ELMV: an Ensemble-Learning Approach for Analyzing Electrical Health Records with Significant Missing Values

Lucas Jing Liu*
Department of Computer Science
University of Kentucky
Lexington, Kentucky, USA
jliu@uky.edu

Jianzhong Di
Department of Metabolic & Bariatric Surgery
Shanghai Jiaotong University Affiliated 6th People’s Hospital
Shanghai, China
dijianzhong@sjtu.edu.cn

Hongwei Zhang*
Department of Metabolic & Bariatric Surgery
Shanghai Jiaotong University Affiliated 6th People’s Hospital
Shanghai, China
zhw00sub@163.com

Jin Chen
Department of Internal Medicine
Department of Computer Science
University of Kentucky
Lexington, Kentucky, USA
chen.jin@uky.edu

ABSTRACT
Many real-world Electronic Health Record (EHR) data contain a large proportion of missing values. Leaving a substantial portion of missing information unaddressed usually causes significant bias, leading to invalid conclusions to be drawn. On the other hand, training a machine learning model with a much smaller nearly-complete subset can drastically impact the reliability and accuracy of model inference. Data imputation algorithms that attempt to replace missing data with meaningful values, inevitably increase the variability of effect estimates with increased missingness, making it unreliable for hypothesis validation. We propose a novel Ensemble-Learning for Missing Value (ELMV) framework, an effective approach to construct multiple subsets with much lower missing rates of the original EHR data as well as to mobilize dedicated support data for ensemble learning, for the purpose of reducing the bias caused by substantial missing values. ELMV has been evaluated on real-world healthcare data for critical feature identification and simulation data with different missing rates for outcome prediction. In both experiments, ELMV outperforms conventional missing value imputation methods and traditional ensemble learning models. The source code of ELMV is available at https://github.com/lucasliu0928/ELMV.

CCS CONCEPTS
• Computing methodologies → Ensemble methods; Feature selection.

KEYWORDS
Machine Learning, Ensemble learning, Missing Values, Electronic Health Record (EHR), Multiple classifier system (MCS)

ACM Reference Format:
Lucas Jing Liu, Hongwei Zhang, Jianzhong Di, and Jin Chen. 2020. ELMV: an Ensemble-Learning Approach for Analyzing Electrical Health Records with Significant Missing Values. In Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB ’20), September 21–24, 2020, Virtual Event, USA. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3388440.3412431

1 INTRODUCTION
Real-world Electronic Health Record (EHR) data have played an essential role in improving patient care and clinician experience and providing rich information for biomedical researches [19, 34, 37]. However, many EHR data contain a significant proportion of missing values, which could be as high as 50%, leading to a substantially reduced sample size even in initially large cohorts if we restrict the analysis to individuals with complete data [21, 28]. On the other hand, leaving a big portion of missing information unaddressed usually cause bias, loss of efficiency, and finally leads to the inappropriate conclusion to be drawn [24].

Data imputation algorithms (e.g., the scikit-learn estimators [31]) attempt to replace missing data with meaningful values, including random values, the mean or median, the spatial-temporal regressed values, most frequent values in the same columns, or representative values identified using k-nearest neighbor [17]. Advanced data imputation algorithms, such as Multivariate Imputation by Chained Equation (MICE) [11], have been developed to fill missing values multiple times. Leveraging the power of GPU and big data, deep neural network models, such as Datawig [9], can estimate more accurate results than traditional data imputation methods [6]. However, as stated in the statistical literature [13, 26, 27], there is inevitably increasing variability of effect estimates with increased missingness; and results may not be reliable enough for hypothesis validation if more than 40% data are missing in important variables [15, 22], indicating that data imputation is not a go-to solution.
Figure 1: Overall framework of ELMV. It includes three stages: predictive model generation, ensemble prediction, and critical feature identification.

methods and traditional ensemble learning models. Overall, ELMV has four algorithmic advantages:

- To our knowledge, ELMV is the first ensemble learning approach that is capable of analyzing large EHR data with significant missing values accurately without using data imputation.
- By constructing multiple maximal subsets of the original EHR data, opportunities are that even if critical features are removed due to high missingness, the generated predictive models using auxiliary features may still maintain a relatively high performance.
- ELMV introduces dedicated support data for ensemble learning for the purpose of reducing the bias caused by substantial missing values.
- ELMV can identify critical features from large EHR data with significant missing values.

2 BACKGROUND

In recent years, techniques have been developed for handling missing values in big data. The simplest and most common strategy is to conduct complete-case analysis (CCA), which refers to removing records with any missing values and focusing only on patients who have the complete records of all parameters [36]. In practice, eliminating patients with any missing values will inevitably introduce biases, given that there is often a huge difference between the true distribution of all patients and that of the patients with complete records [32]. In addition, regarding inference model training, the CCA strategy will significantly reduce the training size, resulting in models under-trained.

