Research Article

BUPNN: Manifold Learning Regularizer-Based Blood Usage Prediction Neural Network for Blood Centers

Lingling Pan,[1,2] Zelin Zang [iD],[3] Siqi Ma,[3] Wei Hu [iD],[1,2] and Zhechang Hu[1]

[1]*Blood Center of Zhejiang Province, 789 Jianye Road, Binjiang District, Hangzhou 310052, Zhejiang, China*
[2]*Key Laboratory of Blood Safety Research, Hangzhou 310052, Zhejiang, China*
[3]*AI Lab, School of Engineering, Westlake University, Hangzhou 310052, Zhejiang, China*

Correspondence should be addressed to Wei Hu; huwei0507@126.com

Blood centers are an essential component of the healthcare system, as timely blood collection, processing, and efficient blood dispatch are critical to the treatment of patients and the performance of the entire healthcare system. At the same time, an efficient blood dispatching system through the high-precision predictive capability of artificial intelligence is crucial for the efficiency improvement of the blood centers. However, the current artificial intelligence (AI) models for predicting blood usage do not meet the needs of blood centers. The challenges of AI models mainly include lower generalization ability in different hospitals, limited stability under missing values, and low interpretability. An artificial neural network-based model named the blood usage prediction neural network (BUPNN) has been developed to address these challenges. BUPNN includes a novel similarity-based manifold regularizer that aims to enhance network mapping consistency and, thus, overcome the domain bise of different hospitals. Moreover, BUPNN diminishes the performance degradation caused by missing values through data enhancement. Experimental results on a large amount of accurate data demonstrate that BUPNN outperforms the baseline method in classification and regression tasks and excels in generalization and consistency. Moreover, BUPNN has solid potential to be interpreted. Therefore, the decision-making process of BUPNN is explored to the extent that it acts as an aid to the experts in the blood center.

1. Introduction

Blood products are an essential part of the treatment of bleeding, cancer, AIDS, hepatitis, and other diseases[1]. Blood is also an indispensable resource in treating injured patients; whether promptly transfusing the blood is critical for rehabilitating injured patients. At the same time, early surgical interventions and rapid blood transfusions are the primary measures to reduce mortality. Unfortunately, these measures require large amounts of blood to support them.

Blood, however, is significantly different from other medical products. Currently, blood cannot be manufactured or synthesized artificially and can only be donated by others. In addition, blood has a short shelf life, making emergency blood more specific and irreplaceable than other medical products. For example, patients in mass casualty events

caused by earthquakes suffer from fractures, fractures accompanied by multiple organ injuries, and crush injuries. Therefore, the peak period of blood consumption occurs 96 hours after the earthquake [2]. In contrast, patients mainly suffer from burns in mass casualties caused by bombings or fires. The peak blood consumption occurs 24 hours after the event. In addition, ongoing blood transfusions may last for several months [3]. Therefore, modeling the prediction of blood use in patients is a meaningful and challenging topic.

Current blood usage prediction models are not well developed. For example, Wang et al. [4] developed an early transfusion scoring system to predict blood requirements in severe trauma patients in the prehospital or initial phase of emergency resuscitation. However, the abovementioned design is more suitable for triage in a

single hospital than in blood centers since the system does not consider how to avoid performance degradation due to domain deviations between different hospitals. Rebecca et al. [5, 6] summarized the currently available studies exploring how to predict the need for massive transfusion in patients with traumatic injuries, listing the blood consumption scoring system (ABC) and the shock index scoring system (SI). These systems use classical or ML-based methods to predict blood consumption during the treatment of patients. Unfortunately, although the progress is remarkable, but it is unsatisfactory in terms of accuracy and generalization, and thus, the blood demand prediction model cannot be widely used. We summarize the problems into the following three points:

(i) Data quality is limited. The partnership between blood centers and hospitals makes it very difficult to establish a rigorous feedback system for patient information. In addition, biases and missing data due to differences in hospital equipment create challenges for a constant blood use prediction model.

(ii) Model generalizability is unsatisfactory. Blood usage prediction models are often built for specific hospitals or cities without considering extending to a wider range of applications and thus have poor generalization performance.

(iii) Model interpretability is inadequate. Most models can only output a category or blood usage forecast but cannot demonstrate the model's decision process. An interpretable model can better work with experts to help blood centers with blood schedules.

This paper proposes a blood usage prediction neural network, BUPNN, to solve the above problems. First, actual patient clinic data and treatment procedures in 12 hospitals are used as training data. Extensively collected data with biases from various hospitals will provide sufficient information to train a high-performance model. We used multiple data complementation schemes to restore the real problem and overcome missing values. In addition, the BUPNN model MDC is augmented with online data by linear interpolation, which increases the diversity of training data and thus improves the stability of the model under training with missing data. Second, to further improve the generalization performance of BUPNN, a similarity-based loss function is introduced to map data with biases to a stable semantic space by aligning samples from different hospitals in the latent space. Third, we analyze the model based on the deep learning interpretation method to enhance interpretability. The proposed analysis is accompanied by the prediction output of the model in real-time to assist one in understanding the process of the model's decision-making. The interpretable study of BUPNN provides the conditions for computers and experts to help each other in blood consumption prediction.

The main contributions of this study are as follows:

(i) Representative samples. A large amount of data from twelve hospitals is collected for this study to investigate the implied relationship between patients' chain indicators and blood consumption.

(ii) Generalizable model. This study designs MDC for online missing data completion, and thus, data augmentation to enhance the model's generalization ability. In addition, the various similarity loss function is designed to improve the model's predictive power across domains.

(iii) Excellent Performance. Experiments on six different settings demonstrate that our method outperforms all baseline methods.

The rest of this paper is organized as follows. In Section 2, we provide a literature review of demand forecasting methods for blood products. Section 3 provides an initial exploration of the data. We provide the data description, model background, model development, and evaluation of four different models for blood demand forecasting in Section. In Section 4, a comparison of the models is provided, and finally, in Section 5, concluding remarks are provided, including a discussion of ongoing work for this problem.

2. Related Work

2.1. ML Techniques for Medical Problems. The integration of the medical field with ML technology has received much attention in recent years. Two areas that may benefit from the application of ML technology in the medical field are diagnosis and outcome prediction. It includes the possibility of identifying high-risk medical emergencies, such as recurrence or transition to another disease state. Recently, ML algorithms have been successfully used to classify thyroid cancer [7] and predict the progression of COVID-19 [8, 9]. On the other hand, ML-based visualization and dimensionality reduction techniques have the potential to help professionals analyze biological or medical data, guiding them to better understand the data [10, 11]. Furthermore, ML-based feature selection techniques [12, 13] have strong interpretability and the potential to find highly relevant biomarkers for output in a wide range of medical data, leading to new biological or medical discoveries.

2.2. Blood Demand Forecasting. There is limited literature on blood demand forecasting; most investigate univariate time series methods. In these studies, forecasts are based solely on previous demand values without considering other factors affecting the demand. Frankfurter et al. [14] developed transfusion forecasting models using exponential smoothing (ES) methods for a blood collection and distribution center in New York. Critchfield et al. [15] developed models for forecasting blood usage in a blood center using several time series methods, including moving average (MA), winter's approach, and ES. Filho et al. [16] developed a Box-Jenkins seasonal

autoregressive integrated moving average (BJ-SARIMA) model to forecast weekly demand for hospital blood components. Their proposed method, SARIMA, is based on a Box-Jenkins approach that considers seasonal and nonseasonal characteristics of time series data. Later, Filho et al. [17] extended their model by developing an automatic procedure for demand forecasting and changing the model level from hospital level to regional blood center to help managers use the model directly. Kumari and Wijayanayake [18] proposed a blood inventory management model for daily blood supply, focusing on reducing blood shortages. Three time series methods, namely, MA, weighted moving average (WMA), and ES, are used to forecast blood usage and are evaluated based on needs. Fortsch and Khapalova [19] tested various blood demand prediction approaches, such as naive, moving average, exponential smoothing, and multiplicative time series decomposition (TSD). The results show that the Box-Jenkins (ARMA) approach, which uses an autoregressive moving average model, results in the highest prediction accuracy. Lestari et al. [20] applied four models to predict blood component demand, including moving average, weighted moving average, exponential smoothing, exponential smoothing with the trend, and select the best method for their data based on the minimum error between forecasts and the actual values. Volken et al. [21] used generalized additive regression and time-series models with exponential smoothing to predict future whole blood donation and RBC transfusion trends.

