IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier

Inpatient Length of Stay and Mortality Prediction Utilizing Clinical Time Series Data

JUNDE CHEN[1], MASON LI[2], Miles Milosevich[2], Tiffany Le[2], Andrew Bahsoun[2] and Yuxin Wen[2] (Member, IEEE)

[1]Schmid College of Science and Technology, Chapman University, Orange, CA 92866 USA
[2]Fowler School of Engineering, Chapman University, Orange, CA 92866 USA

Corresponding author: Yuxin Wen (e-mail: yuwen@chapman.edu).

This research was supported in part by the National Science Foundation under Grant No. 2246158.

ABSTRACT Electronic Health Records (EHRs), which include demographic information, clinical notes, vital signs, laboratory test results, and others, provide rich information for clinical outcome prediction. In this work, we propose a novel attention embedded residual long short-term memory (LSTM) fully Convolutional Network (FCN) to perform the clinical predictions of inpatients' length of stay (LoS) and mortality. The proposed model is uniquely composed of a convolutional neural network (CNN) layer, three residual blocks, an LSTM unit, an FCN module, and a self-attention module. This innovative architecture allows for comprehensive feature extraction, where the CNN and residual blocks enhance clinical data features, the FCN and LSTM separately extract spatial and temporal features, and the self-attention mechanism focuses on pertinent information while filtering out noise. By optimizing the loss function to address class imbalance and overfitting, our model ensures robust and accurate predictions. Experimental results demonstrate that the proposed model outperforms state-of-the-art methods, validating its effectiveness and feasibility in inpatient length of stay and mortality prediction.

INDEX TERMS Convolutional neural network, Long short-term memory, Self-attention mechanism, Electronic Health Records, Length of stay.

I. INTRODUCTION

HEALTHCARE is one of the most exciting frontiers in data mining and machine learning (ML) [1]. Owing to the widespread adoption of Internet and mobile technologies by hospitals and health insurance companies, Electronic Health Records (EHRs) have skyrocketed over the past decade [2]. EHRs contain rich text, visual and time series information, such as patients' diagnostic and medical history, demographic information, and laboratory test results, which are the primary source of patients' health status administration. In the past, EHR data were mainly used for managing the health status of patients. Nevertheless, finding suitable mathematical models among these measurements has the potential for accurate and early predictions of future clinical events. This can help clinicians make more effective medical decisions and promote the economic allocation of hospital resources. So, in recent years, there has been an increased interest in predictive analysis with EHRs.

Data-driven and ML methods have been employed for healthcare data analysis, such as mining signatures from event sequences [3], risk prediction with EHR [4], length of stay (LoS), and mortality predictions for inpatients [5], [6]. Naturally, LoS and mortality predictions are performed primarily with an interest in predicting possible outcomes, which are how long a patient can stay in the intensive care unit (ICU), and whether the patient will die or survive. Using a support vector machine (SVM) as a classifier, Cheng et al. [7] developed a data-driven model to predict the LoS of appendectomy patients and a five-day threshold is applied in their scheme, i.e., either within target when a patient's LoS is inclusively within five days or exceeding the target otherwise. Despite their impressive performance, their method performed the LoS prediction as a dichotomous classification task, and two classes were categorized. Alsinglawi et al. [8] proposed a new regressor architecture called staking regressor to predict LoS for patients diagnosed with heart failure, and their experimental findings indicated that their proposed staking regressor outperformed other methods, even deep learning (DL)-based regressors in their experiments. Nonetheless, their proposed staking regressor is an ensemble learning (EL) approach comprised of multiple regressors, which consumes more computational overhead than a single model. In another research, Bao et al. [9] trained seven ML models, including SVM, decision tree, random forest (RF),

gradients boosting, multiple layer perception, XGBoost, and light Gradient Boosting, to predict mortality or survival of patients during hospitalization. The light Gradient Boosting algorithm showed the best accuracy in their comparative experiments. Likewise, using three ML models, including artificial neural network (ANN), SVM, and RF, Lin et al. [10] performed mortality prediction for acute kidney injury patients in the ICU. Their experimental results indicated the superiority of the RF model compared to other well-known methods. Despite reasonably good findings reported in the literature, traditional ML methods suffer from some bottlenecks, such as reliance on manually designed features, risk of overfitting, lack of robustness, and low accuracy.

