ABSTRACT

Title of Document: USING ARTIFICIAL INTELLIGENCE TO IMPROVE HEALTHCARE QUALITY AND EFFICIENCY

Weiguang Wang, Doctor of Philosophy, 2020

Directed By: Guodong (Gordon) Gao, Professor, Decisions, Operations, and Information Technology
Ritu Agarwal, Professor, Decisions, Operations, and Information Technology

In recent years, artificial intelligence (AI), especially machine learning (ML) and deep learning (DL), has represented one of the most exciting advances in science. The performance of ML-based AI in many areas, such as computer vision, voice recognition, and natural language processing has improved dramatically, offering unprecedented opportunities for application in a variety of different domains. In the critical domain of healthcare, great potential exists for a broader application of ML to improve quality and efficiency. At the same time, there are substantial challenges in the development and implementation of AI in healthcare.

This dissertation aims to study the application of state-of-the-art AI technologies in healthcare, ranging from original method development to model interpretation and real-world implementation. First, a novel DL-based method is developed to

efficiently analyze the rich and complex electronic health record data. This DL-based approach shows promise in facilitating the analysis of real-world data and can complement clinical knowledge by revealing deeper insights. Both knowledge discovery and performance of predictive models are demonstrably boosted by this method.

Second, a recurrent neural network (named LSTM-DL) is developed and shown to outperform all existing methods in addressing an important real-world question, patient cost prediction. A series of novel analyses is used to derive a deeper understanding of deep learning's advantages. The LSTM-DL model consistently outperforms other models with nearly the same level of advantages across different subgroups. Interestingly, the advantage of the LSTM-DL is significantly driven by the amount of fluctuation in the sequential data. By opening the "black box," the parameters learned during the training period are examined, and is it demonstrated that LSTM-DL's ability to react to high fluctuation is gained during the training rather than inherited from its special architecture. LSTM-DL can also learn to be less sensitive to fluctuations if the fluctuation is not playing an important role.

Finally, the implementation of ML models in real practice is studied. Since at its current stage of development, ML-based AI will most likely assistant human workers rather than replace them, it is critical to understand how human workers collaborate with AI. An AI tool was developed in collaboration with a medical coding company, and successfully implemented in the real work environment. The impact of this tool

on worker performance is examined. Findings show that use of AI can significantly boost the work productivity of human coders. The heterogeneity of AI’s effects is further investigated, and results show that the human circadian rhythm and coder seniority are both significant factors in conditioning productivity gains. One interesting finding regarding heterogeneity is that the AI has its best effects when a coder is at her/his peak of performance (as opposed to other times), which supports the theory of human-AI complementarity. However, this theory does not necessarily hold true across different coders. While it could be assumed that senior coders would benefit more from the AI, junior coders’ productivity is found to improve more. A further qualitative study uncovers the underlying mechanism driving this interesting effect: senior coders express strong resistance to AI, and their low trust in AI significantly hinders them from realizing the AI’s value.

USING ARTIFICIAL INTELLIGENCE TO IMPROVE HEALTHCARE QUALITY AND EFFICIENCY

By

Weiguang Wang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2020

Advisory Committee:
Professor Guodong (Gordon) Gao, Chair
Professor Ritu Agarwal, Chair
Assistant Professor Margrét Bjarnadóttir
Assistant Professor Kunpeng Zhang
Professor Ginger (Zhe) Jin, *Dean’s Representative*

Acknowledgements

This dissertation is rooted in the domain of healthcare and incorporates technical elements in computer science. While completing this dissertation has demanded various supports from many areas, there are too many people to thank. I would especially like to thank my two advisors, coauthors from different backgrounds, my dissertation committee, professors who taught me and inspired me, my peer Ph.D. students, colleagues in CHIDS, collaborators in industry, and the Ph.D. office.

I would like to express my deepest gratitude to Professor Gordon Gao for the valuable advice on academic research, the patient guidance in being a good scholar and the immense and unreserved support. I have the same gratitude for Professor Ritu Agarwal, who has provided tremendous advice, guidance and support throughout my Ph.D. program and the dissertation process. It is only with their rich fund of knowledge in healthcare, business, and technology that I am able to complete this dissertation.

I want to truly thank my coauthors Professor Ginger Zhe Jin, Professor Margrét Bjarnadóttir, Professor Kunpeng Zhang, and many others. I learned a lot from them, which contributed enormously to this dissertation. I enjoyed the process of working with them.

Professors at the University of Maryland and other schools who taught me, inspired me, or provided feedback to me are all contributors. My peers who supported me in

different ways should not be neglected. My collaborators in industry also substantially improved the practical value of this dissertation. My colleagues in the Center for Health Information and Decision Systems (CHIDS), especially Dr. Kenyon Crowley and Dr. Michelle Dugas, also played the most important role in connecting me and my research for this dissertation to a broader audience in industry. A special thank you must be given to Justina Blanco in the Ph.D. office, who has helped with all my questions with great patience throughout my Ph.D. program. I sincerely appreciate all these great minds and hands.

My work in the domain of healthcare and technology is also greatly guided and helped by many senior scholars in other fields. Their advice is a huge asset to me. In particular, I would like to express my gratitude and appreciation to Professor Rongping Mu, who was and still is inspiring my thoughts regarding broad topics in science, technology, and society. Without his support, guidance and overall insight in the field, I would never be able to complete this journey.

I also appreciate the financial support from the US National Science Foundation (IIS-1254021).

Finally, I thank my family for their unwavering support.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In the past decade, artificial intelligence (AI), especially machine learning (ML) and deep learning (DL), has represented one of the most exciting advances in science. With unprecedented developments, ML-based AI technologies have dramatically improved performance in many domains, such as computer vision, voice recognition, and natural language processing, to name a few.

While AI holds substantial promises to improve healthcare, significant challenges exist. First, with the recent pervasive digitalization in healthcare, huge amount of data is available in Electronic Health Records (EHR). However, EHR data is quite underutilized, largely due to the data's complexity. The medical codes usually make these complex data fields very hard to analyze. As a result of this "big data challenge", people have to collaborate with domain experts, select a small portion of the EHR data, and throw away all other relevant information. Second, the current applications of complex AI models in healthcare mostly suffer from the "black box" issue. The lack of interpretation hinders the wider adoption of these powerful models. Finally, from the business perspective, it is unclear how AI will interact with human intelligence in a real work environment. There is an urgent need from AI adopters in healthcare to understand how the value of AI could be realized in real practice.

With regard to the three challenges, this dissertation aims to advance the study on state-of-the-art AI technologies in healthcare in the following three ways: 1) developing a novel AI methodology to solve the "big data challenge", 2) shedding light on the “black box” of AI in healthcare predictive models, and 3) gaining deeper insights into real-world AI implementation and its interaction with human intelligence. Each topic is addressed in a different study.

Study 1: Every Bit Counts: Using Deep Learning and Vectorization to Analyze Healthcare Big Data

The rapid digitization of healthcare has generated large volumes of rich and complex data from sources such as claims and electronic health records. Traditional analytic approaches, however, only utilize small subsets of these data and often require deep domain knowledge. We develop a new Deep Learning-based Vectorization (DLV) approach for more comprehensive and efficient analysis of healthcare data. This approach automatically converts data elements into standardized numeric vectors, enabling new types of computing and improving performance in traditional data analysis. We demonstrate the potential of DLV to predict 30-day readmission using discharge records that cover all inpatient hospitalizations in Florida. We find that DLV easily handles large amounts of clinical information (including non-numeric variables), while traditional approaches struggle even to load the data. Furthermore, DLV significantly improves the accuracy of 30-day readmission prediction in the presence of high-dimensional data, boosting the AUC from 0.61 to 0.79. In addition, we demonstrate that the vector representations offered by DLV afford easy visualization for better understanding of the clinical data. Overall, the DLV approach shows great potential in facilitating the analysis of big healthcare data and can complement traditional methods in high-dimensional environments.

Study 2: How AI Plays Its Tricks: Understanding the Superior Performance of a Deep Learning-Based Approach in Predicting Healthcare Costs

This study aims to advance our understanding of deep learning's performance in predicting healthcare costs. We first design a long short-term memory (LSTM) based recurrent neural network (RNN) to incorporate sequential information for more accurate healthcare cost predictions. We then propose a novel approach to explore what drives the advantage of deep learning over major traditional machine learning methods (including linear regression, LASSO regression, ridge regression, and random forest). We find that in most traditional prediction models, greater fluctuation in data leads to deterioration in prediction performance. In contrast, the LSTM model can better incorporate the fluctuation information and actually gain prediction accuracy when fluctuation increases. We further visualize how the LSTM model processes fluctuations in monthly cost information by examining the output signals of the LSTM units. Our work provides insights into the advantages of deep learning models in predicting healthcare costs and also generates practical guidance.

Study 3: Friend or Foe? How Artificial Intelligence Affects Human Performance in Medical Chart Coding

While the impact of AI on jobs has generated considerable discussion and debate, little is known about how AI interacts with workers at different seniority levels across different times of the day. We developed an AI solution for medical chart coding in a publicly traded company and then evaluated its impact on productivity both within

and across individual workers. We find evidence that AI improves worker productivity overall, but the productivity gain is mostly associated with human workers' circadian rhythm. Specifically, AI is most beneficial in the morning when human performance is also at its peak, rather than afternoon or night when human performance slows down. Results also show that the benefits of AI are dependent on the worker's experience level: the productivity of junior workers experiences a significantly higher boost from the use of AI than that of senior workers. Further analysis reveals that the performance discrepancy is attributable to senior user resistance. This paper provides new empirical insights into how AI affects knowledge worker productivity, with important implications for wider adoption and use of AI among knowledge workers.

Chapter 2: Every Bit Counts: Using Deep Learning and Vectorization to Analyze Healthcare Big Data

2.1 Introduction

The past decade saw a sharp increase in the extent to which companies examine large amounts of data to uncover hidden patterns and optimize their business, often referred to as “big data analytics.” In healthcare, increasing digitalization in both the provider and consumer self-health management settings is generating vast amounts of highly granular data. These data come from various sources such as electronic health records (EHR), claims data, and patient-generated content (Mennemeyer et al., 2016). They consist of heterogeneous data elements, including patient demographics, diagnoses, biomarkers, laboratory results, and medical prescriptions as well as unstructured and nontraditional data such as clinical notes and images. The volume, richness, and timeliness of these data represent an unprecedented opportunity for knowledge discovery and quality improvement (Bates et al., 2014; Jensen et al., 2012; Sherman et al., 2016; Jarow et al., 2017). Effective analysis of these data is critical to transforming healthcare into a rapid learning system and guiding the approval and use of new treatments (Galson and Simon, 2016). Therefore, there is a call in the literature to better use patient-level data to generate useful and actionable insights (Angst et al., 2010).

However, traditional healthcare analytics are not designed for high-dimensional data, and have difficulty incorporating a large number of features (Xiao et al., 2018). Most predictive models rely on the expertise of domain experts, who hand select a limited number of variables. This means that despite the availability of variables in our big data era, a large number of available variables are dropped in these models. The resulting models may also have limited generalizability across datasets or institutional settings. Furthermore, traditional healthcare analytics often entail manual coding of non-numeric variables such as codes and texts. The labor-intensive steps in variable selection and coding largely limit the healthcare field's exploitation of big data, which tends to be highly dimensional and non-numeric. There is therefore an urgent need for new methods to improve the efficiency and convenience of big data analysis in healthcare.

In this study, we propose a novel deep learning-based vectorization (hereafter DLV) approach for big data analysis in healthcare. The DLV approach affords substantial advantages compared to traditional data analytic techniques. First, while a typical data analytics project requires a lengthy data preparation and feature extraction process, our DLV requires very little data preparation prior to model training. In DLV, all the data elements are natural occurrences and there is no need to code new variables. Second, DLV converts any structured data element, including demographic variables or medical codes, into a standardized vector that can be easily analyzed. Third, using deep learning architecture, DLV converts high-dimensional data into a reduced dimension with minimum information loss. Finally, the vector representation of variables can be easily visualized. This will allow healthcare researchers and practitioners to quickly spot significant patterns across data elements and then apply their medical knowledge and clinical expertise to generate hypotheses for further analysis. Given these advantages, we expect that DLV can substantially improve the efficiency of healthcare data analyses and spur a new stream of research in this domain.

Our DLV is built on Word2Vec (Mikolov et al., 2013), which was developed for and widely applied to text data analysis. It has revolutionized natural language processing (NLP) (Rush et al., 2015; Pennington et al. 2014) and powered applications such as Google translate and Apple Siri (Sutskever et al., 2014; Bahdanau et al., 2014; Capes et al., 2017). Based on the idea of Word2Vec, new technologies have emerged for other forms of data, such as Bio2Vec for protein (Asgari et al., 2015). Our Word2Vec DLV is similar to Med2Vec (Choi et al., 2016b), although the latter was developed from the perspective of medical code similarity and aims to understand the structure of medical codes such as ICD and CPT. We build on the foundation of a number of studies that have used deep learning for prediction using EHR and claims data such as Doctor AI, GRAM, and Deepr (Choi et al., 2016a; Choi et al., 2017; Nguyen et al., 2016). Compared to these existing models, we aim to leverage the full big data, rather than a subset of them. By vectorizing almost all elements in EHR data, our work extends the scope of DLV to further the understanding of whole clinical practices, and in particular as a method of readmissions prediction (e.g., Rose, 2016; Rajkomar et al., 2018).

Our study combines deep learning and word vectorization for structured healthcare data. This combination of methods may be used for conventional predictive modeling but has a wide range of additional applications. We employ data collected by the Florida Agency for Health Care Administration (AHCA) to provide three use cases that illustrate the potential applications of DLV. First, word vectorization provides an easy framework for data visualization. Second, DLV may be used to map complex clinical relationships. We illustrate how DLV can learn correspondences between ICD-9 and ICD-10 codes without the need for any domain knowledge. This approach could facilitate EHR interoperability. Finally, we use DLV to predict preventable hospital readmissions. We find that DLV has an area under the curve (AUC) of 0.79, which is substantially higher than conventional risk prediction algorithms.

To summarize, by combining Word2Vec and deep learning, DLV can extract, reorganize, and retain the rich information in the complex healthcare big data via vectorization. DLV largely reduces costly manual data preparation and can power a wide spectrum of data analyses ranging from visualization to prediction.

2.2 The Dimensionality Dilemma

Machine learning methods, especially deep learning, are often used for flexible non-parametric predictions in large high-dimensional data environments. They offer two advantages: using more detailed data may improve predictive accuracy and a flexible model will better reflect heterogeneity across individuals. At the same time, however, there are costs to employing flexible and high-dimensional models. Almost all existing predictive models use binary variables to code information like diagnosis, hospital ID, procedures, etc., which often leads to an input matrix too big to load in the models. Even if the models are able to load the data, the prediction will perform poorly due to the sparsity of the input data. Overall, models tend to be computationally burdensome and difficult to implement.

Big data in healthcare therefore creates a dimensionality dilemma: while the high-dimensional data contains rich and useful information, it is often necessary to apply ad hoc variable selection to reduce the dimension of the data to fit into the model, leading to information loss and lower performance. Typically, there are two major strategies for variable selection: data driven and domain knowledge driven. The data driven approach chooses variables based on criteria such as correlations with outcomes (Shameer et al., 2017), the frequency of occurrence (Futoma et al., 2015), or variable selection algorithms (Bayati et al., 2014). Recent machine learning models, particularly those estimated via deep learning, employ a statistical method (Li et al., 2015; Wang et al., 2014; Zhao et al., 2015) for variable selection. These approaches are practical but there is no guarantee that they will keep the most relevant predictors. These data driven approaches to dimension reduction may also be subject to overfitting.

The domain expertise approach is often used to select a logically relevant data subset (e.g., Frizzell et al., 2017; Ouwerkerk et al., 2014; Golas et al., 2018). Predictive models for health policy and practice historically emphasize both dimension reduction and model specification based on domain knowledge (Desai et al., 2002; Hon et al., 2016). While domain knowledge has an intuitive appeal, it has several potential shortfalls. First, this process is labor intensive in high-dimensional environments. Second, even if domain experts are aware of the most important features, they may miss many relevant ones that improve prediction and capture outcome heterogeneity. Third, focusing on what experts know to be important may undermine new knowledge discovery. Finally, model selection based on domain knowledge can easily lead to overfitting and bias (e.g., Ioannidis, 2005; Fanelli et al., 2017; Ioannidis et al., 2017).

Our DLV approach eliminates the need for feature selection and is able to incorporate all information in a standardized way. DLV converts input data into much shorter vectors with much less sparsity. The vectorization in DLV both retains all information from the data and reflects the relative relationship among raw variables. In addition, the vectorization is achieved automatically through a simple prediction task (which we will describe later). This process does not require strong domain knowledge nor excessive computational power.

2.3 Methods

DLV is built on both deep learning and word embedding, and it was actually inspired by the functions and structures of brain cells. DLV models have at least three layers: an initial input layer, one projection layer, and an output layer. The projection layer uses simple modules to transform the data from the preceding layer.[1] Parameters from

[1] Most deep learning models use hidden layers with non-linear functions, such as sigmoid, which is a generalized version of logit, or the rectified linear unit (ReLU). However, for vectorization we use a projection layer with a special linear function, which is explained later.

these modules are *learned* during the training process to try to minimize the difference between the true and predicted outcomes (LeCun et al., 2015). These models may be made more flexible by increasing the number nodes of the projection layer. Leveraging increasing computing power and larger training datasets, deep learning has been able to model complex functions such as natural language translation (Liu and Zhang, 2018).

Word embedding refers to the representation of natural words as vectors. Deep learning approaches learn the complex structure of words within sentences and across data elements, then uses word embedding to convert the words into vectors. These vectors preserve connections among words in the original context and afford the words computability. One well-known example is the vectorization of the natural words Queen, King, Man, and Woman, which are shown to have the following relationship: Queen –Woman = King – Man. In other words, subtracting "Woman" from "Queen" gives us something similar to what is left when "Man" is subtracted from "King" (which is royalty). One can also generate a "King" representation by replacing the gender of "Queen" ("Queen – Woman + Man = King") (Mikolov et al., 2013).

While vectorization was initially developed for natural language processing, we generalize it to structured healthcare data, which includes a wide range of data types such as patient demographics, physician characteristics, insurance benefit design, diagnoses, procedures, costs, prescriptions, etc. We regard all the data associated with one visit as one big “sentence.” Vectorization is carried out in a similar fashion to the vectorization of natural language.

2.3.1 Major Computing Steps in DLV

DLV proceeds through two major steps to complete the vectorization of all data elements. In Step 1, the raw clinical data is preprocessed to the desirable format. In Step 2, a deep learning model is developed to predict the co-occurrence of data elements (such as the co-occurrence of “age 60” and “diabetes”). This step determines the vectors for each data element while maximizing the prediction performance. In other words, DLV achieves the vectorization for all data elements in the data by completing a prediction task using a deep learning model. Details of these two steps are illustrated below.

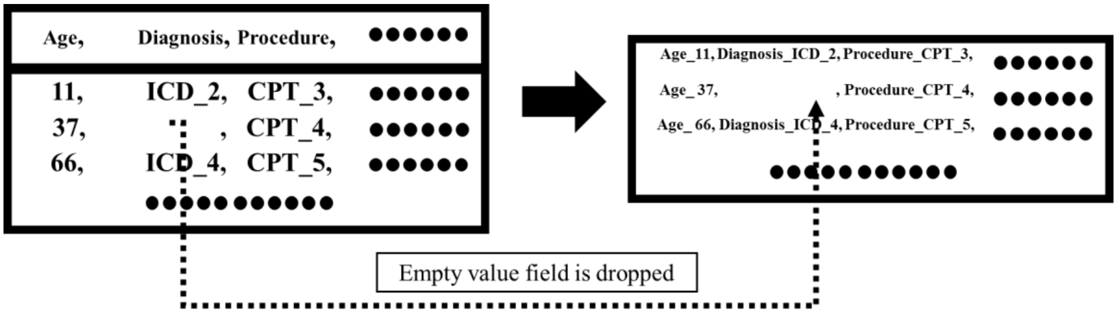

Age, Diagnosis, Procedure, ●●●●●●
11, ICD_2, CPT_3, ●●●●●●
37, , CPT_4, ●●●●●●
66, ICD_4, CPT_5, ●●●●●●
●●●●●●●●●●●
Age_11, Diagnosis_ICD_2, Procedure_CPT_3, ●●●●●●
Age_37, , Procedure_CPT_4, ●●●●●●
Age_66, Diagnosis_ICD_4, Procedure_CPT_5, ●●●●●●
●●●●●●●●●●●
Empty value field is dropped

Figure 1. Data conversion in preprocessing.

In Step 1, DLV begins with a very simple preprocessing step, where variable names and values are combined into a data element which we call a *word*. We illustrate this process using information generated from a patient encounter. In Figure 1, each row comprises data from a single ambulatory visit. The first column is patient age. We combine “Age” and its value “11” in the first row as “Age_11” (Figure 1). Similarly, in that visit, the doctor makes a diagnosis with ICD_2 (where ICD stands for the International Classification of Diseases), then this information is converted to “Diagnosis_ICD_2”.

We next prepare the dataset for the deep learning model in Step 2. The task of the deep learning model is to use a *word* to predict other *words* that would co-occur with it in the same visit. Therefore, the *words* are converted into co-occurrence pairs. For the first instance in Figure 1, for example, there are three *words*: {Age_11, Diagnosis_ICD_2, Procedure_CPT_3}. Since predictions are directional, as illustrated in the bottom of Figure 2, the combination of these three *words* leads to six instances. These instances serve as ground truth for the prediction model using deep learning.

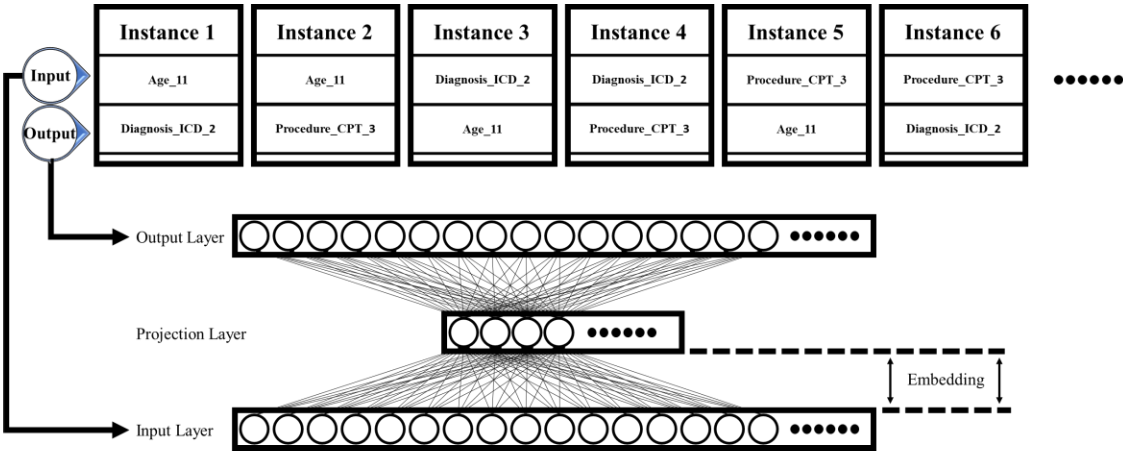


Figure 2. Training data and the architecture of the neural network in DLV.

In Step 2, we first represent the input and output *words* as vectors of dummy variables (which are called one-hot vectors) corresponding to each individual *word*. The input vectors are then transformed by simple linear models that are commonly referred to as the projection layer (similar to the term “hidden layer”, but hidden layer refers to non-linear transformation), and the transformed values from the projection layers are used to generate the output vectors. In other words, input vectors are transformed by the projection layer to generate prediction of output vectors. The structure of the projection layer allows for tremendous flexibility and captures the potentially large set of interactions between data elements. The parameters in the projection layer are estimated via a loss function called cross entropy using backward propagation. This is similar to the idea of maximum likelihood estimation used in conventional regression methods.

The above process can be expressed in the mathematical form below. The occurrence of the output *words*, *O*, given the presence of input *words*, *I*, is a function of a high-dimensional matrix, *B*. This may be thought of as:

$$O = f(B \cdot I) \quad \cdots\cdots(1)$$

B represents the projection layer. Each input *word* has a corresponding vector of parameters from matrix *B* that describes its relative position. Further calculations and transformation, f , can convert this vector into a probability of co-occurrence.[2] After minimizing the loss function, each vector within *B* is taken as the vector representation of a given *word,* where β_1 is the vector representation of the first *word* and $B = [\beta_1, \beta_2, \ldots \beta_N]$.

[2] f includes a matrix multiplication (B, I, and another parameter matrix) and a softmax transformation, which is similar to logistic transformation, converting the results to probabilities. For more technical details, please refer to Mikolov et al., 2013.

These vectors can be used to perform a wide range of calculations and have many potential interpretations. We illustrate this potential in use cases below. First, in Section 2.4 we show that vectorization affords easy visualization, leading to knowledge discovery and easy computing. We then show how DLV substantially improves readmission risk prediction compared to traditional models in Section 2.5.

2.4 DLV Applications – Visualization and Easy Computing

We provide three examples to illustrate DLV’s potential applications. First, we illustrate how word vectorization facilitates data visualization. Second, we show how word vectorization allows for simple but intuitive mappings of complex clinical concepts even in the absence of domain knowledge. Third, we show how DLV can substantially improve clinical prediction (presented in Section 2.5). These applications employ commonly available administrative claims data.

We use hospital outpatient records provided by the Florida Agency for Health Care Administration (AHCA). The outpatient data include encounter-level data for all ambulatory surgeries and emergency department visits from 213 healthcare facilities in the state of Florida, totaling 11,284,760 records. We choose to focus on the outpatient data from Quarter 1 to Quarter 4 of 2015 because the transition from ICD-9 to ICD-10 codes took effect on October 1, 2015 and we leverage this transition as a setting to verify our method.

There are 100 fields in the focused data. The 100 fields fall into the following categories: system identifiers, time stamps, facility characteristics, patient residence, patient demographics, payers, patient discharge status, billing codes (i.e., ICD, Current Procedural Terminology (CPT) or Healthcare Common Procedure Coding System (HCPCS) codes), practitioner identifiers, and various charges. The vector length is 205,790. We exclude system record ID, year, facility Medicare number, facility region, facility county, patient county, patient state, and 15 variables involving different charges. All remaining *words* co-occurring in the dataset form the *word* list. Using DLV, each *word* is represented by a vector drawn from matrix *B*.

2.4.1 Visualizing Vectors for Pattern Discovery

To examine the information contained in the *word* vectors and their ability to reflect the relationships among *words*, we visualize the relationship among all *words* in the dataset. We use a dimension reduction method[3] similar to principal component analysis to plot these high-dimensional vectors in two-dimensional space. Figure 3 shows the plot of the top 1,500 *words* by frequency.[4] The distance between any two dots in the figure represents the relationship of the two corresponding *words*.

Generating a figure like Figure 3 allows us to draw a number of conclusions about the conceptual relationships within the clinical data. First, it is easy to categorize *words* based on the distances between them. In the left bottom area of the map, most of the

[3] t-distributed stochastic neighbor embedding (t-SNE), a technique used for dimensionality reduction in visualization.

[4] Note that users can extend beyond the first 1,500 *words* to plot all that occur with any frequency.

words are about ZIP codes and facilities. These two measures have an intuitive relationship, which DLV has learned from the data. Most *words* in the center of Figure 3 are CPT codes while most *words* on the right are diagnoses. The diagnoses show three distinct subtypes: admission diagnosis toward the bottom, “other diagnosis” at the top, and principal diagnosis in the middle as a link between the two. Within each cluster, the grouping and positions of the *words* are also interpretable. We observe, for example, that diagnosis codes from the same ICD family tend to group together.

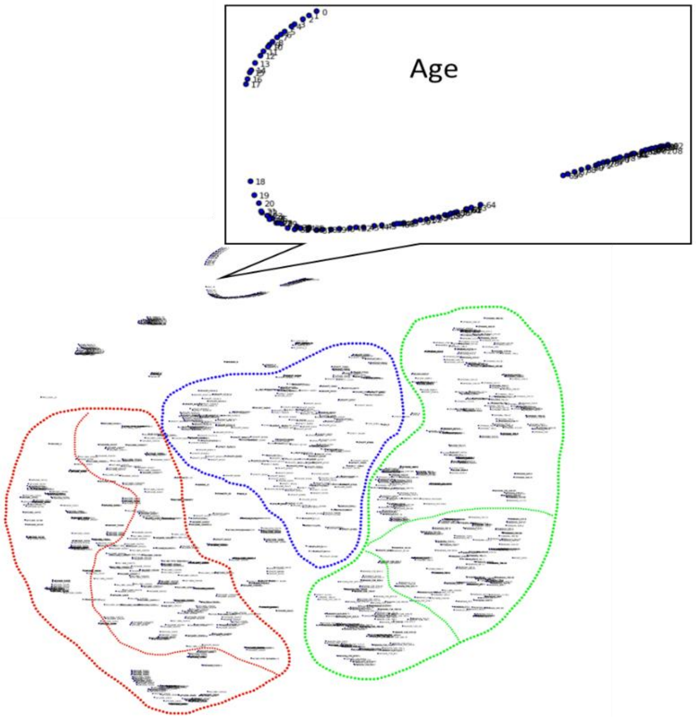


Figure 3. t-SNE of the first 1,500 words, with close-ups of ages.