Several recent studies consider clinically related indicators. For example, Drackley et al. [22] estimated a long-term blood demand for Ontario, Canada, based on previous transfusions' age and sex-specific patterns. They forecast blood supply and demand for Ontario by considering demand and supply patterns and demographic forecasts, assuming fixed patterns and rates over time. Khaldi et al. [23] applied artificial neural networks (ANNs) to forecast the monthly demand for three blood components: red blood cells (RBCs), blood, and plasma, for a case study in Morocco. Guan et al. [24] proposed an optimisation ordering strategy in which they forecast the blood demand for several days into the future and build an optimal ordering policy based on the predicted direction, concentrating on minimising the wastage. Their primary focus is on an optimal ordering policy. They integrate their demand model into the inventory management problem, meaning they do not precisely try to forecast blood demand. Li et al. [25] developed a hybrid model consisting of seasonal and trend decomposition using Loess (STL) time series and eXtreme Gradient Boosting (XGBoost) for RBC demand forecasting and incorporated it into an inventory management problem.

Recently, Motamedi et al. [26] presented an efficient forecasting model for platelet demand at Canadian Blood Services (CBS). In addition, C. Twumasi and J. Twumasi [27] compared k-nearest neighbour regression (KNN), multilayer perceptron (MLP), and support vector machine (SVM) via a rolling-origin strategy for forecasting and backcasting blood demand data with missing values and outliers from a government hospital in Ghana. Abolghasemi et al. [28] treat the blood supply problem as an optimisation problem [29] and find that LightGBM provides promising solutions and outperforms other machine learning models.

3. Blood Centers and Datasets

This section describes how blood center works with an example in Zhejiang Province, China. It includes the responsibilities of the blood center and the partnership between the blood center and the hospital, and shows in detail how the proposed model will assist the blood center in accomplishing its mission better.

3.1. Blood Center. Figure 1 shows that, this study considers a provincial centralized blood supply system, including blood centers, blood stations, and hospitals. The entire centralized blood supply system completes the collection, management, and delivery of blood products. The blood center is responsible for collecting blood products, testing for viruses and bacteria, and supplying some hospitals. At the same time, blood centers assume the management, coordination, and operational guidance of blood collection and supply institutions. Blood stations are responsible for collecting, storing, and transporting blood to local hospitals. While receiving blood, hospitals must collect clinical information on current patients for analysis and decision-making at the blood center to make the blood supply more efficient. Our proposed AI blood consumption prediction system (BUPNN) receives clinical information from each hospital and uses it to predict the future blood consumption of each patient. The proposed system helps blood center specialists perform blood collection and transportation better.

3.2. Data Details and Challenges. We collected data in their actual state to build a practically usable model. The data in this study are constructed by processing BS shipping data and the TRUST (Transfusion Research for Utilization, Surveillance, and Tracking) database from Zhejiang Province Blood Center and 12 hospitals in Zhejiang Province. The data include data from 2025 patients, including 1970 data from emergency trauma patients in 10 hospitals in Zhejiang Province from 2018 to 2020 and another 55 from patients in two hospitals in Wenling's explosion in 2020.

Each dataset mainly included the following parts. (1) General patient information, including case number, consultation time, pretest classification, injury time, gender, age, weight, diagnosis, penetrating injury, heart rate, diastolic blood pressure, systolic blood pressure, body temperature, shock index, and Glasgow coma index. (2) Injury status, including pleural effusion, abdominal effusion, extremity injury status, thoracic and abdominal injury status, and pelvic injury status. (3) Laboratory tests, including hemoglobin, erythrocyte pressure product, albumin, hemoglobin value 24 hours after transfusion therapy, base residual, PH (acidity), base deficiency, oxygen saturation, PT (prothrombin time), and APTT (partial thromboplastin time). (4) Burn situation, including burn area and burn depth. (5) Patient regression, including whether blood use: whether to transfuse suspended red blood cells, the first day of hospital transfusion, and the amount of blood transfusion.

For a more straightforward presentation of the data distribution, preliminary dataset statistics are shown in
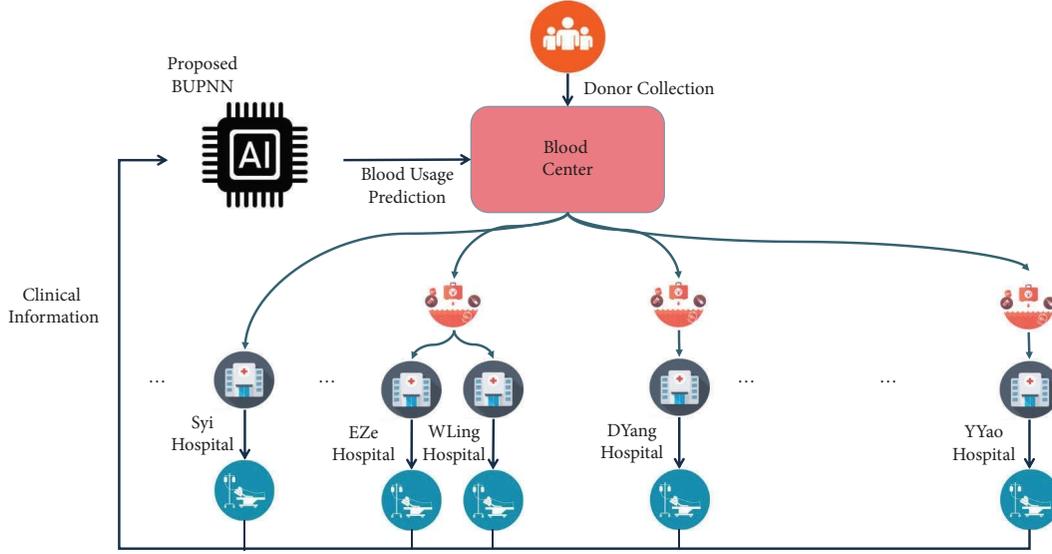
FIGURE 1: BS blood supply chain with one regional blood center and multiple hospitals.

Table 1, and box plots by age, blood consumption, and missing rate are shown in Figure 2.

After a detailed definition of the problem and a description of the dataset, we summarize the main problems faced in this paper.

(i) Data holds missing values. as shown in Table 1 and Figure 2(c), some hospitals have more than 10% missing values. The missing values cause perturbation to the data distribution, severely affecting the model's training and performance.

(ii) Data has domain bias. from Table 1 and Figure 2(c), the missing values rate and blood consumption have obvious differences in different hospitals. Such a data distribution impacts the cross-hospital generalization of the model. In addition, the collection of clinical information from different hospitals may also be biased due to differences in testing devices and habits.

4. Methodology

4.1. Problem Defensions. The following definition is made in this paper to discuss the role of predictors.

Definition 1. (Blood Consumption Prediction Problem, BCPP). Let $(\mathbf{X}^s, \mathbf{y}^s)$ be a training dataset where clinical information $\mathbf{X}^s$ is implicitly linked to the blood usage $\mathbf{y}^s$. The BCPP train a model $F(X \mid \theta)$ with $(\mathbf{X}^s, \mathbf{y}^s)$, and use the model $F(X \mid \theta)$ predict the blood usage of new collected data of clinical information $\mathbf{X}^t$, where θ is the model parameter.

The BCPP includes a classification subproblem and regression subproblem. For classification subproblem $\mathbf{y}^s$ and $\mathbf{y}^t$ are one-hot or category values. For regression subproblem $\mathbf{y}^s$ and $\mathbf{y}^t$ are continuous values.

4.2. Blood Data Complementation. Data complementation is the process of replacing missing data with substituted values. When covering for a data point, it is known as "unit complementation;" when substituting for a component of a data point, it is known as "item complementation." Missing values introduce substantial noise, making data analysis more complex and less efficient. When one or more patient values are missing, most methods discard data with missing values by default, but data complementation attempts to fill in those values. However, missing data are also a reality, and the model should not require that all data be captured well. Therefore, data complementation is introduced to avoid performance degradation with missing values and improve the actual testing data's performance.

In this study, a single data $x_i \in \mathbf{X}^S$ is complemented by

$$\mathbf{x}_i^C = \mathbf{C}(\mathbf{x}_i) = \left\{ \mathbb{C}(\mathbf{x}_{i,1}), \ldots, \mathbb{C}(\mathbf{x}_{i,f}), \ldots, \mathbb{C}(\mathbf{x}_{i,n}) \right\},$$

$$\mathbb{C}(\mathbf{x}_{i,f}) = \begin{cases} \mathbf{x}_{i,f} & \mathbf{x}_{i,f} \text{ is.not.missing}, \\ I_{i,f} & \mathbf{x}_{i,f} \text{ is.missing}, \end{cases} \quad (1)$$

where $\mathbf{x}_{i,1}, \cdots, \mathbf{x}_{i,n}$ are n data components of single data $\mathbf{x}_i$. If any component is missing, the imputed value I_f is used to fill this missing value. The I_f comes from mean value complementation, median value complementation, and KNN complementation:

$$I_{i,f}^{\text{Mean}} = \text{Mean}(\mathbf{X}_f),$$

$$I_{i,f}^{\text{Median}} = \text{Median}(\mathbf{X}_f), \quad (2)$$

$$I_{i,f}^{\text{KNN}} = \text{KNN}\left(\frac{1}{K} \sum_{j \in N_i^{K-NN}} \mathbf{x}_{j,f} \right),$$

TABLE 1: Statistics of datasets.

