Tabular Data Prediction with Heterogeneous Features

Hedong Yan
Computer Department
Hong Kong Baptist University
Hong Kong, China
herdonyan@life.hkbu.edu.hk

I. Motivation

In the domain of machine learning, tabular data exhibits heterogeneous scales, encompassing nominal, ordinal, interval, and ratio types, which are discerned based on their grouping structure. Effectively addressing and leveraging this heterogeneity pose crucial challenges for predictive tasks involving tabular data. Numerous existing approaches have been devised to tackle these challenges by incorporating feature embeddings into downstream backbone models. Nevertheless, deep learning models often struggle to effectively handle this heterogeneity, contributing to their inability to surpass the performance of tree-based models. In this study, we introduce a pioneering plug-in embedding module tailored for downstream deep learning models. This module not only takes into account the value assignment encoding but also considers the probability associated with the given assignment. We employ the inverse probability as the weight for the one-hot encoder applied to nominal data and utilize the inverse occurrence probability as the weight for the thermometer encoder employed with ordinal data. As for continuous data, we adopt a learned Fourier feature methodology based on prior research. Experimental results demonstrate that our proposed approach enhances the performance of the underlying backbone model.

II. Notations

We summarize the symbols we used in this report in table 1. We let capital letter denote variables, and small letters denote values. Similarly, we denote metrics by bold upper letters, and denote vector by bold lower letters.

In this report, we employ a consistent notation convention to facilitate clarity and comprehension. We utilize capital letters to represent variables, while small letters are used to denote specific values. Additionally, bold upper letters are employed to denote various metrics, while bold lower letters are utilized to denote vectors. This systematic symbol representation aids in conveying the information and ensuring consistency throughout the document.

Notation	Description
n	number of instances in the data
x	features of an instance
y	targets of an instance
d_x	number of features in the data
d_y	dimension of target in the data
$\|f_i\|$	number of values of feature i
x_k^i	ith feature of kth instance

TABLE I: Notations

III. Related works

A. Heterogeneous Data Scale

In 1946, Stevens conducted a comprehensive classification of scales of measurement, which included nominal, ordinal, interval, and ratio scales [1]. The nominal scale represents a permutation group, wherein each object is associated with a unique value for the feature. The ordinal scale, on the other hand, is based on a partially ordered set where the comparison of greater or lesser values is defined. Both the interval and ratio scales belong to the general linear group, with the distinction that the ratio scale implies the presence of a "true" zero point. These heterogeneous scales have proven valuable for statisticians in uncovering meaningful insights from data. However, certain statisticians argued in 1993 that the creative approaches and models for data analysis should be constrained by the scales of the data itself [2]. Stevens's measurement theory remains independent of the specific questions posed to the data. Nonetheless, the crucial aspect lies in understanding the data and formulating the pertinent questions that one seeks to address.

B. Heterogeneous Feature Embedding

When dealing with tabular data, the features collected can exhibit a combination of categorical (nominal, ordinal) and numerical (interval, ratio) scales. In order to handle these heterogeneous features, various embedding techniques have been developed to map them into the real number field $\mathbb{R}$ or the interval $[0, 1]$. Existing embedding algorithms for features with heterogeneous scales can be broadly categorized into two classes: unsupervised and supervised (target-aware) methods.

1) Unsupervised Feature Embedding: Unsupervised feature embedding approaches do not rely on label information during the embedding process. Instead, they typically utilize certain priors or assumptions, such as orthogonality, to guide the embedding process. These methods aim to discover inherent patterns or structures within the data itself, without considering the specific labels or target variable. By leveraging these priors, unsupervised embedding techniques can effectively

capture the underlying characteristics of the features and represent them in a transformed space.

Nominal. One-hot encoder is a widely used embedding algorithm for categorical features. It represents each unique value of a feature as a binary number, indicating whether the corresponding value appeared or not. The resulting mapped vector for a feature has a size equal to the number of unique feature values.

Dummy encoding is similar to the one-hot encoder, but it reduces the vector size by one $(n-1)$ by designating one value as the reference category and representing it with a zero vector. The other values are then encoded using binary vectors. For example, a feature with three values would be represented as [0,0] for one value and [1,0] or [0,1] for the other two values.

Binary encoder maps the original feature values into a binary representation using a fixed number of binary bits. The number of bits required is determined by the formula $\lceil \log_2(n) \rceil$, where n is the number of unique feature values.