Vector-based visualization facilitates pattern detection in high-dimensional data. Figure 3, for example, plots all age-related *words* in a line according to numerical value (see the enlarged section in Figure 3). It is interesting to note that the line is split into three discontinuous segments: those under the age of 18, those aged 18 to 64, and those aged 65 and over. These relationships have intuitive explanations. Those aged 65 and older are Medicare eligible and often transition to retirement, while 18-year-olds often transition to school or work and also lose eligibility for Florida's Kidcare insurance program. In summary, the vectorization of age groups successfully captures the dramatic eligibility changes that patients in our dataset undergo at age 18 and age 65.

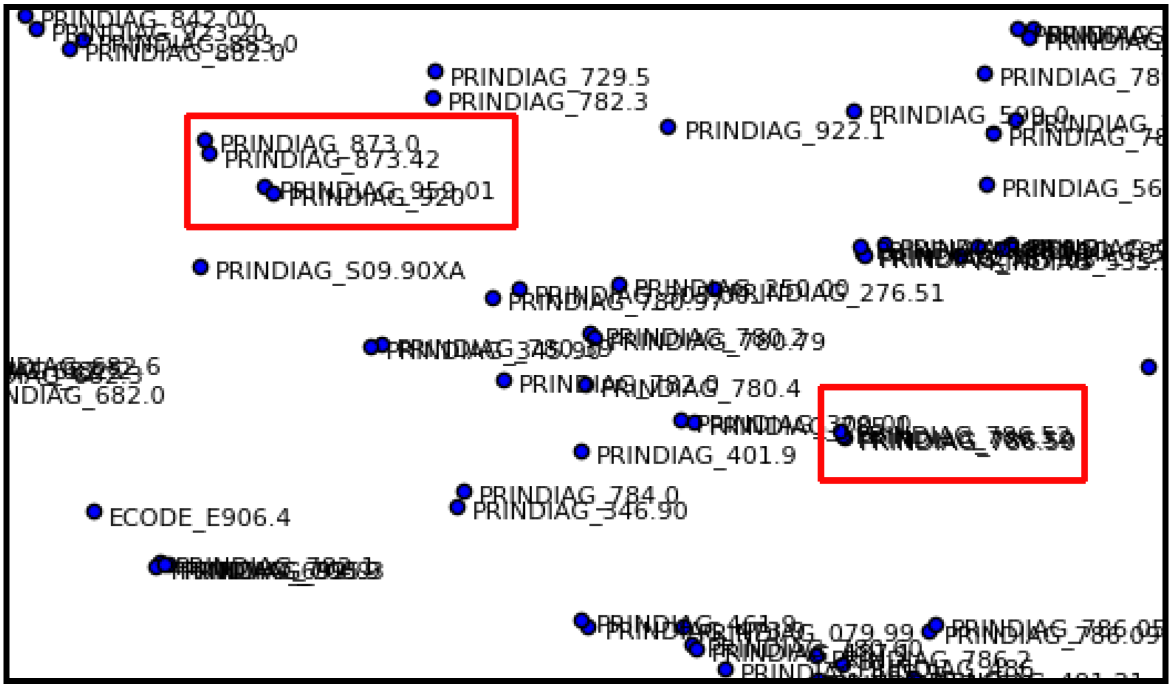


Figure 4. Example principal diagnosis codes in the t-SNE visualization.

Another example of the way DLV can capture diagnostic relationships is the clustering of diagnosis codes as shown in Figure 4. (This figure zooms in on the example region for principal diagnosis codes from Figure 3, i.e. a portion of the middle subtype in the green cluster.) In this figure, we observe that the ICD codes in family 786.5[5] are overlapped (in the bottom right box), indicating the great consistency of medical knowledge (ICD coding system) and clinical practice (DLV based clustering). Another interesting example is in the left box. The four ICD codes are clustered into two groups that are close to each other. ICD 920 is “Contusion of face, scalp, and neck except eye(s)” and ICD 959.01 is “Head injury, unspecified,” while ICD 873.0 and ICD 873.42 represent “Open wound of scalp” and “Open wound of forehead,” respectively. One explanation for this division into two groups may be that both ICD 920 and ICD 959.01 point to a relatively broader scope while ICD 873.0 and ICD 873.42 focus on a specific region of the head. Yet based on their proximity in our visualization, it is clear that these two groups are related.

2.4.2 Computability of Words

To illustrate the computability of the clinical data enabled by DLV, we leverage a policy change that occurred during the span of our data: the conversion of ICD-9 to ICD-10 on October 1, 2015. We demonstrate the computability of DLV by predicting the corresponding ICD-10 code from the existing ICD-9 code. Despite the availability of mapping tools designed to match ICD-9-CM and ICD-10-CM classifications, many codes share complex, entangled and non-reciprocal relationships that may lead to confusion and incorrect coding (Boyd et al., 2013). A study that examined the transition to ICD-10-CM in emergency departments found that 27% of the transitions represented convoluted multidirectional mappings, of which 23% were clinically incorrect (Krive et al., 2015). Another recent study found that 25% of internal

[5] ICD coding structure incorporates the family of codes. Usually, the digits after decimal indicate the subcategory of the diagnoses.

medicine ICD-9-CM codes have convoluted mapping to ICD-10-CM and more than half of those mappings could result in potential clinical inaccuracies or administrative errors (Caskey et al., 2018). The ambiguity and inaccuracy of these mappings may affect physician reimbursement, impact the workflow process, and undermine quality measurement. Our approach can help organizations developing mapping tools to harmonize electronic health measures.

The essential challenge posed by the ICD transition is that our data contains pairs of ICD codes representing the same diseases. Though each pair is for the same disease and maybe even be for the same clinical practice, patient characteristics, and other factors, there are significant differences between them. Differences in the coding system structures (ICD-9 and ICD-10), seasonal differences, and many other differences could all be absorbed by the two codes in one pair. Yet instead of identifying factors driving the differences code by code, DLV considers the differences as a whole and can remove them. Given that there is no way to measure exact differences as a whole, we use the best proxy and leverage the two special *words*: “Quarter 3” and “Quarter 4.” These *words* uniquely map to the use of ICD-9 and ICD-10 respectively. Therefore, if the same disease was represented by ICD-9 (in Quarter 3 of 2015) and then ICD-10 (in Quarter 4 of 2015), we should be able to predict the vector of the ICD-10 code as follows:

$$ICD9\ Code - Quarter\ 3 + Quarter\ 4 = ICD10_{\mathrm{predicted}} \quad \cdots\cdots(2)$$

If our DLV approach works, then the predicted ICD-10 vector should have high similarity with the actual ICD-10 vector. We use the cosine similarity of the vectors, which ranges from completely unrelated at 0 to a perfect correspondence at 1. To illustrate, we first use the case of disease “Epigastric pain.” In ICD-9, the code is 789.06. This corresponds to R10.13 in ICD-10.[6] We find that the cosine similarity between the vector for 789.06 and the vector for R10.13 is 0.85. Applying the simple arithmetic of Equation 2 raises the cosine similarity to 0.97. This illustrates how DLV can be used to map complex clinical relationships without relying on domain knowledge.

We further conduct a systematic examination of the relationship defined in the conversion equation. We test this capability using ICD-9/ICD-10 coding pairs with a unique correspondence. Since many codes occur rarely in any given quarter, we limit our analyses to the 100 most common primary diagnosis codes. We focus on unique mappings as ICD-10 allows for more detail and increases diagnostic distinctions. For example, ICD-9 code 789.09 (“Abdominal pain, other specified site”) corresponds to R10.10 (“Upper abdominal pain, unspecified”), R10.2 (“Pelvic and perineal pain”), and R10.30 (“Lower abdominal pain, unspecified”). Given the data limitation, there might be codes that are not well represented in our data. Therefore we also exclude pairs whose raw similarity scores are lower than 0.25, resulting in the exclusion of 5 pairs. These 5 pairs are really outliers, since we can see that the lowest similarity after

[6] According to ICD10Data.com (https://www.icd10data.com/Convert/789.06), accessed September 30, 2017.

the exclusion is 0.65. Our final sample comprises 34 common and uniquely corresponding ICD-9/ICD-10 codes.

Equation 2 is used to convert the ICD-9 code to an ICD-10 code, and we compare the cosine similarity [between ICD-9 and ICD-10] and that [between computed ICD-10 and ICD-10]. Figure 5 presents the comparison of raw similarity and calculated similarity between ICD-10 and the predicted ICD-10. Across all the 34 pairs, the base cosine similarity across matched *words* is 0.79, and this increases to 0.92 when adjusting for the Quarter 3 and 4 vectors. The quarter vectors increase fit by 16.5% (on average). The increase in similarity is statistically significant at p<0.01, despite the relatively small sample size.

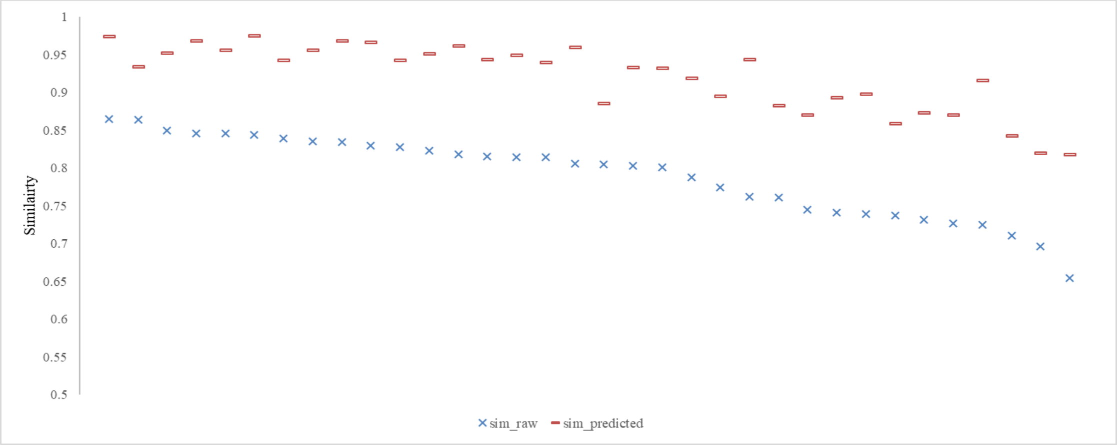


Figure 5. Primary diagnosis code conversion using DLV.

2.5 Using DLV to Predict 30-Day All-Cause Hospital Readmission

We apply DLV to predicting patients' risk of unplanned 30-day hospital readmission. As one of the most important hospital quality measures, the 30-day readmission rate is tied to as much as maximum of 3% Medicare reimbursement. Given this importance, readmission is also becoming a focus of business studies (Senot et al., 2015; Zhang et al., 2016). However, predicting readmission risk remains a difficult task (Ben-Assuli and Padman, 2019). In a literature review published in JAMA in 2011, Kansagara et al. reported that most earlier risk prediction models using retrospective administrative data performed poorly (c-statistics 0.55-0.65).

Models used by the Centers for Medicare and Medicaid Services (CMS) to predict 30-day all-cause readmissions for the three conditions (i.e., heart failure, acute myocardial infarction, and pneumonia) initially targeted by its Hospital Readmissions Reduction Program had c-statistics between 0.61 and 0.63. These three health problems are both common and require high out-of-sample predictive accuracy. DLV has several potential advantages over conventional methods as it requires no parametric or functional form assumptions and can readily handle high-dimensional data. The CMS condition-specific readmission model is used as a benchmark (Chen and Grabowski, 2017; National Quality Forum, 2015).

We compare the out-of-sample predictive performance for 30-day acute myocardial infarction (AMI) readmissions for the CMS and DLV models. Both models are estimated using 2008 hospital inpatient discharge data[7] for AMI patients. The raw data include 285 variables capturing patient demographics, diagnoses, and procedures as well as patient locations and provider identifiers. We identify the target condition

[7] We choose the year 2008 in order to control for confounding effects that might have arisen from the national hospital readmissions reduction program (HRRP) under the Affordable Care Act (ACA). Since July 2009, the Centers for Medicare and Medicaid Services (CMS) Hospital Compare website began publicly reporting hospital performance in mortality and 30-day readmission.

by the principal diagnosis of the index hospitalization, using the ICD-9 codes. More specifically, we identify patients hospitalized with a primary diagnosis of ST-segment elevation myocardial infarction (STEMI) (ICD-9 codes 410.1x, 410.2x, 410.3x, 410.4x, 410.5x, 410.6x, 410.8x, and 410.9x) or non-ST-segment elevation myocardial infarction (NSTEMI) (ICD-9 code 410.7x) during the years 2008. Most variables are categorical and the full range of variation is high-dimensional. Each five-digit ICD-9 code has more than 14,000 possible values (as there could be multiple ICD-9 codes in one visit). Given this task, conventional risk adjustment methods use experts' domain knowledge and ad-hoc statistical tests to select relevant variables and functional forms. Our DLV approach uses the full range of observed variation, including 7 continuous variables and 15,811 dummy variables following the process described in Section 2.2.

The CMS model is estimated by logistic regression while the DLV model is estimated via deep learning.[8] The models are estimated on an 80% subsample of our data. We then use the estimated parameter, or vectors in the case of DLV, to generate predictions on the 20% testing sample. Model fit is evaluated using the AUC for the testing data sample. These out-of-sample results are reported in Table 1. We find that the CMS model has an AUC of 0.61 (Column 1, Table 1), while the DLV model has an AUC of 0.79 (Column 4, Table 1). This represents a 30% improvement.

[8] DLV estimation technical details: The DLV deep learning model employs a cross-entropy loss function – similar to maximum likelihood estimation – and a single hidden layer with 200 nodes. Each *word* is converted to a 300-dimensional vector. The number of hidden layers and the dimensions of the vectors are user-determined parameters. Increasing either the number of layers or the dimensions of the vectors would increase the model's flexibility, analogous to estimating models with more flexible functional forms. The importance of these assumptions can be tested by estimating models with more hidden layers and higher-dimensional vectors. We employ a relatively simple deep learning model and the results about DLV's performance are likely conservative.

As previously stated, DLV gives us two advantages: incorporating a large amount of clinical data via vectorization, and deep learning. To verify whether the above 30% improvement is due to vectorization or simply to deep learning, we also apply a deep learning model using the original predictors in the CMS model and report our results in Column 2 of Table 1 Surprisingly, the deep learning model has an AUC of 0.57, slightly worse than the logistic regression. This is likely due to the fact that the variables included in the CMS model are selected based on their performance in a parametric model, specifically logistic regression. These findings suggest that DLV's superior performance is driven by its ability to incorporate high-dimensional data; however, model flexibility may be important in the variables not selected for the CMS model.

Table 1. Comparison of out-of-sample accuracy for 30-day AMI readmissions.

	(1)	**(2)**	**(3)**	**(4)**
Model	CMS	CMS-DL	Med2Vec	DLV
No. Variables	209	209	3,929	15,811
AUC	0.61	0.57	0.70	0.79
Estimator	Logistic regression	Deep learning without DLV	Deep learning with Med2Vec	Deep learning

Finally, we compare DLV's performance with existing models using Med2Vec (Choi et al., 2016b; Nguyen et al., 2016). While the vectorization is similar, they only vectorized medical codes (diagnosis codes and procedure codes), rather than the whole dataset. We therefore performed the vectorization for medical codes only. As reported in Column 3 of Table 1, Med2Vec leveraged 3,929 variables and achieved AUC of 0.70. Compared to the DLV's AUC at 0.79, it clearly demonstrates the great value of incorporating more data in improving prediction performance.

We further evaluate the model's performance on readmissions for all patients in 2008. In this example we estimate the model using an 80% sample of all hospital admissions and 59,685 variables vectorized by DLV. We attempt to use deep learning with and without DLV. Once again, we evaluate out-of-sample accuracy by calculating the AUC for a 20% testing sample.

Table 2. Performance of readmission predictive models across traditional machine learning, deep learning, and deep learning with DLV.

Model	Deep learning	Deep learning	Deep learning
Configuration	Without DLV	With Med2Vec	With DLV
Input variables	All numerical and dummy variables	Medical code vectors 12,934	All vectorized info 59,698
Performance (AUC)	Failed to load data#	0.63	0.77

#: Data too large to be loaded into a computer with 40 GB of memory.

As a comparison, the healthcare data are coded as binary variables for deep learning without the DLV model, giving us 59,698 variables (including 13 numerical). However, the deep leaning model could not be estimated, even with a relatively high-end computing facility.[9] In contrast, DLV runs smoothly and yields the best

[9] The random forest failed to converge after 100 hours of computing and the penalized regression failed to load the data on a computer with 40 GB of memory.

performance. The DLV model yields an AUC of 0.77 for the entire population, which is significantly better than the previous best performance in the literature: 0.55-0.65 as summarized by Kansagara et al. (2011). Again, we conducted vectorization following Med2Vec and generated 12,934 variables. Using exactly the same deep learning model, the performance of Med2Vec is 0.63. This again demonstrated the advantage of incorporating more data into the model, a key aspect of DLV compared to existing approaches.

2.6 Conclusion and Discussion

The goal of our proposed DLV approach is to facilitate the utilization and analysis of rich healthcare data. This approach leverages cutting-edge deep learning technologies, and it can be easily applied to most administrative claims or EHR datasets. DLV achieves its efficiency and comprehensiveness in healthcare data analytics via: 1) standardized data preparation, which reduces effort and requires less domain knowledge; 2) the format of output vectors, which makes downstream analyses much more efficient; 3) comprehensive dimension reduction, which retains the most information in low dimension vectors; and 4) ease of visualization.

DLV can be used to visualize connections among entities, unveil their relationships, and drive computation for deeper insights. Combined with healthcare domain knowledge, this user-friendly approach can expedite and expand knowledge discovery and hypothesis generation based on readily available real-world data. Future research should examine how the size, quality, and richness of different healthcare datasets might affect the performance of DLV and establish best practices for the standardization and wider use of this approach. We hope our work serves as a starting point of an exciting and productive new area in health services research.

We find that DLV can substantially improve clinical prediction. Our model achieves an AUC of 0.79 – significantly better than conventional risk adjustment models, which have an average AUC ranging from 0.55 to 0.65 (Kansagara et al., 2011). This is also slightly higher than the performance achieved by Rajkomar et al. (2018) (AUC of 0.75-0.76) using much more detailed EHR data to train the model on Google's internal distributed computation platforms.[10] DLV also reduces the dimensionality of this prediction problem, greatly reducing the computational resources required to estimate models.

As a pioneering exploration of applying deep learning for vectorizing all clinical factors, this paper does have some limitations. First, the data used in this study does not include all possible data elements in clinical data. With richer datasets, we believe that DLV can perform even better. Second, more healthcare use cases can be developed using DLV; however, given the limitation of the data, only visualization and readmission prediction were examined. Further studies can use DLV for more tasks in medical knowledge discovery and other predictive models.

[10] Note that the settings of this study and Google's are not quite comparable. But we hope to give audience a sense of the cutting-edge industry research. Though the nature of the prediction tasks could be different, Google uses more detailed measures and should perform better.

Chapter 3: How AI Plays Its Tricks: Understanding the Superior Performance of a Deep Learning-Based Approach in Predicting Healthcare Costs

3.1 Introduction

In recent years, the information systems (IS) community has been actively developing and evaluating new analytical tools to leverage big data in business (Pant and Sheng, 2015; Abbasi et al., 2016) across different domains (Meyer et al., 2014; Adamopoulos et al., 2018), including healthcare (Agarwal and Dhar, 2014; Abbasi et al., 2019). Deep learning, as one of the latest advances in machine learning, is showing great promise due to its superior performance in pattern recognition (LeCun et al., 2015; Najafabadi et al., 2015). However, the level of its applicability to healthcare cost prediction has not been studied. In this study, we aim to contribute to a better understanding of the performance of deep learning models in predicting patients' future costs and to provide guidance to practitioners and researchers alike.

Healthcare cost is an important research topic in the IS community. For instance, Dranove et al. (2014) examined the impact of electronic medical records (EMR) on hospital operating costs. Adjerid et al. (2018) used transaction cost economics to study the reduction of healthcare spending caused by health information exchanges (HIEs). Atasoy et al. (2017) further dug into the spillover effect of health IT investment among hospitals in same regions. As healthcare cost has become a prominent research focus, substantial effort has been exerted to make cost prediction accurate (Morid et al., 2017), and the resulting insights and improvements are utilized across multiple health applications.

Multiple stakeholders now recognize precise prediction of an individual patient's future costs as critically important (Bates et al., 2014). The Centers for Medicare and Medicaid Services (CMS), for instance, has a long history of using cost prediction models for patient risk adjustment in reimbursement (Wynand et al., 2000; Schone and Brown, 2013). Individual and group healthcare cost predictions are used to improve healthcare plan design and to decide which plans employers offer to their employees. Healthcare organizations also find prediction highly useful in designing targeted interventions to improve quality of care, especially in resource-constrained environments (Ganser et al., 2015; Anderson and Bjarnadottir, 2016; Srinivasan et al., 2017). As the US healthcare system moves from volume- to value-based reimbursement mechanisms, cost prediction is playing a central role in initiatives such as population health management and bundled payment. Finally, underwriters and benefit companies use healthcare prediction extensively to manage insured populations. Healthcare cost prediction, of high-cost patients in particular, can drive insights into the variability within the healthcare system, potentially leading to more efficient use of healthcare resources.

The majority of cost prediction models use past cost information, mostly obtained through claims data, as an important predictor of future healthcare costs (Bertsimas et al., 2008; Sushmita et al., 2015; Kim and Park, 2019). The claims data reflects each interaction with the healthcare system (provided the interaction is covered by insurance), so the claims cost information is therefore reflective of both disease burden and utilization patterns. For example, a patient seeking care through the primary care system will have a different cost pattern than a patient with the same condition who mainly seeks care through emergency departments. As a result, the cost information for each patient can be viewed as a sequence of time stamps indicating the patient's health status and utilization of the healthcare system over a period of time. This sequence reflects both patient disease progress and utilization patterns, which can both be valuable in predicting future costs.

Given this sequential nature of cost data, it is surprising that the past use of healthcare costs in the literature has mostly been limited to averages of different cost components (e.g., pharmacy costs and inpatient costs) (Duncan et al., 2016; Sushmita et al., 2015; Kuo et al., 2011; Frees et al., 2013; Morid et al., 2017) while ignoring the more complex patterns in the time series. A few papers have shown that detecting cost patterns can improve prediction (Bertsimas et al., 2008, Morid et al., 2019). Despite the potential usefulness of complex sequential patterns, as reflected in a recent extensive literature review by Morid et al. (2017), most papers still fall back on simple cost averages.

Methodological limitations are one possible reason for the under-utilization of sequential patterns in the existing literature. Traditionally, healthcare cost modeling has utilized regression models (e.g., Ash et al., 2000; Cumming et al., 2002; Zhao et al., 2005); however, the performance of such approaches has been found wanting (Schone and Brown, 2013), and by design regression models are unable to automatically incorporate complex cost patterns. Over the past decade multiple papers have tried to apply traditional machine learning models such as classification and regression trees and gradient boosting (Sushmita et al., 2015; Duncan et al., 2016). Yet these traditional methods often need manual feature engineering, which requires strong domain knowledge and which, even in expert hands, can only cover a limited scope of features.

In this study, we address this gap in the literature by proposing a new LSTM-based approach to better incorporate the sequential patterns in cost prediction in an automatic manner. Compared to standard approaches, deep learning models have significant potential for improving patient cost prediction: studies have shown that recurrent neural network models effectively leverage sequential information in several domains such as natural language processing (Cho et al., 2014; Devlin et al., 2018). Furthermore, deep learning models have the potential to provide this improved prediction accuracy without extensive prior feature engineering. Some previous studies have included neural networks among their supervised learning approaches (Morid et al., 2017), but the modeling has not been at the scale necessary to take advantage of the recent development of deep learning.

Despite the potential benefits of deep learning, there are reasons to doubt whether it necessarily outperforms traditional machine learning models. First, individual healthcare costs pose a challenging prediction problem due to both unforeseeable acute events (such as accidents) and other, sometimes unpredictable fluctuations. It is possible for a patient who is healthy for years, to be suddenly diagnosed with a severe condition that causes their costs to skyrocket. These sometime unexpected fluctuations pose a challenge for any predictive model and it is not clear whether deep learning models can outperform other methods utilizing sequential claims information. Second, deep learning requires a large training dataset. Researchers have not yet studied the appropriate length (time period) of sequential data that facilitates the optimal performance of deep learning approaches. As a result, little is known about the boundary conditions of the data size needed in order for the use of deep learning to be advantageous.

We therefore aim to 1) develop a deep learning model for individual patient cost prediction, and conduct a series of rigorous tests to benchmark its performance; and 2) generate insights that promote a better understanding of deep learning models' advantage in incorporating time series information for precise cost prediction. To our knowledge this is the first study to introduce the deep learning approach for individual patient cost prediction. We suggest a novel way to examine the performance of the deep learning model, demonstrating how it outperforms traditional machine learning models when the fluctuation in the input data is high. Our research also holds significant value for the practice of cost prediction through the introduction of best practices and suitable architecture.

The remainder of the chapter is organized as follows. In Section 3.2 we describe the data and the experimental setup, followed by model description and performance results in Section 3.3. We analyze the model's performance in Section 3.4, provide practical guidelines in Section 3.5 and conclude in Section 3.6.

3.2 Data Sources and Data Modeling

This study is based on claims data which has a long tradition of use for healthcare research. Its efficacy for both medical and healthcare research is due in part to its large scale, potentially long follow-up time, near real-time availability, and relatively low cost compared to other sources. The claims data is generated during the interaction between patients and the healthcare system, and it records the objective information into key information such as diagnosis (up to 10 per visit), procedures, prescription information, provider and point of service details and demographic information. Based on these underlying data, both disease burden and utilization patterns can be derived. While claims data has well-known limitations -- for example, a lack of diagnostic and prognostic information when compared to medical records, as well as variability due to differences in coding practices -- it continues to be an important source of healthcare research data (Bjarnadottir et al., 2016).

The claims dataset used in this study is de-identified HIPAA-compliant insurance claims data for 1,434,912 residents of nine counties in upstate New York (the greater Rochester area). The repository includes claims records from employer-insured, Medicare, and Medicaid members, which is all aggregated by a central agency, the

Finger Lakes Health Systems Agency. The dataset contains much of the standard claims information, including both medical and pharmacy claims, along with demographic information such as age, gender and payer type. The data corresponds to 62,406,379 healthcare encounters that took place between 2007 and 2013. For the purpose of this study, we consider the members' total cost, which is the sum of costs for hospital and other medical services and pharmacy costs. On average, a patient had 43.5 visits during the study period and $252.50 was paid for each encounter.

Table 3. Summary statistics of key cost and demographic variables.

	mean	std	min	q1	median	q3	max
Cost in 2007	$2756	$5334	$0	$467	$1060	$2590	$157574
Cost in 2008	$3101	$6054	$0	$513	$1169	$2880	$288186
Cost in 2009	$3199	$6344	$0	$510	$1169	$2944	$426721
Cost in 2010	$3430	$6778	$0	$545	$1258	$3155	$249050
Cost in 2011	$3689	$7328	$0	$593	$1347	$3329	$348876
Female	59.62%	49.07%	-	-	-	-	-
Age (start of 2012)	47	25	1	22	51	67	89

The goal of this study is to provide accurate future individual cost predictions. As only partial data is available for 2013, we use the members' information from 2007 through 2012. To ensure that we are making cost predictions for members who are actively enrolled during the outcome period, we select 2011 as the outcome year and only include patients who have at least one claim in 2012 (meaning that they did not drop out in the middle of 2011). As a result, our final dataset contains 367,523 patients. We will use patients' information from 2007 to 2010 to predict their total cost in 2011.

The input features include demographic information (birth year and gender) and annual cost for each of the past four years. For the past 48 months, the average cost per encounter for each month, the number of encounters each month, the average number of claims each month, and the average number of claims paid each month are also incorporated. As a result, we have a total of 198 input features. The dependent variable is the member's total cost in 2011.

We randomly split the data into training (80%), validation (10%) and testing data (10%). We analyzed the cost distributions and the distributions of demographic variables to ensure a balanced data split across the three subsets. All training was performed on the training data, model selection utilized the validation data, and the testing data was only used at the end of the experiment to estimate the out-of-sample performance. The performance measure, the mean absolute error, is reported on the testing data.

3.3 Models and Performance

We discuss the development and the performance of our RNN-based deep learning model below and we compare its performance with simple baselines and traditional machine learning models.

3.3.1 Deep Learning Models

When constructing a deep learning architecture, there are no mathematical formulas to fall back on, nor are there universal comprehensive guidelines. The appropriate architecture is determined by the nature of the data at hand as well as the data mining task, and the architecture is constrained by the available computational power. We point the interested reader to two introductory works by Andrej Karpathy in which he shares his experiences and suggests best practices (Karpathy 2019a; Karpathy 2019b).