Hospital name	Hospital abbreviation	Sample size	Average blood usage	Average missing rate	Male/female ratio	Average age
Dongyang Hospital	DYang	95	8.7	0.03	0.34	49.19
Enze Hospital	EZe	13	0.77	0.18	0.86	50.38
Haining Hospital	HNing	57	15.96	0.17	0.68	55.18
Shiyi Hospital	SYi	72	4.65	0.08	0.24	56.40
Shaoyifu Hospital	SYiFu	191	1.46	0.05	0.41	52.50
Shangyu Hospital	SYu	135	7.92	0.07	0.42	51.08
Wenling Hospital	WLing	42	11.62	0.09	0.45	58.98
Xinchang Hospital	XChang	55	11.09	0.03	0.49	57.89
Xiaoshan Hospital	XShan	62	0	0.08	0.44	50.82
Yongkang Hospital	YKang	194	2.36	0.06	0.62	64.43
Yuyao Hospital	YYao	65	9.85	0.03	0.35	54.68
Zheer Hospital	ZEr	1044	1.44	0.03	0.36	53.93
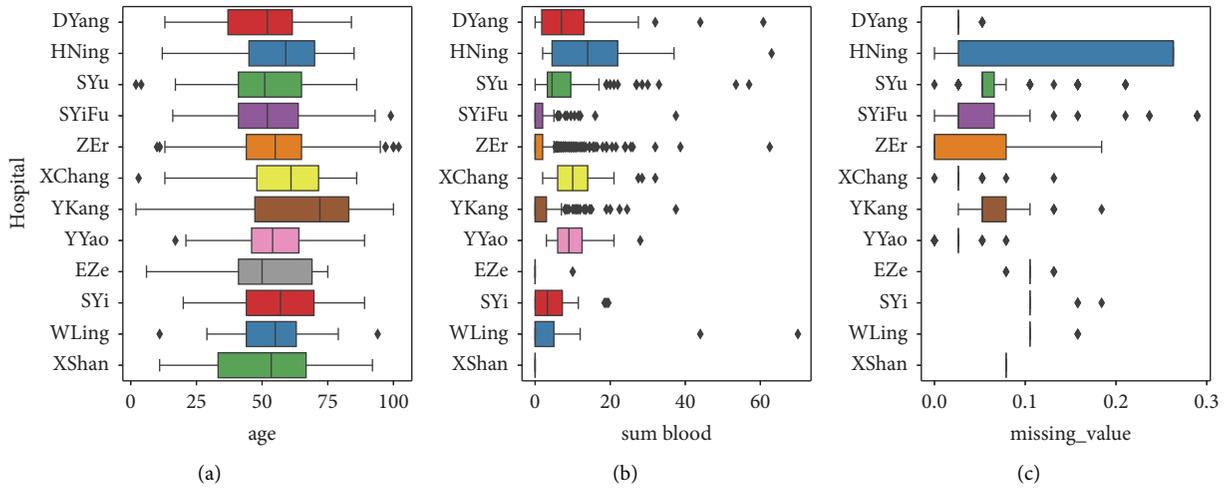
FIGURE 2: Boxplots for the relationship between hospital and age/sum blood/missing value.

where $\text{Mean}(X_f^s)$ and $\text{Median}(X_f^s)$ are the mean value or median value of training datasets on components f. N_i^{K-NN} is the KNN neighborhood of data i in the sense of European distance, $K = 5$ in this paper.

4.3. Cross Hospitals Data Augmentation.
Data augmentation is a well-known neural network (NN) training strategy for image classification and signal processing [30]. Data augmentation improves the performance of the methods by precisely fitting the data distribution. First, data augmentation enhances the diversity of data, thereby overcoming overfitting. Second, data augmentation essentially reinforces the fundamental assumption of DR, i.e., the local connectivity of neighborhoods. Finally, it learns refined data distribution by generating more intra-manifold data based on sampled points.

Cross-hospital data augmentation is introduced in a unified framework to generate new data $\mathbf{x}' = \mathbf{T}(\mathbf{x})$:

$$\mathbf{x}^a = \mathbf{T}(\mathbf{x}^C) = \{\tau(\mathbf{x}_{i,1}^C), \ldots, \tau(\mathbf{x}_{i,f}^C), \ldots, \tau(\mathbf{x}_{i,n}^C)\},$$
$$\tau(\mathbf{x}_f^a) = (1 - r_u) \cdot \mathbf{x}_{i,f}^C + r_u \cdot \overline{\mathbf{x}}_{i,f}^C, \overline{\mathbf{x}} \sim \{N_i^h\}_{\{h \in H/h_i\}}, \tag{3}$$

where the new augmented data $\mathbf{x}^a$ is the combination of each feature $\tau(\mathbf{x}_{i,f}^I)$. $\tau(\mathbf{x}_{i,f}^C)$ is calculated from the linear interpolation of original feature $\mathbf{x}_{i,f}^C$ and augmented feature $\overline{\mathbf{x}}_{i,f}^C$. In addition, the augmented feature $\tilde{\mathbf{x}}$ is sampled from the neighborhood $\mathbf{N}(\mathbf{x})$ of $\mathbf{x}$. N_i^h is the k-NN neighborhood of the data i on the data of hospitals h. H is the set of the hospitals. H/h_i means remove the neighborhood of data i's hospitals. The combination parameter $r_u \sim U(0, p_U)$ is sampled from the uniform distribution $U(0, p_U)$, and p_U is the hyperparameter.

The cross-hospital augmentation generates new data by combining data i with its neighborhoods in different hospitals. It reduces the terrible influence of the missing data and improves the training data's divergence. Thus, the model learns a precise distribution to enhance the performance of our method. In addition, it works together with the loss function to align data from different hospitals, thus overcoming domain bias.

4.4. Architecture of BUPNN.
The proposed BUPNN does not require a unique backbone neural network. The multilayer perceptron network (MLP) is the backbone neural network.

In addition to this, a new network architecture is proposed to enhance generalizability. The proposed neural network architecture is shown in Figure 3.

A typical neural network model uses the network output directly to compute a supervised loss function. It may introduce undesirable phenomena such as overfitting. In this paper, similar to the paper [31], a manifold learning regularizer is proposed to suppress problems such as overfitting using the information in the latent space of the network. As shown in Figure 3, a complete neural network $F(\cdot, w_i, w_j)$ is divided into a latent network $f_i(\cdot, w_i)$ and an output network $f_j(\cdot, w_j)$. The latent network is a preprocessing network that resists the adverse effects of noise and missing value completion. The output network is a dimensional reduction network that maps the data in high-dimensional latent space to the data in low-dimensional latent space. The latent network $f_i(\cdot, w_i)$ maps the data x' input a network latent space and an output network $f_j(\cdot, w_j)$ further map it into the output space:

$$
\begin{aligned}
\mathbf{y_i} &= f_i(\mathbf{x}'_i, w_i), \\
\mathbf{y_j} &= f_i(\mathbf{x}'_j, w_i), \\
\mathbf{z_i} &= f_j(\mathbf{y_i}, w_j), \\
\mathbf{z_j} &= f_j(\mathbf{y_j}, w_j),
\end{aligned}
\tag{4}
$$

where $\mathbf{x}'_i$ and $\mathbf{x}'_j$ are the complementation and augmentation results of origin data x.

Neural networks have powerful fitting performance, but at the same time, there is a risk of overfitting. The typical L2 regularization method can reduce overfitting, but it only limits the complexity of the network without constraining the network in terms of the manifold structure of the data. For example, there is no way to guarantee the distance-preserving and consistency of the network mapping. For this reason, we design a manifold regularizer based on manifold learning to solve this problem, and thus improve the generalization performance of the model and its usability for actual data.

4.5. Loss Function of BUPNN. The loss function of BUPNN consists of two parts; one is the cross-entropy loss which uses label information, and the other is the manifold regularizer loss which uses hospital information and latent space information.

