Predicting Visit Cost of Obstructive Sleep Apnea Using Electronic Healthcare Records With Transformer

ZHAOYANG CHEN1, LINA SILTALA-LI1, MIKKO LASSILA1, PEKKA MALO1, EEVA VILKKUMAA1, TARJA SAARESRANTA2,3, AND ARHO VELI VIRKKI2,3

1Department of Information and Service Management, Aalto University, 02150 Espoo, Finland
2Division of Medicine, Department of Pulmonary Diseases, Turku University Hospital and Sleep Research Centre, University of Turku, 20014 Turku, Finland
3Department of Pulmonary Diseases and Clinical Allergology, University of Turku, 20014 Turku, Finland

(Zhaoyang Chen and Lina Siltala-Li contributed equally to this work.) CORRESPONDING AUTHOR: L. SILTALA-LI (lina.siltala-li@aalto.fi)

This work was supported in part by the Foundation for Economic Education under Grant 16-9442, in part by the Paulo Foundation, and in part by the Helsinki School of Economics Foundation (HSE).

This article has supplementary downloadable material available at https://doi.org/10.1109/JTEHM.2023.3276943, provided by the authors.

ABSTRACT Background: Obstructive sleep apnea (OSA) is growing increasingly prevalent in many countries as obesity rises. Sufficient, effective treatment of OSA entails high social and financial costs for healthcare. Objective: For treatment purposes, predicting OSA patients’ visit expenses for the coming year is crucial. Reliable estimates enable healthcare decision-makers to perform careful fiscal management and budget well for effective distribution of resources to hospitals. The challenges created by scarcity of high-quality patient data are exacerbated by the fact that just a third of those data from OSA patients can be used to train analytics models: only OSA patients with more than 365 days of follow-up are relevant for predicting a year’s expenditures. Methods and procedures: The authors propose a translational engineering method applying two Transformer models, one for augmenting the input via data from shorter visit histories and the other predicting the costs by considering both the material thus enriched and cases with more than a year’s follow-up. This method effectively adapts state-of-the-art Transformer models to create practical cost prediction solutions that can be implemented in OSA management, potentially enhancing patient care and resource allocation. Results: The two-model solution permits putting the limited body of OSA patient data to productive use. Relative to a single-Transformer solution using only a third of the high-quality patient data, the solution with two models improved the prediction performance’s \( R^2 \) from 88.8% to 97.5%. Even using baseline models with the model-augmented data improved the \( R^2 \) considerably, from 61.6% to 81.9%. Conclusion: The proposed method makes prediction with the most of the available high-quality data by carefully exploiting details, which are not directly relevant for answering the question of the next year’s likely expenditure. Clinical and Translational Impact Statement: Public Health–Lack of high-quality source data hinders data-driven analytics-based research in healthcare. The paper presents a method that couples data augmentation and prediction in cases of scant healthcare data.

INDEX TERMS Cost prediction, healthcare data augmentation, obstructive sleep apnea, transformer.

I. INTRODUCTION Obstructive sleep apnea (OSA) is a chronic respiratory disease in which the upper airway repeatedly collapses during sleep. There is no question that this results in poor sleep quality, thereby leading to increased daytime drowsiness, deterioration of cognitive abilities, various comorbidities, and even high rates of traffic and workplace accidents [1], [2]. In addition, many studies attest to high morbidity and mortality associated with the disease [3], [4], [5]. The prevalence of clinically diagnosed OSA was 3.7% in the Finnish adult population [6]. In Finland alone, 1.46 million people are estimated to suffer from moderate to severe sleep apnea,
according to data presented in the Finnish Medical Journal [7]. This represents an astonishingly high percentage of the country’s population of 5.54 million. In the wake of growing public awareness of the serious health consequences possible if OSA is left untreated, it is reported a significant increase in referrals connected with sleep apnea [8].

It goes without saying that sufficient resources must be made available to match. All OSA patients should receive treatment. This requires physicians and healthcare decision-makers to plan budgets accurately and distribute supplies efficiently, for better resource allocation. Hence, they need information about the coming year’s potential costs for OSA-related visits. While electronic healthcare records (EHRs) are ideal for training predictive models with data on visits to physicians, laboratory tests, and therapies, OSA complicates the use of this rich source of data because it is a chronic disease that involves irregular check-up intervals, extensive follow-up, and highly individualized treatments using evolving technologies. Therefore, a dragon of chaos exists in mining EHRs here, brought in by inconsistent coding over the years, large quantities of missing data, human input or measurement error, and loss of follow-ups. To at least some extent, these issues frequently arise in data analytics involving EHRs, which is unsurprising when one considers the messy landscape wrought by the complexity of pathology and epidemiology. Irrespective of these difficulties, the burgeoning quantities of data collected in EHRs renders them one of the best resources for healthcare research, and data-driven studies need to grapple with them [9]. Researchers take many approaches to the problem of EHRs’ “data chaos” with one of the most popular applying state-of-the-art deep learning models since these do not presume any particular stochastic distributions to the data [10], [11], [12], [13], [14]. Still, few studies address predicting the cost of healthcare visits in a way that accounts for both total costs and the visit type at each point in time, let alone focus on making the most of the limited body of data available for particularly complex diseases.

Our study represents five key contributions to the state of the art:

i) We develop a data-augmentation algorithm that preserves the semantic invariance of discrete healthcare data.

ii) We propose a method to augment the input via a subset of the high-quality healthcare data, material that cannot otherwise directly serve addressing the research question.

iii) A multi-task loss function is designed for cost prediction that considers both the sum-total costs and the cost specific to each type of visit.

iv) We combine two Transformer models (one for data augmentation and the other for cost prediction) to achieve better predictive performance while tackling the problem of insufficient data.

v) Our research experiments with and hone the cost-prediction model by working with EHRs from Finland’s Turku University Hospital. The study appears to be among very few projects of this type for OSA. The code from this study is available via https://gitlab.com/lina.siltala/two_model_transformer_predict_cost.