Frequency encoder, also known as Count encoder, maps each feature value to its frequency within the dataset. This encoding technique replaces the original value with its corresponding frequency, effectively representing the value by its occurrence count.

Simple encoder is similar to dummy encoding, but it replaces the binary values 0 and 1 with continuous values. Specifically, it substitutes 0 with $-\frac{1}{n}$ and 1 with $\frac{n-1}{n}$, where n represents the number of unique feature values. This encoding approach retains the ordinal information of the feature values.

These various encoding algorithms provide different strategies for representing categorical features in a numerical format, enabling machine learning models to effectively utilize such features in their training process.

Ordinal. Ordinal encoder is used to map a feature with ordinal scale into an integer value. It assigns a unique integer to each distinct value of the feature, considering the order or ranking among the values.

Rank-hot encoder, also known as thermometer encoder, is similar to one-hot encoding. However, instead of having only one value as hot (1) and the others as cold (0), it sets all values up to and including the current rank as hot. This encoding method captures the ordinal nature of the feature values.

Gray encoder, a type of binary encoder, ensures that adjacent values in the encoded representation differ by only a single bit. This helps in reducing errors or noise during the encoding process.

Several orthogonal encoders are available for linear models. Helmert contrast encoder compares each value of the feature to the subsequent value. For example, the feature sequence [1, 2, 3, 4] would be mapped to [[1, -0.33, -0.33, -0.33], [0, 1, -0.33, -0.33], [0, 0, 1, -1]] using Helmert contrast encoding.

Orthogonal polynomial encoder utilizes linear, quadratic, and cubic trends to fit the ordinal values within the same interval. For example, the sequence [1, 2, 3, 4] would be coded into [[-0.671, -0.224, 0.224, 0.671], [0.5, -0.5, -0.5, 0.5], [-0.224, 0.671, -0.671, 0.224]] using orthogonal polynomial encoding.

Backward difference encoder compares each value of the feature with the mean of the previous values. This encoding technique takes into account the relationship between the current value and the preceding values in the sequence.

These orthogonal encoding methods provide ways to transform categorical features with specific characteristics, such as ordinality or linear trends, into numerical representations suitable for linear models.

Continuous. The common practice is to use the continuous value of a feature directly as input for the backbone model. Another approach involves discretizing the feature values and applying categorical encoders. One such approach is the Piecewise Linear (PLE) encoder, introduced by Gorishniy et al. [3].

The PLE encoder is inspired by the cumulative distribution function of a value. It first discretizes the continuous values of the feature and then applies the rank-hot encoder. Within each bin, the PLE encoder replaces the value $f(x)$ (which is initially set to 1) with a linear transformation $f(x) = \frac{x-b_{t-1}}{b_t-b_{t-1}}$, where b_{t-1} and b_t represent the lower and upper boundaries of the bin, respectively.

By discretizing the feature values and applying the rank-hot encoder with this modified transformation, the PLE encoder captures the relative position or rank of the values within each bin, enabling the model to learn and leverage this ordinal information during training.

Others. Base-N encoder is an encoding method that maps feature values into their base-N representation. In this encoding scheme, the base-N refers to the numerical base used for the representation, where base-1 corresponds to the one-hot encoder, base-2 corresponds to the binary encoder, and base-N corresponds to the ordinal encoder, with N being the number of unique values for the specific feature. This encoding approach leverages the inherent ordinality of the feature values by assigning them integer values based on their order or rank.

On the other hand, hashing encoder is a technique that maps the original feature values into hash values. This encoding method involves applying a hash function to transform the values into a new representation. However, finding an appropriate hash function that yields good results for downstream models can be a non-trivial task. The effectiveness of the hashing encoder depends on the quality of the chosen hash function and its compatibility with the specific downstream models being used.

2) Supervised Embedding: Supervised embedding methods have the potential to enhance model performance by leveraging the supervised label information during the embedding process. These techniques incorporate the target variable or label into the embedding algorithm, allowing the model to learn informative representations that are directly aligned with the prediction task.

However, it is important to be cautious when applying supervised embedding, as it can inadvertently introduce target leakage. Target leakage occurs when the embedding process unintentionally incorporates information from the target variable that would not be available in a real-world prediction scenario. This leakage can lead to inflated performance during

training but can severely degrade the model's generalization ability and performance on unseen data.