4.5.1. Manifold Regularizer Loss. Manifold regularizer loss handles the domain bias in different hospitals during the training phase and provides a manifold constraint to prevent overfitting. Inconsistent medical equipment and subjective physician diagnoses in different hospitals cause domain bias in data between hospitals. The manifold regularizer loss guides the mapping of the neural networks to be insensitive to hospitals, thus overcoming domain bias (shown in Figure 4). Therefore, the pairwise similarity between nodes is defined first. Considering the dimensional inconsistency of the latent space, we use the *t*-distribution with variable degrees of freedom as a kernel function to measure the point-pair similarity of the data:
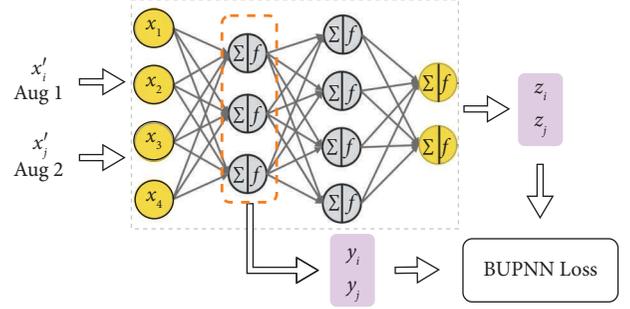
Figure 3: Architecture of BUPNN.
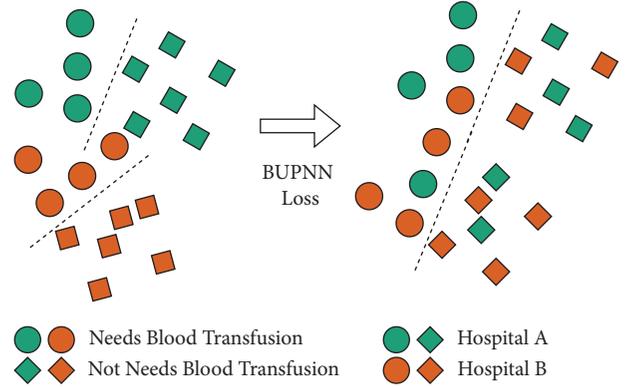
Figure 4: How manifold regularizer loss works. Data from the same hospitals are clustered near each other in latent space due to the significant domain bias possessed by the current data. The manifold regularizer loss guides the neural network model to reduce the domain bias by pulling in neighboring nodes across hospitals to mix data from different hospitals.

$$
\kappa(d, \nu) = \frac{\text{Gam}(\nu + 1/2)}{\sqrt{\nu \pi} \text{Gam}(\nu/2)} \left(1 + \frac{d^2}{\nu}\right)^{-\nu + 1/2},
\tag{5}
$$

where $\text{Gam}(\cdot)$ is the Gamma function, and the degree of freedom ν controls the shape of the kernel function. d is the Euclidean pairwise distance of node pairs.

Based on the defined pairwise similarity in a single space, we minimize the similarity difference between two spaces by fuzzy set cross-entropy loss (two-way divergence) [32] $\mathbf{D}(p, q)$:

$$
\mathbf{D}(p, q) = p \log q + (1 - p) \log(1 - q),
\tag{6}
$$

where $p \in [0, 1]$. Notice that $\mathbf{D}(p, q)$ is a continuous version of the cross-entropy loss. In BUPNN, equation (6) is used to guide the pairwise similarity of two latent spaces to fit each other.

Therefore, the loss function of the manifold regularizer is defined as follows:

$$
L_{\mathrm{D}} = \sum_{i,j}^{B} L(x_i, x_j)
$$

$$
L(x_i, x_j) = \begin{cases} \mathbf{D}\left(1, \kappa\left(d_{ij}^z, \nu_z\right)\right) & \text{if } \mathbf{x}_j = \tau(\mathbf{x}_i), \\ \mathbf{D}\left(\kappa\left(d_{ij}^y, \nu_y\right), \kappa\left(d_{ij}^z, \nu_z\right)\right) & \text{otherwise,} \end{cases}
\tag{7}
$$

where B is the batch size and $\mathbf{x}_j = \tau(\mathbf{x}_i)$ indicates whether $\mathbf{x}_j$ is the augmented data of $\mathbf{x}_i$. If $\mathbf{x}_j$ is the augmented data of $\mathbf{x}_i$, the loss pulls them together in the latent space; otherwise, the loss keeps a gap between them to preserve the manifold structure. The different degrees of freedom v_y and v_z in t-distribution are basic settings according to the dimensions of the space. Equation (7) describes the behavior of a manifold alignment that pairs data collected by different hospitals together in latent space to avoid the detrimental effects caused by domain bias:

$$
\begin{aligned}
d_{ij}^y &= d(y_i, y_j), \\
d_{ij}^z &= d(z_i, z_j),
\end{aligned} \tag{8}
$$

where d_{ij}^y and d_{ij}^z are the distance between data node i and node j in spaces $\mathbb{R}^{d_y}$ and $\mathbb{R}^{d_z}$.

4.5.2. Cross-Entropy Loss. The loss function of the manifold regularizer is essentially an unsupervised term that also requires the use of label information while training the network model. The cross-entropy loss function is introduced simultaneously:

$$
L_{\mathrm{CE}} = -\sum_{i=1}^{N} \left[l_i \ln\left(\sigma(z_i)\right) + (1 - l_i)\ln\left(1 - \sigma(z_i)\right) \right], \tag{9}
$$

where l_i is the label of data node i, $\sigma(z_i)$ is the output of the network model, and $\ln(\cdot)$ is the natural logarithm. When solving the classification subtask, l_i is the category label, and when solving the regression subtask, l_i is the probability label.

The loss function of BUPNN is

$$
L = L_D + \beta L_{\mathrm{CE}}, \tag{10}
$$

where β is a hyperparameter to balance L_D and L_{CE}.

4.6. Pseudocode and Complexity. BUPNN's pseudocode is shown in Algorithm 1. The BUPNN includes the initialization and training phases. In the initialization phase, the kNN neighborhood of every single data is discovered. The time complexity of initialization phases is $O(n^{1.14})$ [33], where n is the number of data. In the training phase, the complexity of the training phases is the same as that of artificial neural networks (ANN). BUPNN calculates the pairwise distance in a batch, so the complexity of training each batch is $O(|B|^2)$, where $|B|$ is the batch size. GPU can well accelerate the pairwise distance, so the training time consumption is the same as that of a typical ANN.

5. Experiments

5.1. Baseline Methods. In this Section, two subtasks, classification subtask and regression subtask, are defined. Seven state-of-the-art baseline classification and regression methods are chosen to discuss the relative advantages of BUPNN. The baseline approach is as follows.

5.1.1. K-Nearest Neighbor Classification/Regression Method (KNN) [37]. The K-nearest neighbor classification/regression method is a nonparametric statistical method. The KNN classification method outputs the prediction by the "majority vote" of its neighbors. The KNN regression method outputs the prediction by the average of its neighbors.

5.1.2. Decision Tree Classification/Regression Method (DT) [35]. A decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while, at the same time, an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches representing values for the attribute tested. The leaf node represents a decision on the numerical target. The root node is the topmost decision node in a tree that corresponds to the best predictor. Decision trees can handle both categorical and numerical data.

5.1.3. Random Forest Classification/Regression Method (RF) [34]. Random forests or random decision forests are ensemble learning methods for classification, regression, and other tasks that operate by constructing many decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct decision trees' habit of overfitting their training set.

5.1.4. Extremely Randomized Trees Classification/Regression Method (ET) [38]. Extremely randomized trees add a further step of randomization to random forest, while similar to ordinary random forests in that they are an ensemble of individual trees, there are two main differences: first, each tree is trained using the whole learning sample (rather than a bootstrap sample), and second, the top-down splitting in the tree learner is randomized. Furthermore, instead of computing the locally optimal cut-point for each feature under consideration (based on, e.g., information gain or the Gini impurity), a random cut-point is selected.

5.1.5. Support Vector Machine Classification/Regression Method (SVM) [39]. Support vector machine is a supervised learning model with associated learning algorithms that analyze data for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new standards to one category or the other, making it a nonprobabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space to maximize the width of the gap between the two categories. New models are then mapped into that space and predicted to belong to a class based on which side of the hole they fall.