More recently, DL techniques, especially convolutional neural networks (CNN), which become preferred methods due to the automatic feature extraction capabilities and competitive performance, have been widely applied for tasks related to inpatients' LoS and mortality prediction [11], [12]. For example, Zolbanin et al. [11] trained four different DL architectures to predict patients' LoS in hospitals, and they confirmed that the CNN outperformed other comparison candidates. Ali et al. [12] proposed a multimodal multitasking DL model to predict both LoS and readmission for patients, and their model outperformed ensemble learning methods such as RF. By training a CNN model named ISeeU2, Caicedo-Torres et al. [13] performed the mortality prediction of patients inside the ICU. Their proposed approach outperformed the traditional baselines while providing enhanced interpretability compared to similar DL methods. More than that, the CNN-based DL method has also proved effective in references [14]–[18]. These research findings indicate the significance of associating DL methods in predicting LoS and mortality of patients in hospitals. Besides, Thakur et al. [19] proposed a fused convolutional neural network long short-term memory (CNN-LSTM) architecture for hemiplegic gait prediction with smart-phone sensor. Similarly, reference [20] leverages the attention mechanism with multihead convolutional neural networks and LSTM for online activity monitoring. Despite impressive performance obtained by these methods, the traditional CNNs they used encounter some bottlenecks, such as redundant network parameters, gradient vanishing risk, and weak generalization ability. In addition, these methods focus on human activity recognition and hemiplegia gait monitoring with complex model designs, reliance on smartphone sensors, and high computational resource requirements, which is impractical for LoS and mortality time-series prediction tasks. In the paper, we propose a novel attention embedded residual long short-term memory (LSTM) Fully Convolutional Network (FCN), termed as ARLF, to perform the clinical time series analysis tasks including the inpatients' LoS and mortality predictions. Concretely, the ARLF is primarily composed of a CNN layer, three residual blocks, an LSTM unit, a FCN module, and a self-attention module, where the CNN layer contains a 1×5 convolution layer, a 1×3 convolutional layer, and two pooling layers. Each residual block consists of two 1×3 convolutional layers, two batch normalization (BN) layers, and one shortcut connection. The convolution kernels in these three residual blocks are 32, 32, and 16, respectively. The clinical data features are extracted by the CNN layer and enhanced by the three residual blocks. Then, the data features are fed into the FCN layer for spatial feature extraction and the LSTM layer for extracting temporal features, respectively. After that, the extracted spatial features and temporal features are concatenated and a self-attention mechanism is embedded into the network to emphasize the features to focus on the final prediction. In brief, the key contributions of this study can be recapitulated below.

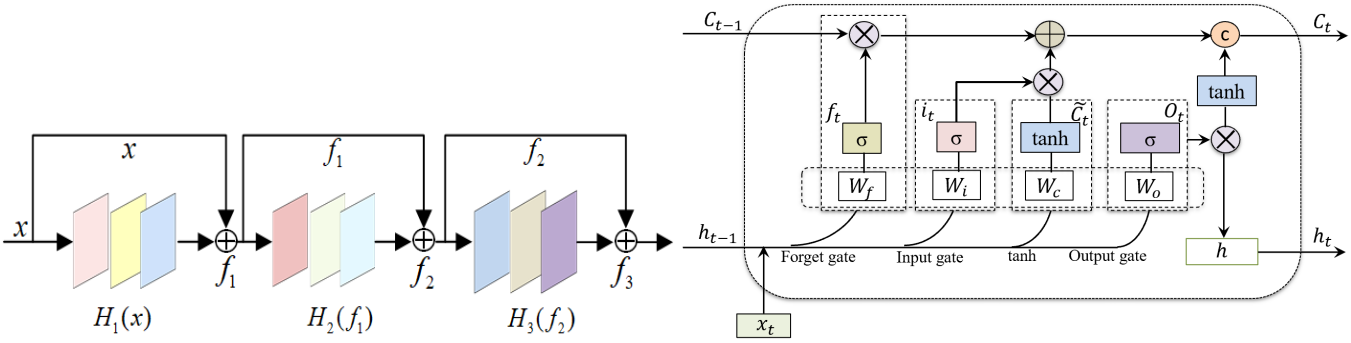
- An efficient attention embedded residual LSTM FCN model, which we termed ARLF, is proposed to perform the clinical time series analysis tasks, including the LoS and mortality predictions for patients in hospitals.
- After extracting the clinical data features from the CNN layer and residual blocks, we used the FCN and LSTM to extract spatial and temporal features separately. Then, a self-attention mechanism was embedded into the network to highlight the useful information while ignoring unwanted noises.
- We enhanced the traditional focal loss function to substitute for the cross-entropy (CE) loss function to address the class-imbalance issue, and an adaptive parameter optimization scheme was developed to determine weights automatically for multiple loss functions.

The rest of this paper is organized as follows. Section II summarizes the relevant work. Section III primarily discusses the methodology for the clinical time series LoS and mortality prediction. Later in Section IV, experiments for investigating the efficiency of the proposed method are presented, and a series of experiments are implemented along with the comparative analysis. Finally, Section V concludes the paper and suggests future work.

II. RELATED WORK

A. FEATURE ENGINEERING

For the clinical time series data, the past studies utilized the sub-timeframe and subsequence-based feature engineering methods such as the logistic regression [21]–[23], since the input sample matrix is very sparse caused by common missing observations. In the literature [1], six different sample statistic features including mean, standard deviation, maximum, minimum, skew, and number of measurements are calculated for any given time series (TS) sample data on seven different subsequences, including the full TS, first 50% of TS, first 25% of TS, first 10% of TS, last 50% of the full TS, last 25% of the full TS, and last 10% of the full TS. By this means, 7×6 features will be generated for each single time series sample. Despite reasonably good results, this approach primarily captures the statistical attributes of the data rather than the sequence dependencies. Its accuracy is consistently outperformed by DL methods such as Recurrent Neural Network (RNN). Motivated by the filters in convolutional neural networks, reference [2] recommended a filter-based feature



(a) The residual network

(b) The basic structural unit of LSTM

FIGURE 1. The residual network and LSTM unit.

engineering (FBFE) approach to apply convolution operation with an M × N filter matrix on the input sample and reshape the output matrix into a one-dimensional feature vector. Their experimental findings indicate that this approach can generate more fine-grained features faster. Nevertheless, their proposed approach is primarily used on the tree boost algorithms such as XGBoost and LightGBM, and it has not been effectively promoted. In addition, there is still room for improvement in the accuracy of the model.

