imaging*, 10(1):40–45, 1997.

[53] Brooke V Wessman, Andrew K Moriarity, Vanda Ametlli, et al. Reducing barriers to timely MR imaging scheduling. *RadioGraphics*, 34:2064–2070, 2014.

[54] B Reiner, E Siegel, and J A Carrino. Workflow optimization: current trends and future directions. *Journal of Digital Imaging*, 15(3):141–152, 2002.

[55] N Kuhnert, O Lindenmayr, and A Maier. Classification of body regions based on MRI log files. In *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, pages 102–109, Cham, 2017. Springer International Publishing.

[56] E Sartoretti, T Sartoretti, C Binkert, et al. Reduction of procedure times in routine clinical practice with Compressed SENSE magnetic resonance imaging technique. *PLoS One*, 14(4):e0214887, 2019.

[57] I A Talati, P Krishnan, and R W Filice. Developing deeper radiology exam insight to optimize MRI workflow and patient experience. *Journal of Digital Imaging*, 32:865–869, 2019.

[58] D Womack, J P Jones, and D T Roos. *The Machine that Changed the World: The Story of Lean Production*. HarperCollins Publishers, New York, USA, 1990.

[59] C J Roth, D T Boll, L K Wall, et al. Evaluation of MRI acquisition workflow with lean six sigma method: case study of liver and knee examinations. *American Journal of Roentgenology*, 195(2):150–156, 2010.

[60] T Melton. The benefits of lean manufacturing: What lean thinking has to offer the process industries. *Chemical Engineering Research and Design*, 83(6):662–673, 2005.

[61] J B Kruskal, A Reedy, L Pascal, et al. Quality initiatives: lean approach to improving performance and efficiency in a radiology department. *RadioGraphics*, 32(2):573–587, 2012.

[62] H B Harvey, E Hassanzadeh, S Aran, et al. Key performance indicators in radiology: you can't manage what you can't measure. *Current Problems in Diagnostic Radiology*, 45(2):115–121, 2016.

[63] P McCullagh. Generalized linear models. *European Journal of Operational Research*, 16(3):285–292, 1984.

[64] U Olsson. *Generalized Linear Models: An Applied Approach*. Studentlitteratur, 2002.

[65] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. In *Proceedings of the 2nd clinical natural language processing workshop*, pages 72–78, 2019.

[66] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[67] Gregory D Avey, Daryn S Belden, Ryan D Zea, and John-Paul J Yu. Factors predicting the time-length variability of identically protocoled MRI exams. *Journal of Magnetic Resonance Imaging*, 49(7):e265–e270, 2019.

[68] Ruolz Ariste and Guy Fortin. Could MRI and CT scanners be operated more intensively in Canada? *Healthcare Policy*, 3(1):e113, 2007.

[69] Anne Holbrook, Harold Jr Glenn, Ramsha Mahmood, Qian Cai, Jian Kang, and Richard Jr Duszak. Shorter perceived outpatient MRI wait times associated with higher patient satisfaction. *Journal of the American College of Radiology*, 13(5):505–509, 2016.

[70] M Muelly, Paul B Stoddard, and Shreyas S Vasanwala. Using machine learning with dynamic exam block lengths to decrease patient wait time and optimize MRI schedule fill rate. In *Proceedings of the 26th Annual Meeting of ISMRM*, Honolulu, 2017.

[71] Bowen Pang, Xiaolei Xie, Feng Ju, and James Pipe. A dynamic sequential decision-making model on MRI real-time scheduling with simulation-based optimization. *Health Care Management Science*, 25(3):426–440, 2022.

[72] Vineeth A Pisharody, Hooman Yarmohammadi, Etay Ziv, et al. Reducing wait times for radiology exams around holiday periods: a Monte Carlo simulation. *Journal of Digital Imaging*, 36(1):29–37, 2023.

[73] Haoqing Qiu, Dujuan Wang, Yanzhang Wang, and Yunqiang Yin. MRI appointment scheduling with uncertain examination time. *Journal of Combinatorial Optimization*, 37(1):62–82, 2019.

[74] Yifei Sun, Usha Nandini Raghavan, Vikrant Vaze, Christopher S Hall, Patricia Doyle, Stacey Sullivan Richard, and Christoph Wald. Stochastic programming for outpatient scheduling with flexible inpatient exam accommodation. *Health Care Management Science*, 24(3):460–481, 2021.

[75] Yu-Chun Sun, Hsiu-Mei Wu, Wan-Yuo Guo, et al. Simulation and evaluation of increased imaging service capacity at the MRI department using reduced coil-setting times. *PLoS One*, 18(7):e0288546, 2023.

[76] Xiaodan Wu, Juan Li, and Mohammad T Khasawneh. Rule-based task assignment and scheduling of medical examinations for heterogeneous MRI machines. *Journal of Simulation*, 2020.