To mitigate target leakage and ensure reliable performance, careful consideration should be given to the design and implementation of supervised embedding methods. It is crucial to ensure that the embedding process only utilizes information that would be available at the time of prediction, preventing any inadvertent incorporation of future or otherwise unavailable information. Thorough validation and evaluation on separate test sets can help detect and address any potential target leakage issues, allowing for more robust and reliable model performance.

Categorical feature The greedy target statistic estimates the expected value of the target variable, denoted as $E(y|x = x_k)$, based on the training dataset. To mitigate potential noise or variability in the estimates, it is common to apply smoothing using parameters a and p. The smoothed estimate can be calculated using the following formula:

$$x_k^i = \frac{\sum_{j=1}^{n} \mathbb{I}_{x_j^i = x_k^i} * y_j + ap}{\sum_{j=1}^{n} \mathbb{I}_{x_j^i = x_k^i} + a} \tag{1}$$

Holdout target statistic use the subset of instances excluding x_k,

$$x_k^i = \frac{\sum_{x_j \neq x_k} \mathbb{I}_{x_j^i = x_k^i} * y_j + ap}{\sum_{x_j \neq x_k} \mathbb{I}_{x_j^i = x_k^i} + a} \tag{2}$$

Ordered target statistic creates a virtual 'time', and use its all available history to calculate the target statistic.

Continuous feature

The one-blob encoding method, proposed by Müller et al. [4], assumes that the feature values follow a Gaussian or Laplace distribution. Each value is represented as a "blob" with a probability distribution centered at that value. The adjacent bins or intervals around the value correspond to the probabilities associated with that value. This encoding approach captures the uncertainty or variability in the feature values by modeling their distributions.

On the other hand, the periodic encoder, introduced by Gorishniy et al. [3], maps a feature into its Fourier forms. The encoding is represented as a vector $f(x) = [\sin(v), \cos(v)]$, where $v = [2\pi c_1 x, .., 2\pi c_k x]$. The frequency vector c_i can be learned from the data. This encoding technique is particularly useful for handling periodic or cyclical features, where the relationship between values wraps around in a circular fashion (e.g., time of day or day of the week). By representing the feature values in their Fourier forms, the periodic encoder captures the underlying cyclical patterns and relationships within the data.

IV. Dimension Reduction of Features

In order to deal with high-dimensional covariates, some dimensionality reduction approaches may be helpful.

Current dimensionality reduction research can be divided into three classes according to the reduction target. The first class is to determine the dimension based on information loss. For example, [5] minimize regression mean squared error (MSE) from cross-validation for a linear model with a kernel. [6] propose a Lagrange loss with a binary mask π for variational autoencoders (VAE) and prove its convergent dimension is a local minimum. However, the hidden distribution is usually in Gaussian space, which is often regarded as an "uninteresting" signal noise due to the *central limit theorem*. The second class evaluates the non-Gaussianity of latent space. For example, [7] assign a stability score to the principal component and regard the change point with the smallest p-value as an indicator. Non-Gaussian component analysis (NGCA) [8]–[10] assumes Gaussian noise is independent of the non-Gaussian subspace, and they discard the Gaussian component to determine the signal space. However, the algorithm is either exponential related to the dimension of the non-Gaussian subspace due to the error of accumulation [10] or the polynomial time is unacceptable. Therefore, it cannot be applied directly to general high-dimensional data. The third class is the end-to-end approach for a specified task. For example, [11] and [12] search for the most discriminative subspace for clustering. Recently, [13] propose a general approach based on probability density function (PDF) estimation without assumption about data structure, although the choice of hidden dimension is empirical. [14] use normalized maximum likelihood to determine the principal component cardinality. Table III illustrates the assumptions of representative dimensionality reduction methods.

PCA [15] [16] is a widely used linear dimensionality reduction technique. It uses orthogonal transformation to obtain the uncorrelated principal components. Autoencoder [17] [18], on the other hand, is a non-linear dimensionality reduction technique that typically uses neural networks and gradient-based optimization to learn the parameters for efficient computation. In autoencoders, the reconstruction error is an important part of the loss function.

V. Methodology

In order to address the challenge of handling heterogeneous features, a transformation of these features into a unified space suitable for the backbone model is necessary.

Our proposed methodology involves assigning lower values to features with higher frequencies after the embedding process, while assigning higher values to features with lower frequencies. This approach is based on the rationale that neural units may experience fatigue when exposed to high-frequency values, resulting in reduced input. Figure 1 provides a visual representation of our proposed inverse probability weighting encoder.