II. RELATED WORK

The worldwide volume of clinical data exceeded 2,300 exabytes in 2020 [15]. This vast body of data holds tremendous potential for data-driven analytics to support decision-making in healthcare, assessment of pathology trajectories, public-health surveillance, precision medicine, and preventive treatment, since EHRs encompass data covering consultations with experts, lab tests, clinical notes, and medication records [16], [17], [18]. Since the sensitivity of the data held in EHRs makes them an obvious target for cyber-attacks and attractive for deliberate data leaks, strict regulations are in place for their use, such as the EU’s General Data Protection Regulation (GDPR), the United States Health Insurance Portability and Accountability Act (or HIPAA), and rules for the Australian government’s My Health Record system [9].

The ironic twist is that, through these, EHRs’ treasure trove of data is not readily amenable to research. Scholars gain access to healthcare data only after a lengthy process for specified research questions, and the research must comply with ethics codes and rules – e.g., the GDPR’s terms for purpose limitation (Article 5(1)(b)), “data minimization” (Article 5(1)(c)), storage restrictions (Article 5(1)(e)), and integrity and confidentiality (Article 5(1)(f)) [9]. It is, without doubt, imperative to protect individuals’ privacy by means of standardization and strict ethics, yet this does bring challenges. For example, it is not easy to obtain the quantities of data needed for solid studies, especially with regard to particular diseases. This marks a stark contrast against natural language processing (NLP), for which Wikipedia, libraries, and social media offer ample material.

Several further factors contribute to the difficulties of scholarly use of EHRs. Firstly, the records are created primarily with physicians and administration in mind, not for research purposes [17]. Also, discrepancies arise, brought on by changes in technology, adjustments to diagnostic codes, and variations in practices between or even within healthcare facilities. Heterogeneous and free-form data create further difficulties for EHR analysis, as do complex intra-patient variations. A fourth important factor is that these records do not cover patients comprehensively: most people visit physicians only when unwell [9]. Of the many challenges bundled with EHRs’ use in healthcare research, the two issues that we most needed to address for our study are the limited body of high-quality data available and the complicated characteristics of the data.

Data augmentation is one technique for solving the first of these problems. Scholars of computer vision have frequently employed it for such purposes as cultivating more image data or applying clipping, rotation, color changes, or blurring. It is easy to understand how such techniques could serve such applications even without domain knowledge, since we know that the image is not converted to something completely different. That is, the post-augmentation body of data has
retained the original’s semantics [19]. It is not so straightforward to apply these techniques to healthcare data. One of the reasons is that EHRs include many discrete variables. With these, keeping the semantic information intact is far more challenging than with the continuous variables that images involve. Reference [20] Work in the NLP domain, in contrast, points to possible ways forward, in that NLP data feature discrete variables and many scholarly efforts in that domain have tackled the problem of insufficient data, with a broad range of methods: random deletion, replacement, or injection [19]; dependency-tree morphing [21]; back-translation [22]; the manifold mixup regularization method [23]; unsupervised data augmentation [24]; kernel methods [25]; semantic augmentation [26]; and others. Work specifically with data augmentation for EHRs has applied contrastive learning to find similarity patterns [27], examined particular types of data (such as images [27] and textual clinical notes for patient-outcome prediction [20]), and explored subfields such as skin-lesion analysis [28]. Scholars have discussed the potential for addressing the data-quantity issue with deep learning via knowledge distilling [29], patient representation [30], vector learning with non-negative restricted Boltzmann machines (eNRBMs) [31], and transfer learning in the EHR context [32]. Workable data augmentation should be easy to implement while still improving the model’s performance for the primary goal. All the aforementioned tactics turned out to be challenging to implement for our goal of predicting the next year’s expenditures. While none of the techniques were directly applicable for our study, they offered inspiration for our augmentation method.

For each patient, the data records in our study are sequential and linked to specific visits, which vary in length. They are very similar to the many-to-many sequence-to-sequence (seq2seq) conditions in NLP [33]. Among the methods traditionally applied for seq2seq modeling are hidden Markov models (HMMs), latent semantic analysis (LSA), latent Dirichlet allocation (LDA), bag-of-words (BOW), skip-gram, words2vec, and global-vector representation [34], [35], [36], [37], [38], [39], [40], [41]. The renaissance of rapidly developing deep learning has channeled current approaches to sequential data mainly into the associated stream, though [42], with special attention surrounding the recurrent neural network and such variants as long short-term memory (LSTM) [43]. Among the state-of-the-art methods applied specifically in NLP-related work are variational autoencoders (VAEs) [44], [45], [46], [47], generative adversarial nets (GANs) [48], [49], adversarial learning for dialogue generation [50], text generation with reinforcement learning [51], [52], Transformer models [53], bidirectional encoder representation from Transformers (BERT) [54], [55], and individual solutions such as Generative Pre-trained Transformer 3 (GPT-2) [56] and ChatGPT [57].

Following in the wake of the deep learning revolution in the NLP field, much research with EHRs has considered scalable deep learning in light of the two domains’ parallels [12], [13], [58], [59], [60]. Many studies subject EHR data to deep learning for risk and disease prediction, data-privacy work, phenotyping, and disease classification, in a shift from labor-intensive feature engineering and other expert-driven methods. The target is data-driven approaches to representing complicated data in lower-dimensional space [61]. Several of these have demonstrated success in applying deep learning models in conjunction with EHRs. Among the tools produced are Deepr, using a conventional neural network (CNN) for deep extraction [62]; DoctorAI, which utilizes a recurrent neural network (RNN) for disease prediction; and the DeepCare system, incorporating LSTM for predicting medicine quantities [63]. Scholars have studied deep learning for numerous aspects of healthcare, such as predicting obesity [64], assessing the likelihood of readmission in cases of congestive heart failure [65], and providing diagnostic decision support via BERT [66]. Also, recent work has directed attention to the general issue of explainability, by means of RNNs and graphing of temporal data [10], [11], [67].