We utilize RNN architecture for our deep learning model; specifically, we use long short-term memory (LSTM) units to capture the sequential claims information. This LSTM model outperformed other RNN models including standard RNN and Gated Recurrent Unit (GRU) in our experiments. The LSTM units have a complex design that incorporates both short memories (information in the last period) and long memories (information in the very early periods) into long sequences. This ability to "remember" long sequences is key to LSTM's advantage and has proven to be a powerful property in number of applications. For example, LSTM-based RNN models are the current technology leaders in Natural Language Processing (NLP) fueling translations and chatbots. It is also the method used by IS researchers for complex sequential data analytics (Liu et al., 2019). Compared to traditional models like hidden Markov models, which incorporate a finite number of prior choices, LSTM has fewer limitations on memory and does not require a prior density distribution of the inputs. Another advantage of LSTM is its ability to manage the gradient vanishing problem, which has hindered successful training of many other

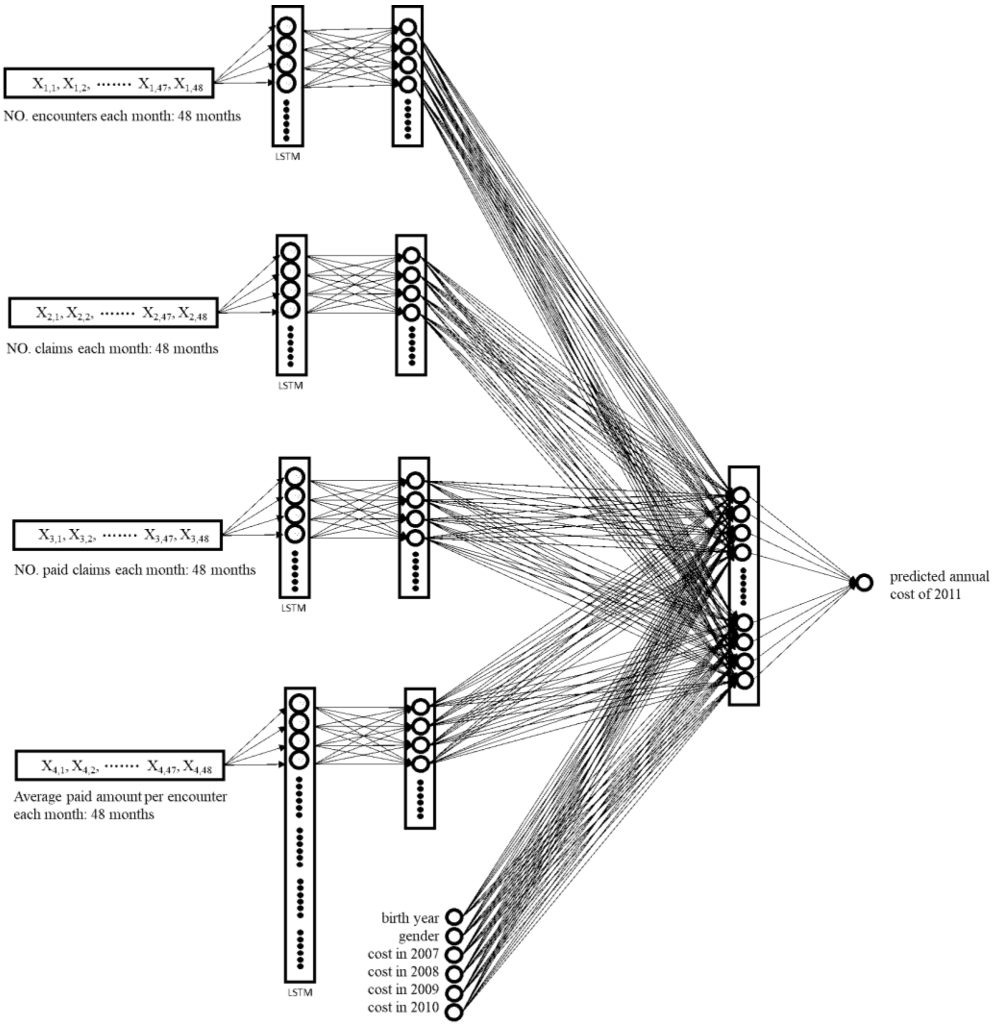
RNN models. Given the high complexity in healthcare cost prediction, an LSTM model may provide better performance than the standard RNN model due to the aforementioned advantages.

We build an LSTM model (hereafter noted as LSTM-DL) to take advantage of the sequential information in the time-series data. For the 48 months in our training data, there are four sequences: (1) the number of encounters of each month; (2) the average cost per encounter in each month; (3) the number of total claims each month; and (4) the number of paid claims each month. We could combine these four sequences into a single time series (and input the data as a four-dimensional vector), or keep the four time series separate. By combining the four sequences into one, we could potentially leverage the interactions among different sequences for better prediction. Alternatively, we could keep the four sequences separate to reduce noise (potentially helping the model to learn unique patterns in each sequence), and then integrate the output using a hidden layer in the later stage. There is no clear guidance suggesting that one approach is always better than the other. We adopted the parallel design due to its performance advantage (a detailed comparison is provided in Appendix A).

The resulting architecture is presented in Figure 6. The monthly payment information (average cost per encounter, 48 features) is input to an LSTM layer with 400 LSTM units. The LSTM output is then fed to a fully-connected layer with 200 nodes. In parallel, the number of encounters each month, number of total claims each month, and number of paid claims each month are each input to a separate LSTM layer followed by a fully-connected layer, each with 50 units. The larger number of units for the average encounter costs reflects the importance of cost information in healthcare cost prediction. The four separate LSTM outputs are then merged, together with the sex, birth year, and the annual total cost for each member. The merged data goes through another fully-connected layer with 300 nodes and then generates the final output layer that generates the predicted cost. We directly use the average absolute error as the loss function. An Adam optimizer (Kingma and Ba, 2014) is adopted to optimize the model parameters on the training data.

The final values of the parameters in the trained model were selected based on the model's performance on validation data: when the model's performance stabilized, we then selected the epoch (iteration) with lowest prediction error on validation data. The performance becomes stable after epoch 217. The average performance on validation data from epoch 217 to 226 is $2,560 (standard deviation is 0.23, indicating that the model is stable). The corresponding average performance on testing data is $2,630 (standard deviation is 0.50). In what follows we use the LSTM model trained after 220 epochs (which performed the best on the validation data) and with a testing error of $2,630.

During our model development and as a post comparison, we contrasted the performance of the LSTM-DL model with the performance of other LSTM architectures. To create these variants, we altered the basic architecture in a number of different ways. First, we merged the four parallel layers and fed the merged sequence into a single LSTM layer (as discussed above); second, we de-emphasized the cost information by changing the number of LSTM units for different input information; and finally we experimented with removing the fully-connected layers. All these variants of the LSTM architecture perform worse than the proposed model, with the difference in MAE ranging from $4 to $9. Detailed results are reported in Appendix A.



$X_{1,1}, X_{1,2},$ ······· $X_{1,47}, X_{1,48}$
NO. encounters each month: 48 months
LSTM
$X_{2,1}, X_{2,2},$ ······· $X_{2,47}, X_{2,48}$
NO. claims each month: 48 months
LSTM
$X_{3,1}, X_{3,2},$ ······· $X_{3,47}, X_{3,48}$
NO. paid claims each month: 48 months
LSTM
$X_{4,1}, X_{4,2},$ ······· $X_{4,47}, X_{4,48}$
Average paid amount per encounter each month: 48 months
LSTM
birth year
gender
cost in 2007
cost in 2008
cost in 2009
cost in 2010
predicted annual cost of 2011

Figure 6. The architecture of the neural network.
Note: Each node (represented by a cycle) in an LSTM layer is a standard LSTM node that takes in the whole sequence (univariate time series of 48 time steps) of input data.

3.3.2 Benchmark Models

In addition to comparing our LSTM-DL model with other possible LSTM architectures, we compared its performance with baselines and with several traditional machine learning models. Since healthcare costs have been shown to offer a strong summary of a member's health and to be a strong prediction of future costs, for our baseline model we use the average of each member's previous years' costs as the prediction for his/her cost in 2011. In other words, we use the past one year cost and the average costs over the past two years, past three years, and past four years as baselines. In the testing data, the average absolute errors of the baseline predictions are $3,239, $3,395, $3,638, and $3,480, respectively.

We then fit a number of traditional machine learning models to the data, including linear regression, regularized regression (LASSO as well as ridge regression), and random forest (RF). All the machine learning models were trained on the same training data as the LSTM-DL model (including the same features). Then, for methods having multiple configurations and training parameters, the model was selected based on the validation data.

We performed a grid search[11] to tune any model parameters. A grid search aids in parameter tuning by running all possible model configurations from possible parameter options and selecting the best combination (based on the validation data).

[11] We used grid search instead of random search or Bayesian model-based optimization because the hyperparameter tuning is for the baseline models, and it is essential to reliably cover the possible combinations. The girds were made small enough to incorporate the possibilities. For baseline group, computational efficiency is not a major concern.

For both LASSO and ridge regression, α is set to seven different values: 0.001, 0.01, 0.1, 1, 10, 100, and 1,000. For the RF model, we used different fine-tuned model parameters using 10 trees, then expanded the size of the forest to 500 trees.

Five parameters were tuned for the RF model: the maximum depth of the tree (3, None), the maximum number of features (1, 3, 10), the minimum number of observations for a node to be split (2, 3, 10), the minimal number required for a leaf-node (1, 3, 10), the objective function (mean squared error, mean absolute error), and the bootstrap option (On, Off). As a result, 216 RF models were generated.

3.3.3 Performance Comparison

Model performance is summarized in Table 4. The mean absolute error of the linear regression model is $3,165, almost the same as the error of ridge regression ($3,165, with α = 1000) and very similar to that of the LASSO regression ($3,158, with α = 100). The best performance for the RF model corresponds to no restriction on the maximum depth of the individual trees, the maximum feature set to 3, the minimum number for a split set to 2, the minimal number required for a leaf set to 10, objective function set to the mean absolute error and bootstrapping utilized. The corresponding absolute error is $2,715, significantly lower than that of the regression models. In addition, we compared the LSTM-DL model to a standard RNN, one-dimensional convolutional neural networks (CNN1d), and two-dimensional convolutional neural networks (CNN2d) using different architectures (including the architecture of the proposed model). The difference between the LSTM-DL model and other deep learning models ranged from 1.0% to 6.0% (details are provided in Appendix A). Among all the models, the LSTM-DL model has the best performance.

We use deep learning for regression; the RNN architecture is utilized. We use LSTM units in deep learning to capture the sequential information. Using exactly the same input information (198 features), the deep learning model achieved the best performance.

Table 4. The mean absolute error (and the mean absolute percent error in parentheses) of all cost prediction methods.

Model	Testing data MAE (MAPE*)	Validation data MAE
Baseline	$3239 (203.27%)	$3156
Linear Regression	$3165 (248.06%)	$3121
Ridge Regression	$3165 (248.05%)	$3121
LASSO Regression	$3158 (252.45%)	$3113
RF	$2715 (135.63%)	$2643
LSTM-DL	$2630 (105.68%)	$2559

*The MAPE calculation excludes two members with zero cost

By design, the LSTM-DL model is significantly more complex that the other machine learning approaches, especially the regression models. If cost fluctuation is important, the regression models are unable to capture that information directly from the time series input. However, feature engineering can incorporate some of the characteristics of the time series, improving the machine learning models. As a step in that direction, we include the standard deviations of each sequence as additional input features. The performance of the regression models improved: the linear regression model from $3,165 to $3,155, ridge regression from $3,165 to $3,155, and LASSO regression from $3,158 to $3,152. Interestingly, the performance of RF remains stable. The improvement in the three regression-based models indicates that incorporating additional variables through key feature engineering could improve their performance. In contrast, one of the advantages of deep learning is its lower need for feature engineering.

The mean of the outcome variable on the testing sample is $3,687. As a result, the LSTM-DL error corresponds to 71.34% of the mean. We heed many caveats when comparing predictive performance across papers that utilize different data; however, this percentage compares favorably with published papers that report the mean absolute error of annual cost prediction using large study cohorts: 93% by Cumming et al. (2002), 98% by Powers et al. (2005), 78.8% by Bertsimas et al. (2008), 75% by Kuo (2011), 146.87% by Frees(2013), and 80.0% by Ramamurthy et al. (2017). While one cannot directly compare predictive performance across datasets, this indirect comparison can serve as a clue to the relative performance of the different approaches in the published literature. We further note that some of these studies use significant (manual) feature engineering in their machine learning models.

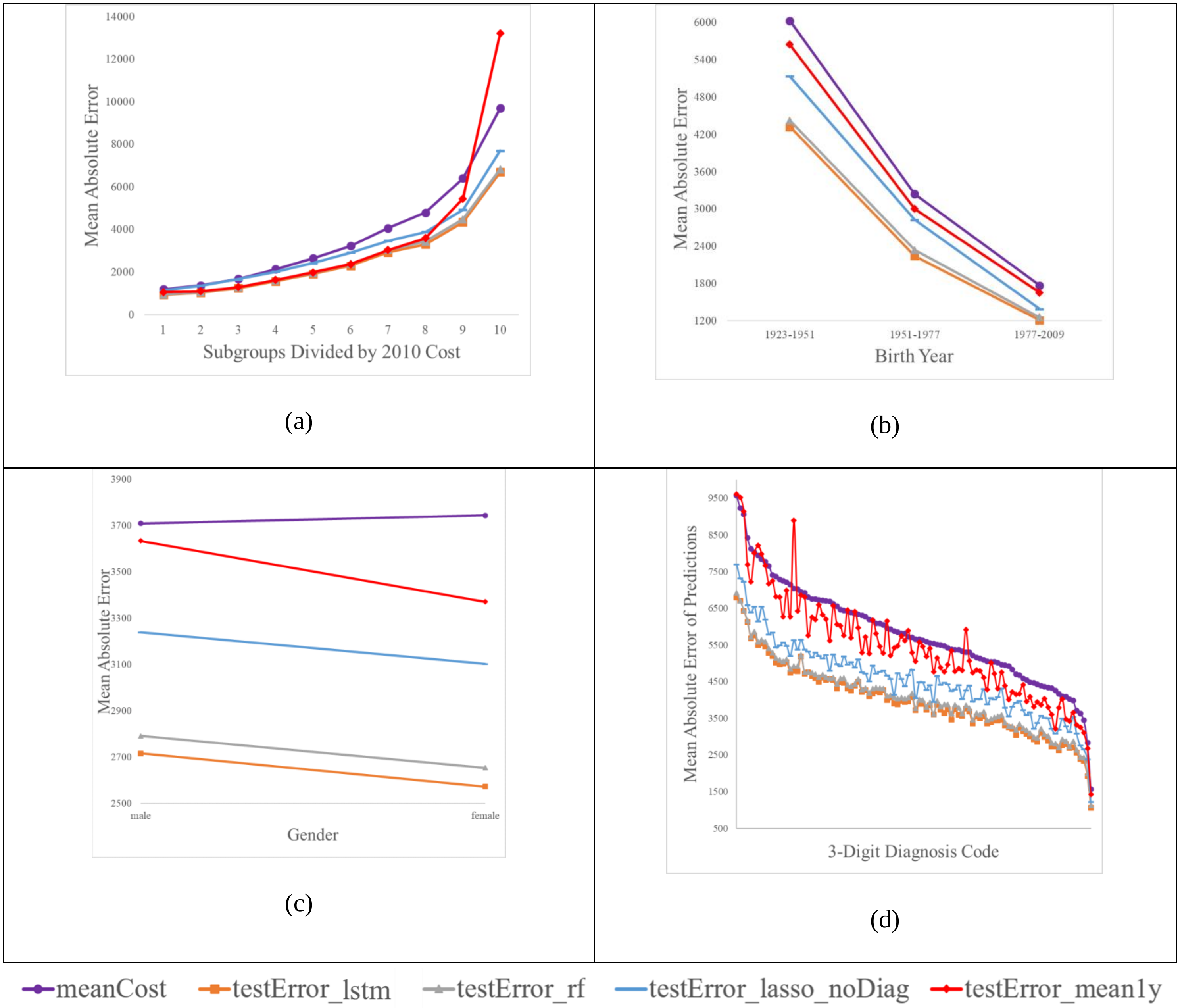
Subgroup Performance. To further examine the robustness of LSTM-DL's performance advantage, we partition the data based on healthcare costs, age, gender and diagnoses for subgroup comparison. Figure 7 summarizes this analysis using members in the testing dataset and their outcomes. In Figure 7 (a), patients are divided into ten equal-sized groups according to their cost in 2010. We observe that the LSTM model is the best model across all the patient groups, but most significantly, it performs the best for high-cost patients (groups 9 and 10). In Figure 7 (b), we compare the performance across patients based on birth year, dividing the patients into three equally sized age groups. Again, while the LSTM-DL model performs best across all groups, the difference is the largest for the oldest members (group 1); we again note that the advantage of LSTM-DL is bigger in the high-cost group (oldest). In Figure 7 (c) we note that the advantage of LSTM-DL is similar for the two genders.

In Figure 7 (d), we compare performance by diagnosis across patients with the 100 most common diagnoses. The minimal frequency is 647/36753 (anemias); the maximal frequency is 9953/36753 (nonallopathic lesions). Note that the same member may be present in multiple subgroups. Again, we observe that LSTM achieves the lowest error consistently across most diagnoses, with the exception of only one single diagnosis group, “malignant neoplasm of prostate” (with a mean cost of $9,226), where RF outperforms LSTM-DL.

We further studied model performance on the dataset broken down by chronic disease burden with each member's chronic diseases identified using the chronic condition coding scheme from AHRQ’s Healthcare Cost and Utilization Project[12]. The performance of the model for members with at least one chronic disease was consistently better than that of the benchmark models. Similarly, we broke down the

[12] Chronic diseases are identified according to the ICD9 diagnosis codes using the translation file downloaded from the AHRQ’s H-CUP project via this link: https://www.hcup-us.ahrq.gov/toolssoftware/chronic/chronic.jsp.

dataset by body system affected (for example the circulatory, the respiratory or the digestive systems); here again, LSTM-DL consistently outperformed the other models.



(a)

(b)

(c)

(d)

Figure 7. Model performance as a function of 2010 costs, age, gender and diagnosis.

Overall, our subgroup analyses confirm that for most subgroups, LSTM-DL consistently achieves better prediction performance than other models.

3.4 Understanding the Role of Fluctuations in Cost Prediction

In this section we present a novel approach to understanding the prediction improvements of the LSTM-DL model: specifically, we use regression modeling to understand what drives errors and differences in performance. Starting with a study of the performance of the LSTM-DL model as a function of the variability in members' monthly costs, we apply this approach to gain insight into the role of such fluctuations in the data and how they affect model performance. We further decompose overall fluctuation into three components: trend, seasonality, and the remainder. We then study the LSTM computational units and their outputs for a more intuitive understanding of how the model responds to different levels of fluctuation.

To avoid selection bias, we interpret the deep learning by not choosing the best on our testing data, which may be less generalizable. We use the trained deep learning model after 210 epochs, whose error is $2597.17. It is a stable model because the two epochs before and after it are all at the same error level: prediction error is $2598.26 for Epoch 208, $2599.02 for Epoch 209, $2599.75 for Epoch 211, and $2597.68 for Epoch 212.

3.4.1 How Fluctuation Affects Prediction Accuracy

We focus on the fluctuation in the time series of healthcare costs; the monthly costs in the past 48 months are used to calculate the fluctuation, where members are indexed by *i*, and time period is indexed by *t*:

$$Fluc_i = \sqrt{\frac{\sum_{t=1}^{47}\left(Cost_{i,t+1} - Cost_{i,t}\right)^2}{47}} \qquad \ldots\ldots(3)$$

We then define the dependent variable *modelErrAbs*, as the absolute error of each of the approaches (LSTM, the strongest baseline, linear, LASSO and ridge regression, and RF). For instance, *LSTMErrAbs*, is the LSTM model's performance (measured by the absolute error).

$$modelErrAbs_i = \beta_0 + \beta_1 Fluc_i + \sum_{j=2}^{4} \beta_j X_{ij} + \varepsilon_i \qquad \ldots\ldots(4)$$

Table 5. Impact of fluctuation on absolute model performance

	(1) LSTMErrAbs	(2) Past1yearErrAbs	(3) LinearErrAbs	(4) LASSOErrAbs	(5) RidgeErrAbs	(6) RFErrAbs
Fluctuation	-0.21***	1.27***	0.24***	0.25***	0.24***	-0.15***
	(0.01)	(0.02)	(0.01)	(0.01)	(0.01)	(0.01)
Paid2011	0.86***	0.64***	0.67***	0.67***	0.67***	0.83***
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Birth Year	5.51***	0.88	-10.47***	-10.94***	-10.47***	3.51***
	(0.36)	(1.06)	(0.6)	(0.59)	(0.6)	(0.41)
Gender	-87.45***	-125.05***	-105.58***	-82.51***	-105.54***	-83.54***
	(8.39)	(24.54)	(13.86)	(13.79)	(13.86)	(9.43)
Constant	-11193.73***	-1830.74***	21022.13***	21918.02***	21018.81***	-7134.94***
	(711.35)	(2081.47)	(1175.41)	(1169.87)	(1175.5)	(799.88)

Notes: Standard errors in parentheses
*Significance is indicated by *** p<0.01, ** p<0.05, * p<0.1*
The columns headed LSTMErrAbs, Past1yearErrAbs, LinearErrAbs, LASSOErrAbs, RidgeErrAbs, and RFErrAbs, show the absolute prediction errors of LSTM-DL, past one year, linear regression, LASSO regression, ridge regression, and RF, respectively.

Table 6. Impact of fluctuation on model performance difference

	(1) LSTM-Past1year	(2) LSTM-Linear	(3) LSTM-LASSO	(4) LSTM-Ridge	(5) LSTM-RF
Fluctuation	-1.48***	-0.45***	-0.46***	-0.45***	-0.06***
	(0.02)	(0.01)	(0.01)	(0.01)	(0.00)
Paid2011	0.22***	0.19***	0.19***	0.19***	0.03***
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Birth Year	4.63***	15.99***	16.45***	15.98***	2.00***
	(0.98)	(0.40)	(0.39)	(0.40)	(0.21)
Gender	37.60*	18.13*	-4.94	18.09*	-3.91
	(22.72)	(9.29)	(9.17)	(9.29)	(4.89)
Constant	-9362.99***	-32215.86***	-33111.75***	-32212.54***	-4058.79***
	(1926.64)	(787.76)	(777.82)	(787.59)	(414.65)

Notes: Standard errors in parentheses
*Significance is indicated by *** p<0.01, ** p<0.05, * p<0.1*
The columns headed LSTM-Past1year, LSTM-Linear, LSTM-LASSO, LSTM-Ridge and LSTM-RF show the differences between the LSTM prediction errors and the baseline, the linear regression, the LASSO regression, the ridge regression and the RF prediction errors, respectively

Equation 4 focuses on model performance (the absolute error), where $modelErrAbs_i$ is the absolute error of a model, $Fluc_i$ is the fluctuation derived from the monthly cost of each member, and $X_{ij}s$ are control variables, gender, birth year, and patient healthcare cost in 2011. The significance of β_1 indicates whether the model's performance is significantly impacted by the monthly cost fluctuations after accounting for age, gender and 2011 costs: a negative sign of β_1 indicates that fluctuation leads to lower absolute error and thus better prediction. The results are reported in Table 5.

The coefficient of $Fluc_i$ in the LSTM-DL model (Column 1 in Table 5) is significantly (p<0.01) negative. This finding demonstrates that LSTM-DL generates significantly better predictions when the monthly costs have high fluctuation even after accounting for overall healthcare cost in the outcome year, birth year, and gender. Interestingly, in most other traditional prediction models (past one year, linear regression, LASSO regression, and ridge regression) the coefficients of fluctuation are all significantly ($p<0.01$) positive, meaning that these models' performance is degraded by the fluctuations (Columns 2-5 in Table 5). The only exception is RF (Column 6), which can leverage the fluctuation to improve prediction, although the magnitude of improvement is smaller than the magnitude of improvement attained by LSTM-DL. This finding is also consistent with our exploration in Section 3.3.3, in which we found that using the standard deviations as additional inputs can improve the performance of regression models but not RF. The results here again indicate that regression models cannot utilize fluctuation information by themselves while RF can leverage fluctuation information to some extent. Overall, the results in Table 5 confirm that LSTM-DL shows superior performance when facing higher fluctuations, while most other models' performance declines.

Next, we examine how the performance advantage of LSTM-DL varies when facing different levels of fluctuation. We set the dependent variable, $\mathit{LSTM\text{-}Model}_i$, as the difference between the absolute error obtained by the LSTM-DL model and that obtained by a machine learning method. For example, when comparing LSTM-DL with linear regression, the dependent variable becomes $\mathit{LSTM\text{-}Linear}_i$, which equals LSTMErrAbs_i - LinearErrAbs_i. Since we are comparing the absolute prediction errors, a negative value of the above variable means a reduction in prediction error, and therefore a greater performance advantage of LSTM.

$$LSTM\_Model_i = \beta_0 + \beta_1 Fluc_i + \sum_{j=2}^{4} \beta_j X_{ij} + \varepsilon_i. \quad ......(5)$$

In Equation 5, β_1 indicates how fluctuation is related to the LSTM-DL model's relative advantage after controlling for costs, age and gender. A negative β_1 means that when facing higher monthly cost fluctuations, LSTM's performance advantage (in reducing the prediction errors) is even greater. As reported in Table 6, the coefficients of fluctuation for the performance difference between the LSTM-DL model and all other models are significantly ($p<0.01$) negative, indicating that the superior performance of LSTM-DL to other methods can in each case be significantly connected to members with high fluctuations. Therefore, we conclude that compared to all other methods we have investigated, LSTM-DL shows the greatest advantage in a context of highly-variable observations.

Further Explorations of Fluctuation: We further study the impact of variability using different definitions of fluctuation. First, we decompose the cost series into its time series components (trend, seasonality and the remainder). Due to their high correlations, we run a separate analysis on each component. We consistently find that the higher the fluctuations in each component, the bigger the advantage of the LSTM-DL model (after controlling for the member's overall cost, age and gender). The estimate of the impact is similar across the three components but highest for trend. Second, we alter the definition of fluctuation to only reflect large increases in costs (defined as at least 100% and at least $50). Using this definition, we again consistently find that the higher the fluctuation, the bigger the advantage of the LSTM-DL model. Details of these additional analyses can be found in Appendix B.

Overall, our findings indicate that LSTM models can do much better than other models when facing high fluctuation in the sequential data of patient costs. It is surprising to see that while greater fluctuation makes other traditional models deliver less precise predictions, it actually makes LSTM models deliver unequivocally better performance.

One possible explanation of this finding may lie in the information offered by fluctuations. The fluctuations in an input cost sequence reflect cost changes caused by many complex patterns. However, most of these patterns are not explicit and are hard to recognize. Since traditional machine learning models cannot easily recognize these patterns, the fluctuations are only noise to them. LSTM models, on the other hand, can leverage the complex patterns contained in the input sequence: to them, the fluctuations are not noise, but useful inputs. Therefore, we believe that LSTM models outperform other models through utilizing the so-called noise.

3.4.2 Unfolding LSTM Processing

To understand how the LSTM-DL model leverages fluctuations to a greater extent than other models, we focus on the LSTM computational units and specifically on the

400 LSTM units that process the cost information and visualize their output[13] for different input sequences. The LSTM outputs will be passed on to the next fully-connected layer (i.e., the 200 nodes in the layer). More specifically, the last value of the output sequence will be passed on (which is a common practice in recurrent neural network design because the last value incorporates the information from all previous values in the output sequence). Each node in the fully-connected layer weights each of the 400 inputs for further processing.

To gain a more intuitive understanding of how LSTM incorporates fluctuation information in its prediction, we randomly select a patient with high fluctuation and a patient with low fluctuation from the testing sample. We then feed their data to the trained model and visualize the output of the LSTM units. In Figure 8, the top row reflects the high-fluctuation patient, and the bottom row the low-fluctuation patient. The first column (A) shows the output of the LSTM-DL model. In each figure, the red line is the actual cost sequence (i.e. input to the LSTM). Each of the 400 green-blue lines reflects the outputs of a single LSTM node. The green-blue color represents the average weight given by the nodes in the following fully-connected layer to the last value in the sequence. The shape of each line reflects the output sequence of the corresponding LSTM unit. In comparing A1 and A2 in Figure 8, it is evident that the trained LSTM-DL model is capable of responding differently to the fluctuation in the

[13] For each individual patient, we input the cost sequence of the past 48 months into the LSTM layer. The cost sequence is fed into all 400 LSTM nodes in the layer. Consequentially, by construction each of the LSTM nodes will output a sequence of the same length. More specifically, each LSTM node processes the input sequentially and calculates an output sequence one value at a time, with each subsequent output value dependent on those before it. In other words, each value in the output sequence is dependent on all the previously calculated output values. Finally, the sequence is passed on to the subsequent layer; however, as is typical in these models, only the last value in the output sequence is utilized by the next layer.

cost sequence: the high fluctuation in the first patient triggers much stronger output. The LSTM-DL model is able to translate the fluctuation to measurable signals and pass them on to the next layers. This is largely consistent with our findings regarding the LSTM-DL model's ability to make use of fluctuation patterns.