A key characteristic shared by heterogeneous features is their occurrence as measurements. Regardless of the feature type (nominal, ordinal, interval, or ratio), these occurrences hold significant meaning. Therefore, we choose to transform each feature into an occurrence space, where each dimension corresponds to a feature value and each point represents a distinct instance. The values within this space reflect the observation probabilities, including conditional probabilities, associated with each instance, ranging from 0 to 1.

	Feature Scale	Encoder	Example
Unsupervised	Nominal	One-hot	[1,2,3]→[[1,0,0],[0,1,0],[0,0,1]]
		Binary	[1,2,3]→[[0,0],[0,1],[1,0]]
		Dummpy	[1,2,3]→[[1,0],[0,1],[0,0]]
		Count	[1,1,3]→[[2],[2],[1]]
		Simple	/
	Ordinal	Ordinal	[1,2,3]→[1,2,3]
		Rank-hot	[1,2,3]→[[1,0,0],[1,1,0],[1,1,1]]
		Gray	[1,2,3]→[[0,0],[0,1],[1,1]]
		Helmert contrast	/
		Orthogonal polynominal	/
		Backward difference	/
		Base-N	/
	Continuous	Bins+Categorical	/
		Piece-wise linear (PLE)	[0.11,0.22,0.31]→[[0.1,0,0],[1,0.2,0],[1,1,0.1]]
	Other	Hashing	/
Supervised	Categorical	Greedy TS (target statistic)	/
		Holdout TS (target statistic)	/
		Ordered TS (target statistic)	/
	Continuous	One-blob	/
		Periodic	/

TABLE II: Heterogeneous Feature Encoders

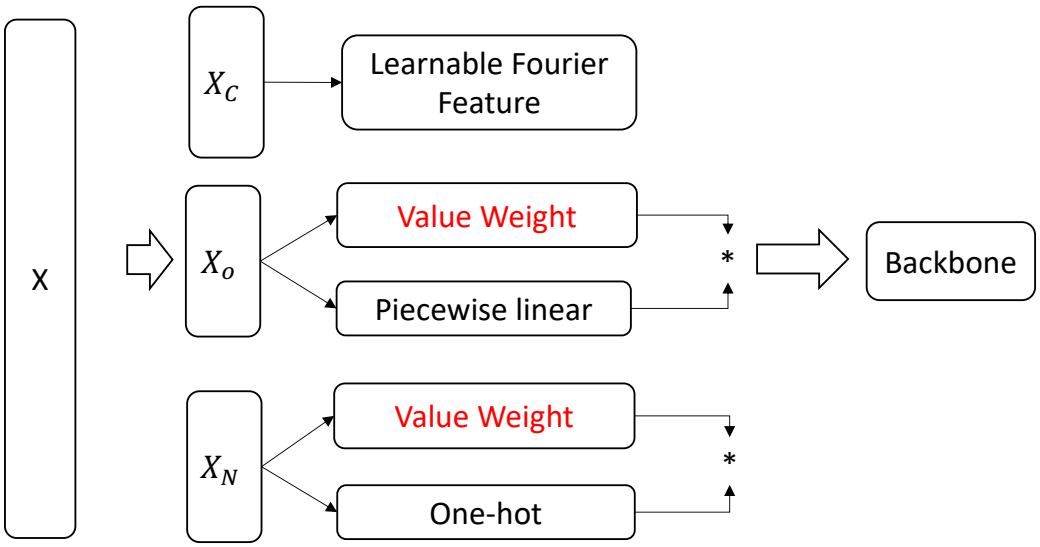

Fig. 1: Proposed Embedding Module. The Embedding Module proposed in this study incorporates the information of occurrence probabilities, as indicated by the red text. The module handles three types of features: continuous (X_C), ordinal (X_O), and nominal (X_N).

TABLE III: Dimensionality reduction assumptions. G: Gaussian; I: independent; nG: non-Gaussian; $\perp$: orthogonal; $\rightarrow$: generate; ANN: additive normal noise; DAG: directed acyclic graph.