Input: data: $\mathscr{X} = \{x_i\}_{i=1}^{|\mathscr{X}|}$, learning rate: η, epochs: E, batch size: B, β, v^z, network: f_θ, g_ϕ,
Output: graph embedding: $\{e_i\}_{i=1}^{|\mathscr{X}|}$.
(1) **while** $i = 0$; $i < E$; i ++ **do**
(2) $\mathbf{x}^{\mathrm{I}} = \mathbf{I}(\mathbf{x})$ ▷ # blood data complementation in equation (1)
(3)   **while** $b = 0$; $b < [|\mathscr{X}|/B]$; b ++ **do**
(4) $\mathbf{x}^{a1}, \mathbf{x}^{a2} = \mathbf{T}(\mathbf{x}^{\mathrm{I}}), \mathbf{T}(\mathbf{x}^{\mathrm{I}})$ ▷ # blood data augmentation in equation (3)
(5) $\mathbf{y}^{a1}, \mathbf{y}^{a2} \longleftarrow f_\theta(\mathbf{x}^{a1}), f_\theta(\mathbf{x}^{a2})$; ▷ # map input data into $\mathbb{R}^{d_y}$ space in equation (4)
(6) $\mathbf{z}^{a1}, \mathbf{z}^{a2} \longleftarrow f_\theta(\mathbf{y}^{a1}), f_\theta(\mathbf{y}^{a2})$; ▷ # map input data into $\mathbb{R}^{d_z}$ space in equation (4)
(7) $d_{ij}^y \longleftarrow d(y^{a1}, y^{a2})$; $d_{ij}^z \longleftarrow d(z^{a1}, z^{a2})$; ▷ #calculate distance in $\mathbb{R}^{d_y}$ and $\mathbb{R}^{d_z}$
(8) $S^y \longleftarrow \kappa(R(B_{ij})d_{ij}^y, v_y)$; $S^z \longleftarrow \kappa(d_{ij}^z, v_z)$; ▷ #calculate similarity in $\mathbb{R}^{d_y}$ and $\mathbb{R}^{d_z}$ in equation (5)
(9) $\mathscr{L}_{\mathrm{D}} \longleftarrow D(S^y, S^z)$; ▷ # calculate loss function in equation (10).
(10) $\theta \longleftarrow \theta - \eta\partial\mathscr{L}_{\mathrm{D}}/\partial\theta$, $\phi \longleftarrow \phi - \eta\partial\mathscr{L}_{\mathrm{D}}/\partial\phi$; ▷ # update parameters.
(11)   **end while**
(12) **end while**
(13) $z_i \longleftarrow f_\theta(g_\phi(x_i))$; ▷ # calculate the embedding result.

ALGORITHM 1: BUPNN algorithm.

5.1.6. Gradient Boost Classification/Regression Method (GB) [40]. Gradient boosting is a machine learning technique used in regression and classification tasks. It gives a prediction model in an ensemble of weak prediction models, typically decision trees. The resulting algorithm is called gradient-boosted trees. When a decision tree is a weak learner, it usually outperforms a random forest.

5.1.7. Adaptive Boost Classification/Regression Method (ADB) [36]. The adaptive boost algorithm obtains a strong learner by combining a series of weak learners and integrating these vulnerable learners' learning capabilities. Adaptive boost changes the weights of the samples based on the previous learners, increasing the importance of those previously misclassified samples and decreasing the weight of correctly classified samples so that the subsequent learners will focus on those misclassified samples. Finally, these learners are combined into strong learners by weighting.

5.1.8. Light Gradient Boosting Machine Classification/Regression Method (LightGBM) [41, 42]. LightGBM is a distributed gradient boosting framework based on the decision tree algorithm. LightGBM is designed with two main ideas in mind: (1) to reduce the use of data in memory to ensure that a single machine can use as much data as possible without sacrificing speed and (2) to reduce the cost of communication to improve the efficiency when multiple machines are in parallel, and achieve linear acceleration in computation. It can be seen that LightGBM was originally designed to provide a fast and efficient data science tool with a low memory footprint, high accuracy, and support for parallel and large-scale data processing.

5.2. Dataset Partitioning and Grid Search. Table 1 and Figure 2 provide basic information about the data. Five data partitioning schemes are provided in this study to provide a detailed comparison of the performance differences between the different schemes.

Three schemes are with data-complement, including COM-Mea, COM-Med, and COM-KNN as defined in equation (2). First, for COM-Mea, COM-Med, and COM-KNN, the missing values are complemented with a specific method. Following that, the training and testing sets are divided by 90%/10%. Two noncomplement schemes (NC-A and NC-B) are introduced to compare with the complement schemes. NC-A deletes all the data items with missing values, following the typical machine learning scheme. Following the data cleaning, NC-A divides the training and testing sets by 90%/10%. NC-B keeps the same training and testing set division as the data complement schemes and only removes all missing data from the training set and obtains a cleaner training set.

The models are evaluated with 10-fold cross-validation for all the training sets and determine the optimal parameters by grid search. For a fair comparison, we control the search space of all baseline methods to be approximately equal. The search space of the compared process is in Table 2.

5.3. Experimental Setup. We initialize the other NN with the Kaiming initializer. We adopt the AdamW optimizer [43] with learning rate of 0.02 and weight decay of 0.5. All experiments use a fixed MLP network structure, $f_{\theta,w}$: (−1, 500, 300, 80), g_ϕ: (80, 500, 2), where −1 is the features number of the dataset. The number of epochs is 400. The batch size is 2048. $v^y = 100$.

5.4. Comparison on Classification Subtask. Although the blood data for each patient is collected for accurate regression subtasks, it is equally important to predict whether a patient needs a blood transfusion. Especially in emergencies, indicating whether a patient needs a blood transfusion in the short term is more important than estimating the amount of blood used throughout the treatment cycle. Therefore, we first evaluate the performance of the proposed BUPNN on the classification subtask. Then, two evaluation metrics, classification accuracy (ACC) and area under the ROC curve (AUC), are used to compare the classifier's

TABLE 2: Details of grid search.

Methods	Abbreviation	Search space	Note
K-nearest neighbor	KNN	$Nei \in [1, 3, 5, 10, 15, 20]$, $L \in [10, 20, 30, 50, 70, 100]$	$Nei \longrightarrow$ neighborssize, $L \longrightarrow$ leaf.size
Random forest	RF	$NE \in [80, 90, 100, 110, 120, 130]$, $MSS \in [2, 3, 4, 5, 6, 7]$	$NE \longrightarrow$ boosted.trees.size, $MSS \longrightarrow$ samples.split.size
Decision tree	DT	$MSL \in [1, 2, 3, 5, 7, 10]$, $MSS \in [2, 3, 5, 7, 10, 15]$	$MSL \longrightarrow$ sample.size.inaleaf, $MSS \longrightarrow$ samples.split.size
Extra tree	ET	$MSL \in [1, 2, 3, 5, 7, 10]$, $MSS \in [2, 3, 5, 7, 10, 15]$	$MSL \longrightarrow$ sample.size.inaleaf, $MSS \longrightarrow$ samples.split.size
Support vector machine	SVM	$M \in [10, 50, 100, 300, 500]$, $T \in [1e^{-4}, 5e^{-4}, 1e^{-3}, 2e^{-3}, 3e^{-3}, 5e^{-3}]$	$NE \longrightarrow$ max iterations, $T \longrightarrow$ tolerance.for.stopping.criteria
Gradient boosting	GB	$NE \in [80, 90, 100, 110, 120, 130]$, $MSS \in [2, 3, 4, 5, 6, 7]$	$NE \longrightarrow$ boosted.trees.size, $MSS \longrightarrow$ samples.split.size
Multilayer perceptron	MLP	$L \in [2, 3, 4, 5, 6]$, $WD \in [0.1, 0.2, 0.3, 0.4]$	$L \longrightarrow$ number.of.layer, $WD \longrightarrow$ weight.declay
Adaptive boost	ADB	$NE \in [40, 50, 60, 70, 80, 90]$, $LR \in [0.8, 0.9, 1, 2, 5,]$	$NE \longrightarrow$ boosted.trees.size, $LR \longrightarrow$ learning.rate\end
Light gradient boosting machine	LGBM	$NE \in [21, 26, 31, 26, 41]$, $L \in [80, 90, 100, 110, 120, 130]$	$NE \longrightarrow$ boosted.trees.size, $L \longrightarrow$ leaf.size\end
Blood usage prediction neural network	BUPNN (ours)	$v^z \in [0.01, 0.03, 0.05, 0.07]$, $\beta \in [0.1, 0.2]$, $K \in [100, 200, 300]$	$NE \longrightarrow$ boosted.trees.size, $L \longrightarrow$ leaf.size\end

performance from multiple perspectives. The performance comparison of BUPNN and eight baseline methods is shown in Tables 3 and 4. In addition, the ROC curves of all the compared methods on different schemes are shown in Figure 5. The scatter plot the prediction of BUPNN for COM-Mea schemes are shown in Figure 6.