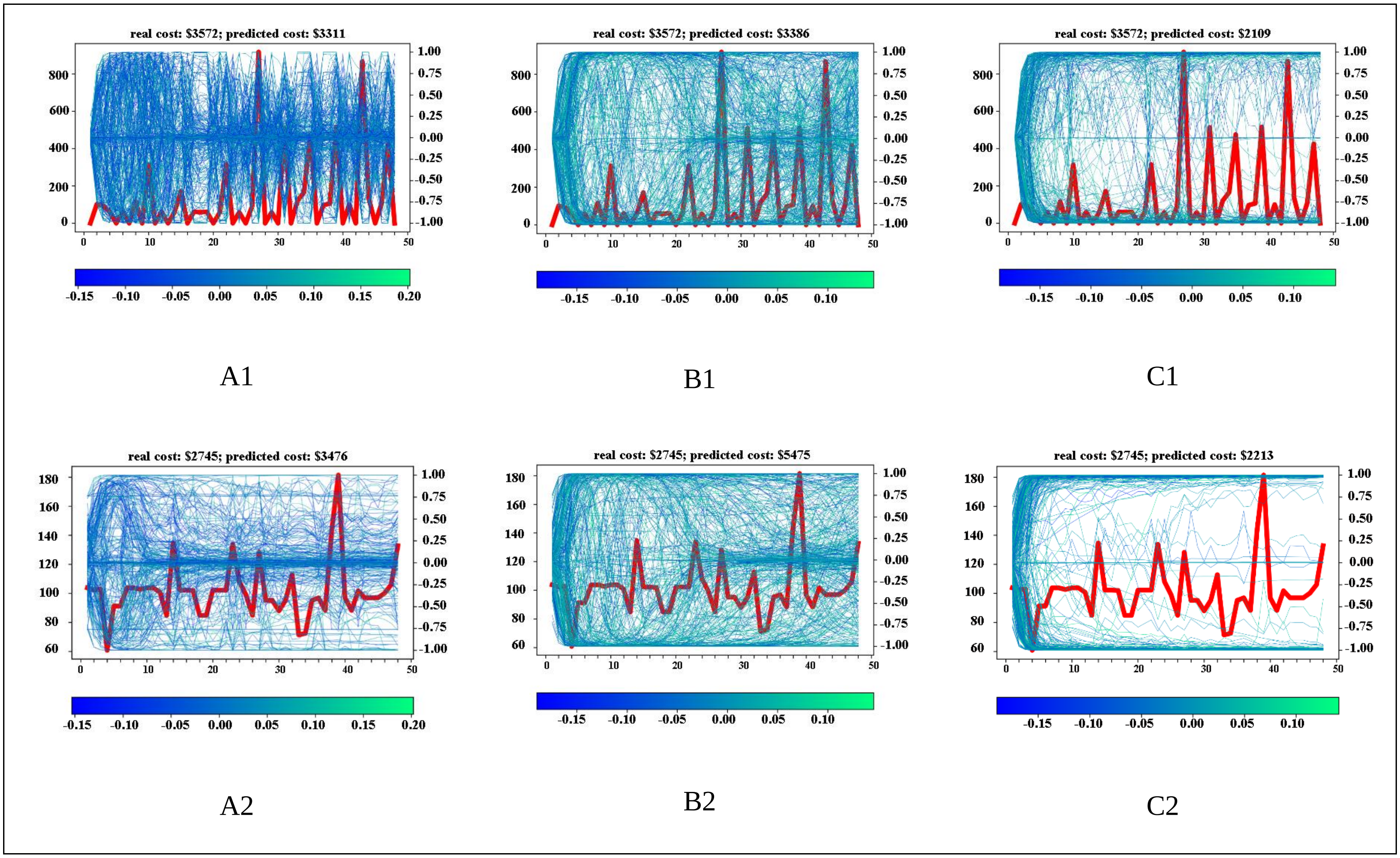


Figure 8. Real patients with high and low fluctuation.

Note: These patients are from the testing data, which was not exposed to any of the models during model training.

We next manipulate the knowledge in the learning step and train two models: LSTM-DL-high, which was trained using only members with high fluctuation data (top 25%), and LSTM-DL-low, which was trained using only data from members with low fluctuation data (bottom 25%). We then examine the output of the LSTM layers of these two models to study the reaction to high- and low-fluctuation cost sequences. LSTM-DL-high (Figure 8 B1) shows a strong reaction to the fluctuations and output sequences, with even larger reactions when compared to LSTM-DL in A1; LSTM-DL-low reduces the fluctuations in the raw sequence and passes relatively flatter signals to the next layers (Figure 8 C1). Similar patterns are observed across the low-fluctuation patient input (B2 and C2 in Figure 8). There are significant distinctions between the three models, suggesting that the ability to identify and amplify useful fluctuation signals is not inherent to LSTM itself. Rather, this is gained during the learning process.

In addition to the real patient data, we also construct input sequences using common functions such as constant, linear, and quadratic functions. The insights obtained are similar and are reported in Appendix C.

3.5 Sensitivity to Sample Size and Heterogeneity

While we show solid evidence that the LSTM-DL model outperforms other models, this performance is achieved by fully utilizing the 48 months of data and the entire data set. In reality, however, high turnover is typical in claims data, as members may switch insurance plans or move causing discontinuity in claims histories. When applying machine learning models, then, decision makers often face the tradeoff between a shorter time sequence with a larger sample size or a longer time sequence but with fewer patients meeting the inclusion criteria.

In this section, we examine the sensitivity of LSTM-DL's advantage with respect to these two key modeling decisions: the duration of the observation period (or, equivalently, the sequence length) and the dataset size. Furthermore, given that there is substantial heterogeneity among patients in their degree of fluctuation, we explore how segmenting, i.e. building different LSTM models for different patient groups, can benefit predictive performance. Findings from these explorations provide useful guidance for the practical application of LSTM models in practice.

3.5.1 Input Data

Length of the Observation Period. One may hypothesize that the longer the observation period, the better the achievable prediction. Yet at the same time, recent information is more relevant to cost prediction (as a member´s current health status is the most relevant for future health expenditures). To examine the impact of the length of the observation period, we vary the observation period for the same cohort of patients by duration: 36, 24, 12, 6, and 3 months, and we contrast the performance with the performance obtained using full 48-month period model. As summarized in Table 7 and visualized in Figure 9, the performance of the LSTM model is remarkably robust to the sequence length. As the observation period is shortened, its performance deteriorates only mildly. For example, reducing the observation period from 48 months to 36 months results in a small performance drop of only $6. We find that LSTM's performance is significantly reduced when the sequence length is 6 months or shorter. Yet it is worth noting that even if only 6 months of data are used, the LSTM model's performance is still better than all other models (including RF). This analysis also leads to the following observation of the RF model's performance: it deteriorates as the input length increases. Given that RF's performance is better than LSTM-DL using the most recent 3 months, extending the sequence length hurts RF's performance (while it benefits LSTM-DL). This is again reflective of the fact that the most recent healthcare costs are most representative of a member's health status, and the fact that RF's performance often deteriorates as the number of (less relevant) features grows. In contrast, the LSTM model can selectively (through its forget gate) "forget" the noise in the early periods. This finding highlights the advantage of the LSTM model as well as its ability to utilize longer inputs.

Table 7. Model performance and input length.

Number of Months	3	6	12	24	36	48
Linear Regression	$3171	$3165	$3162	$3163	$3164	$3121
Ridge Regression	$3171	$3165	$3162	$3163	$3164	$3121
LASSO Regression	$3171	$3163	$3161	$3160	$3158	$3113
RF	$2676	$2680	$2686	$2704	$2712	$2715
LSTM-DL	$2703	$2675	$2655	$2640	$2636	$2630

Table 8. Model performance and training dataset size.

Training Size	6.25%	12.5%	25%	50%	100%
Linear Regression	$3243	$3197	$3182	$3177	$3165
Ridge Regression	$3230	$3193	$3180	$3176	$3165
LASSO Regression	$3178	$3166	$3165	$3167	$3158
RF	$2743	$2737	$2724	$2717	$2715
LSTM-DL	$2793	$2839	$2720	$2642	$2630

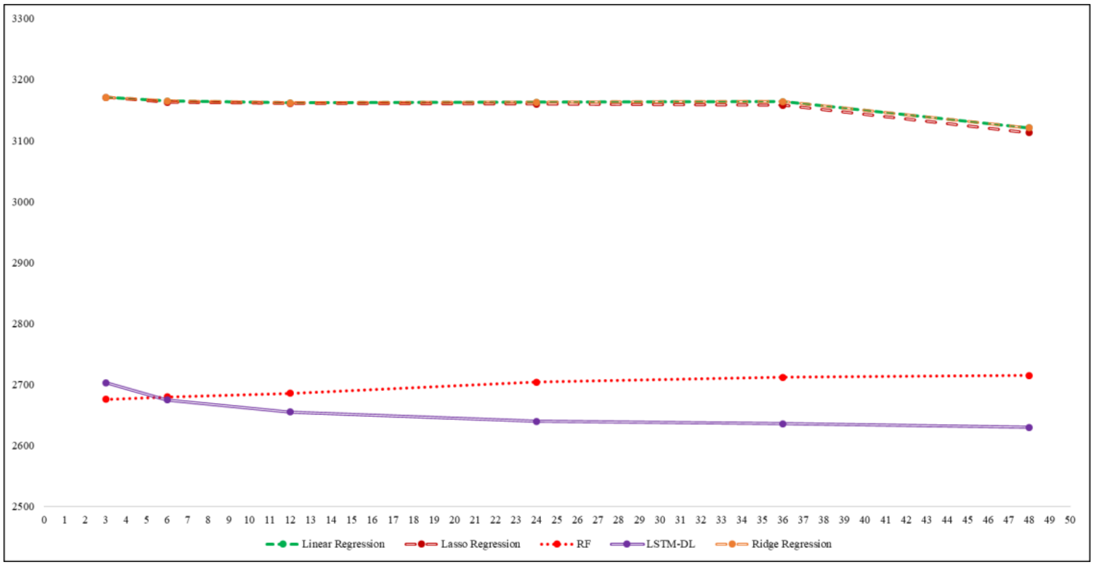


Figure 9. Model performance and input length.

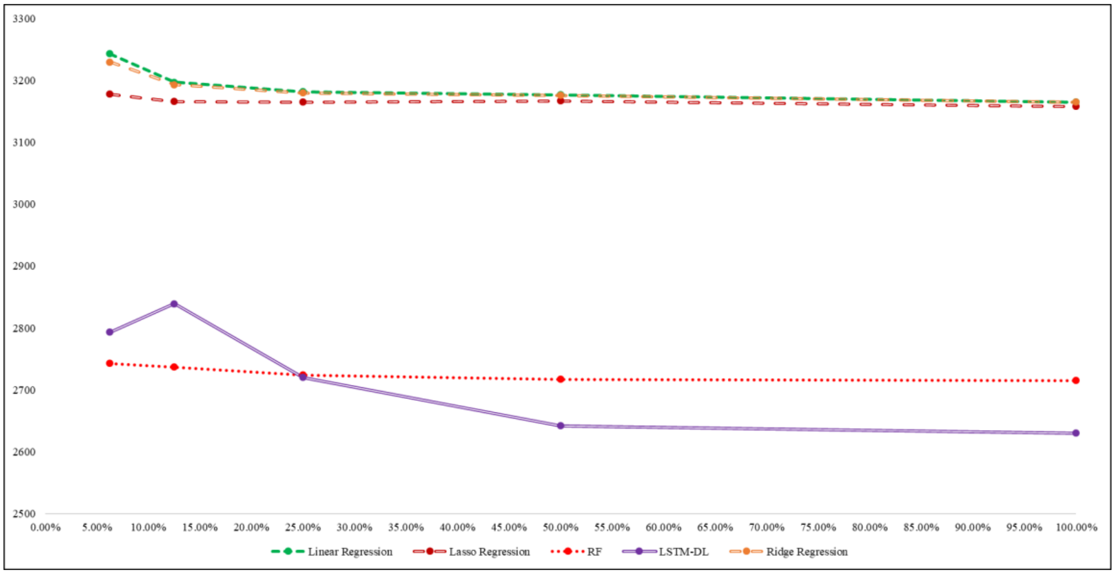


Figure 10. Model performance and training dataset size.

Training Dataset Size. We next vary the size of the training dataset to examine its impact on our proposed LSTM-DL model. The data was down-sampled using random selection. As expected, as the training dataset size decreases, the performance of the LSTM-DL model deteriorates. However, as reported in Table 8 and Figure 10, it remains the best model when the sample size is reduced to 25% of the original sample. When the training dataset size is further reduced to 12.5% and below, the RF model outperforms the LSTM-DL model. This finding highlights the fact that the LSTM-DL model does not always outperform other machine learning models; a sufficient amount of training data is needed to realize the advantage of LSTM models. One practical implication is that in the tradeoff between longer sequence and larger sample size, LSTM seems to be more sensitive to the sample size.

3.5.2 Patient Segmentation

As discussed above, fluctuations are an important aspect of the LSTM-DL model's performance. One may therefore hypothesize that additional value could be gained from fitting separate models on low- and high-fluctuation cohorts. To that end, we retrained two separate LSTM-DL models using only high-fluctuation data (top 25%) and low-fluctuation data (bottom 25%) as discussed in Section 3.4.1. What we found is that the performance of these two fluctuation-specific models, on high- and low-fluctuation members respectively, was no better than the performance of the original model, as detailed in Table 9. Other experiments, including segmenting the population by level of healthcare costs and level of seasonality also show no benefit of segmentation and retraining. While the reduction in training dataset size might affect performance when the population is segmented into multiple sub-cohorts, at the same time, the fact that our original model performs equally well as sub-group models indicates its high robustness and high applicability.

Table 9. Strategic segmentation.

Training set	Model name	Performance on high fluctuation testing data	Performance on low-fluctuation testing data	Average cost in 2011 in testing data	Performance on all testing data
All members	LSTM-DL	$4455	$1641	$3729	$2630
Low fluctuation members	LSTM-DL-low	$4655	$1721	$2199	$2835
High fluctuation members	LSTM-DL high	$4740	$1872	$6472	$2914

Note: the best model for each training data set is selected based its performance on validating data.

3.6 Conclusions and Discussion

Deep learning models have the potential to improve predictions in data-rich environments such as healthcare settings, which makes them highly worthy of research and consideration. Taking advantage of recent developments in deep learning, we develop an LSTM model that captures simple time series information for accurate cost predictions. This LSTM-DL model outperforms other machine learning models in our experiments using a large quantity of claims data. More importantly, this study presents a novel investigation into the performance of LSTM models. We not only show that our LSTM-DL model consistently performs well across different subgroups of patients, but we also acquire insights into how the LSTM-DL model takes advantage of variability in members’ cost structures in order to outperform other methods. This study therefore makes important contributions to the application and understanding of LSTM models for more accurate prediction of healthcare costs.

We would like to note several limitations of this study. Since claims data is widely available, relatively easy to use, and comes at a low cost, it forms the basis for our study and is one of the most widely used data resources for work in population management and health research. Further, since previous research has established the importance of claims cost information for future cost predictions, this information was key in our study. However, other types of claims data, including diagnosis, procedures, prescriptions and potentially even lab information, might also contribute to more accurate, albeit more complex, models. This is an interesting avenue for future research. In addition, claims data does not capture the patient’s clinical details to the same extent as full-scale EMR data, which may also include clinical notes. Therefore, LSTM models taking advantage of additional clinical information through both additional claims information and EMR records have the potential to further improve and advance the science of healthcare cost prediction. We also note that some of our empirical findings above may be limited due to our specific dataset, and further study is needed to confirm their generalizability.

In conclusion, the continuous advancement of deep learning technology provides additional opportunities for future improvement of healthcare cost prediction. We believe that attention mechanisms, capsules, and other emerging deep learning trends may also offer fruitful directions for future research, as they have the potential to make even better use of the available information to generate more accurate predictions. As big data and advanced machine learning models become mainstream in data rich environments, including many IS research areas, translating deep learning technology to these application areas requires care. The approach taken in this study can serve as a road map for considering input data and variables, constructing appropriate architecture, and using regression modeling to understand performance drivers.

Chapter 4: Friend or Foe? How Artificial Intelligence Affects Human Performance in Medical Chart Coding

4.1 Introduction

Artificial intelligence (AI) is shaping our lives dramatically (Brynjolfsson et al., 2018a; Hosanagar, 2019) and is now broadly heralded as a potential stimulus for an economic revolution. Industry projections estimate that AI's contribution to the US economy will be $15.7 trillion by 2030, constituting a boost of up to 26% in GDP (PwC, 2017). The anticipated effect on the workforce is striking: because of AI, up to 400 million workers globally may need to shift jobs by 2030 (Manyika et al., 2017). One of the leading domains for AI application today is healthcare, where the landscape has changed rapidly in the past two decades, fueled by the adoption of new technologies and widespread digitization of health-related data. All major players in the healthcare ecosystem, including government agencies (Talley et al., 2011), healthcare providers (Krittanawong et al., 2017), insurance companies (Kose et al., 2015), and pharmaceutical manufacturers (Ekins, 2016), express enthusiasm about the potential benefits of AI. Given the size (about one-fifth of GDP) and nature (extensive knowledge work) of the healthcare industry, the impact of AI on work in this setting may be substantial.

Unlike past automations that displaced humans in manual work and routine cognitive tasks, AI, especially given recent developments in machine learning (and its subfield of deep learning), is increasingly outperforming humans in high-level cognitive tasks (He et al., 2015; Mnih et al., 2015). For example, while it takes over a decade of training to be a radiologist capable of interpreting mammograms, recent tests have shown that AI can outperform radiologists in diagnosing breast cancer from X-ray images (McKinney et al., 2020), suggesting that it could potentially replace professional experts. AI's superior performance can be attributed to two factors: (1) advances in AI helps better extract insights and knowledge from data, thus rendering professionals' years of training and experience less valuable; and (2) human workers may be prone to physical constraints such as attention deficits and exhaustion due to their natural circadian rhythms (Van Dongen et al., 2000). Indeed, it has been shown that fatigue later in the day leads to poor judgement and diagnostic errors among health providers (Krupinski et al., 2010; Lee et al., 2013). In contrast, AI is indefatigable and consistent in its performance throughout the day.

On the other hand, as an alternative plausible to the perspective of AI replacing the human expert, there are reasons to believe that AI complements, rather than substitutes for, human labor. At its current technological maturity and capability, AI still has a way to go before it can replace professionals (Davenport and Dreyer, 2018). If AI mainly automates explicit and simple tasks, this makes people with rich experience more productive in leveraging AI, as they can focus on the tacit and complex tasks (Pakdemirli, 2019). Similarly, if AI's role is to mainly assist decision making, it can place considerable cognitive burden on a human to digest the information from AI as additional input. Therefore, a human worker might be able to leverage AI better during the peak hours of productivity. Given the magnitude of the potential economic impact of AI (Korinek and Stiglitz, 2017), obtaining a deeper understanding of the interplay between human labor and AI in the context of complex cognitive tasks is important for executives and policy makers seeking to maximize productivity gains.

In this study, we report one of the first empirical studies on how AI affects the performance of workers with different experience levels and at different points in their circadian rhythm. For this investigation, we implement a machine learning-based AI solution in a knowledge work setting in a publicly traded US company in the healthcare sector. Our AI is built for medical coding and automatically identifies patient conditions in medical charts, thereby offering an opportunity to enhance care delivery and reimbursement for insurance companies and healthcare providers.

The AI for the medical coding task offers an ideal setting for studying AI's impact on productivity for several reasons. First, medical coding (specifically risk adjustment coding) is a complex and cognitively non-routine task, which is generalizable to many other knowledge-intensive jobs. Second, the AI used in this study is a typical machine learning-based AI that is considered state-of-the-art for business use.[14] Third, medical coding has well-defined output quantity and quality at the individual coder level, thereby offering appropriate metrics for quantifying AI's impact on

[14] Given that most machine learning models focus on specific and well-defined tasks, the mainstream use of AI in business does not replace human work but rather complements it, as previously discussed. Consistent with current industry practice, the medical coding AI in this study is designed to augment human intelligence.

productivity. Using detailed data from the coding of 1,231,447 patient charts over a one-year period in a natural experiment setting, we measure the impact of AI on medical coders' productivity, followed by an analysis of the differential impact of AI on productivity conditional on workers' experience levels and circadian rhythm. We further conduct a series of qualitative analyses to gain a deeper understanding of the nuances and underlying dynamics of AI's impact.

Our study yields several novel findings. First, we identify an interesting pattern in the AI's performance: AI boosts productivity overall, but rather than helping humans during lower productivity periods (afternoon and night), AI helps most in the morning, an effect that is contemporaneous with human performance peaks. This supports the complementary relationship between AI and human capital. Second, while human/AI complementarity would lead us to predict that humans with rich work experience would most fully realize the benefits of AI, our findings suggest a different relationship between AI and workers' experience: senior workers realize lower benefits from AI than their junior colleagues. Our qualitative analyses help resolve this puzzle: we find that workers' trust in AI is crucial to productivity gain, and senior workers are less trusting of AI.

This study makes significant contributions to AI research relevant for business. First, it is part of a nascent stream of literature that empirically tests the causal effect of AI in improving productivity among knowledge workers in the healthcare field. Second, this paper reveals how AI interacts with human workers of different experience levels and at different times of the day. While our research setting is healthcare, the findings are generalizable to knowledge-intensive work in other domains, and they yield important managerial implications for the wider usage of AI in business.

4.2 Theory and Background

4.2.1 Productivity and AI

There are several well-known mechanisms by which AI can increase productivity. First, AI can free workers from the time and effort required for certain well-defined tasks. To illustrate, in one of the most successful examples of AI in medicine, AI outperformed typical medical staff in making diagnoses based on medical images (Gulshan et al., 2016). By leveraging AI for specific aspects of a job, worker time can be released for other tasks, thereby improving human productivity. Second, AI can reduce well-documented errors in human judgment (Danziger et al., 2011; De Martino et al., 2006) such as anchoring (Tversky and Kahneman, 1974) and recency effects (Tzeng, 1973). Given the significance of human judgment in the economy (Kahneman and Tversky, 1979), AI can further improve productivity by addressing potential bias in human cognition and assuring quality output.[15]

However, despite the optimism and hype surrounding AI, empirical evidence for its positive effects on economic productivity is scant (Case and Deaton, 2017; Syverson, 2017). The handful of studies to date have not found overwhelming evidence for

[15] Although the presence of bias in AI is not the focus of this paper, we acknowledge that, to the extent that an AI may be trained on data from humans, it may encapsulate biases.

increased productivity; in fact, researchers have offered several explanations for why the effect of AI turns out to be insignificant, dubbing this disappointing reality "the AI productivity paradox" (Brynjolfsson et al., 2018b).

We aim to shed light on the mechanism by which AI interacts with different aspects of human performance, examining differences both within and across workers. First, we focus on how AI affects individual worker productivity as the worker's performance varies across time, and we do so by examining one aspect of the artificial-human intelligence complementarity that is less understood. That is, human productivity varies based on circadian rhythm, which is the "self-sustained oscillator with an inherent frequency [that] underlines human 24-hour periodicity" (Aschoff, 1965). This "biological clock" determines the human body's level of functioning and therefore exerts significant impact on human productivity (Folkard, 1975; Dzogang et al., 2018). For example, it has been found that shift workers who need to stay active against their internal circadian rhythm experienced negative impacts on health (Weibel et al., 1995; Kitamura et al., 2002). In another study of about two million students, Pope (2016) found that productivity, as measured by academic performance, was significantly higher in a morning class than in an afternoon class. In healthcare, radiologists have been found to exhibit decision fatigue later in the day, which leads to worse outcomes (Krupinski et al., 2010; Lee et al., 2013).

Since AI has no human circadian rhythm, it can potentially complement and maintain or even improve the productivity of human workers during their innately less productive periods. However, it is not clear whether AI’s largest effect will materialize at the peak or the valley of the human circadian rhythm. Theoretically, the outcome should be driven by the relationship between the human input and the AI input. If human and AI inputs are interchangeable, AI’s impact on productivity would be stronger during the night (the performance valley) than the morning (the performance peak), since human coders slow down at night (see Figure 13) and there would therefore be greater opportunity for AI to contribute. However, if strong complementarities exist between human and AI input, AI should be most effective in the presence of peak human performance. This is especially likely given the current uses of AI in knowledge work, where AI tends to focus on narrow tasks that are part of the input that facilitate final human decision making (Tegmark, 2017). It is therefore possible that AI helps productivity most when the human is at a more capable stage in their circadian rhythm, i.e., in the morning, and can better leverage the AI input.

4.2.2 Heterogeneity in Leveraging AI

All workers might not be equally capable of exploiting the potential of AI. In fact, studies from the history of technological innovation give cause for concern that AI may disproportionately advantage certain types of human workers (David, 2015; Decker et al., 2017). Such heterogeneity in human collaboration with new technologies mainly comes through the pathway of technology-skill complementarity (Goldin and Katz, 1998). This complementarity between human skills and new technologies has been a central focus of many studies, confirming that workers who better complement a new technology will gain more from its adoption (Autor et al., 2003; Bartel et al., 2007).

In recent decades and in comparison with change associated with other types of automation, the changes in IT have been argued to be an especially strong example of skill-biased technology change (SBTC) (Krusell et al., 2000). Using data from the 1990s, Bresnahan et al. (2002) find a strong tendency for IT to favor skilled labor. They posit that IT in terms of computers and network devices makes data more readily available and easier to analyze to obtain business insights, and highly educated employees can most effectively leverage this new capability and generate greater value.

There are reasons to believe that AI will continue the trend that IT started and will also prove to be an instance of SBTC. As such, in order to assess the quality and veracity of AI outputs, the human user needs to possess a breadth of knowledge around the task that AI facilitates. This includes an understanding of practice, data-related knowledge, and exposure to the data. Interpretation involves understanding the real meaning of the output and comprehending the judgment of the AI. It also involves filtering the output of AI and finding ways to check the results (accuracy) and decide which parts of the AI output require further attention and consideration. This reasoning suggests that workers with in-depth knowledge and proficiency around their work might be able to make fuller use of AI. Thus, we would expect seniority to play a role in moderating the impact of AI, and senior workers should be better positioned to exploit AI than their junior colleagues.

However, from an alternative perspective it could also be argued that senior workers would be disadvantaged in collaborating with AI. This would be, first, because previous generations of IT mostly focus on communication (email, web, and social networks) and information processing (office suites, databases, and statistical software). These technologies serve as facilitators to provide necessary information to users, with high-level decision making remaining vested in the human. However, AI has now elevated decision support capabilities to a new level, where rather than simply preparing input to help humans make high-level decisions, it is getting close to making those decisions. Indeed, for the first time in history we see the potential of AI to outsmart humans and pose a potential threaten to our future if out of control, a concern posed in an open letter signed by Stephen Hawking, Elon Musk and more than 8,000 other scholars (Russell et al., 2015).

It is well known that experts, such as senior human workers, are more likely to display a lack of confidence and trust in judgments that are not their own (Liu et al., 2017). The self-confidence engendered by virtue of their experience and expertise could lead them to discount the AI’s recommendations (Bradley, 1981). In other words, senior workers may experience a loss of control when AI is introduced in the workplace and, as a result, resist accepting it.

Further exacerbating senior workers' concern about losing control is the fact that many algorithms employed by advanced AI are notorious for their lack of interpretability (Pasquale, 2015), and studies of the black box issue point to the difficulty humans have in understanding an AI's reasoning processes. Furthermore, AI output can also be biased by the training data, which may result from incompleteness (Beymer et al., 2013), skewness (Gitelman, 2013), non-representativeness (Attenberg et al., 2011), or errors in data preprocessing (Zook et al., 2017). Compared to novice users, experts are more likely to spot errors and imperfections in AI. Such limitations of the AI can further reduce their trust in the AI's output.

To summarize, there are multiple factors at play that would determine how AI's impact is moderated by a human worker's level of experience. Since AI targets high-level complex decision making that requires substantial domain expertise, senior workers enjoy the advantage of leveraging the output of AI. At the same time, however, senior workers tend to have greater difficulty accepting AI and less trust of AI output. Also, if an expert spends extra time evaluating and resolving the conflict between their own judgment and AI recommendations, their productivity gain will be less than that of junior workers. This ambiguity in the theoretical prediction underscores the need for the empirical insights that we offer in this study.

4.3 Research Context

4.3.1 Background

Our research context is the medical coding industry. In the US healthcare system (and in many other countries as well), patient conditions and treatments need to be transformed into standardized codes in the billing process. Accurate medical coding is necessary for both timely and correct payment and for efficient clinical decision making. Historically, medical coding is a labor-intensive job that involves manual code evaluation. It is of considerable economic significance: the market size of the medical coding industry was $10.6 billion worldwide in 2016 and is increasing 10% per year (Grand View Research, 2018).

In our study, we focus on one of the most difficult coding tasks – risk adjustment coding from medical charts, which requires human workers to review the complete medical chart, especially the unstructured physician notes, and make judgments about whether the patient has certain medical conditions such as diabetes. The health conditions identified are designated as risks and used to adjust reimbursements, i.e., for the same clinical procedure, reimbursement for treatments received by patients with higher risks will be higher. The industry has widely adopted the Hierarchical Condition Categories (HCC) coding system created by the Centers for Medicare and Medicaid Services (CMS) (Li et al., 2010). The economic value of coding activity is substantial: an average HCC code has a reimbursement value of several thousand dollars (Pope et al., 2004).