Another common strategy for handling missing values is data imputation. Imputation techniques can be categorized into two groups: single imputation and multiple imputation [14]. The single
imputation refers to replacing a missing value with an estimated value [10]. An example of the simple imputation strategy is the mean imputation [20], where a missing value is replaced with the arithmetic mean. The problem of the simple imputation strategy is that it may significantly underestimate the variance of the data and ignores the complex relationships among explanatory variables [36]. This problem can be addressed using more sophisticated single imputation methods such as regression imputation and the expectation-maximization (EM) algorithm, in which a missing value is assigned by studying the statistical relationships between the target variable and the rest variables in the same dataset [20]. In contrast, multiple imputation techniques estimate a missing value with multiple imputed data. One such technique is Multivariate Imputation by Chained Equations (MICE), where the statistical uncertainty of different imputed data is taken into account [4]. However, none of the existing imputation method outperforms the others on every dataset, indicating that there are no universal model [20] for missing value imputation.

While most machine learning models can only be applied to complete data or will automatically conduct a complete-cases analysis [20], XGBoost [12], the recent implementation of the gradient boosting model can automatically handle missing values with its built-in mechanisms. Specifically, XGBoost handles the missing data problem by adding a default direction for missing values in each tree splitting. The optimal direction for a missing value in each particular explanatory variable at each tree node is learned during model training process with the aim to minimize the regularized loss [12]. XGBoost model chooses the default direction if there is no missing value in any particular explanatory variable in training data, but there are missing values in external validation set. A potential problem in handling missing value in XGBoost is that XGBoost will always choose the default direction for model prediction on the validation set. Thus, the prediction could be a random guess if the missingness patterns in training and validation are entirely different. This could be the case when a large amount of missing value existing especially in the validation data.

Overall, the common problem of existing machine learning approaches is that they do not have the adaptability for handling large missing values. In addition, the discrepancy between training and validation has not been well addressed regarding model inference. In this project, we propose ELMV, an ensemble learning framework that 1) is capable of handling substantial missing values, 2) has the adaptability to different dataset and different predictive models, and 3) considers the discrepancy between training and validation in terms of both missingness and critical features recognition.

3 METHOD
ELMV aims to identify unbiased, precise predictive patterns from EHR data, which, if learned directly, may result in biases caused by substantial missing values [5]. Specifically, given an EHR dataset \( I \) with significant missing values, ELMV first generates a set of subsets of \( I \) with low missing rates, denoted as \( S \), using dynamic programming, and upon these, builds predictive models \( M \) so as to mitigate the overall bias in each dataset in \( S \) for a single predictive model. Second, for every record in the external validation data, ELMV selects the most suitable models from \( M \) for the final prediction using ensemble learning.

Since ELMV is a general machine learning framework for learning from EHR data with significant missing rates, any conventional machine learning model, such as XGBoost [12] and SVM [29], can be used in our framework. For the demonstration purpose, we used XGBoost in the paper. The framework of ELMV involves three stages, namely model generation, critical feature identification, and outcome prediction. The architecture of ELMV is illustrated in Figure 1.

3.1 ELMV Stage 1. Predictive Model Generation
In the predictive model generation stage, we first compute the data missingness of a given EHR dataset, assessing whether it is appropriate to use ELMV. And then, we generate multiple subsets of the original data with lower missing rates. Finally, a predictive model is trained on each subset.

3.1.1 Assessing Data Missingness. Given an EHR dataset \( I \) where rows are patients \( P \), columns are features \( F \) in the EHR, \( N_P \) is the total number of patients, \( N_F \) is the total number of features, a missingness indicator for each patient \( p \), denoted as \( Missing^P \), is defined as a binary vector with the length of \( N_F \) where a one in a specific entry represents that the corresponding feature is missing for patient \( p \).

Specifically, for EHR data with temporal features \( T \times N_F, T.N_P \), where \( T \) is the number of time points of a temporal feature, we define the missingness indicator as a two-dimensional matrix \( Missing^P \) for each temporal feature of patient \( p \), denoted as \( tf^P \in TF \), if a temporal trend-based feature is missing because of the missing data at time point \( t \), let \( Missing^P(t, tf) \) = 1.

A 2-dimensional binary matrix, denoted as \( Missing^I \in \mathbb{R}^{N_P \times N_F} \) can then be generated to store the missingness information of all patients. In 2D case, \( Missing^I[i, j] = 1 \) representing the patient \( i \) has a missing value in the \( j \)th feature. In the case of 3D temporary dataset where the third dimension represents time of records, \( Missing^I[i, j] = 1 \) representing the patient \( i \) at least have one data point missing in the time trajectory of the \( j \)th feature.