5.4.1. Data Complementation Delivers Performance Improvements.
From the performance results of NC-B, COM-Mid, COM-Mea, and COM-KNN in Tables 3 and 4, we observe that the schemes with data complementation yield better performance. We attribute the reasons to the more abundant training data provided by data complementation. Although the complemented data are imperfect, artificial intelligence models can still learn more helpful information.

Although the model trained with only clean data (NC-A) has the highest performance score, scheme NC-A only includes the clean data and cannot be conveniently generated to the actual data with missing values. However, missing values are unavoidable in real-world application scenarios, so data complementation is a better choice to improve the model's performance.

5.4.2. BUPNN Outperforms Almost All Baseline Methods.
From Tables 3 and 4, we observe that the proposed BUPNN has advantages over all the baseline methods. For the AUC metric, BUPNN has the advantage in all five schemes. However, the COM-MID scheme has the most significant benefits, outperforming the second-best method (LGBM) by 3.71%. For the ACC metric, BUPNN has the advantage in all five schemes. However, the COM-KNN scheme has the most significant benefits, outperforming the second-best method (LGBM) by 2.95%. BUPNN has a 1% average advantage over the second-best

method in both metrics. The reason is attributed to the fact that BUPNN is a neural network-based model, which performs better with richer data. In addition, the proposed manifold loss function can improve the model's generalization performance and thus enhance the performance of the testing set.

5.4.3. BUPNN Is Better at Handling Complemented Datasets.
From four schemes with the same testing set (NC-B, COM-Mid, COM-Mea, and COM-KNN) in Tables 3–5, we observe that BUPNN performs better than all the baseline methods when handling complemented datasets. We attribute this to the data augmentation of the proposed BUPNN model. Data augmentation generates new training data from the completed data and attenuates the effect of missing values, thus guiding BUPNN to learn a smooth manifold.

5.4.4. BUPNN Has Better Generalizability between Different Hospitals.
To evaluate the generalization performance of our proposed BUPNN and baseline methods, we tested the ACC with the data from various hospitals (Figure 5). The proposed BUPNN has a dominant performance of five hospitals out of a total of 6 hospitals and has a clear advantage over Shaoyifu Hospital. From Table 1 and Figure 1, we observe that the Shaoyifu Hospital has a relatively obvious domain bias. It has a minimum male-to-female rate, the top three average missing rates, and many outliers in its missing values. We argue that the domain bias influences the performance of the baseline methods, and the proposed BUPNN outperforms the baseline methods due to BUPNN overcoming the domain bias in it. Furthermore, the manifold regularizer loss function provides a good manifold constraint to improve the model's generalizability between different hospitals (as shown in Figure 4).

TABLE 3: Classification AUC comparison with the baseline methods, the best result is shown in bold. The second result is italicized. The brackets at the right end show how much BUPNN exceeds the optimal metrics in the other methods.

	KNN	RF	MLP	ET	SVM	GB	AdaB	LGBM	BUPNN
NC-A	0.8072	*0.9166*	0.8436	0.8443	0.8736	0.8949	0.9072	0.8881	**0.9229** (↑**0.0063**)
NC-B	0.7302	*0.8109*	0.7421	0.8178	0.7116	0.7214	0.8324	0.8119	**0.8349** (↑**0.0240**)
COM-mid	0.8009	*0.8526*	0.7764	0.8437	0.7435	0.8442	0.842	0.8508	**0.8843** (↑**0.0317**)
COM-mea	0.8054	0.8591	0.8399	0.8252	0.7553	0.8470	0.8562	*0.8630*	**0.8797** (↑**0.0167**)
COM-KNN	0.8033	0.8575	0.7912	0.8321	0.7739	0.8446	0.8526	*0.8620*	**0.8761** (↑**0.0141**)
Average	0.7894	*0.8593*	0.7992	0.8326	0.7716	0.8304	0.8581	0.8552	**0.8796** (↑**0.0202**)

TABLE 4: Classification ACC comparison with the baseline methods, the best result is shown in bold. The second result is italicised. The brackets at the right end show how much BUPNN exceeds the optimal metrics in the other methods.

	KNN	RF	MLP	ET	SVM	GB	AdaB	LGBM	BUPNN
NC-A	0.7113	0.8351	0.7367	0.8144	0.7732	0.8454	**0.8557**	0.8454	*0.8454* (↓*0.0103*)
NC-B	0.6539	0.7531	0.7512	0.7428	0.6235	0.7409	0.7557	0.7445	**0.7643** (↑**0.0086**)
COM-mid	0.7340	*0.8030*	0.7589	0.7734	0.5567	0.7931	0.7537	0.7783	**0.8177** (↑**0.0147**)
COM-mea	0.7340	0.7931	0.7546	0.7635	0.6305	0.798	0.7931	*0.8030*	**0.8177** (↑**0.0147**)
COM-KNN	0.7241	*0.7931*	0.7768	0.7685	0.6059	0.7734	0.7783	0.7734	**0.8226** (↑**0.0295**)
Average	0.7115	*0.7955*	0.7560	0.7725	0.6380	0.7902	0.7873	0.7890	**0.8135** (↑**0.0180**)
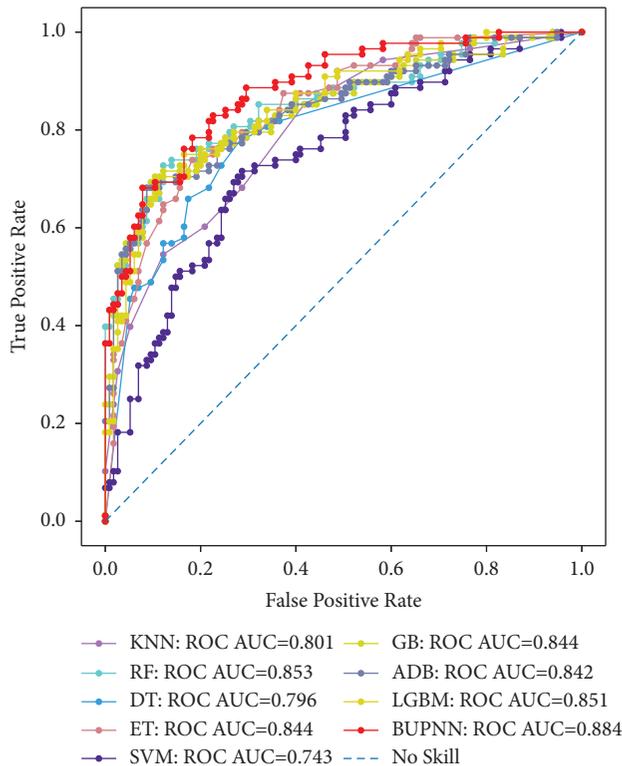
FIGURE 5: The ROC curve comparison for COM-Mea schemes, the missing values are filled with the median value. The closer the curve is to the upper left corner, the better the model's performance. The symmetry of the curve along the line from (0, 1) to (1, 0) indicates the balanced performance of the model.

- KNN: ROC AUC=0.801
- RF: ROC AUC=0.853
- DT: ROC AUC=0.796
- ET: ROC AUC=0.844
- SVM: ROC AUC=0.743
- GB: ROC AUC=0.844
- ADB: ROC AUC=0.842
- LGBM: ROC AUC=0.851
- BUPNN: ROC AUC=0.884
- No Skill

FIGURE 6: The scatter plot the prediction of BUPNN for COM-Mea schemes, the missing values are filled with the median value. The vertical coordinate is used to distinguish different samples, and the horizontal coordinate indicates the predicted value of the BUPNN model for the samples. A predicted value of less than 0.5 indicates that no blood is needed, and a value greater than 0.5 indicates that blood is needed. The color of the scatter indicates whether the prediction is correct or not.

5.5. Comparison on Regression Subtask. Then, we discuss the performance of BUPNN on the regression subtask. Finally, the regression model is asked to predict the total blood usage in the patient's treatment for advance scheduling at the blood center. The experimental setting of the regression subtask is the same as that of the classification subtask. The data are collected from the same patients, but the target variable is the total blood usage in the patient's treatment. The mean square error (MSE) metric is calculated to measure the performance of the seven methods in the five schemes. The performance comparison is shown in Table 6.

TABLE 5: Median classification ACC comparison with the baseline methods for different hospitals' all data, the best result are shown in bold. The second result is italicised. The brackets at the right end show how much BUPNN exceeds the optimal metrics in the other methods.