Encoder–decoder was the most suitable state-of-the-art deep learning technique for our research. Encoders provide an embedding that can successfully learn the latent patient representation while converting the multivariates to a lower-dimensional space. Decoder architecture uses the latent representation discovered in the encoder phase as the context information to learn autoregressively about the following visit. Thus, the model takes age, gender, and other temporal multivariable features of historical patient data as inputs, while its output is temporally univariate (visit costs only). A Transformer model accommodates several inductive biases for sequential forecasting [68]. This model type offers one of the most powerful encoder–decoder architectures because it has multi-head attention and self-attention [53].

Transformers are frequently compared with CNNs and RNNs. A CNN induces the inductive biases of invariance and locality with kernel functions, while an RNN handles temporal-invariance and locality-related inductive biases via its Markovian structure. In contrast against both of these, Transformers do not demand any strong assumption as to the data’s structural nature [68]. We refer the readers to the original paper of Transformer for more details [53]. Although many studies apply Transformers accordingly [68], [69], [70], few of them have produced EHR-based cost prediction that consider not only the sum of all costs but also the type of cost associated with each visit instance. Filling this gap, we took inspiration from the context-learning functionality of encoders that can retain semantic and syntactic information. Our literature-informed approach makes sure that the context of the patients is learned with a similar structure so that the patients’ semantics do not change significantly during data augmentation. Applying two Transformers with the same encoder structure made it easy to implement data augmentation from a subset of the data.
were recorded for each visit instance alongside the number of days between that and each visit for the variable \( \text{diff}_d\text{gn} \) as the start time of the study, then calculated the date of the patient’s first OSA diagnosis (identified as often do [10], [69]. To circumvent the issue, we applied a variables for all patients at the same time points, as studies with large quantities of “missing data” had we processed. On account of patients’ irregular visit intervals and differences in follow-up duration, we would have had to contend with large quantities of “missing data” had we processed variables for all patients at the same time points, as studies often do [10], [69]. To circumvent the issue, we applied a data-processing “trick” from survival analysis [71]: we set the date of the patient’s first OSA diagnosis (identified as G47.3 in EHRs) as the start time of the study, then calculated the number of days between that and each visit for the variable \( \text{diff}_d\text{gn} \), for “difference from diagnosis.” Other variables were recorded for each visit instance alongside \( \text{diff}_d\text{gn} \). These captured both static and time-varying information, such as age, gender, the type of visit, and specialist type, as shown in Fig. 2 and 4. More detailed data descriptions and filtering of OSA cohort are presented in our supplementary material.

Formally, each patient \( i \) is covered via sequential multivariate data until time \( t_i \), where the value of \( t \) at the maximum number of days from diagnosis may differ freely between patients on account of the differences in follow-up duration. All \( \text{diff}_d\text{gn} \) values for patient \( i \) form a set \( T_i = \{0, \ldots t_i, t_i \in \mathbb{N} \} \). The elements of \( T_i \) correspond to the patient’s visits, which can be represented by set \( S_i = \{1, 2, 3, \ldots s_i, s_i \in \mathbb{N} \} \), where \( s_i \) is the total number of visits for patient \( i \) (see Fig. 3). The patients’ data are represented as \( D = \{ \mathbf{d}_{ik} : i = 1, \ldots N, k \in S_i \} \), where \( N \) is the number of patients and \( \mathbf{d}_{ik} \) is a data vector with \( M \) variables corresponding to patient \( i \) and visit \( k \). As Fig. 1 and 3 indicate, we split the patients into two groups: those with under 365 days in all before the last visit (\( D^1 \)) and the other with more than 365 days’ follow-up duration (\( D^2 \)).

1) THE DATA FOR AUGMENTATION \( D^1 \).

The first set covers 11,862 patients, with 118,578 visit records. Because these patients’ consultations spanned less than 365 days, they could not be included in the cost prediction for the next year, which requires more than one year of visits. Hence, only set \( D^2 \) was available for cost prediction. That set contains only 4,885 patients. The low number of patients having data for more than 365 days is explained by the two facts: 1) about 30 % of patients discontinue the treatment within the first year and 2) the number of patients has been much lower before the year 2019. Therefore, predicting a year’s costs from so few patient data is highly challenging. It is natural to turn to data augmentation for a possible solution. In NLP settings, models that have learned the similarity of words can substitute another word for one that carries similar meaning in the context (e.g., “cat” in place of “dog”), or some grammatical (semantic) information may be extracted such that deleting or inserting words yields a new sentence without distorting the meaning. Healthcare-specific data augmentation, in contrast, is problematic, because such patterns of word similarity or grammar have not been found yet, especially with regard to certain diseases; therefore, there is no clear standard of what one can delete, replace, or inject for data augmentation that preserves the patient records’ semantic information. Although we could not use \( D^1 \) directly for the cost prediction, those sleep-apnea patients were treated at the same hospital as members of \( D^2 \). Thanks to the associated similarities in sequence patterns and other characteristics, extracting information from \( D^1 \) represented a feasible route for data augmentation to ameliorate the issue of the restricted pool of data for our prediction.