B. RESIDUAL NETWORK

To improve the capabilities of deep neural networks (DNN), the most direct way is to increase the depth or width of the network. However, with the increase of depth and width, the network contains too many parameters and is hard to train. For this reason, the residual network was first introduced in the ResNet architecture by K. He et al. [24] to alleviate the problem of vanishing gradient and overfitting risk. The basic idea of a residual network is that residual mapping is more straightforward than learning a direct mapping, since the residual error is smaller than the direct learning error. This study proposes a residual network using 3 layers of residual blocks to predict inpatients' LoS and mortality, where the residual block is formed by a shortcut connection that skips the blocks of convolutional layers. The shortcut (residual) connection gives the network a better ability to remember historical information, which weights and fuses the input x into the output $H(x)$ of the residual block to gain the final output o, written as

$$o = Activation[x + H(x)] \tag{1}$$

where *Activation* acts for a function. In this research, the three output features of the residual blocks are designed in our networks, and the formulas are defined as follows,

$$f_1 = x + H_1(x) \tag{2}$$

$$f_2 = f_1 + H_2(f_1) = x + H_1(x) + H_2(x + H_1(x)) \tag{3}$$

$$\begin{aligned} f_3 = f_2 + H_3(f_2) = H_3(x + H_1(x) + H_2(x + H_1(x))) \\ + x + H_1(x) + H_2(x + H_1(x)) \end{aligned} \tag{4}$$

Among them, x is the input feature, and f_1, f_2, and f_3 indicate the output features. H_1, H_2, and H_3 represent a transformation operation such as convolution. Fig.1(a) depicts a 3-layer residual network architecture.

C. LONG SHORT-TERM MEMORY (LSTM)

LSTM is a variant of RNN that deals with time series problems in diverse fields [25], [26]. It overcomes the demerits of the gradient disappearance and the short-term memory feature of RNN, thereby realizing adequate storage and updating the information via an added internal gating mechanism. The basic structural unit of the LSTM model can be shown in Fig. 1(b), where f_t, i_t, c_t, and o_t denote the forgetting gate, cell state, input gate, and output gate, respectively. W and b represent the corresponding weight and bias matrices. *tanh* and σ separately indicate the hyperbolic tangent and sigmoid activation functions. The function of the forgetting gate is to determine which information to ignore and which to retain. The formula of forgetting gate output is expressed by

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{5}$$

where x_t and h_{t-1} denote the input of time t and the output of the hidden layer at time $t-1$. After the selection of the forgetting gate, the information enters the input gate, and the function of the input gate is to decide which parameters need to be updated and how to update them. The output of the input gate is formulated in Eqs. (6-8).

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{6}$$

$$\widetilde{C}_t = tanh(W_c * [h_{t-1}, x_t] + b_c) \tag{7}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{8}$$

where C_t implies the value of the current cell state, and $\odot$ symbolizes the inner product of two vectors. After the screening of input and forgetting gates, the information enters the output gate, and the function of the output gate is to

determine which information to output. The formula of the output gate can be written in Eqs. (9, 10).

$$O_t = \sigma\left(W_o * [h_{t-1}, x_t] + b_o\right) \tag{9}$$

$$h_t = O_t \odot \tanh(C_t) \tag{10}$$

where h_t is the output value of the current unit at time t, W_o and b_o denote the weight and bias of the output gate, respectively. LSTM models can infer temporal dependencies. However, inferring the long-term dependencies of long sequences is a challenging task for using LSTM models, and thus some recent studies have proposed using a hybrid attention mechanism to learn the long-term dependencies [27], [28].

D. SELF-ATTENTION MECHANISM

Note that the temporal features and spatial features are extracted by the LSTM and FCN modules, respectively. The next challenge is to identify which extracted features require more attention in understanding the feature weights for the final class prediction tasks. For this purpose, the proposed ARLF integrates an attention mechanism based on a scaled dot-product attention scheme. The attention mechanism is essentially a neural network within a neural network that learns to weigh portions of a sequence for relative feature importance [29]–[31]. In the model presented here, attention is a multiplicative self-attention mechanism and can be written in the following equation:

$$Atten(X, Y) = softmax(\frac{QK^T}{\sqrt{d_K}}) = softmax(\frac{XY^T}{\sqrt{d_K}}) \tag{11}$$

where the temporal feature extracted by the LSTM module is denoted as $X \in \mathbb{R}^{T\times H}$, and the spatial feature of FCN's output is $Y \in \mathbb{R}^{W\times C}$. Here, T, H, W, and C represent the LSTM timestep (data point), hidden units, width of features, and channels, respectively. Primarily, we define X as the query (Q), and Y as the key (K) and value (V). Thus, the attention scores can be extracted by dividing the dot product of Q and K by the square root of the K's dimension (d_K) to get a matrix, as shown in the above Eq. (11). Once the matrix is obtained, it can be passed to the softmax layer to extract the distribution of attention scores, or attention map, as expressed by:

$$\tilde{X} = Atten(X, Y) \cdot V \tag{12}$$

In Eq. (12), using the dot product of attention map $Atten(X, Y)$ and matrix V, we can obtain a wider global attention map $\tilde{X}$, which is a matrix that contains attention values related to the number of timesteps used in the model.

$$\tilde{Y} = \tilde{X} + Q \tag{13}$$

Ultimately, as in Eq. (13), by adding the $\tilde{X}$ and Q, we can obtain a self-attention feature map via residual connections. Through this process, the detailed knowledge on what global features are crucial can be obtained through the self-attention mechanism.

III. PROPOSED APPROACH

This section presents the proposed multi-view feature integration method of learning multivariate EHR time series for inpatient LoS and mortality predictions. The overall architecture is shown in Fig. 2.