[77] Zheng Zhou, Hao Zhou, Yue Qiao, Zhan Gao, and Yang Yang. An optimization protocol for MRI examination resource allocation based on demand forecasting and linear programming. *Scientific Reports*, 15(1):15076, 2025.

[78] Felipe Silva-Aravena, Juan Morales, Manoj Jayabalan, and Pamela Sáez. Optimizing MRI scheduling in high-complexity hospitals: a digital twin and reinforcement learning approach. *Bioengineering*, 12(6):626, 2025.

[79] Anders N Gullhav, Marielle Christiansen, Bjorn Nygreen, et al. Block scheduling at magnetic resonance imaging labs. *Operations Research for Health Care*, 18:52–64, 2018.

[80] Angela Leung, Jessica Mathews, Mark Crawford, et al. Improving MRI access: a framework for optimizing MRI scheduling templates. *Journal of Medical Imaging and Radiation Sciences*, 53(4):546–553, 2022.

[81] L P G Petroianu, Lun Li, Rebecca J Mieloszyk, Christina M Mastrangelo, Shawn Stapleton, and Christopher Hall. MRI log file analysis for workflow improvement. *Current Problems in Diagnostic Radiology*, 53(2):192–200, 2024.

[82] M Muelly and Shreyas S Vasanawala. MRI schedule optimization through discrete event simulation and neural networks as a means of increasing scanner productivity. In *Radiology Society of North America, 102nd Scientific Assembly and Annual Meeting*, 2016.

[83] Xiang Wang, Samira Nikfam Aski, Frank Uhlemann, Vivek Gupta, and Thomas Amthor. Predicting slot lengths of MRI exams to decrease observed discrepancies between planning and execution. *Current Problems in Diagnostic Radiology*, 53(3):359–368, 2024.

[84] Epic Systems Corporation. Epic electronic health records (EHR) software. Computer software, 2025.

[85] Philips. Philips OneSpace insights. Web page, 2025.

[86] Rui Zhang, Vamsi R Narra, and Akash P Kansagra. Large-scale assessment of scan-time variability and multiple-procedure efficiency for cross-sectional neuroradiological exams in clinical practice. *Journal of Digital Imaging*, 33(1):143–150, 2020.

[87] J Martin Bland and Douglas Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, 327(8476):307–310, 1986.

[88] Tonya S King, Vernon M Chinchilli, and Josep L Carrasco. A repeated measures concordance correlation coefficient. *Statistics in Medicine*, 26(16):3095–3113, 2007.

[89] Lawrence I Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45(1):255–268, 1989.

[90] Richard A Parker, Catherine Scott, Vanda Inácio, and Nathaniel T Stevens. Using multiple agreement methods for continuous repeated measures data: a tutorial for practitioners. *BMC Medical Research Methodology*, 20(1):154, 2020.

[91] Nüvit Özgür Doğan. Bland–Altman analysis: a paradigm to understand correlation and agreement. *Turkish Journal of Emergency Medicine*, 18(4):139–141, 2018.

[92] Davide Giavarina. Understanding Bland Altman analysis. *Biochemia Medica*, 25(2):141–151, 2015.

[93] Anne M Euser, Friedo W Dekker, and Saskia Le Cessie. A practical approach to Bland–Altman plots and variation coefficients for log transformed variables. *Journal of Clinical Epidemiology*, 61(10):978–982, 2008.

[94] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[95] Glen B McBride. A proposal for strength-of-agreement criteria for Lin's concordance correlation coefficient. Technical Report HAM2005-062, NIWA, 2005.

[96] Peter F Watson and Alastair Petrie. Method agreement analysis: a review of correct methodology. *Theriogenology*, 73(9):1167–1179, 2010.

[97] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[98] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[99] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[100] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[101] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[102] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[103] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31, 2018.

[104] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364, 2022.

[105] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[106] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

[107] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[108] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889, 2018.

[109] Shereen Elsayed, Daniela Thyssens, Ahmed Rber, and Max Nikber. Do we really need deep learning models for time series forecasting? *arXiv preprint arXiv:2101.02118*, 2021.

[110] Saverio Vito. Air Quality. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C59K5F.

[111] Husein Harun, Michael Nsor, Edidiong Akpan, Udodirim Ogwo-Ude, Gloria Ogochukwu, Bokolo George, and Josiah Tettey. Time-series forecasting of urban air pollutant levels using deep learning models. 10 2025.

[112] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007.

[113] Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[114] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[115] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[116] Hadi Fanaee-T. Bike Sharing. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5W894.

[117] Florian Knauer and Will Cukierski. Rossmann store sales. `https://kaggle.com/competitions/rossmann-store-sales`, 2015. Kaggle Competition.

[118] Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. In *arXiv preprint arXiv:1604.06737*, 2016.

[119] Tim Januschowski, Yuyang Wang, Kari Torkkola, Timo Erkkilä, Hilaf Hasson, and Jan Gasthaus. Forecasting with trees. *International Journal of Forecasting*, 38(4):1473–1481, 2022.

[120] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631. ACM, 2019.

[121] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.