Based on the definition of data missingness, we compute the missing rate of the entire dataset \( I \), assessing whether ELMV or data imputation techniques should be used. Typically, if the data missingness is low, it is appropriate to impute missing data. However, if the missing rate is above 40%, data imputation may inevitably increase the variability of effect estimates. Instead of imputing missing values directly, ELMV relies on ensemble learning which aggregates predictive models built on multiple subsets with significantly lower missing rates.

Note that although ELMV is still applicable when the missing rate is low (e.g., under 10%), its performance is similar to other state-of-art models.

3.1.2 Generating Subsets with Low Data Missingness. Given an EHR dataset \( I \) and a user-defined data missing rate upper bound \( T_{\text{max--missing}} (e.g., 20\%) \), which is much lower than the missing rate of \( I \), we generate a set of maximal subsets of \( I \) with its missing rate lower than or equal to \( T_{\text{max--missing}} \), saved in \( S \).
A subset $s$ of $I$ ($s \in S$) is a 2-dimensional matrix where rows are patients and columns are features in EHR. We say $s$ is maximal if and only if its missing rate can only increase if its rows or columns are replaced by any new rows or columns in $I$.

Since the total number of possible subsets is $\binom{N_p}{x} \times \binom{N_f}{y}$, where $x$ and $y$ are the numbers of rows and columns of $s$, it is impractical to enumerate all the possibilities and then select the maximal ones. Thus, to identify all the qualified maximal subsets of $I$ with missing rates lower than or equal to $T_{\text{max-missing}}$, we develop a two-step approach.

The approach for generating qualified maximal subsets consists of two steps: 1) to generate all the maximal subsets using dynamic programming, and 2) to filter the maximal subsets with nearly duplicated information. The pseudocode for maximal subsets generation is illustrated in Algorithm 1. In the following section, we explain the steps for generating the qualified maximal subsets.

In the first step, we track the missingness of all the subsets-to-generate using a 2-dimensional matrix $\text{MissingC} \in \mathbb{R}^{N_p \times N_f}$. The value in each entry of $\text{MissingC}(x, y)$ represents the minimum number of missing values of any subset of $I$ with $x$ patients and $y$ features. For instance, $\text{MissingC}(100, 200) = 1300$ means that the minimum number of missing values is 1300 for any sub-matrix of $I$ with 100 patients and 200 features. $\text{MissingC}$ can be used to select maximal subsets (see details in Algorithm 1).

We start to fill $\text{MissingC}$ and to generate the corresponding maximal subsets from the bottom right corner, $\text{MissingC}(N_p, N_f)$. Naturally, it represents the number of missing values when all features and all patients are selected. Hence, the corresponding maximal subset is itself. And then, we repeatedly remove one feature or one patient that has the maximum number of missing values at a time until the subset reaches the smallest required number of features and the smallest number of patients. By removing a feature or a patient with the maximum missing values at each time step, the generated subset is ensured to have the missing rate corresponding to the required number of features and patients. The whole process is achieved using dynamic programming [7].

The second step of subset generation is to identify and remove subsets conveying nearly identical information. For all the subset with a similar missing rate, we keep the subsets with the maximum number of features if the number of patients is identical, or keep the subsets with the maximum number of patients if the number of features is identical.

The final outcome of this step is a set of maximal subsets of the original EHR dataset with missing ratio smaller than or equal to $T_{\text{max-missing}}$.

3.1.3 Training Predictive Models. Using every qualified maximal subset of the original data $I$, we train a traditional classification model and save all the trained models in model set $M$. Since ELMV is a general framework for learning predictive patterns from data with significant missingness, any classification model, such as support vector machine and gradient boosting, can be used in this step. We expect that the classification model deployed here is capable of handling a few missing values. Otherwise, we recommend to employ a data imputation method before calling a classification model.

### Algorithm 1: Algorithm For Generating Maximal Subsets

**Input**: 2D DataMatrix[$N_p \times N_f$] or 3D Temporal DataMatrix [$N_p \times N_f \times N_t$]

**Intermediate**: MissingI, MissingI_List, MissingC,

**Output**: Max_S # Maximal Subsets

**Function** ConstructMissingI(DataMatrix):

```plaintext
for i = 1 to N_p do
    for j = 1 to N_f do
        if \(\sum_{t=1}^{N_t} \text{MissingI}(t, i, j) \geq 1\) then
            MissingI_i,j = 1
        else
            MissingI_i,j = 0
    end
end
return MissingI
```

**Function** Order(MissingI):

Order input by the missing percentage of patients and features ascendingly from left to right and from top to bottom

```plaintext
return ordered MissingI
```

**Function** CountMissings(MissingI):

Count the total number of ones in input

```plaintext
return Total_Number_Of_Missing_Values
```