	KNN	RF	ET	MLP	GB	LGBM	SVMVC	ADB	BUPNN
DYang	**1.000**	**1.000**	0.571	0.723	0.786	0.714	0.857	0.929	**1.000** (↑**0.000**)
SYi	**1.000**	0.667	**1.000**	**1.000**	**1.000**	0.667	**1.000**	**1.000**	**1.000** (↑**0.000**)
SYiFu	0.796	0.778	0.815	0.735	0.556	0.593	0.630	0.778	**0.926** (↑**0.111**)
WLing	0.500	**1.000**	0.750	0.250	**1.000**	**1.000**	**1.000**	**1.000**	**1.000** (↑**0.000**)
ZEr	0.748	0.800	0.743	0.786	0.707	0.833	0.828	0.801	**0.834** (↑**0.001**)
Average	0.765	0.834	0.764	0.698	0.775	0.750	0.833	**0.854**	**0.909** (↑**0.055**)

TABLE 6: MSE comparison for all data, best results are shown in bold; results with clear advantage are shown in underline. The second result is italicised. The brackets at the right end show how much BUPNN exceeds the optimal metrics in the other methods.

	LR	SVM	MLP	ET	GB	RF	LGBM	Adab	BUPNN
NC-A	0.0095	0.0112	0.0121	0.0194	0.0093	*0.0090*	0.0090	0.0161	**0.0079** (↑0.0012)
COM-mid	*0.0042*	0.0075	0.0167	0.0103	0.0040	0.0043	0.0042	0.0175	**0.0038** (↑0.0004)
COM-mea	0.0042	0.0076	0.0076	0.0074	*0.0039*	0.0041	0.0044	0.0151	**0.0037** (↑0.0002)
COM-KNN	0.0041	0.0065	0.0082	0.0142	0.0040	*0.0039*	0.0043	0.0109	**0.0036** (↑0.0003)
Average	0.0055	0.0082	0.0111	0.0128	*0.0053*	0.0053	0.0054	0.0149	**0.0048** (↑0.0005)
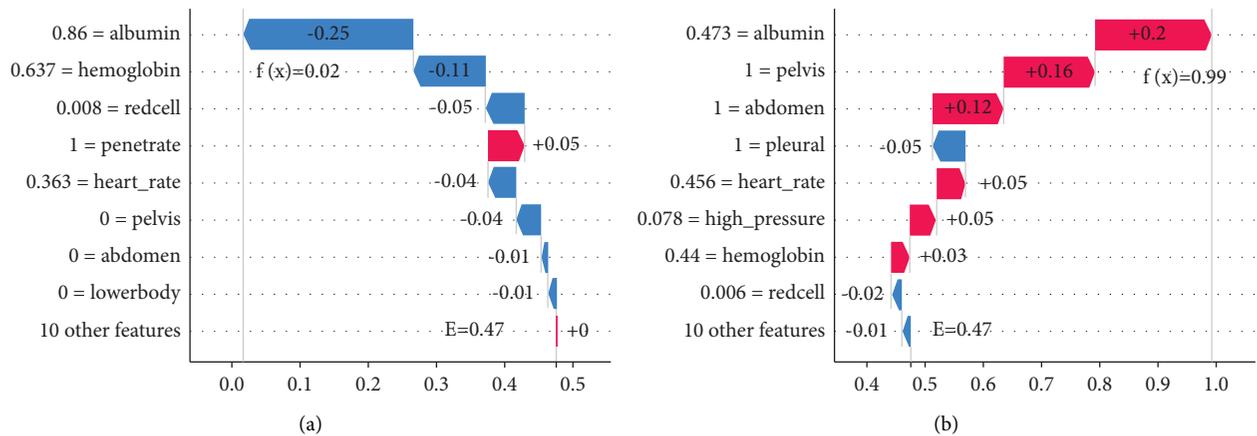
FIGURE 7: Explanation of the decision process of BUPNN for a single sample.

5.5.1. BUPNN Shows a Consistency Advantage on Regression Subtasks. From Table 6, we observe that the proposed BUPNN outperforms all the baseline methods on all four schemes. BUPNN has the most significant benefits for the COM-mid scheme, outperforming the second-best method (LGBM) by 0.004 (10.5%). The percentage of improvement shows that the advantage of BUPNN is more evident in the regression subtask. Furthermore, it indicates that the proposed BUPNN is more suitable for handling more difficult regression tasks and that BUPNNs have strong application potential when collecting richer data.

5.6. The Explanatory Analysis of BUPNN. Artificial neural network-based models are considered to have a strong performance. Still, their black-box nature leads to difficulty in interpretability, thus making it difficult for the model to be trusted by domain experts. On the other hand, the proposed BUPNN has stronger interpretability because the smoothness of its mapping leads to models that profound model

interpreters can easily interpret. An easy-to-use neural network interpretive tool is introduced to explain the decision process of BUPNN. The visual presentation of the interpretation results is shown in Figure 7.

In Figure 7, the decision processes of BUPNN for three samples are explained by "shap" tool. The vertical coordinates represent the clinical indicators, and the horizontal coordinates represent the predicted values of BUPNN. The images show the decision process of the model from bottom to top. Specifically, the model predicts $E = 0.47$ for an average patient when no information about the patient is observed, i.e., no transfusion is needed. We believe this is reasonable because only patients who are sufficiently harmed need blood transfusions to be treated. Next, using Figure 7(a) as an example, the BUPNN model observed additional patient information, such as "no injury to the lower body," "no injury to the abdomen," "no injury to the pelvis," and the "patient's heartbeat is not accelerated." This evidence made the BUPNN firm the original judgment that blood transfusion is unnecessary. Although the penetration
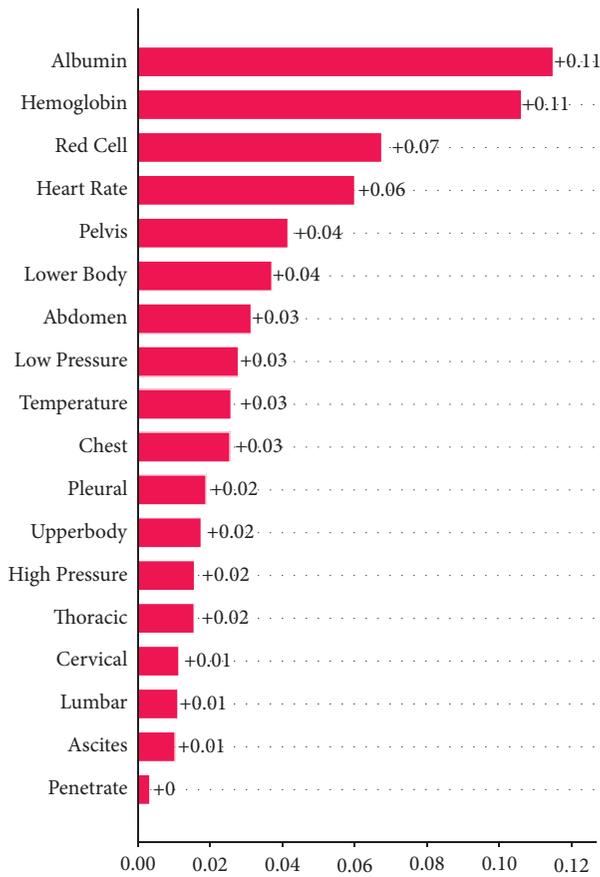
FIGURE 8: Important indicators about blood transfusion obtained by interpretable analysis of BUPNN.

injury raised the probability of transfusion, several essential features (e.g., albumin and hemoglobin) ultimately guided the model to predict $f(x) = 0.02$.

Figure 7(b) provides more examples of how to interpret the decision process of the BUPNN. These two examples show what kind of information needs to be observed by BUPNN that would predict a sample as needing a blood transfusion. Among them, we found some necessary signatures of the need for transfusion, such as the abdomen, pelvis, and pleural.

Next, we calculated the importance of all features in determining whether a blood transfusion is needed or not and displayed them in Figure 8 in the form of a bar chart.

6. Conclusions

In this paper, a neural network model-based blood usage predictor, called deep blood usage neural network (BUPNN), is proposed to serve the scheduling of blood supply in provincial blood centers. The proposed BUPNN receives clinical information from hospital patients and predicts whether a patient needs blood and the amount of blood used. The proposed BUPNN receives clinical data from hospital patients and indicates whether a patient needs blood and the blood consumption amount. The proposed BUPNN mainly solves the problem of predicting blood

usage with high availability and generalization in real-life situations. It can accomplish the prediction task well in different hospitals' missing values and domain bias. BUPNN proposes a manifold learning-based regularizer for the blood prediction problem to improve the model's generalization performance on data from different hospitals and enhance the model using data augmentation and data complementation.