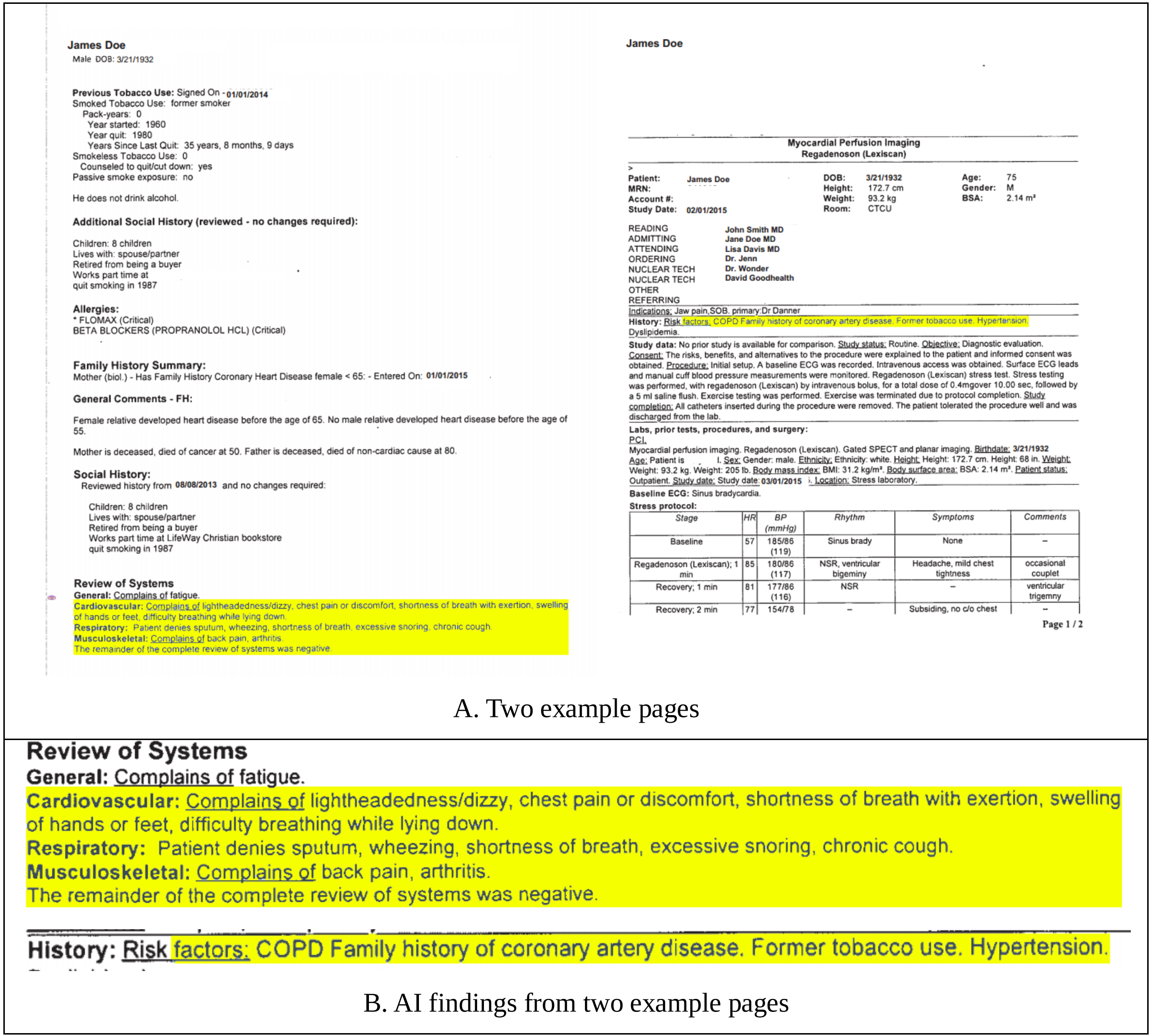
We note that the practice of medical chart coding is representative of the activities that typical knowledge workers do in other industries. The job is a non-routine task because HCC codes are not directly included in the medical charts, so coders need to read, understand, and interpret the information in order to decide which HCC codes should be reported. Moreover, every medical chart includes large amounts of patient-specific information, requiring a coder to exercise comprehensive reasoning, judgment, and decision making for every medical chart (Dimick, 2010). To tackle this complex task, our collaborator, a leading public healthcare analytics company that provides medical chart coding services to multiple insurance companies, employs hundreds of coders who have collectively coded over 36 million medical charts in the past decade.

The coding process is as follows: first, every time a coder requests a new chart to code, a medical chart is *randomly* assigned to them. Once assigned, the medical chart is displayed on the coder's desktop screen. The coder first spends some time browsing the chart and forms a basic understanding of the patient's medical profile, such as whether the patient has diabetes, cancer, or hypertension. In the second step, the coder then goes through the chart to identify evidence for specific codes. Each coder is subject to post-reviews of randomly selected coded charts to minimize coding errors and to ensure quality. According to the company's policy, all coders must maintain over 90% accuracy in their reported HCC findings. Otherwise, the coder will be asked to complete a training program (usually lasting several days) and will not be assigned work for the duration of training.

Based on discussion with experts and management teams in the company, we identified the time spent reviewing a medical chart as the measure of a coder’s productivity. While one might think that the HCC codes are the ultimate output, the number of HCC codes that can be detected is not purely driven by coder efforts but is also determined by the nature of information in the chart. In addition, for every possible code identified, a coder expends almost as much effort to determine whether that code is a false positive to ensure quality. Given that the charts are randomly assigned and the coding quality is well maintained across coders, the average time taken to code a chart reflects a natural measure of productivity.

4.3.2 Development of the AI

We developed a machine learning-based AI to facilitate the labor-intensive process of medical chart coding. Specifically, the task that this AI accomplishes is to highlight sentences with at least one potential HCC code. To do this, the AI first processes all sentences in the chart through a filter. This filter relies on a dictionary, developed and maintained by experts in the company, to capture all possible keywords that could indicate HCC-related health conditions. However, keyword matching yields too many false positives. In the second step, then, a machine learning model is deployed to evaluate the probability that the focal sentence contains valid HCC codes, and it then highlights that sentence for the coder to preview. This process is illustrated in Figure 11.



James Doe
Male DOB: 3/21/1932

Previous Tobacco Use: Signed On - **01/01/2014**
Smoked Tobacco Use: former smoker
Pack-years: 0
Year started: 1960
Year quit: 1980
Years Since Last Quit: 35 years, 8 months, 9 days
Smokeless Tobacco Use: 0
Counseled to quit/cut down: yes
Passive smoke exposure: no

He does not drink alcohol.

Additional Social History (reviewed - no changes required):

Children: 8 children
Lives with: spouse/partner
Retired from being a buyer
Works part time at
quit smoking in 1987

Allergies:
* FLOMAX (Critical)
BETA BLOCKERS (PROPRANOLOL HCL) (Critical)

Family History Summary:
Mother (biol.) - Has Family History Coronary Heart Disease female < 65: - Entered On: **01/01/2015**

General Comments - FH:

Female relative developed heart disease before the age of 65. No male relative developed heart disease before the age of 55.

Mother is deceased, died of cancer at 50. Father is deceased, died of non-cardiac cause at 80.

Social History:
Reviewed history from **08/08/2013** and no changes required:

Children: 8 children
Lives with: spouse/partner
Retired from being a buyer
Works part time at LifeWay Christian bookstore
quit smoking in 1987

Review of Systems
General: Complains of fatigue.
Cardiovascular: Complains of lightheadedness/dizzy, chest pain or discomfort, shortness of breath with exertion, swelling of hands or feet, difficulty breathing while lying down.
Respiratory: Patient denies sputum, wheezing, shortness of breath, excessive snoring, chronic cough.
Musculoskeletal: Complains of back pain, arthritis.
The remainder of the complete review of systems was negative.

James Doe

Myocardial Perfusion Imaging
Regadenoson (Lexiscan)

>

Patient:	James Doe	**DOB:**	3/21/1932	**Age:**	75
MRN:		**Height:**	172.7 cm	**Gender:**	M
Account #:		**Weight:**	93.2 kg	**BSA:**	2.14 m²
Study Date:	02/01/2015	**Room:**	CTCU		

READING	**John Smith MD**
ADMITTING	**Jane Doe MD**
ATTENDING	**Lisa Davis MD**
ORDERING	**Dr. Jenn**
NUCLEAR TECH	**Dr. Wonder**
NUCLEAR TECH	**David Goodhealth**
OTHER	
REFERRING	

Indications: Jaw pain,SOB. primary:Dr Danner
History: Risk factors: COPD Family history of coronary artery disease. Former tobacco use. Hypertension. Dyslipidemia.

Study data: No prior study is available for comparison. Study status: Routine. Objective: Diagnostic evaluation. Consent: The risks, benefits, and alternatives to the procedure were explained to the patient and informed consent was obtained. Procedure: Initial setup. A baseline ECG was recorded. Intravenous access was obtained. Surface ECG leads and manual cuff blood pressure measurements were monitored. Regadenoson (Lexiscan) stress test. Stress testing was performed, with regadenoson (Lexiscan) by intravenous bolus, for a total dose of 0.4mgover 10.00 sec, followed by a 5 ml saline flush. Exercise testing was performed. Exercise was terminated due to protocol completion. Study completion: All catheters inserted during the procedure were removed. The patient tolerated the procedure well and was discharged from the lab.

Labs, prior tests, procedures, and surgery:
PCI.
Myocardial perfusion imaging. Regadenoson (Lexiscan). Gated SPECT and planar imaging. Birthdate: **3/21/1932** Age: Patient is l. Sex: Gender: male. Ethnicity: Ethnicity: white. Height: Height: 172.7 cm. Height: 68 in. Weight: Weight: 93.2 kg. Weight: 205 lb. Body mass index: BMI: 31.2 kg/m². Body surface area: BSA: 2.14 m². Patient status: Outpatient. Study date: Study date: **03/01/2015** . Location: Stress laboratory.

Baseline ECG: Sinus bradycardia.

Stress protocol:

Stage	*HR*	*BP (mmHg)*	*Rhythm*	*Symptoms*	*Comments*
Baseline	57	185/86 (119)	Sinus brady	None	–
Regadenoson (Lexiscan); 1 min	85	180/86 (117)	NSR, ventricular bigeminy	Headache, mild chest tightness	occasional couplet
Recovery; 1 min	81	177/86 (116)	NSR	–	ventricular trigemny
Recovery; 2 min	77	154/78	–	Subsiding, no c/o chest	–

Page 1 / 2

A. Two example pages

Review of Systems
General: Complains of fatigue.
Cardiovascular: Complains of lightheadedness/dizzy, chest pain or discomfort, shortness of breath with exertion, swelling of hands or feet, difficulty breathing while lying down.
Respiratory: Patient denies sputum, wheezing, shortness of breath, excessive snoring, chronic cough.
Musculoskeletal: Complains of back pain, arthritis.
The remainder of the complete review of systems was negative.

History: Risk factors: COPD Family history of coronary artery disease. Former tobacco use. Hypertension.

B. AI findings from two example pages

Figure 11. Example of ai findings in a medical chart.
Note: The highlighted areas in A are done by AI and are magnified in B.

For model development, we used 26,000 labeled medical charts, out of which 24,000 were randomly selected as training data. After training, the model was tested on the remaining 2,000 charts. We developed several versions of the AI using different approaches, including support vector machine (SVM), convolutional neural networks (CNN) and recurrent neural networks (RNN). Our implementation is based on the SVM version due to its superior computing efficiency. The area under the curve (AUC) is 0.97. Our SVM model outputs a probability, which allows us to customize the threshold to control the recall (how many HCC codes could be missed by the AI). The company set the threshold as 0.90 (recall as roughly 95%) for the best outcome.

Given its superior performance, this AI is a potential game changer in the industry. Before the introduction of machine learning-based AI, both academia and industry had expended significant effort in entity recognition from clinical notes using rule-based models. One of the most famous tools is cTAKES (Savova et al., 2010). During our model development, then, we benchmarked our AI models with the performance of cTAKES on our data. Given a level of recall (approximately 90%) similar to that required of our SVM model, the precision of cTAKES is only 6.5%. Given such a high false positive rate, it is not feasible for coders to use rule-based models in real coding work, and prior to this AI, our collaborator had not successfully implemented any models for HCC coding. Our AI achieves a precision of about 30% while maintaining a recall of roughly 95%. With its exclusive ability to handle the cognitively complex task of HCC coding, it is making a big difference in coder performance. The company has implemented it in daily practice, and executives attribute significant revenue generation to its use.

This AI system is representative of current state-of-the-art applications of AI in knowledge work: the AI assists the human who makes the final decisions. For example, one of the most successful use cases of AI in medicine is diagnoses from imaging, where AI suggests the diagnosis and human doctors make the final determination (Hainc et al., 2017; Hosny et al., 2018). Similarly, the AI developed for this study highlights sentences where a code might be identified and suggests the HCC code that may apply, but it still requires human coders to review its findings. As noted by industry experts (Williams, 2015), the singularity of AI is still decades away, and the majority of current AI applications are similar to our use case, in which AI facilitates rather than completely replaces human decision making. Therefore, findings from this study are pertinent to the current practice of AI use in business.

4.3.3 AI Implementation

The AI system went online in July 2018. With help from management, 80 coders were selected to represent the full spectrum of coding seniority levels.[16] The remaining 468 coders who did not have access to the AI constitute our control group. To help us understand the impact on productivity, we collected data for a year before AI was used. This pre-treatment period runs from July 16, 2017 to April 30, 2018 (May and June 2018 were excluded due to the adjustment for transition). We also have coder performance data from July 1, 2018 to October 31, 2018, which is defined as the post-treatment period.[17] During our study period, no major changes were made to the work procedures aside from the implementation of AI. The 80 coders in the treatment group reviewed and coded 196,995 charts. The control group (468 coders)

[16] While one might prefer using a random sample of coders as the treatment group, some practical concerns preclude us from doing so. There is a relatively high turnover in coding jobs, and there is a concern that the randomization process would enlist coders who might not have sufficient history for the pre-trend data or who might drop out soon, which would introduce bias in the analysis of long-term productivity. We therefore opt for a more stable treatment group and perform extensive validation tests that are described later.

[17] Our sample includes a longer pre-treatment period to better examine the trends of the treatment and control groups before the AI implementation. In our robustness test, we also use a shorter pre-treatment period (the same length as the post-treatment period) and confirm that all the results remain the same (results in Table 24).

coded 1,034,452 medical charts in the same period for a total of 1,231,447 medical charts in the study sample.[18]

Table 10. Summary statistics.

	Treated		Control	
	Pre	Post	Pre	Post
Num of Charts	138,937	58,058	803,707	230,745
Num of Charts per Coder	1736.71 (293.12)	774.11 (301.00)	1770.00 (306.64)	744.34 (320.44)
Review Time (in min)	14.46 (26.55)	14.08 (27.88)	12.87 (27.33)	12.87 (27.54)
Num of Pages	34.57 (51.08)	35.24 (50.28)	34.39 (48.71)	35.06 (49.55)
Num of HCC Codes	0.42 (0.80)	0.27 (0.60)	0.58 (0.94)	0.33 (0.69)
Percentage of Round 1 Coding	99.99% (1.00%)	100.00% (0.00%)	96.59% (18.15%)	98.11% (13.62%)
Percentage of Charts Finished in Early Morning	0.89% (9.39%)	0.05% (2.16%)	9.22% (28.93%)	15.26% (35.96%)
Percentage of Charts Finished in Morning	19.01% (39.24%)	23.98% (42.70%)	22.64% (41.85%)	25.43% (43.55%)
Percentage of Charts Finished in Afternoon	38.72% (48.71%)	43.66% (49.60%)	32.63% (46.89%)	29.14% (45.44%)
Percentage of Charts Finished at Night	41.38% (49.25%)	32.32% (46.77%)	35.51% (47.85%)	30.17% (45.90%)
Percentage of CMS Coding	56.69% (49.55%)	70.74% (45.49%)	29.54% (45.62%)	66.52% (47.19%)

Standard Deviation in Parentheses

Pooling both groups together, the average time to code one medical chart is 13.11 minutes with a standard deviation of 27.32; the average number of pages is 34.58 with standard deviation of 49.21. Focusing only on treated medical charts, the average time spent on one medical chart is 14.35 minutes with a standard deviation of

[18] The number of charts per coder in the two study groups is different, which is largely due to the high turnover rate of this job. As previously mentioned, we selected stable coders for the treatment group, resulting in a lower turnover rate and thus a higher number of charts per coder.

26.95; the average number of pages is 34.77 with a standard deviation of 50.85. The summary statistics of each group in both pre-AI and post-AI periods are reported in Table 10.

4.4 Results: Impact of AI on Productivity

4.4.1 Model-Free Evidence

We first present model-free evidence of the impact of AI on coder productivity. For each month, Figure 12 plots the average time (in minutes) that it took coders in both the treatment and control groups to code one medical chart. As shown in the figure, the trends of the control group and treatment group are very similar with modest fluctuations. In the pre-period (July 2017–April 2018), coders in the treatment group spent 1.69 more minutes on each medical chart than coders in the control group. Leveraging this average difference between the two groups in the pre-treatment period, we can calculate the reduction of coding time due to AI implementation. The dotted line in Figure 12 shows the time that the treatment group would have spent on an average medical chart if that 1.69 minute difference had remained consistent throughout the study. Comparing this dotted-line projection to the observed trend, we see that AI reduced the coding time for an average medical chart by 0.41 minutes (2.95%) in August, 1.04 minutes (6.61%) in September, and 1.57 minutes (9.92%) in October. Overall, the average coding time per chart is 14.60 minutes for the treatment group in the pre-treatment period, which further decreased to 14.18 minutes in the post-period. Meanwhile, the average coding time for the control group increased from

12.91 minutes to 13.50 minutes. Comparing the two differences, we conclude that coding time was reduced by 1.01 minutes (6.92%) per medical chart after AI implementation.

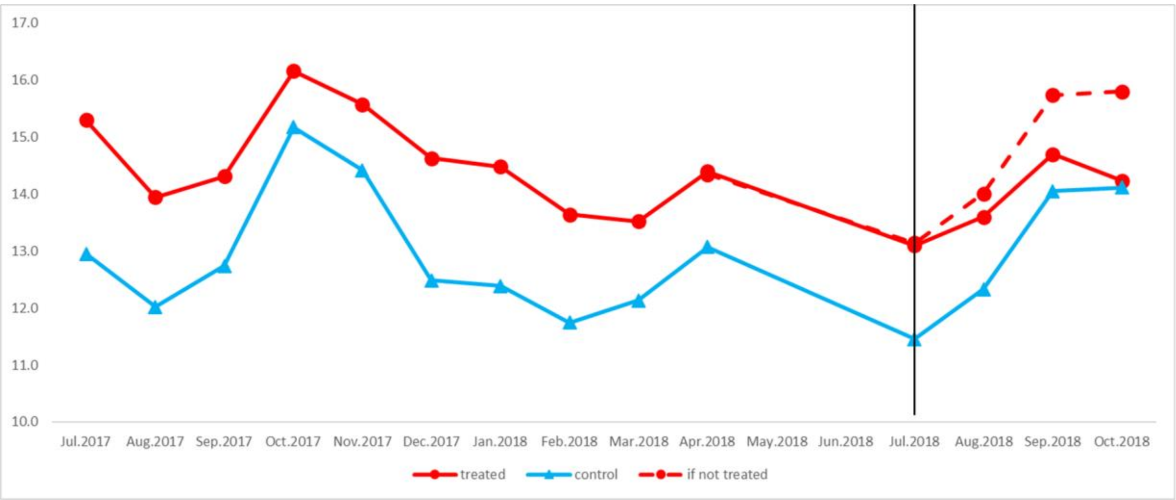


Figure 12. Trends in medical chart coding time.

To examine the circadian rhythm of human workers, we introduce dummy variables to indicate the time of day (according to local completion time) that each medical chart was coded. Time of day is categorized as early morning (12 a.m.–7 a.m.; 1,263 charts completed by the treatment group during our study period), morning (7 a.m.–1 p.m.; 40,335 charts completed by the treatment group), afternoon (1–5 p.m.; 79,140 charts completed by the treatment group), or night (5 p.m.–12 a.m.; 76,257 charts completed by the treatment group). As presented in Figure 13, the average coding time during the pre-treatment period varies across different times of day, suggesting the presence of variance in productivity of human workers. Statistics show that the average coding time is the shortest in the morning (12.2 minutes). Compared with this best period, the productivity of coders dropped 13.9% in the afternoon and 31.1% at night. This finding also indicates that the highest human productivity (the peak of human circadian rhythm) is in the morning and the lowest human productivity (the valley of human circadian rhythm) is at night.

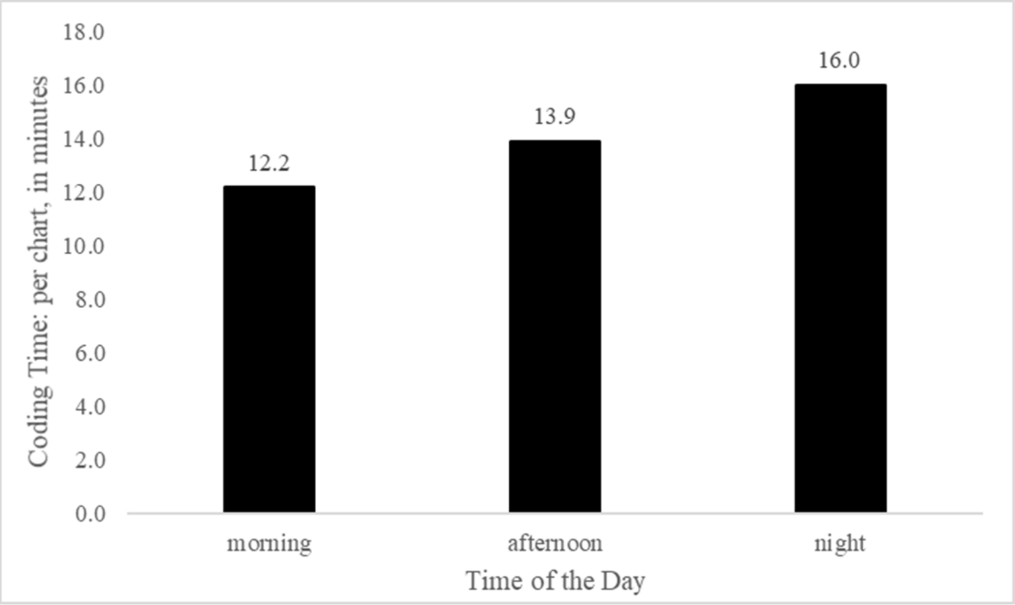


Figure 13. Medical chart coding time by human circadian rhythm.

4.4.2 AI Impact on Productivity

Equation 6 depicts our formal empirical specification. To eliminate potential confounders, we conducted coder-level fixed effects analysis. In our model, the dependent variable (Y_i) is the time (in minutes) spent on a medical chart. *Post* is a dummy variable that takes the value 0 for the pre-AI period and 1 for the post-AI period. *AI* (omitted in the fixed effects model) is 1 for the treatment group, i.e., those 80 coders who are assigned to use the AI, and 0 for the remaining coders, who do not use AI. The interaction term *Post * AI* captures the effect of AI on review time.

$$Y_i = \beta_0 + \beta_1\, Post + \beta_2\, Post * AI + \beta_3\, X_i + \varepsilon_i \quad \ldots\ldots(6)$$

We also included a set of medical chart-level characteristics as control variables. We controlled for the time of day coding by including three dummy variables, morning, afternoon, and night (early morning as default). The length of medical charts was controlled for using number of pages (*NumPage*). Finally, the type of coding (i.e., different versions of HCC codes from CMS and HHS), the round of coding (the company uses a second round of coding for quality control), and month when coding was performed were also incorporated into our model. Individual coder characteristics are controlled using fixed effects.

The estimation of AI's impact on productivity is reported in Table 11. To validate the consistency of the results across specifications, we used an OLS model with chart-level controls (Column 1) and coder fixed effects with chart-level controls (Column 2). As reported in Column 2, the coefficient of *Post * AI* is -1.62, which is statistically significant (p<0.01). On average, the AI reduces coding time by 1.62 minutes (11.10% in pre-treatment period) per medical chart. Concerns might be raised about whether the productivity increase was due to a reviewer's rush to complete the job, thereby lowering output in terms of the number of codes extracted. We therefore further controlled for the number of HCC codes found in the medical chart and estimated the model again. The coefficient of *Post * AI* is again negative and significant (-1.61, see Column 3), further affirming the result. Our analysis indicates that combining AI with human coders does boost productivity in knowledge work.

Table 11. AI impact on medical chart coding time.

	(1) Main Result (OLS)	(2) Main Result (Coder Fixed Effects)	(3) Main Result (Coder Fixed Effects)
Dependent Variable:		Review Time	Review Time
Post	1.19***	2.64***	2.35***
	(0.22)	(0.66)	(0.65)
AI	0.75***		
	(0.08)		
Post X AI	-0.56***	-1.62***	-1.61***
	(0.15)	(0.57)	(0.58)
NumHCC			5.30***
			(0.14)
Constant	0.41	-11.17	-10.84*
	(1.06)	(7.72)	(6.56)
Control Variables:	NumPage, Round of Coding, Type of Coding, Time of Day		
Observations	1,231,447	1,231,447	1,231,447
R-squared	0.07	0.06	0.09
Number of Coders	548	548	548

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

4.4.3 AI and Workers' Circadian Rhythm

To formally test the interaction between the AI and workers' circadian rhythm, we used the model specified in Equation 7.

$$Y_i = \beta_0 + \beta_1\, Post + \beta_1\, Post * AI * early\, morning + \beta_2\, Post * AI * morning + \beta_3\, Post * AI * afternoon + \beta_4\, Post * AI * night + \beta_5\, X_i + \varepsilon_i \qquad \ldots\ldots(7)$$

Results are reported in Table 12. Since it is not common for coders to work in the early morning (less than 1% of the medical charts are processed during this time period) our regression results focus on morning, afternoon and night. Overall, it is interesting to note that only the morning and afternoon coefficients in Column 1 are significant (p<0.01), suggesting that the effects of AI are not detectable at all times of day. Regarding the magnitude, the coefficient of morning is -2.88 while that of afternoon is -1.65, indicating that AI's boost to productivity is much larger in the morning. In the meantime, the coefficient for night is only -0.39, indicating that AI's effect at night is only 23.6% of its effect in the afternoon and 13.5% of its effect in the morning. Comparing the relative effects at different times of day, we see that AI leads to lower (-1.68, p<0.1) review time in the morning (Column 2 of Table 12), i.e., greater productivity boost, while the effect is not as strong in the afternoon or night (Columns 3 and 4 of Table 12). This finding is consistent with the human circadian rhythm (Pope 2016), indicating that the strongest effects of AI are realized at the peak of human productivity. As human productivity declines to its lowest point (night), AI's effect becomes insignificant. At their lowest point of performance, coders' ability to collaborate effectively in reviewing AI recommendations might not be good enough to trigger the full realization of AI's value, and the benefits of complementarity are not discernible. Overall, the findings in Table 12 strongly suggest that rather than AI displacing human input, human and artificial intelligence are complementary, further confirming the collaborative nature between human and AI in the medical coding tasks.

Table 12. AI impact and circadian rhythm (with coder fixed effects).

	(1)	(2)	(3)	(4)
	Moderator	Moderator	Moderator	Moderator
	Time of day	Morning	Afternoon	Night
Dependent Variable:	Review Time	Review Time	Review Time	Review Time
Post		2.48***	2.29***	2.38***
		(0.68)	(0.67)	(0.67)
Post X AI		-1.20*	-1.59**	-2.17***
		(0.68)	(0.69)	(0.56)
Post X EarlyMorning	2.91***			
	(0.85)			
Post X Morning	1.94**	-0.55		
	(0.80)	(0.67)		
Post X Afternoon	2.48***		0.22	
	(0.67)		(0.34)	
Post X Night	2.25**			-0.11
	(0.88)			(0.74)
Post X AI X EarlyMorning	-2.02			
	(1.44)			
Post X AI X Morning	-2.88***	-1.68*		
	(0.77)	(0.90)		
Post X AI X Afternoon	-1.65***		-0.10	
	(0.60)		(0.63)	
Post X AI X Night	-0.39			1.75
	(1.20)			(1.21)
Morning	4.30***	4.17***	3.97***	4.00***
	(0.68)	(0.56)	(0.51)	(0.51)
Afternoon	3.74***	3.57***	3.58***	3.65***
	(0.63)	(0.48)	(0.50)	(0.48)
Night	5.62***	5.48***	5.51***	5.48***
	(0.63)	(0.52)	(0.52)	(0.51)
Constant	-11.00*	-10.89*	-10.81	-10.84*
	(6.56)	(6.54)	(6.56)	(6.57)
Control Variables:	NumPage, NumHCC, Round of Coding, Type of Coding			
Observations	1,231,447	1,231,447	1,231,447	1,231,447
R-squared	0.09	0.09	0.09	0.09
Number of Coders	548	548	548	548

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

4.4.4 Collaboration between AI and Workers of Different Experience Levels

If technology and experience are complementary, one would expect that senior workers will benefit more from AI than their less experienced colleagues. Alternatively, as discussed in Section 4.2.2, they may feel uneasy with the AI's recommendations. We now report analyses to test these competing conjectures. Based on the recommendation of company management, we measure workers' experience based on their tenure (number of years in the coding job) and classify coders into senior (10 or more years' experience) and junior (less than 10 years' experience) categories.[19] In the treatment group, there are 27 senior coders who completed 83,900 medical charts during the study period. The 53 junior coders completed 113,095 medical charts in the same period. Correspondingly, in the control group, 101,346 charts were completed by 65 senior coders and 933,106 charts by 403 junior coders. We create a dummy variable for each seniority level, then interact the two seniority level dummies with the treatment effect *Post * AI*.