Method	Mapping	$p(\mathbf{z})$	$p(\mathbf{x})$
PCA	Linear	IG	IG
ICA	Linear	InG	InG+G
t-SNE	Nonlinear	Local continuty	Local continuty
βVAE	Nonlinear	IG with β	\
NGCA	Linear	G$\perp$nG	ANN
LinGAM	Linear	G$\rightarrow$nG	ANN with DAG

However, a significant challenge arises due to the combinatorial nature of both the space dimensions and the probabilities. To overcome this challenge, we posit that the occurrence space can be separated based on features. Consequently, we derive a set of d_x subspaces denoted as $\mathcal{S}1, ..\mathcal{S}d_x$. Each subspace encapsulates dimensions represented by $|f_1|, ..|f_n|$. As a result, the dimensionality of the occurrence space is effectively reduced to $\sum_{i=1}^{n} |f_i|$. To approximate the combinatorial observation probabilities, we utilize zero-order conditional probabilities $p(x_1^k), ...p(x_{d_x}^k)$.

The subsequent aspect involves the integration of information from the occurrence space and the original value representations. Nominal features are encoded using a one-hot encoder, facilitating the capture of occurrence probabilities. For ordinal features, we employ a rank-hot encoder, as the occurrence of higher rank values implies the occurrence of lower rank values. Continuous features are represented using a Fourier-based form, enabling the modeling of occurrence patterns across different frequencies.

VI. Experiment

A. Task 1: Predicting Outcome in Randomized Trial

Predicting individual outcomes in randomized trials is a fundamental objective in causal inference. In datasets derived from randomized trials, instances are characterized by three classes of features: treatment, pre-treatment, and post-treatment. The values of the treatment feature are not determined by the data collector but are instead assigned by a random generator controlled by the researchers. Subsequently, the treatment is implemented according to the assigned value by the trial executor. The pre-treatment features are measured prior to the assignment of the treatment value, while the post-treatment features are measured subsequent to the treatment assignment. Among the post-treatment features, three categories can be distinguished: main outcome, secondary outcome, and additional post-treatment variables. However, in this context, the term "outcome" is often used to refer specifically to the post-treatment variable of primary interest.

1) Dataset: We have curated a comprehensive collection of randomized trial datasets and made them publicly accessible on the website: https://github.com/herdonyan/RandomizedTrialDataset.
The datasets encompass a wide range of randomized trials, representing a valuable resource for researchers in various fields. Despite the significant cost associated with designing and executing randomized trials, there is a growing trend among funding agencies and journals to mandate the availability of these datasets, while ensuring privacy protection measures are in place. This initiative aims to promote transparency, reproducibility, and collaboration in the scientific community by facilitating access to randomized trial data and fostering further research advancements.

The AKIAlert dataset is a valuable resource derived from a randomized trial that recorded electronic health record (EHR) data of patients. This dataset follows a double-blinded, multicenter, and parallel design. The primary focus of the trial is to evaluate the impact of an acute kidney injury (AKI) alert provided by the electronic system compared to usual care without an alert. The participants were identified electronically and randomized using a simple randomization approach with allocation concealment.

The dataset comprises a total of 6,030 adult inpatients with AKI, which is defined based on the Kidney Disease: Improving Global Outcomes (KDIGO) creatinine criteria. It includes 49 pre-treatment variables, 1 main outcome variable, and additional post-treatment variables. Among the 49 pre-treatment variables, there are 9 nominal features, 19 ordinal features, 3 interval features, and 20 ratio features.

Within the cohort of 6,030 patients, 948 individuals experienced AKI progression within a 14-day period, while 5,082 patients did not exhibit such progression. The dataset provides a valuable resource for conducting analyses and exploring the impact of the AKI alert on various post-treatment variables, aiding researchers in gaining insights into the management and outcomes of AKI in the context of electronic health records.

The primary task of interest in the AKIAlert dataset is to predict the occurrence of AKI progression within 14 days of randomization based on the available pre-treatment variables. This task can be formulated as a classical binary classification problem, where the objective is to distinguish between patients who will experience AKI progression within the specified timeframe and those who will not.

By leveraging the 49 pre-treatment variables present in the dataset, researchers can develop predictive models and algorithms to identify patterns and relationships that may contribute to the prediction of AKI progression. These variables, including the 9 nominal features, 19 ordinal features, 3 interval features, and 20 ratio features, offer a rich set of information to analyze and extract relevant predictors for the binary classification task.

The successful development of a predictive model for AKI progression within 14 days in randomized trials can have significant clinical implications, enabling early identification and intervention for alert-benefited patients while avoid the for alert-harmful patients. Moreover, it can contribute to advancing the field of acute kidney injury research and improving patient outcomes in healthcare settings.