**Initialization**

```plaintext
MissingI_List_{N_p, N_f} = ConstructMissingI(DataMatrix)
MissingC_{N_p, N_f} = CountMissings(MissingI_List_{N_p, N_f})
MissingI_List_{N_p, N_f} = Order(MissingI_List_{N_p, N_f})
```

for $i = 1$ to $N_p$

```plaintext
for $j = 1$ to $N_f$

if $i = N_p$ and $j = N_f$ then
    if MissingC_{i+1, j} < MissingC_{i, j+1} or MissingC_{i+1, j+1} is empty then
        MissingI_List_{i, j+1} = Order(MissingI_List_{i, j+1})
        last_step = MissingI_List_{i, j+1}
        /* then remove the last feature */
        MissingI_List_{i, j} = last_step[-last_column]
    else if MissingC_{i+1, j} \geq MissingC_{i, j+1} or MissingC_{i, j+1} is empty then
        MissingI_List_{i+1, j} = Order(MissingI_List_{i+1, j})
        last_step = MissingI_List_{i+1, j}
        /* then remove the last patient */
        MissingI_List_{i, j} = last_step[-last_row]
end
```

```plaintext
return Max_S_i,j = Patient and Features in MissingC_{i, j}
```

```plaintext
```

BCB '20, September 21–24, 2020, Virtual Event, USA Liu and Zhang, et al.
For the demonstration purpose, the XGBoost implementation [12] “xgboost” in R library is used in this step. Specifically, we choose a tree-based model called “gbtree” booster with a softmax objective “multi:softprob” for relatively easier classification tasks. Also, we choose a linear model called “glinear” with a logistic objective “binary:logistic” for relative harder classification tasks, such that a multi-class task can be converted into binary classification using the one vs. rest approach [2]. Finally, each trained predictive model is evaluated using leave-one-out cross validation (LOOCV) [33] approach. Model validation performance is saved for later use.

3.2 ELMV Stage 2. Ensemble Prediction

In the ensemble prediction stage, ELMV aggregates multiple selected predictive models trained in stage one to make predictions for records in an external validation set.

Here, each predictive model is trained with a qualified maximal subset with its missing rate smaller than or equal to $T_{\text{max-missing}}$. If $T_{\text{max-missing}}$ is significantly smaller than the missing rate of the original data $I$, the qualified maximal subsets could be much smaller subsets of the original data. Therefore, a predictive model can successfully capture the local but not the global properties of the original data. Directly using these predictive models individually may not result in optimal results. Meanwhile, for the records in the external validation set, they may differ regarding which distributions the records are drawn from, indicating that we may not obtain the best performance by aggregating all the pre-trained models. Hence, in the ensemble prediction stage, we develop a novel strategy to select pre-trained predictive models according to data representation and ensemble them for external validation.

3.2.1 Constructing Support Set. To estimate the distribution of the external validation records, a support set is generated. Mathematically, the support set $SS_{N_{\text{ss}}}$ generated by randomly select $N_{\text{ss}}$ rows from the original dataset $I$. Similar to $I$, $SS_{N_{\text{ss}}}$ may have a significant missingness. For $SS_{N_{\text{ss}}}$, a binary missingness matrix $\text{MissingSS}$ is obtained using the same method described in Section 3.1.1.

3.2.2 Measuring Patients Similarity. For each external validation record, we measure the similarities between it and all the records in the support set $SS_{N_{\text{ss}}}$ pair-wisely. Top $k_1$ similar records in $SS_{N_{\text{ss}}}$ are selected. ELMV assigns a set of dedicated pre-trained models to each external validation record by selecting all the pre-trained models that can successfully predict at least $k_2$ top records ($k_2 \leq k_1$). Both $k_1$ and $k_2$ are a user-defined parameter.

Formally, the similarity between a external validation record and all records in the support set $SS_{N_{\text{ss}}}$ is defined in Equation 1:

$$\text{Sim} = W_F * \text{Softmax}(-\text{Dist}_F) + W_M * \text{Softmax}(-\text{Dist}_M)$$

(1)

where $\text{Sim} \in \mathbb{R}^{1 \times N_{N_{\text{ss}}}}$ represents the similarity between each individual validation record and all the records in the support set, $\text{Dist}_F \in \mathbb{R}^{1 \times N_{N_{\text{ss}}}}$ represents the Euclidean distance of the corresponding feature vectors, $\text{Dist}_M \in \mathbb{R}^{1 \times N_{N_{\text{ss}}}}$ represents the Hamming distance of the missingness indicator vectors $\text{Missing}$. $N_{N_{\text{ss}}}$ represents the number of records in support set $SS_{N_{\text{ss}}}$, and the overall similarity score is a weighted sum of the two distances normalized using softmax. Here, weights $W_F$ and $W_M$ are user-adjustable parameters. Larger $W_F$ indicates ELMV pays more attention to feature vectors similarity, and likewise larger $W_M$ indicates the missingness vectors similarity is more important.