Data augmentation is designed to expand the input dataset in a manner that fills the material out with noise alongside the semantic information preserved, thereby improving the performance of the model [22]. The augmentation in our case entailed changing only the details of one or two visits, so as to keep the characteristic sequence pattern of sleep-apnea patients intact. Hence, when we trained Transformer model

---

**FIGURE 1.** For analysis, the data were filtered by the number of visits and total follow-up duration. \( D^1 \) is the set of patients with fewer than 365 days of follow-up, and \( D^2 \) contains those patients with more than 365 days’ follow-up.

**III. MATERIAL AND METHODOLOGY**

Our data processing and methods are detailed below. The overall aim is to predict visit expenditures with a combination of original and augmented data. We address the data-augmentation and the cost-prediction element separately in relation to both the data and the model.

**A. DATA**

The filtering and preparation of data for this study are presented in Fig. 1, 2, and 3. The data included the years from 2002 to 2019. Its procedures were approved under research permit T164/2019 from Turku University Hospital. On account of patients’ irregular visit intervals and differences in follow-up duration, we would have had to contend with large quantities of “missing data” had we processed variables for all patients at the same time points, as studies often do [10], [69]. To circumvent the issue, we applied a data-processing “trick” from survival analysis [71]: we set the date of the patient’s first OSA diagnosis (identified as G47.3 in EHRs) as the start time of the study, then calculated the number of days between that and each visit for the variable \( \text{diff}_d\text{gn} \), for “difference from diagnosis.” Other variables were recorded for each visit instance alongside \( \text{diff}_d\text{gn} \). These captured both static and time-varying information, such as age, gender, the type of visit, and specialist type, as shown in Fig. 2 and 4. More detailed data descriptions and filtering of OSA cohort are presented in our supplementary material.

Formally, each patient \( i \) is covered via sequential multivariate data until time \( t_i \), where the value of \( t \) at the maximum number of days from diagnosis may differ freely between patients on account of the differences in follow-up duration. All \( \text{diff}_d\text{gn} \) values for patient \( i \) form a set \( T_i = \{0, \ldots t_i, t_i \in \mathbb{N} \} \). The elements of \( T_i \) correspond to the patient’s visits, which can be represented by set \( S_i = \{1, 2, 3, \ldots s_i, s_i \in \mathbb{N} \} \), where \( s_i \) is the total number of visits for patient \( i \) (see Fig. 3). The patients’ data are represented as \( D = \{ \mathbf{d}_{ik} : i = 1, \ldots N, k \in S_i \} \), where \( N \) is the number of patients and \( \mathbf{d}_{ik} \) is a data vector with \( M \) variables corresponding to patient \( i \) and visit \( k \). As Fig. 1 and 3 indicate, we split the patients into two groups: those with under 365 days in all before the last visit (\( D^1 \)) and the other with more than 365 days’ follow-up duration (\( D^2 \)).

1) THE DATA FOR AUGMENTATION \( D^1 \).

The first set covers 11,862 patients, with 118,578 visit records. Because these patients’ consultations spanned less than 365 days, they could not be included in the cost prediction for the next year, which requires more than one year of visits. Hence, only set \( D^2 \) was available for cost prediction. That set contains only 4,885 patients. The low number of patients having data for more than 365 days is explained by the two facts: 1) about 30 % of patients discontinue the treatment within the first year and 2) the number of patients has been much lower before the year 2019. Therefore, predicting a year’s costs from so few patient data is highly challenging. It is natural to turn to data augmentation for a possible solution. In NLP settings, models that have learned the similarity of words can substitute another word for one that carries similar meaning in the context (e.g., “cat” in place of “dog”), or some grammatical (semantic) information may be extracted such that deleting or inserting words yields a new sentence without distorting the meaning. Healthcare-specific data augmentation, in contrast, is problematic, because such patterns of word similarity or grammar have not been found yet, especially with regard to certain diseases; therefore, there is no clear standard of what one can delete, replace, or inject for data augmentation that preserves the patient records’ semantic information. Although we could not use \( D^1 \) directly for the cost prediction, those sleep-apnea patients were treated at the same hospital as members of \( D^2 \). Thanks to the associated similarities in sequence patterns and other characteristics, extracting information from \( D^1 \) represented a feasible route for data augmentation to ameliorate the issue of the restricted pool of data for our prediction.

Data augmentation is designed to expand the input dataset in a manner that fills the material out with noise alongside the semantic information preserved, thereby improving the performance of the model [22]. The augmentation in our case entailed changing only the details of one or two visits, so as to keep the characteristic sequence pattern of sleep-apnea patients intact. Hence, when we trained Transformer model...
with 

\( \mathcal{M}^1 \) with \( \mathcal{D}^1 \), we processed the data as inputs and target in the following way: the patient history data \( \mathcal{D}^1_{\text{inp}} = \{d_i : i = 1, \ldots, N_1, 0 \leq k \leq s_i - 2, k \in S_i\} \), constitute the input, and the target (the patient’s visit cost type) is \( \mathcal{V}^1_{\text{out}} = \{v_{ik} : i = 1, \ldots, N_1, s_i - 2 < k \leq s_i, k \in S_i\} \) as shown in Fig. 3, where \( N_1 \) is the number of patients in \( \mathcal{D}^1 \), and \( v_{ik} = [v_{ikc}] \in \{0, 1\}^C \) is a binary vector, where \( C \) is the number of unique visit cost types, and \( v_{ikc} = 1 \) if and only if the cost type of visit \( k \) for patient \( i \) is \( c \).

2) THE DATA FOR COST PREDICTION \( \mathcal{D}^2 \).