Results are reported in Table 13. As shown in Column 1, we find the moderating effect for junior coders to be negative and significant (p<0.01), which means that AI helps junior workers to shorten chart review time and improve productivity. Interestingly, the moderating effect of senior coder is small and positive, while not statistically significant. The fact that this value is positive suggests that the AI might even be slowing senior coders down (i.e., it takes them longer to review a chart when using AI). The statistical significance of a greater productivity boost for junior coders

[19] We also use multiple cutoffs such as 8 years, 9 years, 11 years, and 12 years to define senior coders, and obtain consistent results. More details are provided in Appendix G.

is further supported by a Wald test (p<0.05). We also confirm this result by analyzing the effect within the treatment group (80 coders). Based on the regression results in Column 1, we plot the review time for the two seniority levels before and after AI implementation in Figure 14, which clearly shows that junior coders enjoying a greater productivity boost.

Table 13. AI benefit to workers at different seniority levels by circadian rhythm (with coder fixed effects).

	(1) Full Sample	(2) Morning	(3) Afternoon	(4) Night	(5) Morning Constant Sample	(6) Afternoon Constant Sample	(7) Night Constant Sample
Dependent Variable:	Review Time	Review Time	Review Time	Review Time	Review Time	Review Time	Review Time
Post X Senior	0.98	-0.60	0.52	2.00	-0.83	0.51	1.21
	(0.78)	(0.67)	(0.67)	(1.39)	(0.69)	(0.73)	(1.42)
Post X Junior	2.56***	1.45**	1.06*	4.19***	1.28**	0.53	3.62**
	(0.67)	(0.62)	(0.59)	(1.41)	(0.56)	(0.63)	(1.48)
Post X AI X Senior	0.12	-1.00	-0.82	2.15	-1.23*	-1.24*	1.67
	(0.73)	(0.61)	(0.65)	(1.66)	(0.67)	(0.75)	(2.04)
Post X AI X Junior	-2.14***	-2.68***	-1.12	-2.43	-3.10***	-1.09	-2.05
	(0.78)	(0.77)	(0.78)	(1.51)	(0.82)	(0.88)	(1.56)
Constant	-10.74*	3.40	-1.57***	5.36	4.45**	3.03*	6.53***
	(6.40)	(4.33)	(0.20)	(4.93)	(2.16)	(1.82)	(1.67)
Control Variables:	NumPage, NumHCC, Round of Coding, Type of Coding, Time of Day						
Observations	1,231,447	280,953	408,631	431,257	232,210	320,886	318,290
R-squared	0.09	0.08	0.17	0.06	0.12	0.17	0.05
Number of Coders	548	479	541	519	251	251	251

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

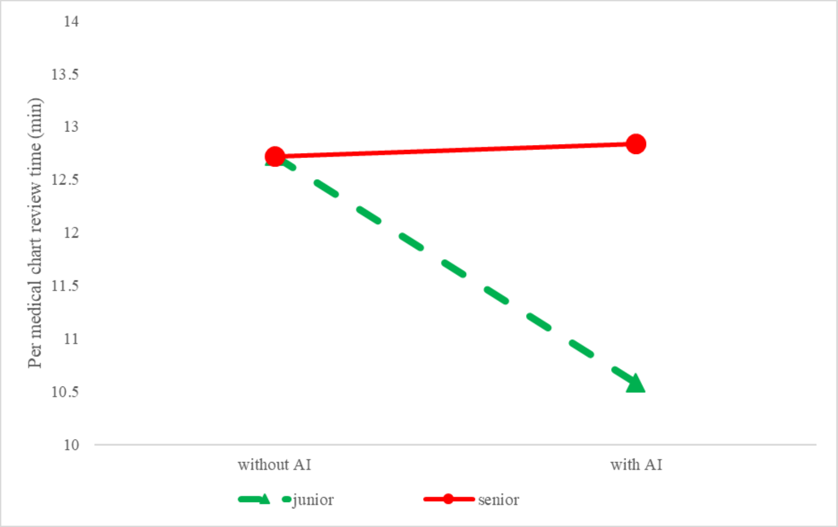


Figure 14. Visualization of the moderating effect of seniority level.

Given that the coder's circadian rhythm significantly affects productivity, we further examine the above findings for each time period of the day. Results are reported in Table 13. Theoretically, if human input is substitutable by machine intelligence, AI should be more able to make up the loss of human performance in the "valley" of the circadian rhythm. However, our findings do not support this conjecture. We find that AI's benefit to junior coders is significantly higher in the morning (Column 2 of Table 13), which is the peak of human circadian rhythm (as reflected in Figure 13). This suggests that junior coders need to be at maximal human performance in order to extract more value from AI. When human performance is at lower levels (afternoon and night), the impact of AI on junior coders becomes insignificant. We also confirm that the moderating effect of seniority is consistently insignificant across different time intervals (Columns 2–4 of Table 13). Overall, our findings show that AI helps junior coders more, and is most effective when the human performance is at peak level.

One point worth mentioning is that each coder works independently and is not assigned to work at a specific time of day. We confirmed that the findings are not driven by certain coders working certain scheduled shifts by comparing the distribution of medical charts across different times of day for individual coders and for the whole sample. First, it is not the case that any single coder works predominantly at a particular time of day. Focusing on the treatment group, we observed that with only three exceptions, almost all of the coders coded at all three time intervals (morning, afternoon, and night). Second, the medical charts finished in each time interval were not predominantly completed by specific coders. For morning, afternoon, and night, no coder's work accounted for more than 7% of the total charts. Nevertheless, we further constructed a constant sample of 251 coders who coded more than 200 charts in all of the three time intervals. In the resulting constant sample, 957,614 charts were completed. Of those charts, 160,382 were completed by 46 coders in the treatment group. We utilized the constant sample and conducted the same analyses described above. The outcomes are similar to our original results (Columns 5–7 of Table 13), reinforcing the robustness of the findings.

4.5 Robustness and Falsification Check

We demonstrated a significant improvement in medical chart coding productivity due to the use of AI. As discussed in Section 4.4.2, one might be concerned that the increased productivity comes at the cost of a decrease in the output; that is, coders might speed up their work but identify fewer HCC codes. To uncover the impact of the AI on output quality, we modify Equation 6 by using the number of detected HCC codes as the dependent variable. The number of detected HCC codes was used as the measure for quality under the condition that the error rate of the detected codes did not change significantly in the post-reviews. This result is reported in Column 1 of Table 14. *Post * AI* has a positive and insignificant coefficient, which means that AI does not lead to a deterioration in quality. If anything, the positive sign suggests that AI tends to help workers find slightly more HCC codes on average. Meanwhile, we do not observe significant differences in quality gains across the two seniority levels (Column 2 of Table 14).

To further confirm that our results are driven by AI rather than by common environmental factors that affect coding in general, we conducted a falsification test. Note that besides HCC coding, coders in this study also work on other chart coding tasks, such as CRG, HEDIS, and CDPS coding. Since our AI was developed to improve HCC coding, these non-HCC coding tasks should not be affected by the AI. Therefore, they serve as good candidates for the falsification test.

During our study period, 246,847 of these non-HCC medical charts were coded by all coders (52,208 by the treatment group and 194,639 by the control group). We use the same regression model (Equation 6) for these non-AI coding tasks; results are in Columns 3 and 4 of Table 14. The coefficient of *Post*AI* is insignificant, indicating that worker productivity in non-AI coding tasks did not increase in the same period. Also, the moderating effects of experience levels are not significant, further confirming the insignificant influence of AI implementation on non-AI coding tasks. Therefore, our main finding is less likely to be due to factors unrelated to the AI.

Table 14. Robustness checks (with coder fixed effects).

	(1) NumHCC	(2) NumHCC - Seniority Level	(3) Other Coding Types	(4) Other Coding Types - Seniority Level
Dependent Variable:	NumHCC	NumHCC	Review Time	Review Time
Post	0.04*		6.62	
	(0.02)		(6.73)	
Post X AI	0.01		3.07	
	(0.02)		(2.08)	
Post X Senior		-0.01		5.08
		(0.03)		(6.71)
Post X Junior		0.05**		7.01
		(0.02)		(6.74)
Post X AI X Senior		0.06*		7.75
		(0.04)		(4.88)
Post X AI X Junior		-0.01		1.76
		(0.02)		(2.20)
Constant	-0.00	0.00	-26.95**	-27.08**
	(0.22)	(0.22)	(11.76)	(11.72)
Control Variables:	NumPage, NumHCC, Review Time, Round of Coding, Type of Coding, Time of Day			
Observations	1,231,447	1,231,447	246,852	246,852
R-squared	0.10	0.10	0.17	0.17
Number of Coders	548	548	392	392

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

We also conducted a series of additional robustness checks, which are reported in the Appendix D - H. In these tests we address the following concerns: the distinction between the treated coders and coders in the control group stays the same over time (Appendix D); 2) the treatment effect is consistent when comparing only with the most recent months in the pre-treatment period (Appendix E); 3) the advantage of junior coders in utilizing AI is not driven by their rapid reaction to the AI implementation, but rather, the advantage is consistent over time (Appendix F); 4) the findings persist if different thresholds are used to split senior and junior coders (Appendix G); and 5) the findings are not driven by the learning effects of new coders (Appendix H).

4.6 What Mechanisms Underlie Differences in Response to AI?

Our finding that AI helps junior workers more during their peak performance period supports the theoretical conjecture that AI complements, rather than substitutes, human workers. However, if AI and human capital are truly complementary, it is puzzling why senior workers, with more experience, benefit less from AI compared to junior coders. We previously suggested that "experts" may have a tendency to disregard the recommendations of an AI, preferring instead to rely on their own judgment. In this section, we conduct additional qualitative analyses to further uncover the mechanism of AI's impact on senior coders.

We first collected qualitative feedback from senior coders through focus groups and a formal survey after the AI was implemented. We learned that senior coders do not trust AI to perform as well as humans and tend to focus on the errors made by the AI. Senior coders also complained more about the errors than junior coders did. Indeed, one senior coder commented:

> "*I don't fully trust the tool to identify codes. I haven't been told if it is supposed to highlight knowns or not so when I see a known not highlighted I question if the tool is working correctly."*

Also, given that the senior coders have higher skill levels, they are more likely to detect imperfections in the AI output, which further deteriorates their trust in the AI:

> "*Many areas of the record are highlighted that are not appropriate for coding. Once one area of the record, whether or not appropriately, is highlighted, I need to review the entire record. I have not found this to be helpful."*

The comments suggest that as a result of their low trust in AI, these senior coders opted to review all the information in the charts rather than solely focusing on the areas highlighted by AI. The role of trust is also supported by an additional small-scale lab study we conducted in the company's work environment. The nine coders[20] in this lab study were typical coders recruited from the coding team, and they had no prior exposure to the AI. They were asked to code 100 pre-selected charts (randomly selected from the medical chart pool). While all nine coders worked on the 100 medical charts independently, they were randomly assigned into three groups (three coders per group) that used different coding methods.

[20] The goal of this lab study is to explore the role of trust on the effect of this AI. Overall, we want to examine the impact of the provided instructions, which are used to mimic the actions with and without trust. We don't expect that senior and junior coders will act significantly differently in following the detailed instructions. Therefore, the coders do not need to be senior coders. These 9 coders are selected because they were not exposed to this AI yet, and it has a mixture of seniority level that are balanced across study groups.

Specifically, Group 1 was the control group without AI, which involved reading through the whole medical chart to find HCC codes, replicating the standard coding practice before AI implementation. The method for Group 2 was designed to mimic the case of coders who do not trust AI: although AI findings were provided to them, these coders were instructed to not rely on the AI results but to still review all the information in the chart. Group 3 was designed to replicate the scenario in which coders trust AI. Therefore, coders in Group 3 only validated the AI findings, and the validated HCC codes constituted their final coding result.

Table 15. Verifying the mechanism of low productivity due to lack of trust.

	Group 1 Control group			Group 2 No trust in AI			Group 3 High trust in AI		
Coder	1	2	3	4	5	6	10	11	12
Coder tenure (years)	7	4	6	7	2	9	5	7	9
Per chart time (min)	15.77	11.43	18.91	18.97	22.96	23.4	9.14	7.98	8.29
NumHCC per chart	1.54	1.62	1.68	1.64	1.61	1.66	1.54	1.46	1.54
Num of charts	100	100	100	100	100	100	99	98	99
Num of AI findings	/	/	/	1095	1095	1095	1095	1084	1095
Per chart time (min)	15.37			21.78			8.45		
NumHCC per chart	1.61			1.64			1.51		
NumHCC per min	0.1047			0.0753			0.1787		

Note: we used *NumHCC per chart* as a proxy of quality measure, which shows no statistically significant differences across the three groups. In addition, as indicated in the last row, the number of HCC codes per minutes also suggests that the comprehensive productivity is the highest for high trust in AI and the lowest for no trust in AI.

The results are reported in Table 15. Group 3, whose coding method placed a high level of trust in AI, achieved much higher productivity ((21.78-8.45)/21.78=61.2% less coding time) than Group 2 (no trust). Group 3 also demonstrated 45.0% ((15.37-

8.45)/15.37) more productivity than the control group. This result confirms that AI can significantly improve productivity if coders trust it. It is also noteworthy that the average medical chart coding time of Group 2 is much higher (41.7%) than the control group. This demonstrates one critical point: AI is not necessarily beneficial – in fact, a lack of trust in AI could lead to a negative impact on productivity. The findings from this lab study therefore show the importance of trust in leveraging AI for high productivity.

Lastly, we note that in this lab study we sought to create contexts with sharp differences: i.e., coders were instructed to fully trust or not trust the AI. In reality, coders' trust in AI more likely exists on a continuum between full trust and no trust. This low trust might explain why some senior coders in our main study still benefit from AI, but less so than junior coders.

Overall, evidence from the qualitative feedback as well as the lab study indicated that trust in AI plays an important role in AI productivity gain. Since senior coders tend to believe that they have better expertise than AI and possess greater confidence in their own judgment, they display a more emphatic lack of trust. Therefore, increasing senior coders' trust in AI, or at least in the good intentions of those responsible for its implementation, could persuade and enable these highly experienced individuals to realize the full potential of AI in their work.

4.7 Discussion and Conclusion

AI is one of the most powerful new technologies at this point in history, and it could significantly affect the economy and change the way we work. However, we have limited knowledge about how AI affects individual workers' productivity. In this empirical study in a knowledge-intensive work setting, we find that AI has a positive impact on knowledge workers' productivity, and crucially, we find that the magnitude of the benefit depends on the coders' circadian rhythm. Human experts perform best on medical chart coding in the morning and worst at night; correspondingly, AI boosts performance most in the morning and least at night. We also find that the benefits of AI are heterogeneous among knowledge workers, with junior coders in our study experiencing greater gains in productivity. Senior coders, although more highly experienced and knowledgeable, are less likely to trust AI and therefore benefit less from it.

The findings in this study also offer indications on one fundamental issue, which is whether AI is displacing human labor. The overall positive effect of AI, the heterogeneity regarding circadian rhythm, and the high trust group (Group 3) in the lab study are all confirming the complementarity between AI and human experts. This strongly suggests that while AI can replace a significant portion of human input, this replacement cannot be extended to all human work, which means part of human input is irreplaceable in this setting. Moreover, as we find that high performance (the peak of circadian rhythm) can better realize AI's value, thus, it is reasonable to believe senior coders are capable of gaining more than their junior colleague. In other words, AI does not erase the important role of work experience. Since our study setting is a typical knowledge work environment and the level of AI is quite representative, we expect these implications to be highly applicable to other contexts.

Our work contributes to research at the nexus of healthcare and IT. IS researchers have studied the impact of different technologies in healthcare (Ganju et al., 2016; Appari et al., 2018), and the human ability to utilize technology is a core focus of this literature (Atasoy et al., 2017; Karahanna et al., 2019). Our study highlights the need to better understand the potential of AI in the healthcare field. Broadly speaking, this study also contributes to the literature on the business value of information technology and on SBTC theory.

We acknowledge several limitations of this study. First, the sample is drawn from one company, which may cause concerns about the generalizability of our findings. However, the coding task performed by this company's employees is typical of knowledge work tasks. In addition, the AI created for this study is representative of AI at the present developmental stage. Therefore, the conclusions we derive from this study can inform a broader spectrum of contexts in which AI is used. Second, the coding industry is characterized by a high turnover rate. To ensure the continuity of productivity before and after the AI adoption, coders in the treatment group were not randomly selected but drawn from the company's more stable workers. We conducted extensive checks (including the falsification test, the placebo test, and the parallel trend) to strengthen the robustness of our findings. One interesting avenue for future research would therefore be to examine the effect of AI in a randomized field experiment.

Appendices

Appendix A: Details of Model Comparisons

Deep learning models

During our model development and as post comparison, we compared the performance of the proposed LSTM-DL model with different LSTM architectures as well as with other neural network models, as summarized in Table 16 and detailed below.

In *LSTM-combine*, instead of inputting the four sequential features into four parallel LSTM layers, we combined the four sequences and fed the merged sequence into an LSTM layer that is the same size as the summation of the four parallel layers in our original model. This adjustment in the design allows interactions of the four sequences. However, the unique patterns in each sequence would be less likely to be identified and utilized. This model achieved slightly worse performance than our proposed model.

LSTM-equalWeight does not emphasize the importance of the cost information. Instead of a 400-node layer for the cost sequence and 50-node layers for the other three sequences, the *LSTM-equalWeight* model reduces the cost LSTM layer to 50 nodes, which is the same weight as other sequences. This model's performance is $9 worse than the original model.

LSTM-equalNumNodes has the same number of nodes as the original LSTM-DL model. One concern with the *LSTM-equalWeight* model is that the total number of LSTM nodes was reduced, which could be a reason for its lower performance. Therefore, in *LSTM-equalNumNodes*, we increased the number of nodes for the four LSTM layers from 50 to 138, which combined is equal to the number of LSTM nodes in the original model. This model achieved a slightly better performance than *LSTM-equalWeight*, but it was still worse than our proposed model.

LSTM-noFC compares our original model to a model without the fully-connected layers immediately after the initial LSTM layers. In the design of our proposed model, these fully-connected layers are used to assemble the features identified by LSTM nodes. Given that one may question the necessity of using the fully-connected layers, we removed them for comparison purposes. The resulting performance is on par with the *LSTM-equalWeight* model.

Finally, we compared the LSTM model with a standard RNN, CNN1d, and CNN2d, using the exact architecture of the proposed model as well as the architecture of *LSTM-combine*. Please refer to Table 16 for details; the highest performing model in this group is *CNN2d-combine*, with an MAE that is $26 higher than the original proposed model.

Table 16. Performance of deep learning models.

Model Name	Final performance on testing data	Best performance on validation data	Model description
LSTM-combine	$2634	$2562	In LSTM-DL, the input series were taken by parallel LSTM layers first. This model merges the input series first and feeds them into a single LSTM layer.
LSTM-equalWeight	$2639	$2565	In LSTM-DL, the cost LSTM layer has 400 nodes while the other 3 LSTM layers have 50 nodes each. In this model, we reduce the number from 400 to 50 nodes.
LSTM equalNumNodes	$2637	$2563	LSTM-equalWeight reduces the number of nodes for one LSTM layer. Therefore, the total number of nodes is smaller than for our proposed model. In this model, we use approximately the same number of nodes (138 *4) as in the proposed model (400+50*3).
LSTM-noFC	$2639	$2567	One may ask whether it was necessary to use a fully-connected layer right after each LSTM layer. This model removes that fully-connected layer.
Standard RNN	$2657	$2583	This uses standard RNN instead of LSTM. Other architecture is exactly the same as our proposed model.
Fully-connected	$2710	$2634	This model uses fully-connected layers to replace LSTM layers for the deep learning model. Other architecture is exactly the same as our proposed model.
Fully connected-combine	$2789	$2712	This model uses fully-connected layers to replace LSTM layers for the deep learning model. Other architecture is exactly the same as the LSTM-combine model.
CNN1d	$2664	$2590	This model uses CNN1d instead of LSTM. Other architecture is exactly the same as our proposed model.
CNN1d-combine	$2666	$2588	This model uses CNN1d instead of LSTM. Other architecture is exactly the same as the LSTM-combine model.
CNN2d-combine	$2655	$2580	This model uses CNN2d instead of LSTM. Other architecture is exactly the same as the LSTM-combine model.

Note: The best model for each training dataset is selected based on its performance on the validating data.

Time series models

Beyond demonstrating that our proposed LSTM model performs best among deep learning models, we also compared it with time series models. Since our finding focuses on the fluctuations and other factors related to the input sequences, it is important to show that this performance cannot be easily achieved by ordinary time series models. More specifically, as reported in Row 1 of Table 17, we used the 48 monthly costs to predict the total cost in 2011, which is the summation of the 12 predicted monthly costs in 2011. We used the Python package "pyramid" to develop ARIMA models for each of the patients and select the best hyper-parameters for the ARIMA model. The performance of the *ARIMA-month* model yielded a worse performance than the baseline. In Row 2 of Table 17, our input was aggregated to the annual level. Using the four annual costs as a short time series input and using the same model development strategy, *ARIMA-year* achieved a performance of $3,128, which is better than the most basic baselines but still significantly worse than the proposed models. This comparison suggests that time series models perform far worse than the proposed LSTM-DL model in this task, which in part is explained by the high variability in the time series.

Table 17. Performance of ARIMA models.

Model Name	Final performance on testing data	Best performance on validation data	Model description
ARIMA-month	$3362	$3304	This ARIMA model uses the past 48 monthly costs to predict the 12 monthly costs in 2011. Grid search is applied to find the best model for each individual patient.
ARIMA-year	$3128	$3055	This ARIMA model uses the past 4 yearly costs to predict the annual cost in 2011. Grid search is applied to find the best model for each individual patient.

Appendix B: Study of Other Decompositions of Fluctuations

In the main body of the paper we studied the performance of the different models broken down by each patient's level of fluctuation. In this appendix we further study the impact of fluctuations broken down by time series components, then we analyze fluctuation using an alternative definition of fluctuation.

Time series decomposition

We use time series decomposition techniques to separate each patient's cost sequence into three components: trend, seasonality, and the remainder (Cleveland et al., 1990). The trend of a patient cost is the overall tendency of the healthcare cost to change over time, revealing its "changing direction." The seasonality of a member's cost series is the periodic variation within a fixed period (12 months, in this study). Seasonality is influenced by certain recurring factors, such as annual checkups, weather patterns, the patient's work calendar, etc. The remainder or noise comprises the leftover factors that cannot be explained by either trend or seasonality.

We decompose the monthly cost series using an additive time series model; each patient's monthly cost series (representing 48 months) is decomposed to trend, seasonality, and remainder series (each 48 months long). To implement this decomposition, we use a two-step approach. First, we de-trend the series utilizing changes in the moving average with a 12-month window. Second, we estimate seasonality on this de-trended series. The remainder is the original value with the time dependent trend and seasonality subtracted. Finally, we calculate the fluctuation in each of the time series components (utilizing Equation 3 in the main body).

Using the dependent variables defined in the main text, we fit the following regression models:

$$LSTMErrAbs_i = \beta_0 + \beta_1 Component\ Fluc_i + \sum_{j=2}^{4} \beta_j X_{ji} + \varepsilon_i \quad \ldots\ldots(8)$$

$$LSTM\_Model_i = \beta_0 + \beta_1 Component\ Fluc_i + \sum_{j=2}^{4} \beta_j X_{ji} + \varepsilon_i \quad \ldots\ldots(9)$$

where *Component Fluc*$_i$ refers to the fluctuation in each of the three time series components. *Trend Fluc*$_i$ is the fluctuation derived from the trend component of the monthly cost series of each patient; *Seasonal Fluc*$_i$ is each member's fluctuation derived from the seasonality component; and *REM Fluc*$_i$ is the fluctuation derived from the remainder component.

Similar to the analysis in the main body, we use two dependent variables to understand 1) how LSTM-DL performs given the fluctuation of different components, and 2) whether LSTM-DL performs better or worse than other methods when facing different levels of fluctuation in the three components. The significance of β_1 indicates 1) whether the LSTM-DL model is significantly impacted by the fluctuations of each of the three components (for Equation 8) and 2) whether the LSTM-DL model performs significantly better (negative coefficient) or worse (positive coefficient) than other methods when faced with fluctuations in each of the three components (after controlling for overall cost, age and gender) (for Equation 9).

Since all three components are highly correlated, we report how LSTM-DL performance is influenced by each of the three components independently in Column 1 of Tables 18 through 20. To make the coefficients of fluctuation comparable across all three components, we normalize the three fluctuations as independent variables. We note that the LSTM-DL model performs better when the fluctuation of all three components is high (after controlling for the member's overall cost, age and gender). These predictive advantages for members with high fluctuation may result from the ability of our LSTM-DL model to capture sequential changes as discussed in the main text. The similarity of magnitude among the three components reflects their high correlations but suggests that the LSTM-DL model can exert its maximum advantage in situations where there is high trend fluctuation (-293.12) compared to high seasonal fluctuation (-253.12) and high remainder fluctuation (-260.97).

Columns 2 through 6 of Tables 18 through 20 compare the performance of the LSTM model to that of the other machine learning models as a function of the fluctuation in decomposed time series. In summary, high trend fluctuation, seasonal fluctuation, and remainder fluctuations are all correlated to the high performance of our LSTM-DL model.

Table 18. Impact of trend fluctuation on LSTM model performance.

	(1) LSTMErrAbs	(2) LSTM-Past1year	(3) LSTM-Linear	(4) LSTM-LASSO	(5) LSTM-Ridge	(6) LSTM-RF
Trend Fluc	-293.12***	-1737.61***	-619.91***	-631.22***	-619.44***	-83.43***
	(8.99)	(24.94)	(9.96)	(9.83)	(9.95)	(5.24)
Paid2011	0.86***	0.21***	0.19***	0.19***	0.19***	0.03***
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Birth Year	5.51***	7.90***	15.98***	16.46***	15.98***	2.02***
	(0.36)	(1.00)	(0.40)	(0.39)	(0.40)	(0.21)
Gender	-87.14***	44.31*	18.81**	-4.24	18.77**	-3.79
	(8.39)	(23.26)	(9.28)	(9.17)	(9.28)	(4.89)
Constant	-11385.82***	-17173.97***	-32643.72***	-33561.31***	-32638.96***	-4155.25***
	(709.22)	(1967.13)	(785.18)	(775.34)	(785.01)	(413.52)

Notes: Standard errors in parentheses
*Significance is indicated by *** p<0.01, ** p<0.05, * p<0.1*
LSTMErrAbs is the absolute prediction error of the LSTM model; the columns headed LSTM-Past1year, LSTM-Linear, LSTM-LASSO, LSTM-Ridge and LSTM-RF show the differences between the prediction errors of the LSTM model and the baseline, the linear regression, the LASSO regression, the ridge regression and the RF prediction errors, respectively.

Table 19. Impact of seasonal fluctuation on LSTM model performance.

	(1) LSTMErrAbs	(2) LSTM-Past1year	(3) LSTM-Linear	(4) LSTM-LASSO	(5) LSTM-Ridge	(6) LSTM-RF
Seasonal Fluc	-253.12***	-1486.24***	-535.17***	-545.88***	-534.81***	-73.73***
	(8.87)	(24.92)	(9.91)	(9.79)	(9.91)	(5.16)
Paid2011	0.86***	0.20***	0.18***	0.18***	0.18***	0.03***
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Birth Year	6.10***	11.54***	17.23***	17.72***	17.23***	2.17***
	(0.36)	(1.01)	(0.40)	(0.40)	(0.40)	(0.21)
Gender	-85.47***	54.37**	22.34**	-0.66	22.29**	-3.33
	(8.41)	(23.63)	(9.39)	(9.28)	(9.39)	(4.89)
Constant	-12533.67***	-24278.59***	-35074.02***	-36015.91***	-35066.39***	-4446.31***
	(708.16)	(1988.97)	(790.83)	(781.39)	(790.64)	(411.82)

Notes: Standard errors in parentheses
*Significance is indicated by *** p<0.01, ** p<0.05, * p<0.1*
LSTMErrAbs is the absolute prediction error of the LSTM model; the columns headed LSTM-Past1year, LSTM-Linear, LSTM-LASSO, LSTM-Ridge and LSTM-RF show the differences between the prediction errors of the LSTM model and the baseline, the linear regression, the LASSO regression, the ridge regression and the RF prediction errors, respectively.