3.2.3 Ensemble Prediction. Finally, we select multiple pre-trained predictive models and aggregate them by adopting the ensemble prediction approach. The model selection procedure can be described as a multi-objective optimization problem that considers the following objectives: the model prediction performance on support records similar to the target external validation records, the model performance on all records in the support set, the model cross-validation performance such as accuracy, precision, recall, and $F_1$, as well as the characteristics of the subset that is used to train the model including the number of features, the number of patients, and the missing rate.

Given a list of model selection criterion \{C1, C2, ..., Cn\} and a list of candidate models \{M1, M2, ..., Mn\} $\in M$, let $T_{\text{Best}}(C_M)$ be a binary vector indicating whether model $M$ performs the best under criteria $C$. Mathematically,

$$T_{\text{Best}}(C_M) = \begin{cases} 1, & \text{if } C_M \text{ performs the best under criteria } C_i \\ 0, & \text{otherwise} \end{cases}$$

(2)

A pre-trained model is selected if and only if it performs the best on at least one criterion formulated in Equation 3 or the overall performance in all criterion is the highest (see Equation 4). The number/type of the objectives $K_{\text{obj}}$ are user adjustable.

$$\exists C : T_{\text{Best}}(C_M) = 1$$

(3)

$$\arg \max_M \sum_{i=1}^{n} T_{\text{Best}}(C_M)$$

(4)

In the last step, the final prediction for each record in the external validation set can be obtained by integrating all the selected models. For the demonstration purpose, a majority voting of all the selected models is used here, which can be replaced with other ensemble learning approaches with a simple modification.

3.3 ELMV Stage 3. Critical Feature Identification

Each predictive model trained with a qualified maximal subset produces its own critical features in its local context. In order to identify the critical features of the entire data, we repeatedly apply the leave-one-out cross validation (LOOCV) [33] on each qualified maximal subset. Finally, we aggregate the most critical features of each predictive model using a weighted voting mechanism. The critical feature identification process is shown in Figure 2. Through this process, domain experts can examine the validity and reliability of ELMV by checking whether the critical features found is reasonable under both the local and global context.

In the weighted voting process, the weight of a critical feature is determined by three factors, i.e. the local LOOCV performance of the pre-trained predictive model, missing rate of the qualified maximal subset used to train the predictive model, and local feature importance.

Generally speaking, the higher the local LOOCV performance, the more weight is put on the features found by that predictive
model. Specifically, for each predictive model, the top-$k_3$ local critical features are determined by model feature importance. And then, all the top-$k_3$ critical features of every predictive model with a similar missing rate are sorted and ranked. The feature ranking is based on the ratio between the number of times a given feature being selected as a critical feature by individual predictive models and the number of times it is available. Given the ranked feature list, we select top-$k_4$ critical features using weighted voting where weights are determined by the averaged local LOOCV model performance.

The description of all the user-defined hyperparameters is provided in Table 1. The source code of ELMV is available at: https://github.com/lucasliu0928/ELMV.

4 EXPERIMENTS

Multiple experiments were carried out on both simulation datasets and a real-world EHR dataset to validate the usefulness of ELMV. For performance comparison, XGBoost was used as the base predictive model. We compared ELMV with three models: 1) to impute missing values with the mean imputation and to train XGBoost with the imputed data, 2) to impute missing values with MICE [11] and to train XGBoost with the imputed data, and 3) to train XGBoost directly without using any data imputation method.

4.1 Data Preprocessing

4.1.1 Simulation Data. To simulate EHR data with a significantly high missing rate, we selected a complete data and constructed multiple simulation datasets with a wide range of missing rates. On the simulation data, we test whether adopting ELMV can achieve performance comparable to that of a predictive model trained on the complete dataset. Specifically, the complete dataset obtained was the IRIS dataset widely used in machine learning education from the UCI repository [16]. The IRIS data consists of four features, 150 records, and three outcome labels. The LOOCV accuracy of XGBoost on the IRIS data is as high as 0.97.

In total, 18 simulation datasets were generated using the IRIS data, each having 40 features and 150 records, while the missing rate varying from 5% to 70%. All the simulation datasets were constructed similarly, except for the missing rates. First, using each of the original features in the IRIS data, we generated nine additional features with their correlation coefficient to the original feature ranging from 0.1 to 0.9. The purpose was to test whether the model performance can be retained using auxiliary (highly correlated) features when original features are missing. In addition, the additional features were used to test whether the model can identify and retain high-quality features while discarding low-quality features. Finally, we randomly removed 5% to 70% entries from every simulation dataset.

4.1.2 Real World Healthcare Data. The real-world EHR data we used was collected in a follow-up study of 240 type 2 diabetes (T2DM) patients who went through the Laparoscopic Roux-en-Y Gastric Bypass (LRYGB) surgery [1] in the Shanghai Jiaotong University Affiliated 6th People’s Hospital. The data have been de-identified before use.