A. DATA REPRESENTATION

For each subject n, we denote a set of D-dimensional multivariate time series as $X^{(n)} = [x_{t_1}^{(n)}, \cdots, x_{t_j}^{(n)}, \cdots, x_{t_{T_n}}^{(n)}]^T \in \mathbb{R}^{T_n\times D}$ where T_n indicates the length of time series in $t^{(n)} = [t_1, \cdots, t_j, \cdots, t_{T_n}]$ time points, and $x_{t_j}^{(n)} \in \mathbb{R}^D$ means the observation (also known as measurements) of all variables at the t_j-th time point. $x_{t_j}^{(n)}$ contains D features $x_{t_j,1}^{(n)}, x_{t_j,2}^{(n)}, \cdots, x_{t_j,D}^{(n)}$ and $x_{t_j,d}^{(n)}$ denotes the observation of the d-th variable of $x_{t_j}^{(n)}$. Let $s_{t_j}^{(n)}$ be the time-stamp of observation $x_{t_j}^{(n)}$, and we assume that the first observation is at the time point $t = 0$, i.e., $s_1 = 0$. Owing to irregular sampling, the time intervals between different timestamps may not be the same. To effectively denote the missing values in $x_{t_j}^{(n)}$, we introduce a masking vector $l_{t_j,d}^{(n)} \in \{0, 1\}$, which forms the time series $L^{(n)}$, to represent whether the variable is missing at time step t.

$$l_{t_j,d}^{(n)} = \begin{cases} 0, \ \textit{If } x_{t_j,d}^{(n)} \textit{ is not observed} \\ 1, \ \textit{Otherwise} \end{cases} \tag{14}$$

In most cases, some features are continuously missing over a period of time, and the trend is very evident. We follow the fact: when the data lacks subsequent observations, the impact of the last observation is big while the impact of distant observations is small [32]. Therefore, for the missing value of a certain variable, we use the last observation value and normal value (e.g., normal body temperature 37°C) to impute the missing $m_{t_j,d}^{(n)}$, which forms the imputation sequence of $M^{(n)}$. Mathematically, the missing value is imputed by

$$m_{t_j,d}^{(n)} = \begin{cases} x_{t_{j-1},d}^{(n)}, & t_j > 1, \ l_{t_{j-1},d}^{(n)} = 0 \\ N_d^{(n)}, & t_j = 1, \ l_{t_{j-1},d}^{(n)} = 0 \\ 0, & others \end{cases} \tag{15}$$

where $x_{t_{j-1},d}^{(n)}$ and $N_d^{(n)}$ represent the last observation value and normal value, respectively.

In this study, we frame the LoS prediction task as a classification problem with 10 classes/buckets, which are one for ICU stays shorter than a day, seven day-long classes for each day of the first week, one for stays of over one week but less than two, and one for stays of over two weeks, respectively [1]. Besides, referring to Y. Hu et al. (2020) work [2], a filter-based feature engineering approach inspired by filters in CNNs is used in our scheme. We take the time series samples after one-hot encoding and normalization processing as input. Then, the convolution operation with an $M \times N$ filter matrix is applied to the input samples, and the output matrix is reshaped into a one-dimensional feature vector. In this manner, more fine-grained features are generated at a faster speed and input to the proposed ARLF model for the

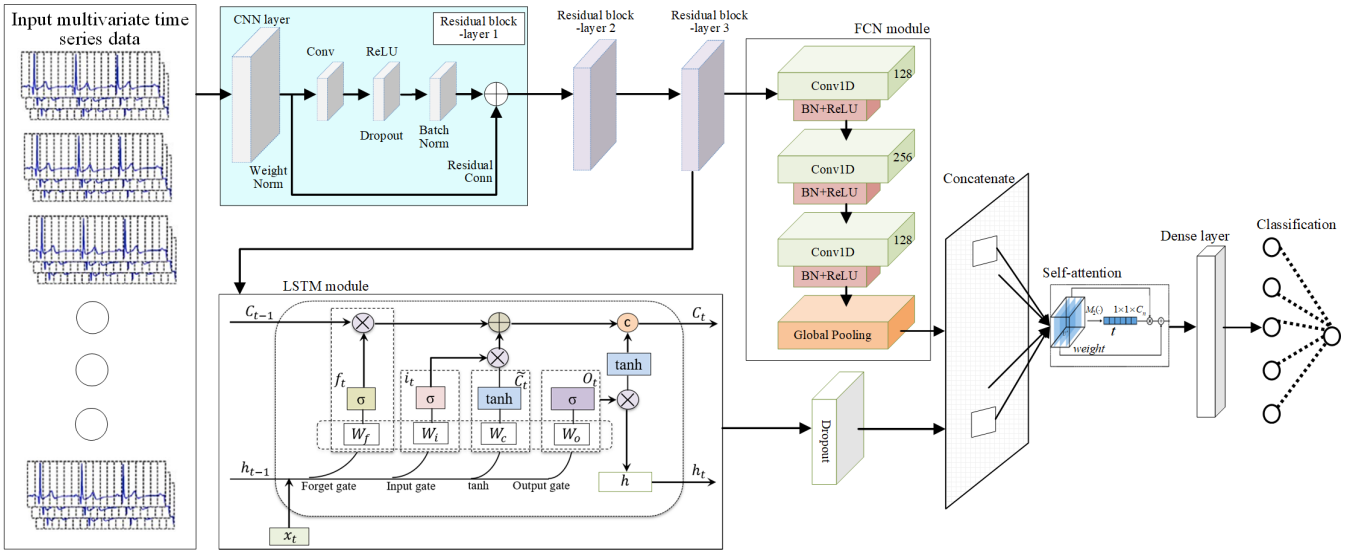


FIGURE 2. Architecture of the proposed ARLF model.