Data Availability

The blood data used to support the findings this study are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] A. B. Cardillo, J. M. Heal, K. F. Henrichs et al., "Reducing the need for hla-matched platelet transfusion," *New England Journal of Medicine*, vol. 384, no. 25, pp. 2451-2452, 2021.

[2] H. Hu, X. Lai, C. Tan, N. Yao, and L. Yan, "Factors associated with in-patient mortality in the rapid assessment of adult earthquake trauma patients," *Prehospital and Disaster Medicine*, vol. 37, no. 3, pp. 299–305, 2022.

[3] O. P. S. P. Sickle Cell Anemia (Stop 2) Trial Investigators, "Discontinuing prophylactic transfusions used to prevent stroke in sickle cell disease," *New England Journal of Medicine*, vol. 353, no. 26, pp. 2769–2778, 2005.

[4] H. Wang, J. Umejiego, R. D. Robinson et al., "A derivation and validation study of an early blood transfusion needs score for severe trauma patients," *Journal of Clinical Medicine and Research*, vol. 8, no. 8, p. 591, 2016.

[5] T. C. Nunez, I. V. Voskresensky, L. A. Dossett, R. Shinall, W. D. Dutton, and B. A. Cotton, "Early prediction of massive transfusion in trauma: simple as abc (assessment of blood consumption)?" *Journal of Trauma and Acute Care Surgery*, vol. 66, no. 2, pp. 346–352, 2009.

[6] R. Schroll, D. Swift, D. Tatum et al. "Accuracy of shock index versus abc score to predict need for massive transfusion in trauma patients," *Injury*, vol. 49, no. 1, pp. 15–19, 2018.

[7] Y. Sun, S. Selvarajan, Z. Zang et al., "Protein classifier for thyroid nodules learned from rapidly acquired proteotypes," *medRxiv*, 2020.

[8] F. Saberi-Movahed, M. Mohammadifard, A. Mehrpooya et al., "Decoding clinical biomarker space of covid-19: exploring matrix factorization-based feature selection methods," *Computers in Biology and Medicine*, vol. 146, Article ID 105426, 2022.

[9] K. Zhou, Y. Sun, L. Li et al., "Eleven routine clinical features predict covid-19 severity uncovered by machine learning of longitudinal measurements," *Computational and Structural Biotechnology Journal*, vol. 19, no. 2001-0370, pp. 3640–3649,

2021, https://www.sciencedirect.com/science/article/pii/S2001037021002609.

[10] Z. Zhang, S. Cheng, L. Lu et al., "EVNet: an explainable deep network for dimension reduction," *in IEEE Transactions on Visualization and Computer Graphics*, 2022.

[11] A. Mehrpooya, F. Saberi-Movahed, A. N, M. Rezaei-Ravari, M. Eftekhari, and I. Tavassoly, "High dimensionality reduction by matrix factorization for systems pharmacology," *bioRxiv*, vol. 23, no. 1, 2021.

[12] F. Saberi-Movahed, M. Eftekhari, and M. Mohtashami, "Supervised feature selection by constituting a basis for the original space of features and matrix factorization," *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 1405–1421, 2020.

[13] Z. Zang, Y. Xu, Y. Geng, S. Li, and S. Z. Li, "Udrn: unified dimensional reduction neural network for feature selection and feature projection," 2022, http://arxiv.org/abs/2207.03809.

[14] G. M. Frankfurter, K. E. Kendall, and C. C. Pegels, "Management control of blood through a short-term supply-demand forecast system," *Management Science*, vol. 21, no. 4, pp. 444–452, 1974.

[15] G. C. Critchfield, D. P. Connelly, M. S. Ziehwein, L. S. Olesen, C. E. Nelson, and E. P. Scott, "Automatic prediction of platelet utilization by time series analysis in a large tertiary care hospital," *American Journal of Clinical Pathology*, vol. 84, no. 5, pp. 627–631, 1985.

[16] O. S. S. Filho, W. Cezarino, and G. R. Salviano, "A decision-making tool for demand forecasting of blood components," *IFAC Proceedings Volumes*, vol. 45, no. 6, pp. 1499–1504, 2012.

[17] O. S. S. Filho, M. A. Carvalho, W. Cezarino, R. Silva, and G. Salviano, "Demand forecasting for blood components distribution of a blood supply chain," *IFAC Proceedings Volumes*, vol. 46, no. 24, pp. 565–571, 2013.

[18] D. Kumari and A Wijayanayake, "An efficient inventory model to reduce the wastage of blood in the national blood transfusion service," in *Proceedings of the 2016 Manufacturing & Industrial Engineering Symposium (MIES)*, pp. 1–4, IEEE, Colombo, Sri Lanka, October 2016.

[19] S. M. Fortsch and E. A. Khapalova, "Reducing uncertainty in demand for blood," *Operations Research for Health Care*, vol. 9, pp. 16–28, 2016.

[20] F. Lestari, U. Anwar, N. Nugraha, and B. Azwar, "Forecasting demand in blood supply chain (case study on blood transfusion unit)," *Proceedings of the World Congress on Engineering*, vol. 2, pp. 16–28, 2017.

[21] T. Volken, A. Buser, D. Castelli et al. "Red blood cell use in Switzerland: trends and demographic challenges," *Blood transfusion*, vol. 16, no. 1, pp. 73–82, 2018.

[22] A. Drackley, K. B. Newbold, A. Paez, and N. Heddle, "Forecasting ontario's blood supply and demand," *Transfusion (Philadelphia)*, vol. 52, no. 2, pp. 366–374, 2012.

[23] R. Khaldi, A. El Afia, R. Chiheb, and R. Faizi, "Artificial neural network based approach for blood demand forecasting: fez transfusion blood center case study," in *Proceedings of the 2nd International Conference on Big Data*, pp. 1–6, Cloud and Applications, Association for Computing Machinery, Tetouan, Morocco, March 2017.

[24] L. Guan, X. Tian, S. Gombar et al., "Big data modeling to predict platelet usage and minimize wastage in a tertiary care system," *Proceedings of the National Academy of Sciences*, vol. 114, no. 43, pp. 11368–11373, 2017.

[25] N. Li, F. Chiang, D. G. Down, and N. M. Heddle, "A Decision Integration Strategy for Short-Term Demand Forecasting and Ordering for Red Blood Cell Components," *Operations Research for Health Care*, vol. 29, 2020.

[26] M. Motamedi, N. Li, D. G. Down, and N. M. Heddle, "Demand forecasting for platelet usage: from univariate time series to multivariate models," arXiv: Learning, 2021.

[27] C. Twumasi and J. Twumasi, "Machine learning algorithms for forecasting and backcasting blood demand data with missing values and outliers: a study of tema general hospital of Ghana," *International Journal of Forecasting*, vol. 38, no. 3, pp. 1258–1277, 2022.

[28] M. Abolghasemi, B. Abbasi, T. Babaei, and S. Z. Hosseinifard, "How to Effectively Use Machine Learning Models to Predict the Solutions for Optimization Problems: Lessons from Loss Function," 2021, https://arxiv.org/abs/2105.06618.

[29] Z. Zang, W. Wang, Y. Song et al., "Hybrid deep neural network scheduler for job-shop problem based on convolution two-dimensional transformation," *Computational Intelligence and Neuroscience*, vol. 2019, 2019.

[30] D. A. Van Dyk and X. L. Meng, "The art of data augmentation," *Journal of Computational & Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[31] Z. Zang, S. Li, D. Wu et al., "Dlme: Deep Local-Flatness Manifold Embedding," in *Computer Vision-ECCV 2022: 17th European Conference*, pp. 576–592, Springer, Berlin, Germany, 2022.

[32] S. Z. Li, Z. Zang, and L. Wu, "Deep Manifold Transformation for Dimension Reduction," 2020, https://arxiv.org/abs/2010.14831.

[33] L. McInnes and J. Healy, "Umap: uniform manifold approximation and projection for dimension reduction," 2018, https://arxiv.org/abs/1802.03426.

[34] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] L. I. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees. wadsworth," *Biometrics*, vol. 40, p. 358, 1984.

[36] C. C. Chang and C. J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[37] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, *Knn model-based approach in classification*, pp. 986–996, Lecture Notes in Computer Science, Springer, Berlin, Germany, 2003.

[38] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[39] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[40] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," *Advances in Neural Information Processing Systems*, vol. 12, 1999.

[41] Q. Meng, G. Ke, T. Wang et al., "A communication-efficient parallel algorithm for decision tree," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[42] G. Ke, Q. Meng, T. Finley et al., "Lightgbm: a highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[43] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, https://arxiv.org/abs/1711.05101, 2017.