In order to address the challenge posed by label imbalance, we employ the average precision score (PR-AUC) as the performance metric for evaluating the models. The PR-

TABLE IV: Heterogeneous datasets for outcome prediction

Dataset	Instance	Outcome	Treatment
Safety and Preliminary Efficacy of Intranasal Insulin for Cognitive Impairment in Parkinson Disease and Multiple System Atrophy	16	Parkinson disease	Intranasal insulin
	https://physionet.org/content/inipdmsa/1.0/		
Tai Chi, Physiological Complexity, and Healthy Aging - Gait	60	Gait and EMG data	Tai Chi
	https://physionet.org/content/taichidb/1.0.2/		
ECG Effects of Dofetilide, Moxifloxacin, Dofetilide+Mexiletine, Dofetilide+Lidocaine and Moxifloxacin+Diltiazem	22	ECG	Dofetilide, Moxifloxacin, Dofetilide+Mexiletine, Dofetilide+Lidocaine and Moxifloxacin+Diltiazem
	https://physionet.org/content/ecgdmmld/1.0.0/		
ECG Effects of Ranolazine, Dofetilide, Verapamil, and Quinidine	22	ECG	Ranolazine, Dofetilide, Verapamil, and Quinidine
	https://physionet.org/content/ecgrdvq/1.0.0/		
CAST RR Interval Sub-Study Database	734	Cardiac arrhythmia suppression	Encainide, flecainide, moricizine (antiarrhythmic drugs) or a placebo
	https://physionet.org/content/crisdb/1.0.0/		
Randomized trial of AKI alerts in hospitalized patients	6030	Acute Kidney Injury	Electronic AKI alert versus usual care
	https://datadryad.org/stash/dataset/doi:10.5061%2Fdryad.59zw3r27n		
Telerehabilitation program for COVID-19 survivors (TERECO) - Randomized controlled trial	120	Exercise capacity, lower-limb muscle strength (LMS), pulmonary function, health-related quality of life (HRQOL), and dyspnoea	Telerehabilitation program for COVID-19 survivors
	https://datadryad.org/stash/dataset/doi:10.5061%2Fdryad.59zw3r27n		
Bicycling comfort video experiment	15289	Bicycle rating	Video Type
	https://datadryad.org/stash/dataset/doi:10.25338%2FB8KG77		
Megafon uplift competition	1.5 million	User conversion	Exposure
	https://ods.ai/tracks/df21-megafon/competitions/megafon-df21-comp/data		
Infant Health and Development Program	1090	Cognitive development, Behavior problems, Health status	Home visits, attendance at a special child development center
	https://www.icpsr.umich.edu/web/HMCA/studies/9795		
National Supported Work Evaluation Study	6600	effects of the Supported Work Program	Offered a job in supported work
	https://www.icpsr.umich.edu/web/ICPSR/studies/7865		
CPAP Pressure and Flow Data from a Local Trial of 30 Adults at the University of Canterbury	30	Breathing	Continuous positive airway pressure
	https://physionet.org/content/cpap-data-canterbury/1.0.1/		

AUC, a commonly utilized measure in binary classification tasks, provides a comprehensive evaluation of the model's effectiveness in scenarios where there is a significant disparity in the class distribution.

Unlike conventional evaluation metrics such as accuracy or F1-score, the PR-AUC takes into account both precision and recall, which are particularly relevant in imbalanced datasets. Precision quantifies the proportion of correctly predicted positive instances out of all instances classified as positive, while recall captures the proportion of correctly predicted positive instances out of the total number of actual positive instances.

By calculating the area under the precision-recall curve, the PR-AUC delivers a comprehensive assessment of the model's performance across a range of classification thresholds. This approach proves advantageous when dealing with imbalanced datasets, as it focuses on the performance of the minority class (AKI progress) and is less influenced by the dominance of the majority class (non-AKI progress).

Employing the PR-AUC as the evaluation metric in the assessment of models on the AKIAlert dataset ensures a robust estimation of their predictive capabilities, mitigating the effects of label imbalance. Higher PR-AUC scores signify superior performance in accurately identifying instances of AKI progression, thereby contributing to enhanced patient management and facilitating informed clinical decision-making.

2) Implementation details: To address the potential impact of randomness in dataset splitting, all experiments were conducted five times using different random splits of the

dataset. This approach helps mitigate the influence of splitting randomness and provides a more robust evaluation of the models' performance.

Four different models were evaluated on the dataset: CatBoostTree, HetMLP (our proposed model), Vanilla MLP, and stochastic prediction. Each model was trained and tested using the randomized dataset splits, ensuring a comprehensive assessment of their respective performance.