The 4,885-patient dataset contains 89,571 visit records. These data were processed as inputs and targets (per Fig. 3 and 4) to predict, in our research setting, the visit costs over the last year (ignore leap year and only assume it has 365 days for simplicity). Since patients’ visit history and intervals may be different in length, their time indices for the allocation of inputs and targets vary accordingly. We set \( t'_i = \max\{x \in T_i : 0 \leq x < t_i - 364\} \) as the last time for the inputs and \( t''_i = \min\{x \in T_i : t_i - 364 \leq x \leq t_i\} \) as the first time for the targets. The visits corresponding to \( t'_i \) and \( t''_i \) are \( s'_i \) and \( s''_i \). Then, the inputs get expressed as \( \mathcal{D}^2_{\text{inp}} = \{d_{ik} : i = 1, \ldots, N_2, 0 \leq k \leq s'_i, k \in S_i\} \), and the targets (the patient’s visit cost type) are \( \mathcal{V}^2_{\text{out}} = \{v_{ik} : i = 1, \ldots, N_2, s''_i \leq k \leq s_i, k \in S_i\} \). For simplicity, we denote the times corresponding to the inputs and targets as set \( T_{\text{inp}} = \{0, \ldots, t'_i\} \) and \( T_{\text{out}} = \{t''_i, \ldots, t_i\} \).

Each unique visit cost type has an associated cost value. The patients’ visit costs can be represented as cost vectors, \( C^2_{\text{out}} = \{c_{ik} : i = 1, \ldots, N_2, s''_i \leq k \leq s_i, k \in S_i\} \), where the elements of \( c_{ik} \in \mathbb{R}^C \) represent the costs attributed to different cost types. If the visit is not attributed to a cost type \( c \in \{1, \ldots, C\} \), then \( c_{ikc} \) is set equal to zero. When feeding the data to the neural networks, special tokens [CLS] and [SEP] are inserted to indicate the first and final visit by the patient, respectively.

It is worth noting that, because \( \mathcal{D}^1 \) contains many more patients than \( \mathcal{D}^2 \), and, consequently, also many more visit cost types (91 cost types in \( \mathcal{D}^1 \) versus 50 cost types in \( \mathcal{D}^2 \)). To address the variation in the length of target sequences, all sequences are padded to have the same length.

B. THE MODEL ARCHITECTURE

Encoder–decoder models are very popular for seq2seq prediction problems, and Transformers are among the most powerful tools in this class [70]. Our choice to develop two Transformer models for enabling the efficient use of patient data in solid prediction of coming costs led to the architecture depicted in Fig. 5. This design addresses the two main differences between our work and application of seq2seq in NLP. 1) Rather than mostly univariate data (words in the NLP case), we had to factor in the multivariate nature of healthcare data (with variables such as age, gender, and specialists visited). For predicting next year’s visit costs, historical cost information is not the only relevant variable. Others too are important, because they reflect between-patient differences and within-patient variance during follow-up – demographic information plays important roles in visit patterns and trajectories. For our design to consider all of the most influential variables, we applied multivariable embedding in the encoder for \( \mathcal{M}^1 \) and \( \mathcal{M}^2 \), as shown in Fig. 4. 2) Because of the small quantity of data for training and the challenges created for transfer learning by the complexity of pathology and epidemiology, we needed a solution for efficiently putting data to use in healthcare studies. The trained model \( \mathcal{M}^1 \) with \( \mathcal{D}^1 \) for data augmentation held promise to solve this two-horned problem via more data and an alternative solution for transfer learning.

1) THE MODEL FOR DATA AUGMENTATION: \( \mathcal{M}^1 \).

As the diagram in Fig. 5 indicates, \( \mathcal{M}^1 \) must be trained and evaluated before it gets employed for data augmentation. Although, as Fig. 6 shows, \( \mathcal{M}^1 \) and \( \mathcal{M}^2 \) have identical encoder portions, the decoder in \( \mathcal{M}^1 \) uses only a fixed visit horizon (for augmentation inferring the next one or two visits) while the decoder part of \( \mathcal{M}^2 \) accounts for differences in
FIGURE 4. Split the data into inputs and target, and inputs’ embedding. The patient represented made eight visits, and the\n\[\text{diff_dgn} = \{0, 30, 70, 248, 690, 810, 1100, 1140\}\]. We have records extending until 1,140 days after this patient’s sleep-apnea diagnosis. The goal is to predict the visit\ncost over the last year (1,140 - 364 = 776) on the basis of the previous visits’ records (from the date of G47.3 diagnosis to day 776). Therefore, the time\nindices for the inputs and target are the maximum between 0 and 776 and the minimum between 776 and the last visit point, day 1,140. In line with the\ndefinition\n\[t'_i = \max\{x \in T_i : 0 \leq x < t_i - 364\}\] and \[t''_i = \min\{x \in T_i : t_i - 364 \leq x \leq t_i\}\], the inputs for this patient are the records from day 0 to day 690 from\ndiagnosis, and the targets are the visit costs from day 810 to 1,140. The autoregressive prediction mechanism of Transformer is shown at right.

FIGURE 5. The model architecture. One of the two components is for data augmentation (applied with sub-sample \(D_1\)), and the other is for\ncost prediction (utilizing augmented data and \(D_2\) in combination). The paper presents results from comparisons with the raw data, randomly\naugmented data, and Transformer \(M_1\) augmented data, to clarify the better prediction performance.

visit length, since patients’ visit frequencies in the next year\nwill differ. The loss function in \(M_1\) measures cross-entropy,\nwhich is often used for multi-label classification.