TABLE 1. The major parameters of the proposed model.

Module (layer)	Input shape	Filter no.	Kernel size	Output shape	Total Params	Repeat times
InputLayer	(None, None, 76)	-	-	(None, None, 76)	-	1
Masking	(None, None, 76)	-	-	(None, None, 76)	-	1
Bid-LSTM	(None, None, 76)	-	-	(None, None, 16)	18,240	9
Conv1D	(None, None, 16)	-	-	(None, None, 16)	34,000	3
BN layer	(None, None, 16)	-	-	(None, None, 16)	832	3
Add (shortcut conn)	[(None, None, 16)]×2	-	-	(None, None, 16)	-	1
Conv1D	(None, None, 16)	-	-	(None, None, 32	6,240	3
Add (shortcut conn)	[(None, None, 32)]×2	-	-	(None, None, 32)	0	1
Activation	(None, None, 32)	-	-	(None, None, 32)	-	1
Conv1D	(None, None, 32)	-	-	(None, None, 64)	24,768	3
Add (shortcut conn)	[(None, None, 64)]×2	-	-	(None, None, 64)	0	1
Activation	(None, None, 64)	-	-	(None, None, 64)	-	1
Conv1D	(None, None, 64)	-	-	(None, None, 32)	15,456	3
Add (shortcut conn)	[(None, None, 32)]×2	-	-	(None, None, 32)	0	1
Activation	(None, None, 32)	-	-	(None, None, 32)	-	1
Conv1D	(None, None, 32)	-	-	(None, None, 128)	295,424	3
BN layer	(None, None, 128)	-	-	(None, None, 128)	2,048	3
LSTM	(None, None, 32)	-	-	(None, 16)	3,136	1
Activation	(None, None, 128)	-	-	(None, None, 128)		1
Dropout	(None, 16)	-	-	(None, 16)	0	1
GAP1D	(None, 128)	-	-	(None, 128)		
Concatenate	[(None, 16),(None, 128)]	-	-	(None, 144)	0	1
Attention layer	(None, 144)	-	-	(None, 144)	20,880	1
Softmax	(None, 144)	-	-	(None, 144)	0	1
Lambda	[(None, 144)]×2	-	-	(None, 144)	0	1
Dense	(None, 144)	-	-	(None, 10)	1,450	1

LoS and mortality predictions of patients in hospitals.

B. ARLF MODEL

As indicated previously, the DNN's training becomes more complex with the increase of network depth due to the vanishing gradients and degradation problems. The solutions to

both these problems can be solved by using Resnet blocks [24]. From this perspective, the paper proposes the ARLF, an attention embedded residual LSTM FCN, to predict the in-hospital LoS for patients. The ARLF model with fused residual blocks is shown in Fig. 2, which primarily consists of a CNN layer, three residual blocks, an LSTM unit, an FCN module, and a self-attention module. The CNN layer contains a 1×5 convolution layer, a 1×3 convolutional layer, and two pooling layers, and each residual block contains two 1×3 convolutional layers, two batch normalization (BN) layers, and one shortcut connection. The convolution kernels in these three residual blocks are 32, 32, and 16, respectively. The clinical data features are extracted by the CNN layer and enhanced by the three residual blocks. Then, the data feature matrices are separately input to the FCN layer for spatial feature extraction and input to the LSTM layer for extracting temporal features. After that, the extracted spatial features along with temporal features are concatenated. A self-attention mechanism is embedded into the network to highlight the useful features while suppressing unwanted noises for the final classification prediction tasks of inpatients' LoS and mortality.

More specifically, the FCN block of the proposed ARLF comprises a one-dimensional kernel convolutional layer, followed by a BN layer and a ReLU activation layer. Three of these blocks are included in the ARLF model, and the number of convolutional kernels (filters) is 128, 256, and 128 without striding, and the filter sizes of 8, 5 and 3, respectively. There is no pooling operation to prevent over-fitting, and the BN layer is employed to accelerate the convergence speed and enhance the model's generalization ability. After being processed by the FCN block, the extracted features are fed into a global average pooling (GAP) [33] layer to create one feature map for each category and take the average of each feature map. Usually, the extracted features are input into a completely linked layer. By using a GAP layer, the model's parameters can be reduced significantly. Subsequently, the spatial features extracted by FCN and temporal features extracted by LSTM are concatenated and input into a self-attention layer to perform adaptive feature recalibration. As mentioned earlier, the attention mechanism is essentially a neural network within a neural network that learns the importance of features. Correspondingly, the self-attention mechanism used in this study is modeled as a feed forward neural network such that

$$e_{ij} = tanh\left(\theta_h h_{ij} + b_h\right),\ e_{ij} \in [-1, 1] \tag{16}$$

where h_{ij} refers to the output of the hidden layer, θ terms denote the weight matrices, and b terms denote the bias vectors. Then, the relative importance of each hidden state or attention weight λ_{ij} is scaled in a [0, 1] interval using the softmax function, which produces a vector "alignment score" weighting the importance of the individual parts of the batched input sequences, expressed as

$$\lambda_{ij} = softmax(e_{ij}) = \frac{exp(e_{ij})}{\sum_{j=1}^{t} exp(e_{ij})},\ \sum_{j=1}^{t} \lambda_{ij} = 1 \tag{17}$$

Finally, the context vector c_i can be computed as the weighted sum of the hidden states with the calculated attention weights:

$$c_i = \sum_{j=1}^{t} \lambda_{ij} h_{ij} \tag{18}$$

which can be seen as a high-level representation of the input sequence. In this manner, the concatenated spatial and temporal (spatio-temporal) features are weighted and input into the densely-connected (DC) layer with the Softmax activation function to get the final class prediction results. Fig. 2 portrays a structural schematic diagram of the proposed ARLF model, and the major parameters are summarized in Table 1.