By employing multiple executions of the experiments and evaluating different models, we aim to obtain reliable and statistically significant results. This approach allows us to analyze the performance of each model across multiple iterations, capturing variations in their predictive capabilities and facilitating a more comprehensive understanding of their strengths and weaknesses.

The utilization of this experimental methodology enhances the reliability and validity of our findings, enabling us to draw robust conclusions regarding the comparative performance of the evaluated models on the given dataset.

For the CatBoostTree model, which is considered state-of-the-art (SOTA) for our experiment, all nominal and ordinal features were categorized as categorical features. The default parameters of the CatBoostClassifier were employed during the training process.

The model was trained for a total of 500 iterations, with a learning rate set to 0.03. The depth of the trees used in the model was set to 6, while the l_2 regularization applied to the leaves had a value of 3.0. The loss function utilized for training was the logarithmic loss.

During the tree construction process, a maximum of 4 combinations of categorical features were considered, providing flexibility in capturing potential interactions between these features. Additionally, the minimum number of training samples required in each leaf node was set to 1, ensuring the model's adaptability to various sample sizes.

To control the complexity of the resulting tree, the maximum number of leaf nodes was limited to 31. The cosine function was employed as the score function to guide the selection of the next split in the tree construction process, contributing to an effective and efficient model.

Furthermore, the scale position weight was utilized to address any potential data imbalance issues, ensuring that the model's performance was not skewed by uneven class distributions.

By specifying these parameters and employing the CatBoostTree model, we aimed to leverage its advanced capabilities for handling categorical features and effectively modeling the dataset under consideration.

The VilliaMLP model utilized a multi-layer perceptron (MLP) architecture as its backbone. The MLP consisted of three layers, with hidden units of 1024, 512, and 256, respectively. Dropout regularization was applied with a rate of 0.1 to prevent overfitting.

The learning rate for training the model was set to 0.001, promoting efficient optimization during the learning process. Weight decay regularization with a coefficient of 0.000001 was incorporated to prevent excessive model complexity and enhance generalization.

To assess the model's performance and prevent overfitting, a train/test splitting ratio of 8:2 was employed, with 80% of the data allocated for training and 20% for testing. Additionally, a train/validation splitting ratio of 9:1 was used to further evaluate the model's performance during training. Validation was performed at each epoch, and an early stop policy was implemented to select the best-performing model based on the validation dataset. The early stop epoch was set to 20.

For handling the different types of features, specific encoders were employed. Nominal features were encoded using one-hot encoding, while ordinal features were encoded using an ordinal encoder. Continuous features were encoded using a learnable Fourier encoder, allowing the model to effectively capture and represent the underlying patterns in the data. The maximum frequency number for the Fourier encoder was set to 200.

To address label imbalance, the loss function employed a weighting scheme based on the labels. This approach helped mitigate any potential degradation in performance resulting from imbalanced class distributions, ensuring fair and accurate model evaluation.

By leveraging the VilliaMLP model with these configurations and techniques, we aimed to effectively leverage the MLP architecture and appropriate feature encoders to achieve accurate predictions while mitigating common challenges such as overfitting and label imbalance.

In contrast to VilliaMLP, HetMLP shares a similar architecture and configuration, with the exception of the feature encoders used. In HetMLP, we adopted a different approach by assigning weights to the transformed features based on their inverse probabilities.

Specifically, the weight assigned to each transformed feature was determined by its frequency in the dataset. If a feature appeared frequently, it was assigned a smaller weight, whereas features with lower frequencies were given larger weights. This weighting scheme aimed to address the heterogeneity in feature frequencies and ensure that each feature contributed appropriately to the overall model representation.

By incorporating these weighted transformed features, HetMLP aimed to capture the varying importance and impact of different features based on their frequencies. This approach allowed the model to effectively handle the heterogeneous nature of the dataset, providing a more nuanced and informative representation of the input data.

Overall, HetMLP and VilliaMLP shared similar architectural configurations, but their respective feature encoding strategies differed. HetMLP leveraged the inverse probability weighting of transformed features to effectively address the heterogeneity of feature frequencies and enhance the model's predictive performance.

Stochastic prediction was employed as a baseline method to compare against the performance of the proposed models. In this approach, the probability $p(y)$ was utilized to make

Algorithm	PR-AUC
HetMLP	.2117±.0009
VilliaMLP	.2087±.0164
Baseline (Stochastic)	.1568±.0089

TABLE V: PR-AUC of different models

predictions regarding whether a patient would experience acute kidney progress in the future.