The LRYGB dataset consists of 79 variables including HbA1c and the other 78 biomedical variables collected at six different time points, i.e. before the LRYGB surgery, 3-month, 6-month, 12-month, 24-month, and 36-month after the surgery. In total, 240 T2DM patients participated the study. 24 out of the 78 biomedical variables, such as CysC, weight index, and direct bilirubin, were pre-selected based on domain knowledge for further studies. The purpose of the study is to predict the HbA1c trajectories that are defined as follows. The types of HbA1c trajectories that are defined as follows. The types of HbA1c trajectories were determined using clustering, followed by manual curation. Specifically, we adopted the reversed K-nearest neighbor (rKNN) [30] to remove outliers and adopted the agglomerative hierarchical clustering with Ward’s method [35] to separate all the patients into nine clusters. The Elbow method was then used to determine the optimal number of clusters, on which the decreasing rate of Within-Sum-of-Squares
We applied ELMV, as well as three baseline algorithms, i.e., mean imputation, MICE, and XGBoost without data imputation, on both the simulation data and the LRYGB data. For performance comparison, conventional classification metrics were used, including accuracy, precision, recall, and F-1. Additionally, domain experts manually reviewed the critical features selected by ELMV, assessing whether they are clinically reasonable for predicting the HbA1c trajectory.

### 4.2 Experimental results

We evaluated ELMV by testing whether it can identify critical features for predicting the trajectory of HbA1c. As part of the data preprocessing, we imputed a small portion of the missing values using domain knowledge and simple statistics such as linear interpolation. Also, we copied the 6th month values to the 3rd month, if the 6th month values were missing. We removed patients whose HbA1c values at both 3rd month and 6th month are missing. After this step, the LRYGB follow-up data consists of 202 patients, 24 features, and 6 labels. The missingness of all the features of the LRYGB data is shown in Figure 3. The missing ratio at every time point is 3%, 33%, 18%, 18%, 37%, and 56% respectively. Clearly, patient dropout is a main issue that resulted in high missing rates at later time points. Using this real-world data, we aim to test ELMV at the non-random missing data situation. Specifically, we evaluated ELMV by testing whether it can identify critical features for predicting the trajectory of HbA1c.

![Figure 3: In the LRYGB follow-up study, the distribution of the missing values of all the 24 variables at six time points. In general, more values are missing towards the end of the follow-up study. Red indicates higher missing ratio towards 100%, green is for lower missing ratio towards 0%, and black indicates 50% missing ratio.](image)

We compared model prediction accuracy of the four methods on the simulation datasets. When the missing rate was low (5% to 20%), all the models can achieve nearly perfect performance (accuracy ≥ 0.93). However, if the missing rate was in the range of 60% and 70%, the accuracy of all other methods was reduced significantly below 75% no matter how the missing values were handled while ELMV still can maintain its accuracy above 75%.

A moving average of accuracy and F-1 on the finer granularity without data imputation were systematically compared. Table 2 compares model prediction accuracy of the four methods on the simulation datasets. When the missing rate was low (5% to 20%), all the models can achieve nearly perfect performance (accuracy ≥ 0.93). However, if the missing rate was in the range of 60% and 70%, the accuracy of all other methods was reduced significantly below 75% no matter how the missing values were handled while ELMV still can maintain its accuracy above 75%.

A moving average of accuracy and F-1 on the finer granularity of missing rates shown in Figure 4 and Figure 5 reveal that ELMV is not affected by the high missing rates as bad as the other models. The performance trends suggest that ELMV achieved the best performance towards larger missing rates steadily, and XGBoost had the best performance if the missing rate was relatively low. MICE

Table 1: Definition of hyperparameters used in ELMV.

| Parameter       | Definition                                                                 | Suggested Value                                                                 |
|-----------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| $l_{\text{max-missing}}$ | Subset missing rate upper bound                                             | $\leq$ missing rate of original dataset I                                      |
| $k_1$           | Number of similar records in Support Set (SS)                              | $k_3$ and $k_4$ can be chosen according to the distribution of similarity scores (e.g., top 20% similarity scores) |
| $k_2$           | Number of similar records that were predicted correctly by each generated model |                                                                                 |
| $k_3$           | Number of local critical features identified by each qualified maximal subset | $k_3$ and $k_4$ can be chosen according to the total number of features (e.g., 10% of the total number of features) |
| $k_{\text{obj}}$ | Number/type of model selection criterion                                   | $k_{\text{obj}} \geq 1$                                                      |
| $W_F$           | Similarity weights for feature vector                                       | $0 \leq W_F \leq 1$                                                           |
| $W_M$           | Similarity weights for missingness vector                                   | $0 \leq W_M \leq 1$                                                           |