C. OPTIMIZATION OF THE ARLF

To alleviate the class imbalance problem, Lin et al. [34] introduced a focal loss (FL) function that assigns different weights to negative and positive samples. The formula of the FL function is expressed by

$$\mathcal{L}_{FL} = -\alpha_K \left(1 - p\left(k \,|x\right)\right)^{\varphi} log(p\left(k \,|x\right)) \tag{19}$$

where α_K and φ separately denote the hyper-parameters of weighting and modulating factors, K indexes the number of classes, and $p\left(k \,|x\right)$ implies the probability that a specific sample x is predicted to belong to class k. The classical FL function is designed to address binary classification issues in the object detection field. However, as previously stated, the in-hospital LoS prediction for patients belongs to a multi-class problem. To overcome this challenge, the traditional FL function is improved in our scheme to make it suitable for the multi-class problem. The enhanced FL (EFL) function is formularized by

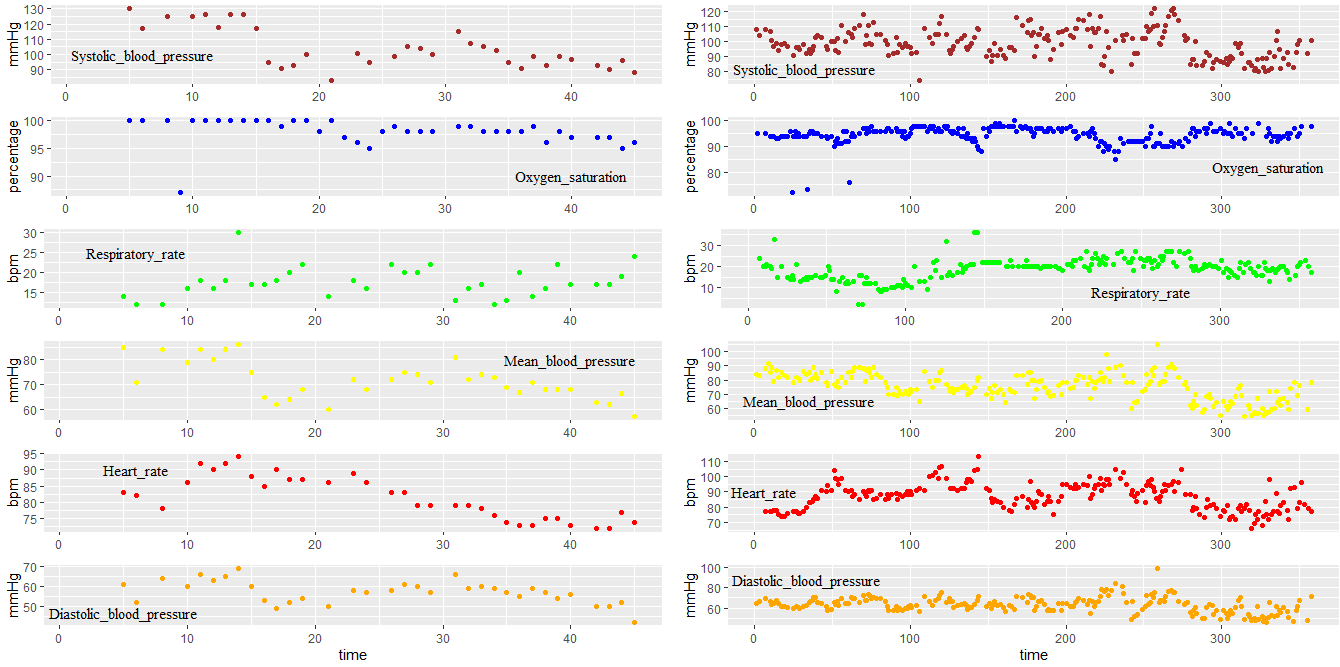
$$\mathcal{L}_{EFL} = -\sum_{k=1}^{K} \alpha_K (1 - p(k \,|x))^{\varphi} q(k \,|x) log(p(k \,|x)) \tag{20}$$

$$\alpha_K = count(x)/count(x \in k) \tag{21}$$

$$q(k \,|x) = \begin{cases} 0, & k \neq y \\ 1, & k = y \end{cases} \tag{22}$$

where x means the sample, and y indicates the ground-truth class. Additionally, considering the overfitting to noisy labels in multi-class problems, a Symmetric Cross-Entropy (SCE) loss function that uses a noise-robust counterpart Reverse Cross-Entropy (RCE) boosts Cross-Entropy (CE) symmetrically is proposed by [35] to suppress the overfitting or underfitting risk on some classes. The SCE loss function is written as

$$\mathcal{L}_{RCE} = -\sum_{k=1}^{K} p\left(k \,|x\right) logq\left(k \,|x\right) \tag{23}$$



(a) Short LoS sample by vital sign measures

(b) Long LoS sample by vital sign measures

FIGURE 3. Time series data sample from the MIMIC III v1.4 database.