By employing stochastic prediction, we aimed to establish a reference point for evaluating the effectiveness of the other models. The baseline method provided a benchmark against which the performance improvements of the CatBoostTree, VilliaMLP, and HetMLP models could be assessed.

Through this comparative analysis, we sought to highlight the advancements and enhancements achieved by the proposed models over the stochastic prediction baseline. The evaluation of the models against this baseline allowed for a comprehensive understanding of their predictive capabilities and demonstrated the potential improvements that can be achieved in predicting acute kidney progress within the given dataset.

B. Result Analysis

In order to assess the effectiveness of our algorithm, we compared its performance with that of VilliaMLP using the average precision score (PR-AUC) metric. The PR-AUC metric was chosen to account for the label imbalance in the dataset.

Upon evaluating the results, it can be observed that our algorithm outperformed VilliaMLP in terms of PR-AUC. The higher PR-AUC score achieved by our algorithm indicates its superior ability to accurately predict the occurrence of acute kidney progress within the specified timeframe.

The comparison with VilliaMLP serves as empirical evidence supporting the effectiveness and improved performance of our algorithm in addressing the given task. The results demonstrate the potential of our algorithm as a valuable approach for predicting acute kidney progress in randomized trial, surpassing the performance of the previously established VilliaMLP model.

VII. Future plan

In tabular data, there are not only the four scales datatype. Other data is also useful, such as text, feature names, questions of survey. In future, we plan to combine the API of big model to deal with the heterogeneity outside the Steven's scales theory.

Another challenge is time-series data, most algorithms and models addressing on data heterogeneity are often assume independently identified distribution. However, time-series data is also very important in machine learning. In future, we will extend our algorithms to hetergeneous time-series data.

REFERENCES

[1] S. S. Stevens, "On the theory of scales of measurement," *Science*, vol. 103, no. 2684, pp. 677–680, 1946.

[2] P. F. Velleman and L. Wilkinson, "Nominal, ordinal, interval, and ratio typologies are misleading," *The American Statistician*, vol. 47, no. 1, pp. 65–72, 1993.

[3] Y. Gorishniy, I. Rubachev, and A. Babenko, "On embeddings for numerical features in tabular deep learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 991–25 004, 2022.

[4] T. Müller, B. McWilliams, F. Rousselle, M. Gross, and J. Novák, "Neural importance sampling," *ACM Transactions on Graphics (ToG)*, vol. 38, no. 5, pp. 1–19, 2019.

[5] Y. Xia, H. Tong, W. K. Li, and L.-X. Zhu, "An adaptive estimation of dimension reduction space," in *Exploration of A Nonlinear World: An Appreciation of Howell Tong's Contributions to Statistics*. World Scientific, 2009, pp. 299–346.

[6] Y. Dong and C. Gao, "An adaptive dimension reduction algorithm for latent variables of variational autoencoder," *arXiv preprint arXiv:2111.08493*, 2021.

[7] G. Darnell, S. Georgiev, S. Mukherjee, and B. E. Engelhardt, "Adaptive randomized dimension reduction on massive data," *Journal of Machine Learning Research*, 2017.

[8] G. Blanchard, M. Kawanabe, M. Sugiyama, V. Spokoiny, K.-R. Müller, and S. Roweis, "In search of non-gaussian components of a high-dimensional distribution." *Journal of Machine Learning Research*, vol. 7, no. 2, 2006.

[9] D. M. Bean, *Non-Gaussian component analysis*. University of California, Berkeley, 2014.

[10] N. Goyal and A. Shetty, "Non-gaussian component analysis using entropy methods," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 840–851.

[11] C. Ding and T. Li, "Adaptive dimension reduction using discriminant analysis and k-means clustering," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 521–528.

[12] T. Luo, C. Hou, F. Nie, and D. Yi, "Dimension reduction for non-gaussian data by adaptive discriminative analysis," *IEEE transactions on cybernetics*, vol. 49, no. 3, pp. 933–946, 2018.

[13] P. M. Baggenstoss and S. Kay, "Nonlinear dimension reduction by pdf estimation," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1493–1505, 2022.

[14] A. Tavory, "Determining principal component cardinality through the principle of minimum description length," in *International Conference on Machine Learning, Optimization, and Data Science*. Springer, 2019, pp. 655–666.

[15] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[16] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[17] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[18] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.