(WSS) was the slowest. Two clinicians examined the obtained clusters independently and defined six types of HbA1c trajectory. In summary, after semi-automatic labeling, the LRYGB data consists of 214 patients, 24 features, and six labels. The missingness of all the features of the LRYGB data is shown in Figure 3. The missing ratio at every time point is 3%, 33%, 18%, 18%, 37%, and 56% respectively. Clearly, patient dropout is a main issue that resulted in high missing rates at later time points. Using this real-world data, we aim to test ELMV at the non-random missing data situation. Specifically, we evaluated ELMV by testing whether it can identify critical features for predicting the trajectory of HbA1c.
Table 2: Averaged accuracy of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).

| Missing Rate | XGBoost | Mean Imputation | MICE Imputation | ELMV |
|--------------|---------|-----------------|-----------------|------|
| 5%           | 0.97    | 0.97            | 0.97            | 0.97 |
| 10%          | 1.00    | 0.97            | 0.93            | 1.00 |
| 20%          | 0.97    | 0.97            | 0.97            | 0.97 |
| 60%          | 0.67    | 0.63            | 0.73            | 0.80 |
| 65%          | 0.70    | 0.73            | 0.70            | 0.77 |
| 70%          | 0.70    | 0.67            | 0.63            | 0.77 |

Table 3: Averaged precision of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).

| Missing Rate | XGBoost | Mean Imputation | MICE Imputation | ELMV |
|--------------|---------|-----------------|-----------------|------|
| 5%           | 0.97    | 0.97            | 0.97            | 0.97 |
| 10%          | 1.00    | 0.97            | 0.95            | 1.00 |
| 20%          | 0.97    | 0.97            | 0.97            | 0.97 |
| 60%          | 0.65    | 0.64            | 0.75            | 0.83 |
| 65%          | 0.70    | 0.80            | 0.72            | 0.78 |
| 70%          | 0.71    | 0.69            | 0.63            | 0.76 |

Table 4: Averaged recall of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).

| Missing Rate | XGBoost | Mean Imputation | MICE Imputation | ELMV |
|--------------|---------|-----------------|-----------------|------|
| 5%           | 0.95    | 0.95            | 0.95            | 0.95 |
| 10%          | 1.00    | 0.95            | 0.90            | 1.00 |
| 20%          | 0.95    | 0.95            | 0.95            | 0.95 |
| 60%          | 0.64    | 0.59            | 0.73            | 0.83 |
| 65%          | 0.70    | 0.71            | 0.69            | 0.76 |
| 70%          | 0.70    | 0.63            | 0.61            | 0.74 |

had the overall lowest accuracy and its accuracy trend dropped steadily when the missing rate was increased. Surprisingly, the mean imputation had a relatively stable performance, probably because the missingness was generated completely randomly. Both mean imputation and MICE have lower accuracy than XGBoost, indicating that the two imputation methods tested failed to reinforce XGBoost to handle missing values.

Feature Selection on Real World EHR Data. We applied ELMV on the LRYGB data (78 features and 202 T2DM patients), aiming at identifying critical features for the HbA1c trajectory prediction.

4.2.2 Feature Selection on Real World EHR Data. We applied ELMV on the LRYGB data (78 features and 202 T2DM patients), aiming at identifying critical features for the HbA1c trajectory prediction.

Figure 4: The moving average of accuracy of ELMV, XGBoost, and two imputation methods on the simulation data with missing rate increasing from 60% to 70%.

All the qualified maximal subsets of the LRYGB data generated by ELMV are shown in Figure 6. Every point in the figure represents a qualified maximal subset of the LRYGB dataset. The X-axis indicates the number of patients, and the y-axis indicates the number of features of the qualified maximal subset.

In Figure 6, the points with the same color have a similar missing rate. We generated all the qualified maximal subsets of the LRYGB data so that any combinations of features of interest can be evaluated in the critical feature identification stage of ELMV. Note that since the goal of this experiment is to identify the critical features among 24 pre-selected features, we only used the qualified maximal subsets of the pre-selected features in the following analysis. Several early-stage biomarkers, such as serum Ca2+ and cholesterol level measured at 3-month found by ELMV were supported well by the literature [3, 8] to be critical for predicting HbA1c trajectory in the first three years after the LRYGB surgery.

In addition, the overall accuracy of ELMV on the LRYGB data is 0.93, significantly higher than that of XGBoost (0.63), Mean imputation (0.30), and MICE (0.28). The performance of ELMV on all the qualified maximal subsets with the missing rate ranging from
ELMV: an Ensemble-Learning Approach for Analyzing Electrical Health Records with Significant Missing Values

5 DISCUSSION AND CONCLUSION

In this paper, we developed a novel ensemble learning model called ELMV to predict patient outcome using EHR data with substantial missing values. In our experimental results, ELMV outperformed two widely used data imputation methods and an ensemble learning method on both patient outcome prediction and critical feature identification. We also demonstrated that ELMV is novel on model selection which takes into account data and missingness distributions in training and validation.