$$\mathcal{L}_{SCE} = \mathcal{L}_{CE} + \mathcal{L}_{RCE} = -\sum_{k=1}^{K} q\left(k\,|x\right) logp\left(k\,|x\right) - \sum_{k=1}^{K} p\left(k\,|x\right) logq\left(k\,|x\right) \tag{24}$$

where $\mathcal{L}_{CE}$ and $\mathcal{L}_{RCE}$ represent the CE and the RCE losses, respectively. Referring to the above analysis, we propose an integrated loss function that leverages the merits of the EFL and SCE functions to optimize the proposed network. Considering that only the result of one loss function can be updated during backpropagation, the two different loss functions are fused into a joint loss function, and the most-used weighted sum approach is employed in our scheme. The weighted total loss function can be defined by

$$\mathcal{L}_{ES}^{(s)} = \sum_{j=1} \lambda_j^{(s)} L_j^{(s)} = \lambda_1^{(s)} \mathcal{L}_{EFL}^{(s)} + \lambda_2^{(s)} \mathcal{L}_{SCE}^{(s)} \tag{25}$$

where s indicates the s-th epoch of training, and λ is the weight hyperparameter that controls the ratio between two losses. The training performance of the model heavily relies on the assigned weights between losses, while manually tuning these weights is undoubtedly a challenging task and consumes significant costs. Therefore, we included the loss weights in the definition of the loss function itself and developed a self adaptive way to update the loss weights, thereby managing changes internally. The formula of loss weight update is written by

$$\lambda_j^{(s+1)} \leftarrow \lambda_j^{(s)} - \gamma \nabla_{\lambda_j} \mathcal{L}_{grad} \tag{26}$$

In Eq. (26), $\mathcal{L}_{grad}$ represents the gradient loss, which is used to describe the loss caused by the loss weight λ_j; γ is a constant hyperparameter. The gradient loss $\mathcal{L}_{grad}$ can be calculated by

$$\mathcal{L}_{grad}\left(s, \lambda_j^{(s)}\right) = \sum_j \left| G_j^{(s)} - \overline{G^{(s)}} \times \left[\mu_j^{(s)}\right]^{\alpha} \right|_1 \tag{27}$$

$$G_j^{(s)} = \left\| \nabla_\theta \lambda_j^{(s)} \mathcal{L}_j^{(s)} \right\|_2, \ \mu_j^{(s)} = \frac{\mathcal{L}_j^{(s)} / \mathcal{L}_j^{(0)}}{E_{task}\left[\mathcal{L}_j^{(s)} / \mathcal{L}_j^{(0)}\right]} \tag{28}$$

Among them, $G_j^{(s)}$ represents the gradient normalization value of the j-th loss function in the s-th epoch of training, which is computed by the L2 norm of the weighted loss gradient; $\overline{G^{(s)}}$ is the average gradient normalization of all the losses in the s-th epoch of training; $\mu_j^{(s)}$ denotes the relative training speed of the j-th loss function in the s-th epoch of training. In brief, the loss weight is used as an optimization parameter in this scheme and the $\mathcal{L}_{grad}$ of loss weights is established at each epoch of the update. The initial weight parameters of the $\mathcal{L}_{EFL}$ and $\mathcal{L}_{SCE}$ are both set to 0.5 in the networks, and the update of $\mathcal{L}_{grad}$ is implemented at each epoch of training. Algorithm 1 summarizes the detailed training procedure of the proposed method.

IV. EXPERIMENTS

In this section, numeric experiments are conducted on the benchmarking MIMIC-III v1.4 dataset to investigate the performance of the proposed approach. The detailed data transformation and preprocessing of the MIMIC-III v1.4

Algorithm 1: The detailed training procedure of the proposed method.

Input: The training samples $T = \{s_1, s_2, ..., s_n\}$ where $i = 1, 2, ..., n$, $s \in R^n$
Output: Obtain the best model parameters θ_{best}.
Randomly initialize the model parameters $\theta \in R^d$;
while *not done* **do**
  Generate the predictions using T sample set;
  Evaluate the loss $L_{EFL}(\theta)$ and $\nabla_\theta L_{EFL}(\theta)$; // $L_{EFL}(\theta)$ refers to Eq.(20);
  Evaluate the loss $L_{SCE}(\theta)$ and $\nabla_\theta L_{SCE}(\theta)$; // $L_{SCE}(\theta)$ refers to Eq.(24);
  Calculate the total loss $L_{ES}(\theta)$; // L_{ES} refers to Eq.(25), and initial λ_j is set to 0.5;
  Evaluate the gradient loss $L_{grad}(\lambda_j)$ and $\nabla_{\lambda j} L_{grad}(\lambda_j)$ using T samples; // refer to Eq.(27);
  Calculate adapted parameters using gradient descent;
  Continuously update the model parameters $\theta \leftarrow \theta - \eta \nabla_\theta L_{ES}(\theta)$ using T sample set. // η is the learning rate
end while

dataset are introduced in Section IV.*A*. Then, we evaluate the accuracy of the proposed ARLF compared to state-of-the-art (SOTA) methods. Subsequently, we assess the efficacy of fused modules and optimized loss function for the proposed approach via ablation study.