Table 20. Impact of remainder fluctuation on LSTM model performance.

	(1) LSTMErrAbs	(2) LSTM- Past1year	(3) LSTM-Linear	(4) LSTM- LASSO	(5) LSTM-Ridge	(6) LSTM-RF
REM Fluc	-260.97***	-1475.10***	-528.99***	-538.37***	-528.67***	-73.64***
	(8.87)	(24.95)	(9.92)	(9.81)	(9.92)	(5.16)
Paid2011	0.86***	0.20***	0.18***	0.18***	0.18***	0.03***
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Birth Year	6.02***	11.71***	17.31***	17.81***	17.31***	2.17***
	(0.36)	(1.01)	(0.40)	(0.40)	(0.40)	(0.21)
Gender	-85.23***	56.36**	23.08**	0.11	23.03**	-3.24
	(8.41)	(23.65)	(9.40)	(9.29)	(9.40)	(4.89)
Constant	-12385.26***	-24612.27***	-35239.74***	-36210.36***	-35231.44***	-4453.11***
	(707.54)	(1990.12)	(791.43)	(782.17)	(791.24)	(411.74)

Notes: Standard errors in parentheses

*Significance is indicated by *** p<0.01, ** p<0.05, * p<0.1*

LSTMErrAbs is the absolute prediction error of the LSTM model; the columns headed LSTM-Past1year, LSTM-Linear, LSTM-LASSO, LSTM-Ridge and LSTM-RF show the differences between the prediction errors of the LSTM model and the baseline, the linear regression, the LASSO regression, the ridge regression and the RF prediction errors, respectively.

Table 21. Impact of fluctuation on absolute model performance.

	(1) LSTMErrAbs	(2) Past1yearErrAbs	(3) LinearErrAbs	(4) LASSOErrAbs	(5) RidgeErrAbs	(6) RFErrAbs
Fluctuation	-42.62***	99.03***	9.56**	4.86	9.57**	-30.04***
	(-2.74)	(-8.36)	(-4.50)	(-4.49)	(-4.50)	(-3.06)
Paid2011	0.85***	0.69***	0.68***	0.68***	0.68***	0.83***
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Birth Year	7.27***	-15.15***	-13.83***	-14.60***	-13.82***	4.77***
	(-0.36)	(-1.11)	(-0.60)	(-0.60)	(-0.60)	(-0.41)
Gender	-71.91***	-178.71***	-113.42***	-89.37***	-113.38***	-72.57***
	(-8.50)	(-25.97)	(-13.99)	(-13.93)	(-13.99)	(-9.50)
Constant	-14385.92***	29669.92***	27713.72***	29266.98***	27697.60***	-9413.51***
	(-723.85)	(-2210.45)	(-1190.85)	(-1185.94)	(-1190.92)	(-808.69)

Notes: Standard errors in parentheses

*Significance is indicated by *** p<0.01, ** p<0.05, * p<0.1*

The columns headed LSTMErrAbs, Past1yearErrAbs, LinearErrAbs, LASSOErrAbs, RidgeErrAbs, and RFErrAbs show the absolute prediction errors of LSTM-DL, past one year, linear regression, LASSO regression, ridge regression, and RF, respectively.

Table 22. Impact of fluctuation on model performance difference.

	(1) LSTM-Past1year	(2) LSTM-Linear	(3) LSTM-LASSO	(4) LSTM-Ridge	(5) LSTM-RF
Fluctuation	-141.64***	-52.17***	-47.47***	-52.18***	-12.58***
	(-7.96)	(-3.14)	(-3.11)	(-3.14)	(-1.58)
Paid2011	0.17***	0.17***	0.17***	0.17***	0.03***
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Birth Year	106.79***	41.51***	17.46*	41.46***	0.66
	(-24.72)	(-9.76)	(-9.67)	(-9.75)	(-4.92)
Gender	22.42***	21.10***	21.88***	21.10***	2.51***
	(-1.06)	(-0.42)	(-0.41)	(-0.42)	(-0.21)
Constant	-44055.84***	-42099.64***	-43652.90***	-42083.53***	-4972.42***
	(-2103.82)	(-830.29)	(-822.84)	(-830.06)	(-418.52)

Notes: Standard errors in parentheses
*Significance is indicated by *** p<0.01, ** p<0.05, * p<0.1*
The columns headed LSTM-Past1year, LSTM-Linear, LSTM-LASSO, LSTM-Ridge and LSTM-RF show the differences between the prediction errors of the LSTM model and the baseline, the linear regression, the LASSO regression, the ridge regression and the random forest prediction errors, respectively.

An alternative measure of fluctuation

Fluctuation, time series decomposition and variability in costs are not the only way to represent cost changes. Motivated by the realities of health care costs – where small fluctuations may not reflect a change in the underlying health condition of the member – we define an alternative measure of fluctuation focusing on "large increases" within a period. Specifically, given a sequence of monthly costs $seq = [x_1, x_2, \ldots, x_{48}]$, we define the month over month change as change = $[diff_1, diff_2, \ldots, diff_{48}]$, where $diff_i = x_i - x_{i-1}$. When considering changes in health care costs, however, just considering the relative change is not sufficient, as a change from $1 to $5 is a 400% increase but is not meaningful. We therefore impose a twofold condition to specify a large change; the absolute difference needs to be larger than $50 and the relative increase needs to be larger than 100%. For each member, we summarize the number of large increases over the 48-month period. Tables 21 and 22 summarize the results. We perform the same regression analyses as before and find the results to be consistent with the other fluctuation analysis. In this case and after controlling for age, gender and the overall costs in 2011, the higher the number of meaningful increases, the better the LSTM model performance and its relative performance compared to other methods.

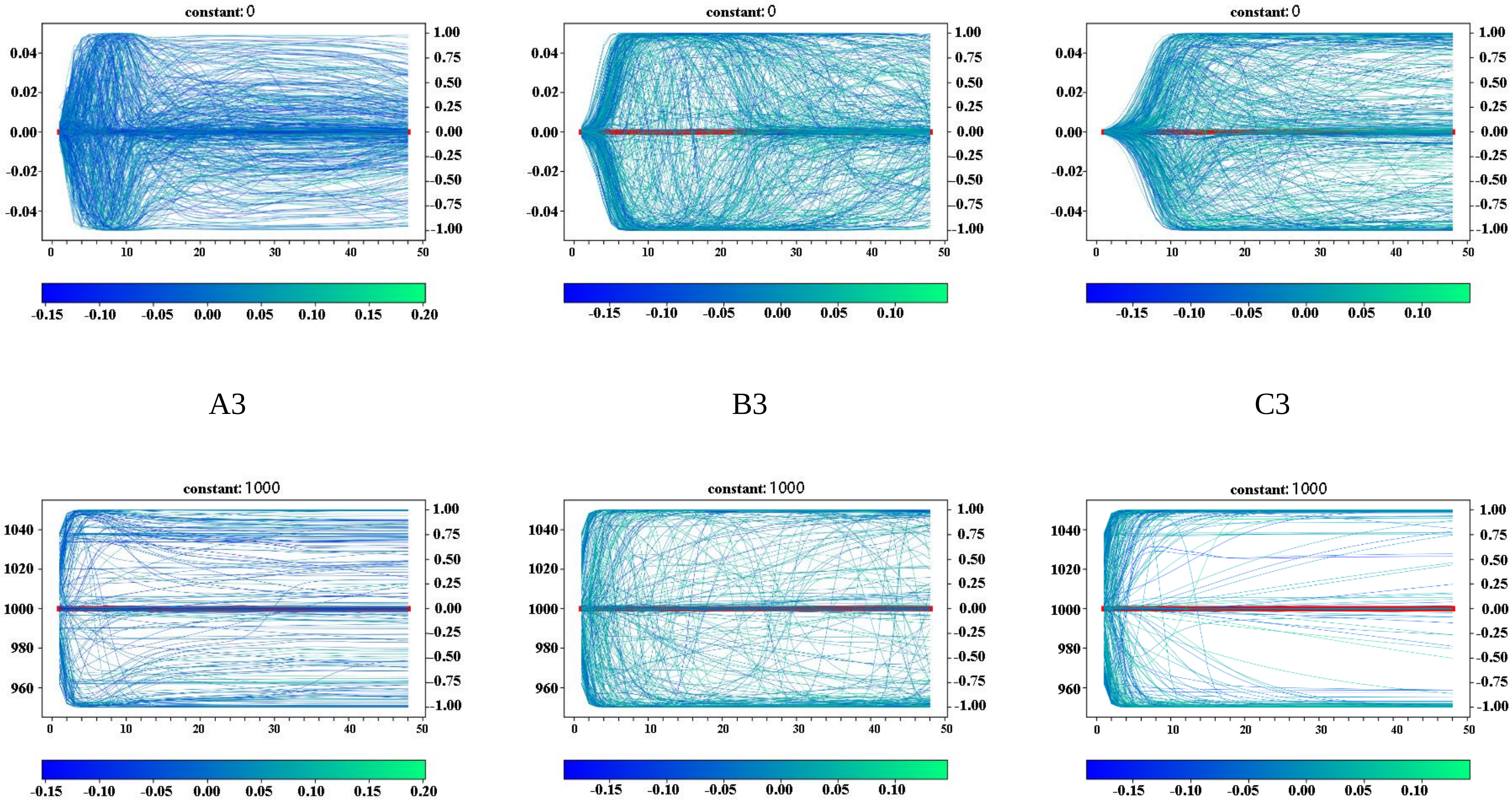
Appendix C: Signal Analysis of Polynomial and Step Functions

To further study the reactions of the LSTM units to the input sequences, we study their reactions to simple functions, and we include some examples here along with insights obtained through this analysis.

As explained in Figure 8 in the main text, we focus on the 400 LSTM units that process the cost information and visualize their output. The red line in each figure is the cost sequence that is input. Each of the 400 green-blue lines reflects the output of a single LSTM unit in response to the input. The shape of each line reflects the output sequence of the corresponding LSTM unit. The green-blue color represents the average weight given by the nodes in the fully-connected layer to the last value in the sequence; the darker the color, the more weight that is given to the output of the corresponding LSTM unit.

This study reinforces the findings in the main text about the LSTM units' ability to handle fluctuation in cost sequences of real patients. LSTM-DL-high consistently reacts with a high level of fluctuation to even constant or simple inputs. As for LSTM-DL-low, any input patterns are less reflected in the output. Across these cases, we consistently observe the LSTM-DL model abstracting patterns from the fluctuating inputs at a level between the levels of LSTM-DL-high and LSTM-DL-low. Using these simulated input sequences, we demonstrate that the ability to react to fluctuations is learned and not inherent from LSTM structure. This is consistently demonstrated in various sequences (generated by different functions).

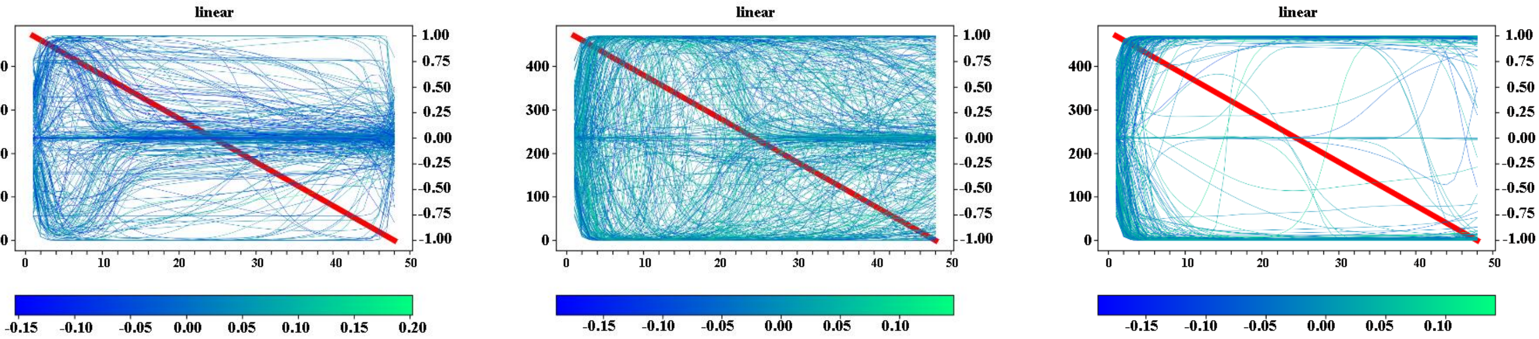
One special and interesting insight is that when the input is a constant (see Figure 15), the LSTM-DL model reflects the most stable output signal. LSTM-DL-high outputs signals with high fluctuations. LSTM-DL-low's output signals also exhibit more fluctuations than those of LSTM-DL.



A3 B3 C3

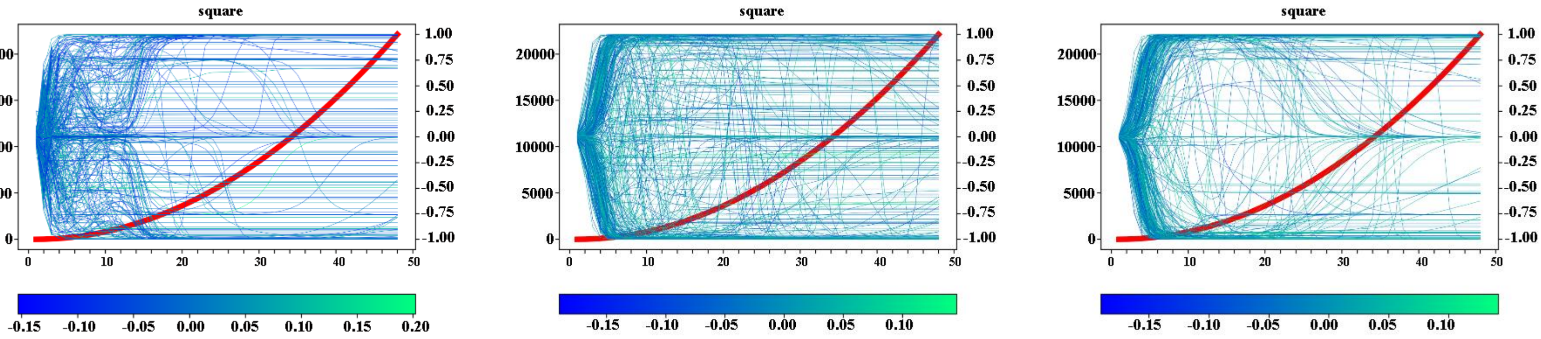
A4 B4 C4

Figure 15. Simulated input – constant.



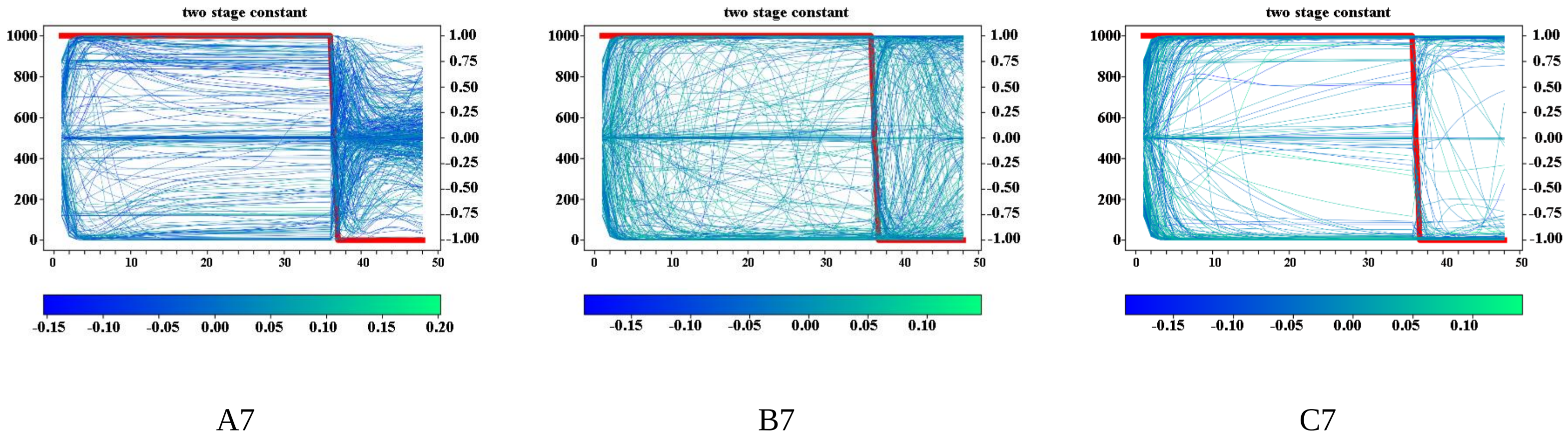
A5 B5 C5

Figure 16. Simulated input – linear.



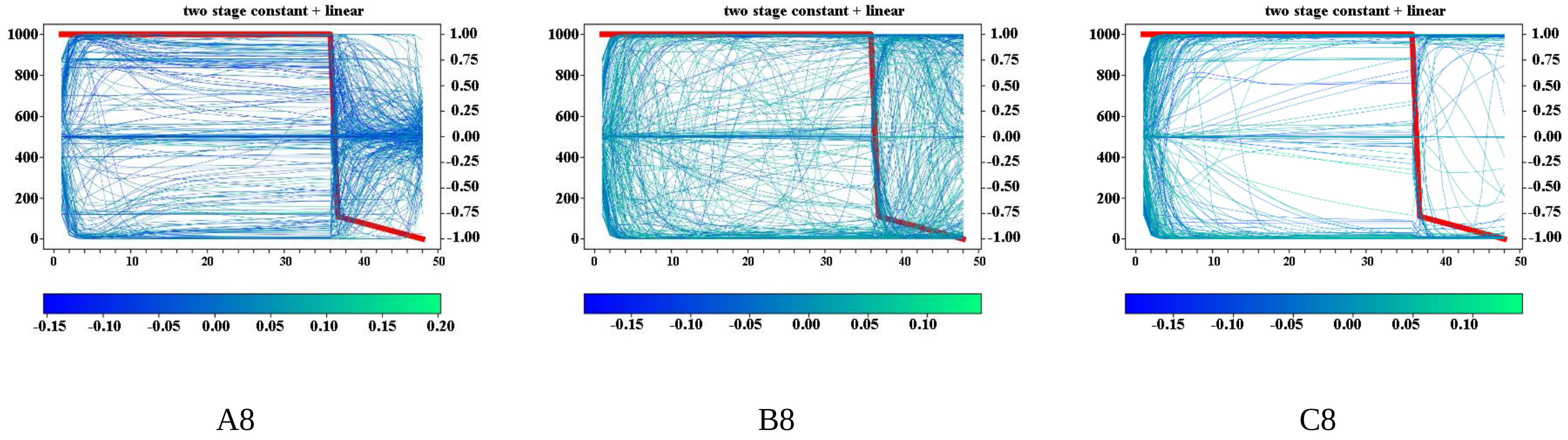
A6 B6 C6

Figure 17. Simulated input – quadratic.



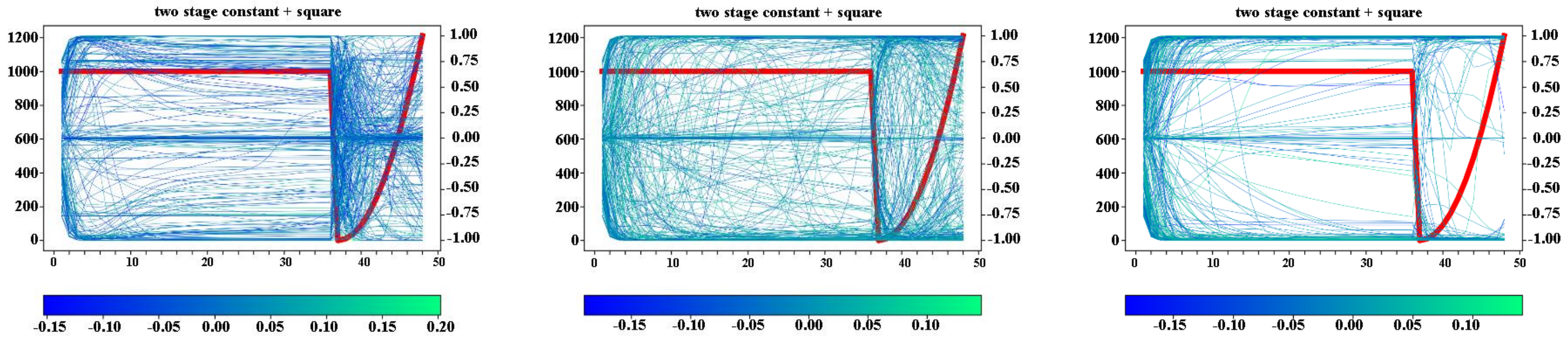
A7 B7 C7

Figure 18. Simulated input – two stage constant.



A8 B8 C8

Figure 19. Simulated input – two stage constant + linear.



A9 B9 C9

Figure 20. Simulated input – two stage constant + quadratic.

Appendix D: Placebo Test for Trends in the Pre-Treatment Period

Table 23. Additional robustness checks – parallel trend (with coder fixed effects).

	(1)	(2)
	Placebo	Placebo
Dependent Variable:	Review Time	Review Time
Post	-0.03	
	(0.43)	
Post X AI	0.24	
	(0.55)	
Post X Senior		0.16
		(0.66)
Post X Junior		-0.03
		(0.44)
Post X AI X Senior		-0.16
		(0.86)
Post X AI X Junior		0.46
		(0.74)
Constant	-4.62	-4.62
	(4.02)	(4.02)
Control Variables:	NumPage, Round of Coding, Type of Coding, Time of Day	
Observations	942,644	942,644
R-squared	0.06	0.06
Number of Coders	534	534

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

Our data covers a 17-month period, from July 2017 to October 2018. One potential concern is that the distinction between the control group and the treated group could change over time. Especially during the pre-period, the difference-in-differences estimation requires that the difference between the treatment group and control group is constant (i.e., the parallel trend assumption). Therefore, we conducted a placebo test in which we split the pre-period into two by considering the new calendar year as a “treatment.” In this placebo test, July–December 2017 is the pre-treatment period and January–April 2018 (months before the actual AI treatment) is the post-treatment period. We use the same models as in the main findings and report the results of this placebo treatment in Table 23. Neither the main treatment effect nor the heterogeneity across seniority levels are significant, indicating that distinction between the two groups is stable across the pre-period. This placebo test provides further evidence that the findings in the paper are driven by AI rather than by temporal trends.

Appendix E: Comparison with Recent Performance

Table 24. Additional robustness checks – comparison with recent performance (with coder fixed effects).

	(1) Same Year	(2) Same Year
Dependent Variable:	Review Time	Review Time
Post	2.96*** (0.84)	
Post X AI	-1.97*** (0.68)	
Post X Senior		1.45 (0.93)
Post X Junior		3.80*** (0.88)
Post X AI X Senior		-0.21 (0.68)
Post X AI X Junior		-3.00*** (0.95)
Constant	2.89 (1.76)	2.65 (2.06)
Control Variables:	NumPage, NumHCC, Round of Coding, Type of Coding, Time of Day	
Observations	602,152	602,152
R-squared	0.07	0.09
Number of Coders	483	483

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

We set the pre-treatment period at 10 months in order to confirm the parallel trend of the two groups. However, the length of the pre-treatment period may also raise a concern, as one may argue that the actual treatment effect should be assessed by comparing the post-period with fewer distal months in the pre-treatment period. We therefore exclude the medical charts completed in the year 2017 and assess the treatment effects by comparing the post-period with only the final months of the pre-treatment period. As reported below, we find that our main findings are consistent and robust (Table 24), suggesting that our results are not driven by a comparison of the post-treatment period with a distant pre-period.

Appendix F: Temporary vs. Persistent Advantage of Junior Coders

Table 25. Additional robustness checks – effect on worker seniority over time (with coder fixed effects).

	(1) Excluding July 2018	(2) Excluding July 2018	(3) Excluding July&August 2018	(4) Excluding July&August 2018
Dependent Variable:	Review Time	Review Time	Review Time	Review Time
Post	-0.50		-1.17**	
	(0.46)		(0.46)	
Post X AI	-1.60***		-1.34**	
	(0.61)		(0.66)	
Post X Senior		-1.94***		-1.97***
		(0.60)		(0.61)
Post X Junior		-0.17		-0.41
		(0.50)		(0.51)
Post X AI X Senior		0.31		0.32
		(0.77)		(0.81)
Post X AI X Junior		-2.21***		-2.05**
		(0.84)		(0.95)
Constant	-10.32	-10.21		-3.84
	(6.57)	(6.40)		(4.05)
Control Variables:	NumPage, NumHCC, Round of Coding, Type of Coding, Time of Day			
Observations	1,148,554	1,148,554	1,075,923	1,075,923
R-squared	0.09	0.09	0.06	0.09
Number of Coders	546	546	543	543

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

One alternative explanation for why senior coders lag behind junior coders in realizing the benefits of AI is that it takes longer for senior coders to learn and adapt to new technologies. If that is the case, then junior coders' advantage should be temporary and would be mostly driven by the starting period (e.g., their younger age and higher technological aptitude means that they would adopt the technology faster).

To address this concern, we exclude the starting month (July) from the post-period and re-estimate the AI’s differential impact with regard to seniority levels. As presented in Table 25, all the results are consistent. We further exclude the first two months in the post-period (July and August 2018) and use only the last two months in the post-period (September and October 2018); this analysis yields the same result. This robustness check suggests that senior coders’ disadvantage in leveraging AI is persistent rather than temporary.

Appendix G: Thresholds for Seniority

Although the company’s managers suggested 10 years of experience as the threshold for senior coders, we conduct further analysis to reinforce the generalizability of the findings about seniority level. We use alternate thresholds of 8 years, 9 years, 11 years, and 12 years to define senior coders, and we estimate the same regression model as before to examine how seniority level influences the AI’s effects. The results are reported in Table 26. The coefficient of Post X AI X Junior is significant (p<0.05) across all thresholds, while that of Post X AI X Senior is insignificant. We also see that the magnitude is bigger for junior coders than senior coders.

Table 26. Additional robustness checks – threshold for seniority (with coder fixed effects).

	(1)	(2)	(3)	(4)
	8+ Years as Senior	9+ Years as Senior	11+ Years as Senior	12+ Years as Senior
Dependent Variable:	Review Time	Review Time	Review Time	Review Time
Post X Senior	0.81	0.82	1.40	2.20***
	(0.72)	(0.73)	(1.49)	(0.63)
Post X Junior	2.90***	2.82***	2.38***	2.35***
	(0.69)	(0.69)	(0.65)	(0.65)
Post X AI X Senior	-0.79	-0.83	-0.73	-1.44
	(0.79)	(0.83)	(1.68)	(1.52)
Post X AI X Junior	-1.26**	-1.25**	-1.64***	-1.61***
	(0.62)	(0.62)	(0.61)	(0.59)
Constant	-10.84*	-10.82*	-10.86*	-10.84*
	(6.36)	(6.37)	(6.56)	(6.56)
Control Variables:	NumPage, NumHCC, Round of Coding, Type of Coding, Time of Day			
Observations	1,231,447	1,231,447	1,231,447	1,231,447
R-squared	0.09	0.09	0.09	0.09
Number of coders	548	548	548	548

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

Appendix H: New Coders

One concern is that our findings regarding AI's effect could be driven by newly hired coders whose productivity is not stable. In this case, the findings would include the learning efforts of new coders (new coders in the treated group would become more productive, but new coders in the control group would become less productive). To address this concern, we exclude all coders who joined the coding team in 2018; 1 coder in the treated group and 43 coders in the control group were excluded. We examined AI's main effect and its heterogeneity on both seniority level and time of day. All the results remain consistent (Table 27).

Table 27. Additional robustness checks – new coders (with coder fixed effects).

	(1) No New Coders	(2) No New Coders Seniority Level	(3) No New Coders Time of Day
Dependent Variable:	Review Time	Review Time	Review Time
Post	2.35*** (0.65)		
Post X AI	-1.62*** (0.58)		
Post X Senior		0.96 (0.78)	
Post X Junior		2.56*** (0.67)	
Post X AI X Senior		0.11 (0.73)	
Post X AI X Junior		-2.16*** (0.78)	
Post X EarlyMorning			2.89*** (0.85)
Post X Morning			1.95** (0.81)
Post X Afternoon			2.43*** (0.68)
Post X Night			2.28** (0.90)
Post X AI X EarlyMorning			-2.05 (1.44)
Post X AI X Morning			-3.00*** (0.78)
Post X AI X Afternoon			-1.59*** (0.60)
Post X AI X Night			-0.42 (1.21)
Constant	-10.84* (6.57)	-10.74* (6.42)	-11.00* (6.58)
Control Variables:	NumPage, NumHCC, Round of Coding, Type of Coding, Time of Day		
Observations	1,214,320	1,214,320	1,214,320
R-squared	0.09	0.09	0.09
Number of Coders	504	504	504

Robust Standard Errors in Parentheses
*** p<0.01, ** p<0.05, * p<0.1

Bibliography

Abbasi A, Sarker S, Chiang R. H. (2016). "Big Data Research in Information Systems: Toward an Inclusive Research Agenda," Journal of the Association for Information Systems 17(2), p. 3.