The algorithm developed for data augmentation includes\ndeletion, replacement, and insertion, all of which are common\ntactics, especially for augmentation in situations with discrete\nvariables [22]. Considering a function \(F\) that embodies the\ndata augmentation process, \(F\) accepts a subset of the original\ndata \(D_2\) as input and produces the augmented data \(\tilde{D}_2\) as\noutput. The concept of semantic invariance, as it pertains\nto our research, signifies that the probability distribution of\nsubsequent visits \(\tilde{D}_2\) closely approximates the distribution\nfound in \(D_2\). Given that the patients in \(D_1\) and \(D_2\) pre-\ndominantly share the same ethnicity, are treated at the same\nregional hospital, and all have OSA, the clinical relevance of\nthe augmented data \(\tilde{D}_2\) relies on the similarity between these\ntwo datasets with respect to domain-specific information.\nAdditionally, the high predictive performance of the trained\nmodel \(M_1\), featuring a constrained parameter range of visit\ntypes and limited to only two subsequent visit predictions,
Algorithm 1 – Using Transformer $M^1$ for Data Augmentation

Require: $M^1$, $D^2$, sample ratio

Ensure: augmented data

Step 1: Preparing the sample
1. Sample patients $D^2_s$ from $D^2$ with sample ratio
2. Define $T^2_s$ as a set of visit indices in $D^2_s$ that have taken place in less than 365 days
3. Define $V^2$ as the set of unique visit cost type vectors in $D^2$
4. Define $V^2_s$ as the visit cost type vectors for the patients included in sample $D^2_s$

Step 2: Predicting cost types of next visits in $D^2_s$
5. for $d_{ik} \in D^2_s$ do
6. # Check that visit index can be used as input
7. if $k \in T^2_s$ then
   # Predict the cost types of next visits
   $\hat{\nu}_{ik}, \hat{\nu}_{(k+1)} \leftarrow M^1.predit(d_{ik})$
8. else go to line 4
9. end if
10. # Check that predicted visit cost is supported in $D^2$
11. # Note: model $M^1$ supports visit cost types that are
12. # not found in $D^2$
13. if $\hat{\nu}_{ik} \notin V^2$ then
   delete $d_{ik}$ from $D^2_s$
   go to line 4
14. end if
15. # Augment predicted cost types to $D^2_s$
16. if ($\hat{\nu}_{ik} == \nu_k$) and ($\hat{\nu}_{(k+1)} \neq \text{null}$) then
   # Case 1: next visit is correctly predicted by $M^1$
   inject the 2nd predicted visit $\hat{\nu}_{(k+1)}$ in $D^2_s$
   go to line 4
17. else
   # Case 2: next visit is predicted incorrectly but found
   # as a visit cost type supported by $D^2$
   $\hat{\nu}_{ik} \leftarrow M^1.prob(d_{ik})$
   $c \leftarrow \arg\max\limits_{c \in \{1, \ldots, C\}} \{\hat{p}_{ikc}, \exists \nu_{ik} \in V^2, \nu_{ikc} = 1\}$
   define $\nu$ such that $\nu_c = 1$ and $\nu_j = 0$ if $j \neq c$.
   replace $\nu_{ik}$ with $\nu$ in $D^2_s$
18. end if
19. end for

2) THE MODEL FOR COST PREDICTION: $M^2$.
Taking an approach similar to that in NLP, we compute the conditional probability of a patient’s visit cost data $C_i \subseteq C_{out}$ for each patient $i$ given the corresponding input data $D_i \subseteq D_{inp}$ and previous visit costs

$$P(C_i \mid D_i) = P(c_{i,0}^{s_i} \mid D_i) \prod_{k=s_i+1}^{s_i} P(c_{i,k} \mid c_{i,k-1}, \ldots, c_{i,0}, D_i). \quad (1)$$

3) THE LOSS FUNCTION FOR $M^2$.
Although our primary objective is to predict the sum-total visit costs for the next year on the basis of demographic

details and information on visits (which occur sequentially during the last year of follow-up), it is important also to
predict the type of visit cost at each time point. This kind of prediction is necessary for two reasons: 1) From a practical perspective, it is highly informative for healthcare decision-makers. With this information, they not only can calculate annual costs accurately for budget purposes but also can efficiently allocate specific resources to individual departments in accordance with the predicted visit cost types. 2) From the computation standpoint, predicting individual visits serves to help calculate the labeled and predicted annual costs efficiently allocate specific resources to individual departments.

Because our objectives encompassed predicting two distinct major elements – the sum total of costs in the next year and each visit’s cost type – we used two sets of metrics for model evaluation.

C. EVALUATION METRICS

As the mean-squared error of the annual true cost and the annual predicted cost, where \( K_i \) is the number of visits for patient \( i \), \( \hat{p}_{ikc} \) is the predicted probability that the cost type of the visit \( k \) is \( c \), and \( e_c \) is the cost for a visit with type \( c \). Since the model \( M^2 \) does not directly predict the costs, we use the estimated probabilities \( \hat{d}_{ik} = M^2 \text{.prob}(d_{ik}) \), where \( d_{ik} \) is the data of patient \( i \) associated with the visit \( k \), together with expected visit type costs to approximate the total costs for a visit. Here, the actual cost of visit \( k \) by patient \( i \) is \( c_{ikc} \), when the visit is known to be of cost type \( c \). The predicted probabilities are calculated using soft-max function.

The multi-label cost \( L_2 \) is defined as the cross entropy loss

\[
L_2(C_b, D_b) = -\frac{1}{N_b} \sum_{i=1}^{N_b} \left( \frac{1}{K_i} \sum_{k=1}^{K_i} \sum_{c=1}^{C} \log \left( \frac{\exp(o_{ikc})}{\sum_{c'=1}^{C} \exp(o_{ikc'})} \right) \right), \tag{3}
\]

where \( o_{ikc} \) are the unnormalized logits produced by the last linear layer of the model, \( o_{ik} = M^2 \text{.logits}(d_{ik}) \).