Since extra steps have been taken in ELMV, an interesting question is whether ELMV is significantly slower than the other models. We compared the computational time between ELMV with the baseline methods on both the simulation data and the healthcare data. As shown in Table 8, the mean imputation was the fastest on both datasets, while ELMV was the slowest (101 secs) on the simulation data when the number of features was relatively small. On the real healthcare data where the number of features was relatively large, MICE took more than 300 minutes while ELMV spent only around 18 minutes, and most of its time (70%) was spent on generating the maximal subsets using dynamic programming. This issue could be further addressed by clustering patients with similar missingness.

In ELMV, a novel approach is introduced to estimate the distribution of external validation data and to guide the ensemble learning using a support set. An interesting question is to what extent the support set can contribute to the ensemble learning since it is useful only when the external validation data are known. To this end, we compared ELMV with the k-nearest neighbor (kNN) model, which
simply assigns each external validation record to the label of most similar records in the support set. The results shown in Table 7 indicate that the kNN-based voting approach is unlikely to provide the correct prediction most of the time. This experiment further confirms that it is critical to integrate the support set with ensemble learning rather than simple voting.

To address the issue of different missingness patterns such as MCAR, MAR, and MNAR in the qualified maximal subsets, ELMV provides users the flexibility to employ any data imputation or predictive models. In addition, the final prediction is collected by aggregating results from multiple ELMV selected models. In the future, we will test and further improve ELMV on both MNAR and MAR data.

In future work, we plan to address the parameter tuning problem. In ELMV, there are many user-adjustable parameters, including the number of similar records in support set SS, weights for similarity scoring, and the number/type of objectives used in ensemble learning. It would be beneficial if part of these parameters can be learned and automatically tuned during the model training process.

ACKNOWLEDGMENTS
This project is supported by the NIH National Cancer Institute (grant no. 1R21CA231911) and the Kentucky Lung Cancer Research Program (grant no. KLCC-3048113817) to JL and JC and by the Clinical Retrospective Study of Shanghai Jiaotong University Affiliated 6th People’s Hospital (grant no. YNHG201912) to HZ and JD.

Chinese Clinical Trial Registry Number: ChiCTR-ONN-17012895.

REFERENCES
[1] Theodore K Alexandrides, George Skroubis, and Fotis Kalfarentzos. 2007. Resolution of diabetes mellitus and metabolic syndrome following Roux-en-Y gastric bypass and a variant of bilipancreatic diversion in patients with morbid obesity. Obesity surgery 17, 2 (2007), 176–184.
[2] Mohamed Aly. 2005. Survey on multiclass classification methods. Neural Netw 19 (2005), 1–9.
[3] Carina Ämmälä, Frances M Ashcroft, and Patrik Rorsman. 1993. Calcium-independent potentiation of insulin release by cyclic AMP in single β-cells. Nature 363, 6427 (1993), 356–358.
[4] Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. 2011. Multiple imputation for direct and indirect effects. Biostatistics 12, 1 (2011), 40–49.
[5] Brett K Beaulieu-Jones, Daniel R Lavage, John W Snyder, Jason H Moore, Sarah A Pendergrass, and Christopher R Baurer. 2018. Characterizing and managing missing structured data in electronic health records: data analysis. JMIR medical informatics 6, 1 (2018), e11.
[6] Brett K Beaulieu-Jones and Jason H Moore. 2017. Missing data imputation in the electronic health record using deeply learned autoencoders. In Pacific Symposium on Biocomputing 2017. World Scientific, 207–218.
[7] Richard Bellman. 1966. Dynamic Programming. Science 153, 3731 (1966), 34–37.
[8] Michael J Berridge, Peter Lipp, and Martin D Bootman. 2000. The versatility and universality of calcium signalling. Nature reviews Molecular cell biology 1, 1 (2000), 11–21.
[9] Felix Biessmann, Tammo Rukat, Phillipp Schmidt, Prathik Naidu, Sebastian Schelten, Andrey Taptuiev, Dustin Lange, and David Salinas. 2019. DataWig: Missing Value Imputation for Tables. Journal of Machine Learning Research 20, 175 (2019), 1–6.
[10] Matthijs Blankers, Maarten WJ Koeter, and Gerard M Schippers. 2010. Missing data approaches in eHealth research: simulation study and a tutorial for nonmathematically inclined researchers. Journal of medical Internet research 12, 5 (2010), e54.
[11] S van Buuren and Karin Groothuis-Oudshoorn. 2010. mice: Multivariate imputation by chained equations in R. Journal of statistical software 33, 7 (2010), 1–68.
[12] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Francisco, California, USA) (KDD ’16). ACM, New York, NY, USA, 785–794.