Abbasi, A., Li, J., Adjeroh, D., Abate, M. and Zheng, W., (2019). "Don't Mention It? Analyzing User-Generated Content Signals for Early Adverse Event Warnings," Information Systems Research 30(3), pp.1007-1028.

Adamopoulos, P., Ghose, A. and Todri, V. (2018). "The Impact of User Personality Traits on Word of Mouth: Text-Mining Social Media Platforms," Information Systems Research 29(3), pp.525-777.

Adjerid, I., Adler-Milstein, J. and Angst, C. (2018). "Reducing Medicare Spending through Electronic Health Information Exchange: The Role of Incentives and Exchange Maturity," Information Systems Research 29(2), pp. 341-361.

Agarwal, R. & Dhar, V., (2014). "Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research," Information Systems Research, pp. 443-448.

Anderson, D., & Bjarnad´ottir, M. (2016). "When Is an Ounce of Prevention Worth a Pound of Cure? Identifying High-Risk Candidates for Case Management," IIE Transactions on Healthcare Systems Engineering 6(1), 22-32.

Angst, C.M., Agarwal, R., Sambamurthy, V. and Kelley, K., (2010). "Social Contagion and Information Technology Diffusion: The Adoption of Electronic Medical Records in US Hospitals," Management Science 56(8), pp.1219-1241.

Appari, A., Johnson, M.E. and Anthony, D.L., (2018). "Health IT and Inappropriate Utilization of Outpatient Imaging: A Cross-Sectional Study of US Hospitals," International Journal of Medical Informatics 109, pp.87-95.

Aschoff, J. (1965). "Circadian Rhythms in Man," Science 148(3676), pp. 1427-1432.

Asgari, E., and Mofrad, M. R. (2015). "Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics," PloS one 10(11), e0141287.

Ash, A.S., Ellis, R.P., Pope, G.C., Ayanian, J.Z., Bates, D.W., Burstin, H., Iezzoni, L.I., MacKay, E. and Yu, W. (2000). "Using Diagnoses to Describe Populations and Predict Costs," Health Care Financing Review 21(3), p. 7.

Atasoy, H., Chen, P.Y. and Ganju, K., (2017). "The Spillover Effects of Health IT Investments on Regional Healthcare Costs," Management Science 64(6), pp. 2515-2534.

Attenberg, J., Melville, P., Provost, F., and Saar-Tsechansky, M. (2011). "Selective Data Acquisition for Machine Learning," Cost-Sensitive Machine Learning, p. 101.

Autor, D. H., Levy, F., and Murnane, R. J. (2003). "The Skill Content of Recent Technological Change: An Empirical Exploration," The Quarterly Journal of Economics 118(4), pp. 1279-1333.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv preprint arXiv:1409.0473.

Bartel, A., Ichniowski, C., and Shaw, K. (2007). "How Does Information Technology Affect Productivity? Plant-Level Comparisons of Product Innovation, Process Improvement, and Worker Skills," The Quarterly Journal of Economics 122(4), pp. 1721-1758.

Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A., and Escobar, G. (2014). "Big Data in Health Care: Using Analytics to Identify and Manage High-Risk and High-Cost Patients," Health Affairs 33(7), pp. 1123-1131.

Bayati, M., Braverman, M., Gillam, M., Mack, K. M., Ruiz, G., Smith, M. S., and Horvitz, E. (2014). "Data-Driven Decisions for Reducing Readmissions for Heart Failure: General Methodology and Case Study," PloS one 9(10), e109264.

Ben-Assuli, O., and Padman, R. (2019). "Trajectories of Repeated Readmissions of Chronic Disease Patients: Risk Stratification, Profiling, and Prediction," MIS Quarterly, forthcoming.

Bertsimas, D., Bjarnadottir, M. V., Kane, M. A., Kryder, J. C., Pandey, R., Vempala, S., & Wang, G. (2008). "Algorithmic Prediction of Health-Care Costs," Operations Research 56 (6), pp. 1382-1392.

Beymer, D. J., Brannon, K. W., Chen, T., Hardt, M. A. W., Kumar, R. K., and Syeda-Mahmood, T. F. (2013). Machine Learning with Incomplete Data Sets, US.

Bjarnadottir, M.V., D. Czerwinsky & Guan Y. (2016). "The History and Modern Applications of Insurance Claims Data in Health Care Research," in Healthcare Data Analytics pp. 523-553.

Boyd, A.D., Li, J.J., Burton, M.D., Jonen, M., Gardeux, V., Achour, I., Luo, R.Q., Zenku, I., Bahroos, N., Brown, S.B. and Vanden Hoek, T., (2013). "The Discriminatory Cost of ICD-10-CM Transition between Clinical Specialties: Metrics, Case Study, and Mitigating Tools," Journal of the American Medical Informatics Association, 20(4), pp. 708-717.

Bradley, J.V., (1981). "Overconfidence in Ignorant Experts," Bulletin of the Psychonomic Society 17(2), pp. 82-84.

Bresnahan, T.F., Brynjolfsson, E. and Hitt, L.M., (2002). "Information Technology, Workplace Organization, and the Demand for Skilled Labor: Firm-Level Evidence," The Quarterly Journal of Economics 117(1), pp. 339-376.

Brynjolfsson, E., Hui, X. and Liu, M., (2018a). "Does Machine Translation Affect International Trade? Evidence from a Large Digital Platform," National Bureau of Economic Research.

Brynjolfsson, E., Rock, D., and Syverson, C. (2018b). "Artificial Intelligence and the Modern Productivity Paradox: A Clash of Expectations and Statistics," in The Economics of Artificial Intelligence: An Agenda, University of Chicago Press.

Capes, T., Coles, P., Conkie, A., Golipour, L., Hadjitarkhani, A., Hu, Q., Huddleston, N., Hunt, M., Li, J., and Neeracher, M. (2017). "Siri On-Device Deep Learning-Guided Unit Selection Text-to-Speech System," Proc. Interspeech 2017, pp. 4011-4015.

Case, A., and Deaton, A. (2017). "Mortality and Morbidity in the 21st Century," Brookings Papers on Economic Activity (2017), p. 397.

Caskey, R.N., Abutahoun, A., Polick, A., Barnes, M., Srivastava, P. and Boyd, A.D., (2018). "Transition to International Classification of Disease Version 10, Clinical Modification: The Impact on Internal Medicine and Internal Medicine Subspecialties," BMC Health Services Research 18(1), p. 328.

Chen, M., and Grabowski, D. C. (2017). "Hospital Readmissions Reduction Program: Intended and Unintended Effects," Medical Care Research and Review, 1077558717744611.

Cho, K., Van Merri¨enboer, B., Bahdanau, D. and Bengio, Y., (2014). "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," arXiv preprint arXiv:1409.1259.

Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F. and Sun, J., (2016a). "Doctor AI: Predicting Clinical Events via Recurrent Neural Networks," Machine Learning for Healthcare Conference, pp. 301-318.

Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., and Sun, J. (2016b). "Multi-Layer Representation Learning for Medical Concepts," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM2016, pp. 1495-1504.

Choi, E., Bahadori, M.T., Song, L., Stewart, W.F. and Sun, J., (2017). "GRAM: Graph-Based Attention Model for Healthcare Representation Learning," Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 787-795.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). "STL: A Seasonal Trend Decomposition," Journal of Official Statistics 6(1), pp. 3-73.

CMS. (2018). "NHE Fact Sheet," https://www.cms.gov/research-statistics-data-andsystems/statistics-trends-and-reports/nationalhealthexpenddata/nhe-fact-sheet.html. Retrieved May 5th, 2018.

Cumming, R., D. Knutson, B. Cameron, B. Derrick. (2002). "A Comparative Analysis of Claims-Based Methods of Health Risk Assessment for Commercial Populations," https://www.soa.org/Files/Research/Projects/2005-comp-analysis-methods-commercialpopulations.pdf. Retrieved June 25, 2018.

Danziger, S., Levav, J., and Avnaim-Pesso, L. (2011). "Extraneous Factors in Judicial Decisions," Proceedings of the National Academy of Sciences 108(17), pp. 6889-6892.

Davenport, T.H. and Dreyer, K.J., (2018). "AI Will Change Radiology, But It Won't Replace Radiologists," Harvard Business Review, pp. 1-5.

David, H. (2015). "Why Are There Still So Many Jobs? The History and Future of Workplace Automation," Journal of Economic Perspectives 29(3), pp. 3-30.

De Martino, B., Kumaran, D., Seymour, B., and Dolan, R. J. (2006). "Frames, Biases, and Rational Decision-Making in the Human Brain," Science 313(5787), pp. 684-687.

Decker, M., Fischer, M., and Ott, I. (2017). "Service Robotics and Human Labor: A First Technology Assessment of Substitution and Cooperation," Robotics and Autonomous Systems (87), pp. 348-354.

Desai, M. M., Bogardus Jr, S. T., Williams, C. S., Vitagliano, G., and Inouye, S. K. (2002). "Development and Validation of a Risk‐Adjustment Index for Older Patients: The High‐Risk Diagnoses for the Elderly Scale," Journal of the American Geriatrics Society 50(3), pp. 474-481.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., (2018). "Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805.

Dimick, C., (2010). "Achieving Coding Consistency," Journal of AHIMA 81(7), pp. 24-28.

Dranove, D., Forman, C., Goldfarb, A. and Greenstein, S. (2014). "The Trillion Dollar Conundrum: Complementarities and Health Information Technology," American Economic Journal: Economic Policy 6(4), pp. 239-70.

Duncan, I., Loginov, M., & Ludkovski, M. (2016). "Testing Alternative Regression Frameworks for Predictive Modeling of Health Care Costs," North American Actuarial Journal 20 (1), pp. 65-87.

Dzogang, F., Lightman, S. and Cristianini, N., (2018). "Diurnal Variations of Psychometric Indicators in Twitter Content," PloS One, 13(6), e0197002.

Ekins, S., (2016). "The Next Era: Deep Learning in Pharmaceutical Research," Pharmaceutical Research 33(11), pp. 2594-2603.

Florida Kidcare. (Accessed September 21, 2017, at https://www.floridakidcare.org/.)

Folkard, S., (1975). "Diurnal Variation in Logical Reasoning," British Journal of Psychology 66(1), pp. 1-8.

Frees, E. W., Jin, X., & Lin, X. (2013). "Actuarial Applications of Multivariate Two-Part Regression Models," Annals of Actuarial Science 7(2), 258-287.

Frizzell, J. D., Liang, L., Schulte, P. J., Yancy, C. W., Heidenreich, P. A., Hernandez, A. F., Bhatt, D. L., Fonarow, G. C., and Laskey, W. K. (2017). "Prediction of 30-Day All-Cause Readmissions in Patients Hospitalized for Heart Failure: Comparison of Machine Learning and Other Statistical Approaches," JAMA Cardiology 2(2), pp. 204-209.

Futoma, J., Morris, J., and Lucas, J. (2015). "A Comparison of Models for Predicting Early Hospital Readmissions," Journal of biomedical informatics (56), pp. 229-238.

Galson, S., and Simon, G. (2016). "Real-World Evidence to Guide the Approval and Use of New Treatments," National Academy of Medicine Washington, DC.

Ganju, K.K., Pavlou, P.A. and Banker, R.D., (2016). "Does Information and Communication Technology Lead to The Well-Being of Nations? A Country-Level Empirical Investigation," MIS Quarterly 40(2), pp. 417-430.

Ganser, M., Dhar, S., Kurup, U., Cunha, C., & Gacic, A. (2015). "Patient Identification for Telehealth Programs," 14th International Conference on Machine Learning and Applications (ICMLA) pp. 360-363. IEEE.

Gitelman, L. (2013). “Raw data” Is an Oxymoron, MIT press Cambridge, MA.

Golas, S. B., Shibahara, T., Agboola, S., Otaki, H., Sato, J., Nakae, T., Hisamitsu, T., Kojima, G., Felsted, J., and Kakarmath, S. (2018). "A Machine Learning Model to Predict the Risk of 30-Day Readmissions in Patients with Heart Failure: A Retrospective Analysis of Electronic Medical Records Data," BMC Medical Informatics and Decision Making 18(1), p. 44.

Goldin, C., and Katz, L. F. (1998). "The Origins of Technology-Skill Complementarity," The Quarterly Journal of Economics 113(3), pp 693-732.

Grand View Research. (2018). "Medical Coding Market Size, Share & Trends Analysis Report by Classification System (International Classification of Diseases, Healthcare Common Procedure Code System), By Component, And Segment Forecasts, 2018 - 2025."

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., and Cuadros, J. (2016). "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs," JAMA 316(22), pp. 2402-2410.

Hainc, N., Federau, C., Stieltjes, B., Blatow, M., Bink, A., and Stippich, C. (2017). "The Bright, Artificial Intelligence-Augmented Future of Neuroimaging Reading," Frontiers in Neurology (8), p. 489.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification," Proceedings of the IEEE International Conference on Computer Vision, pp. 1026-1034.

Hon, C. P., Pereira, M., Sushmita, S., Teredesai, A., and De Cock, M. (2016). "Risk Stratification for Hospital Readmission of Heart Failure Patients: A Machine Learning Approach," Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, ACM2016, pp. 491-492.

Hosanagar, K., (2019). A Human's Guide to Machine Intelligence: How Algorithms are Shaping Our Lives and how We Can Stay in Control, Viking.

Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H., and Aerts, H. J. (2018). "Artificial Intelligence in Radiology," Nature Reviews Cancer, p. 1.

Jarow, J. P., LaVange, L., and Woodcock, J. (2017). "Multidimensional Evidence Generation and FDA Regulatory Decision Making: Defining and Using “Real-World” Data," JAMA 318(8), pp. 703-704.

Jensen, P. B., Jensen, L. J., and Brunak, S. (2012). "Mining Electronic Health Records: Towards Better Research Applications and Clinical Care," Nature reviews. Genetics 13(6), p. 395.

Kahneman, D., and Tversky, A. (1979). "Prospect Theory: An Analysis of Decision under Risk," Econometrica 47(2), pp. 263-291.

Kansagara, D., Englander, H., Salanitro, A., Kagen, D., Theobald, C., Freeman, M., and Kripalani, S. (2011). "Risk Prediction Models for Hospital Readmission: A Systematic Review," JAMA 306(15), pp. 1688-1698.

Karahanna, E., Chen, A., Liu, Q.B. and Serrano, C., (2019). "Capitalizing on Health Information Technology to Enable Advantage in US Hospitals," MIS Quarterly 43(1), pp. 113-140.

Karpathy, A., (2019a). "Hacker's guide to Neural Networks," http://karpathy.github.io/neuralnets/. Retrieved December 2, 2019.

Karpathy, A., (2019b). "A Recipe for Training Neural Networks," http://karpathy.github.io/2019/04/25/recipe/. Retrieved December 2, 2019.

Kaushik, S., Choudhury, A., Dasgupta, N., Natarajan, S., Pickett, L. A., & Dutt, V. (2017). "Using LSTMs for Predicting Patient's Expenditure on Medications," International Conference on Machine Learning and Data Science (MLDS) pp. 120-127. IEEE.

Kim Y.J. and Park H. (2019). "Improving Prediction of High-Cost Health Care Users with Medical Check-Up Data," Big Data 7(3), pp. 163-175.

Kingma, D. P., & Ba, J. (2014). "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980.

Kitamura, T., Onishi, K., Dohi, K., Okinaka, T., Ito, M., Isaka, N. and Nakano, T., (2002). "Circadian Rhythm of Blood Pressure Is Transformed from a Dipper to a Non-Dipper Pattern in Shift Workers with Hypertension," Journal of Human Hypertension 16(3), p.193.

Korinek, A., and Stiglitz, J. E. (2017). "Artificial Intelligence and Its Implications for Income Distribution and Unemployment," National Bureau of Economic Research.

Kose, I., Gokturk, M. and Kilic, K., (2015). "An Interactive Machine-Learning-Based Electronic Fraud and Abuse Detection System in Healthcare Insurance," Applied Soft Computing 36, pp. 283-299.

Krive, J., Patel, M., Gehm, L., Mackey, M., Kulstad, E., Lussier, Y.A. and Boyd, A.D., (2015). "The Complexity and Challenges of the International Classification of Diseases, Ninth Revision, Clinical Modification to International Classification of Diseases, 10th Revision, Clinical Modification Transition in Eds," The American Journal of Emergency Medicine 33(5), pp. 713-718.

Krupinski, E.A., Berbaum, K.S., Caldwell, R.T., Schartz, K.M. and Kim, J., (2010). "Long Radiology Workdays Reduce Detection and Accommodation Accuracy," Journal of the American College of Radiology 7(9), pp.698-704.

Krittanawong, C., Zhang, H., Wang, Z., Aydar, M. and Kitai, T., (2017). "Artificial Intelligence in Precision Cardiovascular Medicine," Journal of the American College of Cardiology 69(21), pp. 2657-2664.

Krusell, P., Ohanian, L. E., Ríos ‐ Rull, J. V., and Violante, G. L. (2000). "Capital ‐ Skill Complementarity and Inequality: A Macroeconomic Analysis," Econometrica 68(5), pp. 1029-1053.

Kuo, R. N., Dong, Y. H., Liu, J. P., Chang, C. H., Shau, W. Y., & Lai, M. S. (2011). "Predicting Healthcare Utilization Using a Pharmacy-Based Metric with the WHO’s Anatomic Therapeutic Chemical Algorithm," Medical Care, pp. 1031-1039.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). "Deep Learning," Nature 521(7553), pp. 436-444.

Lee, C.S., Nagy, P.G., Weaver, S.J. and Newman-Toker, D.E., (2013). "Cognitive and System Factors Contributing to Diagnostic Errors in Radiology," American Journal of Roentgenology 201(3), pp. 611-617.

Li, P., Kim, M.M. and Doshi, J.A., (2010). "Comparison of the Performance of the CMS Hierarchical Condition Category (CMS-HCC) Risk Adjuster with the Charlson and Elixhauser Comorbidity Measures in Predicting Mortality," BMC Health Services Research 10(1), p. 245.

Li, Y., Chen, C.-Y., and Wasserman, W. W. (2015). "Deep Feature Selection: Theory and Application to Identify Enhancers and Promoters," International Conference on Research in Computational Molecular Biology, Springer2015, pp. 205-217.

Liu, X., Stoutenborough, J. and Vedlitz, A., (2017). "Bureaucratic Expertise, Overconfidence, and Policy Choice," Governance 30(4), pp. 705-725.

Liu, Y., and Zhang, J. (2018). "Deep Learning in Machine Translation," in Deep Learning in Natural Language Processing, Springer, pp. 147-183.

Liu, X., Zhang, B., Susarla, A. and Padman, R., (2019). "Go to YouTube and Call Me in the Morning: Use of Social Media for Chronic Conditions," Forthcoming, MIS Quarterly.

Manyika, J., Lund, S., Chui, M., Bughin, J., Woetzel, J., Batra, P., Ko, R., and Sanghvi, S. (2017). "Jobs Lost, Jobs Gained: Workforce Transitions in a Time of Automation," McKinsey Global Institute.

McKinney, S.M., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H., Back, T., Chesus, M., Corrado, G.C., Darzi, A. and Etemadi, M., (2020). "International Evaluation of an AI System for Breast Cancer Screening," Nature 577(7788), pp. 89-94.

Mennemeyer, S. T., Menachemi, N., Rahurkar, S., and Ford, E. W. (2016). "Impact of the HITECH Act on Physicians’ Adoption of Electronic Health Records," Journal of the American Medical Informatics Association 23(2), pp. 375-379.

Meyer, G., Adomavicius, G., Johnson, P.E., Elidrisi, M., Rush, W.A., Sperl-Hillen, J.M. and O’Connor, P.J. (2014). "A Machine Learning Approach to Improving Dynamic Decision Making," Information Systems Research 25(2), pp. 239-263.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). "Distributed Representations of Words and Phrases and Their Compositionality," Advances in Neural Information Processing Systems 2013, pp. 3111-3119.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., and Ostrovski, G. (2015). "Human-Level Control through Deep Reinforcement Learning," Nature 518(7540), p. 529.

Morid, M. A., Kawamoto, K., Ault, T., Dorius, J., & Abdelrahman, S. (2017). "Supervised Learning Methods for Predicting Healthcare Costs: Systematic Literature Review and Empirical Evaluation," American Medical Informatics Association Annual Symposium Proceedings (2017), p. 1312.

Morid, M. A., Sheng, O. R. L., Kawamoto, K., Ault, T., Dorius, J., and Abdelrahman, S. (2019). "Healthcare Cost Prediction: Leveraging Fine-Grain Temporal Patterns," Journal of Biomedical Informatics 91, p. 103113.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). "Deep learning Applications and Challenges in Big Data Analytics," Journal of Big Data 2(1), p. 1.

National Quality Forum (NQF) Report: All-Cause Admissions and Readmissions Measures—Final Report. Accessed October 9th, 2015, at: http://www.qualityforum.org/Publications/2015/04/All-Cause_Admissions_and_Readmissions_Measures_-_Final_Report.aspx.

Nguyen, P., Tran, T., Wickramasinghe, N. and Venkatesh, S., 2016. "Deepr: A Convolutional Net for Medical Records," IEEE Journal of Biomedical and Health Informatics 21(1), pp. 22-30.

Ouwerkerk, W., Voors, A. A., and Zwinderman, A. H. (2014). "Factors Influencing the Predictive Power of Models for Predicting Mortality and/or Heart Failure Hospitalization in Patients with Heart Failure," JACC: Heart Failure 2(5), pp. 429-436.

Pakdemirli, E., (2019). "Artificial Intelligence in Radiology: Friend or Foe? Where Are We Now and Where Are We Heading?" Acta Radiologica Open 8(2), p.2058460119830222.

Pant, G. and Sheng, O.R. (2015). "Web Footprints of Firms: Using Online Isomorphism for Competitor Identification," Information Systems Research 26(1), pp. 188-209.

Pasquale, F. (2015). The Black Box Society, Harvard University Press.

Pennington, J., Socher, R., and Manning, C. (2014). "Glove: Global Vectors for Word Representation," Proceedings of the 2014 Conference on Empirical Methods nn Natural Language Processing (EMNLP)2014, pp. 1532-1543.

Pope, G.C., Kautter, J., Ellis, R.P., Ash, A.S., Ayanian, J.Z., Iezzoni, L.I., Ingber, M.J., Levy, J.M. and Robst, J., (2004). "Risk Adjustment of Medicare Capitation Payments Using the CMS-HCC Model," Health Care Financing Review 25(4), p.119.

Pope, N. G. (2016). "How the Time of Day Affects Productivity: Evidence from School Schedules," Review of Economics and Statistics 98(1), pp. 1-11.

Powers, C. A., Meyer, C. M., Roebuck, M. C., & Vaziri, B. (2005). "Predictive Modeling of Total Healthcare Costs Using Pharmacy Claims Data: A Comparison of Alternative Econometric Cost Modeling Techniques," Medical Care 43(11), 1065-1072.

PwC (2017). "Sizing the Prize: What's the Real Value of AI for Your Business and How Can You Capitalise?"

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., and Sun, M. (2018). "Scalable and Accurate Deep Learning with Electronic Health Records," npj Digital Medicine 1(1), p. 18.

Ramamurthy, K. N., Wei, D., Ray, E., Singh, M., Iyengar, V., Katz-Rogozhnikov, D. & YuenReed, G. (2017). "A Configurable, Big Data System for On-Demand Healthcare Cost Prediction," 2017 IEEE International Conference on Big Data, pp. 1524-1533. IEEE.

Rose, S. (2016). "A Machine Learning Framework for Plan Payment Risk Adjustment," Health services research 51(6), pp. 2358-2374.

Rush, A. M., Chopra, S., and Weston, J. (2015). "A Neural Attention Model for Abstractive Sentence Summarization," arXiv preprint arXiv:1509.00685.

Russell, S., Dewey, D. and Tegmark, M., (2015). "Research Priorities for Robust and Beneficial Artificial Intelligence," AI Magazine, 36(4), pp. 105-114.

Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Kipper-Schuler, K.C. and Chute, C.G., (2010). "Mayo Clinical Text Analysis and Knowledge Extraction System (Ctakes): Architecture, Component Evaluation and Applications," Journal of the American Medical Informatics Association 17(5), pp. 507-513.

Schone, E., & Brown, R. (2013). "Risk Adjustment: What Is the Current State of the Art and How Can It Be Improved?" POLICY 1, p. 6.

Senot, C., Chandrasekaran, A., Ward, P. T., Tucker, A. L., and Moffatt-Bruce, S. D. (2015). "The Impact of Combining Conformance and Experiential Quality on Hospitals’ Readmissions and Cost Performance," Management Science 62(3), pp. 829-848.

Shameer, K., Johnson, K. W., Yahi, A., Miotto, R., Li, L., Ricks, D., Jebakaran, J., KOVATCH, P., Sengupta, P. P., and GELIJNS, S. (2017). "Predictive Modeling of Hospital Readmission Rates Using Electronic Medical Record-Wide Machine Learning: A Case-Study Using Mount Sinai Heart Failure Cohort," Pacific Symposium on Biocomputing 2017, World Scientific2017, pp. 276-287.

Sherman, R. E., Anderson, S. A., Dal Pan, G. J., Gray, G. W., Gross, T., Hunter, N. L., LaVange, L., Marinac-Dabic, D., Marks, P. W., and Robb, M. A. (2016). "Real-World Evidence—What Is It and What Can It Tell Us," New England Journal of Medicine 375(23), pp. 2293-2297.

Srinivasan, K., Currim, F., & Ram, S. (2017). "Predicting High Cost Patients at Point of Admission Using Network Science," IEEE Journal of Biomedical and Health Informatics 22(6), pp.1970-1977.

Sushmita, S., Newman, S., Marquardt, J., Ram, P., Prasad, V., Cock, M. D., & Teredesai, A. (2015). "Population Cost Prediction on Public Healthcare Datasets," Proceedings of the 5th International Conference on Digital Health 2015, pp. 87-94.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to Sequence Learning with Neural Networks," Advances in Neural Information Processing Systems 2014, pp. 3104-3112.

Syverson, C. (2017). "Challenges to Mismeasurement Explanations for the US Productivity Slowdown," Journal of Economic Perspectives 31(2), pp. 165-186.

Talley, E.M., Newman, D., Mimno, D., Herr II, B.W., Wallach, H.M., Burns, G.A., Leenders, A.M. and McCallum, A., (2011). "Database of NIH Grants Using Machine-Learned Categories and Graphical Clustering," Nature Methods 8(6), p. 443.

Tegmark, M., (2017). Life 3.0: Being Human in the Age of Artificial Intelligence, Knopf.

Tversky, A. and Kahneman, D., (1974). "Judgment under Uncertainty: Heuristics and Biases," Science 185(4157), pp. 1124-1131.

Tzeng, O. J. (1973). "Positive Recency Effect in a Delayed Free Recall," Journal of Verbal Learning and Verbal Behavior 12(4), pp. 436-439.

Van Dongen, H. P., and Dinges, D. F., (2000). "Circadian Rhythms in Fatigue, Alertness, and Performance," Principles and Practice of Sleep Medicine 20, pp. 391-399.

Wang, Q., Zhang, J., Song, S., and Zhang, Z. (2014). "Attentional Neural Network: Feature Selection Using Cognitive Feedback," Advances in Neural Information Processing Systems 2014, pp. 2033-2041.

Weibel, L., Brandenberger, G., Goichot, B., Spiegel, K., Ehrhart, J. and Follenius, M., (1995). "The Circadian Thyrotropin Rhythm Is Delayed in Regular Night Workers," Neuroscience Letters 187(2), pp. 83-86.

Williams, C. (2015). "AI Guru Ng: Fearing a Rise of Killer Robots Is Like Worrying about Overpopulation on Mars."

Wynand, P. M. M., De Ven, V., & Ellis, R. P. (2000). "Risk Adjustment in Competitive Health Plan Markets," in Handbook of Health Economics (1) pp. 755-845. Elsevier.

Xiao, C., Choi, E. and Sun, J., (2018). "Opportunities and Challenges in Developing Deep Learning Models Using Electronic Health Records Data: A Systematic Review," Journal of the American Medical Informatics Association 25(10), pp.1419-1428.

Zhang, D. J., Gurvich, I., Van Mieghem, J. A., Park, E., Young, R. S., and Williams, M. V. (2016). "Hospital Readmissions Reduction Program: An Economic and Operational Analysis," Management Science 62(11), pp. 3351-3371.

Zhao, Y., Ash, A. S., Ellis, R. P., Ayanian, J. Z., Pope, G. C., Bowen, B., & Weyuker, L. (2005). "Predicting Pharmacy Costs and Other Medical Costs Using Diagnoses and Drug Claims," Medical Care 43(1), pp. 34-43.

Zhao, L., Hu, Q., and Wang, W. (2015). "Heterogeneous Feature Selection with Multi-Modal Deep Neural Networks and Sparse Group Lasso," IEEE Transactions on Multimedia 17(11), pp. 1936-1948.

Zook, M., Barocas, S., Crawford, K., Keller, E., Gangadharan, S. P., Goodman, A., Hollander, R., Koenig, B. A., Metcalf, J., and Narayanan, A. (2017). "Ten Simple Rules for Responsible Big Data Research," PLoS Computational Biology 13(3), e1005399.