FIGURE 6. Loss function. To compute \( L_1 \), we permute the dimension of the output of the linear layer, i.e., [sequence length, batch size, output embedding size], into [batch size, sequence length, output embedding size]. Then the output of the new dimension is passed to the Softmax layer to calculate the possibilities of the dimension of the output embedding size. Next, the cost calculation function helps calculate the labeled and predicted annual costs based on the output target and the Softmax output. The process is much simpler for calculating \( L_2 \). The output of the linear layer is permuted into the dimension [batch size, output embedding size, sequence length]. Then, the cross-entropy loss function from PyTorch takes the permuted output and the output target to calculate the cross-entropy loss. Finally, since there is a large difference in magnitude between \( L_1 \) and \( L_2 \), we choose the common logarithm (log10) to scale down \( L_1 \) and then add it to \( L_2 \), i.e., \( \mathcal{L} = \log_{10}(L_1) + L_2 \).
evaluation: 1) for the regression modeling, root mean-squared error (RMSE) and $R^2$ to measure cost-prediction performance and 2) top-k accuracy indicators ($k=3,5,10$) for evaluation of the classification performance.

1) METRICS FOR CLASSIFICATION ($\mathcal{M}_1$ and $\mathcal{M}_2$).

Recommendation systems’ ability to find the best options is often judged in terms of top-k accuracy [72], healthcare analytics work has often followed design philosophy with such a “best bet” concept because it reflects the mindset of physicians performing diagnosis as they assess which diseases could be considered and whether, upon examination, the culprit might indeed be one of the candidates found [10]. In calculation of top-k accuracy, the prediction is deemed correct if the true label is among the model’s $k$ prediction with the highest predicted likelihood [72].

The formula is presented in Eq. 4. To avoid confusion with our earlier notation, where $k$ is used as an index for the patient’s visits, we will use $h$ instead of $k$ to denote the rank of the prediction, and we will use $H$ as the number of prediction allowed for every true label. Let $\hat{v}_h^i$ denote the predicted cost type vector for patient $i$ at visit number $j$, where the cost type is selected based on the $h$-th highest predicted likelihood. Let $v_j$ be the corresponding true label of visit cost type for patient $i$ at visit number $j$. Indicator function $\mathbb{I}(\hat{v}_h^i = v_j)$ has a value of 1 if $\hat{v}_h^i = v_j$; otherwise, the value is 0. In our study, we took the values 3, 5, and 10 as $h$ for prediction of any single cost. 

$$
\text{Top-k accuracy}(v, \hat{v}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{H} \sum_{s=1}^{N} \mathbb{I}(\hat{v}_h^i = v_j) \tag{4}
$$

2) METRICS FOR REGRESSION ($\mathcal{M}_2$).

For measuring how close the predicted cost $\hat{C}_{\text{out}}^2$ is to the actual cost $C_{\text{out}}^2$, we chose two commonly used metrics suited to evaluating regression models [73]. In this evaluation, $N$ was the number of patients.

Firstly, RMSE (the square root of the mean-squared error) gave us the expected value of the squared error or loss and it was computed as the square root of $L_1$. It enjoys widespread use because it is expressed in the same units as the response variable [73].

$R^2$, in turn, expresses the proportion of the variance explained by the independent variables in the model [74]. Via the proportion of the variance explained, it shows how well the model can predict the unseen data. The best possible $R^2$ value is 1.0, while a value of 0.0 indicates that the model does not aid in explanation (i.e., it predicts the average value $\bar{c}_i$). [73] Eq. 5 presents the calculation of $R^2$, where $\hat{c}_{ikc} = \hat{p}_{ikc}c_i$.

$$
R^2 = 1 - \frac{\sum_{i=1}^{N} \left( \sum_{k=1}^{K_i} \sum_{c=1}^{C} (\hat{c}_{ikc} - c_{ikc}) \right)^2}{\sum_{i=1}^{N} \left( \sum_{k=1}^{K_i} \sum_{c=1}^{C} (c_{ikc} - \bar{c}_i)^2 \right)} \tag{5}
$$

**IV. RESULTS**

Our design split the dataset into training, validation, and testing sets. The purpose of validation is to prevent overfitting during training and to guarantee that the model gets evaluated in relation to an entirely unseen set of data [75]. The batch size and the learning rate were set as 64 and 0.0001 respectively. More information on experiments of hyperparameters can be found in the supplementary material. The performance of model $\mathcal{M}_1$ is presented in Table 1. Since all the top-k values for the test set exceed 91%, we apply the train $\mathcal{M}_1$ model for data augmentation based on the model design architecture shown in Fig. 5.

For the baseline models required as a reference for judging our two-model design, we have chosen four seq2seq models based on LSTM because LSTM is commonly used for seq2seq as one of the most popular encoder-decoder models. These four baseline models are: 1) a general LSTM-based encoder–decoder model, 2) an attention-oriented one, 3) a bidirectional one, and 4) an attention-based bidirectional one. Taking a combination approach, the last of these encoder–decoder models employs a bidirectional LSTM encoder to encode the historical time series while another LSTM decoder produces the future time series, with an attention mechanism implemented for coordinating the input and output time series and dynamically selecting the most pertinent contextual data for prediction purposes [76]. The other three baseline models are variants of the attention-based bidirectional LSTM (BiLSTM) encoder–decoder model. All four baseline models can learn from historical records and, thereby, create representative material to inform prediction.

We compared these models with our Transformer-based prediction, assessing the performance of each with the original raw data, randomly augmented data, and data augmented via Transformer model $\mathcal{M}_1$. This section details the results, presented concisely in Table 2. Since our goal entailed giving the highest priority to predicting regression cost ($L_1$), special emphasis is placed on regression metrics ($R^2$).

As the table attests, models without an attention mechanism display excellent top-k accuracy but have negative $R^2$ values, suggesting that these models fail to predict the year’s total visit costs. In contrast, LSTM with attention and our $\mathcal{M}_2$ achieves an $R^2$ of more than 0.88. Transformer $\mathcal{M}_2$ stands out from all the other models for every indicator. As for performance with randomly augmented data for training and evaluation, the LSTM models do worse, BiLSTM models function slightly better, and – surprisingly – $\mathcal{M}_2$ has a higher $R^2$ value: 0.944. Finally, when compared to the original data, data augmented via the same Transformer model ($\mathcal{M}_2$) afford
TABLE 2. Models performance with original data and augmented data with transformer.

<table>
<thead>
<tr>
<th>Model</th>
<th>Original Data</th>
<th>Randomly Augmented Data</th>
<th>Augmented Data with Transformer $M^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train Set</td>
<td>Top3</td>
<td>Top5</td>
</tr>
<tr>
<td>LSTM</td>
<td>Train</td>
<td>57.87</td>
<td>73.32</td>
</tr>
<tr>
<td></td>
<td>Val</td>
<td>57.92</td>
<td>72.42</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>60.46</td>
<td>75.48</td>
</tr>
<tr>
<td>LSTM</td>
<td>Train</td>
<td>60.72</td>
<td>74.02</td>
</tr>
<tr>
<td></td>
<td>Val</td>
<td>60.96</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>60.46</td>
<td>75.48</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>Train</td>
<td>62.95</td>
<td>73.49</td>
</tr>
<tr>
<td></td>
<td>Val</td>
<td>59.07</td>
<td>72.53</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>60.54</td>
<td>75.74</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>Train</td>
<td>61.26</td>
<td>74.10</td>
</tr>
<tr>
<td></td>
<td>Val</td>
<td>59.74</td>
<td>72.15</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>63.08</td>
<td>76.29</td>
</tr>
<tr>
<td>Transformer $M^2$</td>
<td>Train</td>
<td>80.62</td>
<td>86.30</td>
</tr>
<tr>
<td></td>
<td>Val</td>
<td>81.36</td>
<td>86.27</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>83.64</td>
<td>87.76</td>
</tr>
</tbody>
</table>

FIGURE 7. Comparison of mean and median values of true costs and predicted costs per patient per year.

better prediction of cost for all regression-model conditions except LSTM with attention (though the latter still is able to reach an $R^2$ of 0.858). The $R^2$ of Transformer $M^2$ rises to 0.975 in this condition, and that of BiLSTM increases from 0.616 to 0.819.

Our designed model performs the best based on the results shown in Table 2. Therefore, we only apply this model to predict visit costs and compare them with true visit costs. The sum of the true total cost for all patients in the test set is $\text{e}146,815$. In comparison, the prediction by the trained model with Transformer augmented data is $\text{e}144,244$. Fig. 7 illustrates the comparison of mean and median values.

V. DISCUSSION AND CONCLUSION

In this paper, we have developed a method for predicting (1) the cost types for individual OSA patients’ future visits and (2) the total costs resulting from these visits using EHR data. This method utilizes a versatile Transformer-based architecture which helps make the most of the limited EHR data. While Transformers are applied in many fields – computer vision, NLP, and also modern healthcare [70], our approach presents advances relative to existing work by using two Transformers ($M^1$, $M^2$). The two-model system is easy to implement in this context since the encoder portion is identical and enables retaining the sequential patterns of the OSA patients who live in the same region and receive treatment at the same hospital. Model $M^1$ is designed to augment the input by using material that is not suitable as-is for our prediction task. The second component, Transformer $M^2$, outputs not only the prediction of the next year’s total costs but also itemization by visit cost type for each visit instance, with the aid of two subsidiary loss functions. Our system outperforms all the baseline models covered in Table 2 both in top-k accuracy and by regression metrics. We empirically demonstrated prediction improvements arising from our model-informed data augmentation, which enriched the input relative to the original longer-follow-up a third of the high-quality data. We found also that summation of embeddings and utilization of special tokens can serve as an effective way to deal with multivariate sequences of healthcare data. The design sheds new light on approaches to tackle the problem of small bodies of relevant healthcare data and offers a different perspective on cost prediction for better decision-making.

Data augmentation is one of the best ways to make additional data available for research, and many studies already attest to the effectiveness of a corresponding strategy, via empirical evidence and design insight. However, the difficulty of assessing any given technique in the absence of a large-scale study creates obstacles: quantitative measurement of how well it fits the data is rare, and there is little research into why it works. Further issues often arise from a lack of variety in the augmented body of data. Frequently, the augmentation is accomplished under supervision, which may result in overfitting or bias in the prediction task [22]. Although a seq2seq model’s ability to retain long-term relationships enables it to handle tasks involving lengthy sequences quite well, it is unable to store contextual data. In contrast, the Transformer-based approach preserves data related to context. While it performs all tasks better than...
The authors would like to thank Anna Shefl for proofreading. Our future work will consider such issues. We recommend exploring other data-augmentation techniques and see how they affect prediction accuracy with datasets of various sizes. Further studies could also extend beyond the healthcare decision-making framework, delving into prediction outcomes at other levels and probing/cultivating direct linkages throughout the systems involved. For example, we should study integration of the cost-prediction model into healthcare’s treatment-optimization process. For data specific to OSA, multitasking with EHRs could be handled more effectively. In one example, one might implement mortality/survival evaluation and cost prediction jointly in such a way that not only the financial element but also OSA patients’ quality of life would be considered. Since abundant research has shown the power of Bayesian optimization for improved modelling, its use should be considered for informing prediction projects, although the method may prove time-consuming as the models grow larger and more complicated [78]. Our study exclusively emphasizes complete data, while ignoring variables containing missing data, such as medical markers including BMI and blood pressure. However, in our future research, we propose to address the issue of missing data and utilize the data more effectively.

ACKNOWLEDGMENT

The authors would like to thank Anna Shefl for proofreading the article.

REFERENCES