A. DATASET DESCRIPTION AND PREPROCESSING

MIMIC, short for the Medical Information Mart for Intensive Care, is a large database of clinical records for patients admitted to the Beth Israel Deaconess Medical Center (BIDMC). All the data are de-identified, where patient identifiers are removed according to the Health Insurance Portability and Accountability Act (HIPAA) Safe Harbor provision. The MIMIC database contains a wide range of patient records, such as patients' demographic information, laboratory test results, medication orders, free-text notes authored by clinicians, procedures and diagnoses, etc. In this study, we used the MIMIC-III v1.4 [36], released in September 2016. This database contains a cohort of 46,520 unique patients with 58,976 admissions. We followed [1] to extract clinical time series. Each sample includes the LoS (stay hours) and 17 vital sign indicators such as diastolic blood pressure (normal 60-90mmHg), Eye opening (Glascow coma scale, GCS), Scale total (GCS total, score), Verbal response (normal 5, no response 1), Glucose (normal <100 mg/dL), heart rate (normal 60-100 times/minute), mean blood pressure (normal 70-105mmHg), oxygen saturation (normal 95%-100%), respiratory rate (normal 12-20 times/minute), systolic blood pressure (normal 90-140mmHg), temperature, among others. Our training dataset includes 2,000 samples, the validation dataset includes 1,000 samples, and the testing dataset includes 508 samples for the LoS prediction. Fig. 3 portrays the time series distribution of vital sign measurement sample data, where Fig. 3(a) shows a time series trend of vital signs for a patient with short LoS (shorter than a day, class 1), and Fig. 3(b) depicts a time series trend of vital signs for a patient with long LoS (over one week, class 8).

As seen in Fig. 3, though the vital sign measure indicators can reflect the length of stay for inpatients to a certain extent, the characteristics are not very significant when observing these indicators directly from the original data. Therefore, after data transformation and preprocessing, the proposed method is used to perform the class prediction of LoS for inpatients on the MIMIC-III v1.4 dataset, and the influential SOTA methods are selected to compare models. To scientifically and objectively evaluate the models, the standard measure metrics are utilized to investigate the performance of the LoS and mortality predictions using different methods. The detailed contents are described in subsequent sections.

B. EXPERIMENTAL SETUP AND EVALUATION METRICS

All methods involved in this paper are developed using Python 3.6 deep learning framework, where the commonly-used ML libraries including Keras, Scikit-learn, Matplotlib, and TensorFlow are utilized and sped up by a graphics processing unit (GPU). The experiments are performed on a server with an AMD EPYC 7502P 32-Core Central Processing Unit (CPU) @ 2.50 GHz, 32 GB memory, and RTX A6000 GPU.

To investigate the performance of the model for LoS and mortality predictions, i.e., multi-label and binary classification tasks, we use the area under the receiver operating characteristic (ROC) curve (ROC-AUC) and area under the precision–recall curve (PR-AUC) as the evaluation metrics. Besides that, well-known metrics such as *Accuracy* (*Acc*), *Precision* (*Pre*), *Recall* (*Rec*), and F_1-*Score* (F_1) are also utilized to measure the performance of the models for the LoS and mortality predictions, which can be computed using the following equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{29}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{30}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{31}$$

$$F1 = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \tag{32}$$

where true positive (*TP*) and true negative (*TN*) refer to the positive data label and negative data label that are predicted correctly. False positive (*FP*) and false negative (*FN*) refer to the negative data label and positive data label that are predicted incorrectly. Tables III-VIII present the results.

TABLE 2. The LoS prediction performance on the training and validation sets.

No.	Methods	Training set						Validation set					
		ROC-AUC	PR-AUC	Acc	Pre	Rec	F1	ROC-AUC	PR-AUC	Acc	Pre	Rec	F1
1	XGBoost	0.989	0.929	0.926	0.926	0.926	0.927	0.822	0.434	0.378	0.421	0.378	0.398
2	LightGBM	0.970	0.856	0.827	0.843	0.827	0.834	0.817	0.434	0.366	0.427	0.366	0.394
3	LSTM	0.859	0.456	0.489	0.471	0.489	0.479	0.711	0.337	0.305	0.302	0.305	0.303
4	GRU	0.872	0.405	0.465	0.407	0.465	0.434	0.733	0.346	0.315	0.323	0.315	0.318
5	ClinicNet	0.868	0.458	0.523	0.505	0.523	0.514	0.696	0.298	0.276	0.336	0.276	0.303
6	Transformer	0.815	0.416	0.413	0.435	0.413	0.424	0.793	0.392	0.392	0.310	0.392	0.346
7	Proposed ARLF	0.862	0.567	0.565	0.565	0.565	0.556	0.801	0.402	0.401	0.421	0.401	0.411

TABLE 3. The test results of different methods for LoS prediction, (.) denote the ranking.

No.	Methods	Testing set						Average rank	Time (s)
		ROC-AUC	PR-AUC	Acc	Pre	Rec	F1		
1	XGBoost	0.813 (3)	0.329 (4)	0.307 (4)	0.329 (4)	0.307 (4)	0.317 (4)	3.833	121.146
2	LightGBM	0.829 (2)	0.355 (2)	0.338 (3)	0.358 (3)	0.338 (3)	0.347 (2)	2.500	151.887
3	LSTM	0.753 (6)	0.247 (6)	0.195 (7)	0.224 (6)	0.195 (7)	0.208 (7)	6.500	79.167
4	GRU	0.776 (5)	0.255 (5)	0.220 (5)	0.209 (7)	0.220 (5)	0.214 (6)	5.500	73.793
5	ClinicNet	0.729 (7)	0.245(7)	0.211 (6)	0.257 (5)	0.211 (6)	0.231 (5)	6.000	75.727
6	Transformer	0.806 (4)	0.352 (3)	0.363 (2)	0.376 (2)	0.363 (2)	0.322 (3)	2.667	119.135
7	Proposed ARLF	0.936 (1)	0.417 (1)	0.572 (1)	0.507 (1)	0.572 (1)	0.538 (1)	1.000	89.038

C. RESULTS AND DISCUSSION

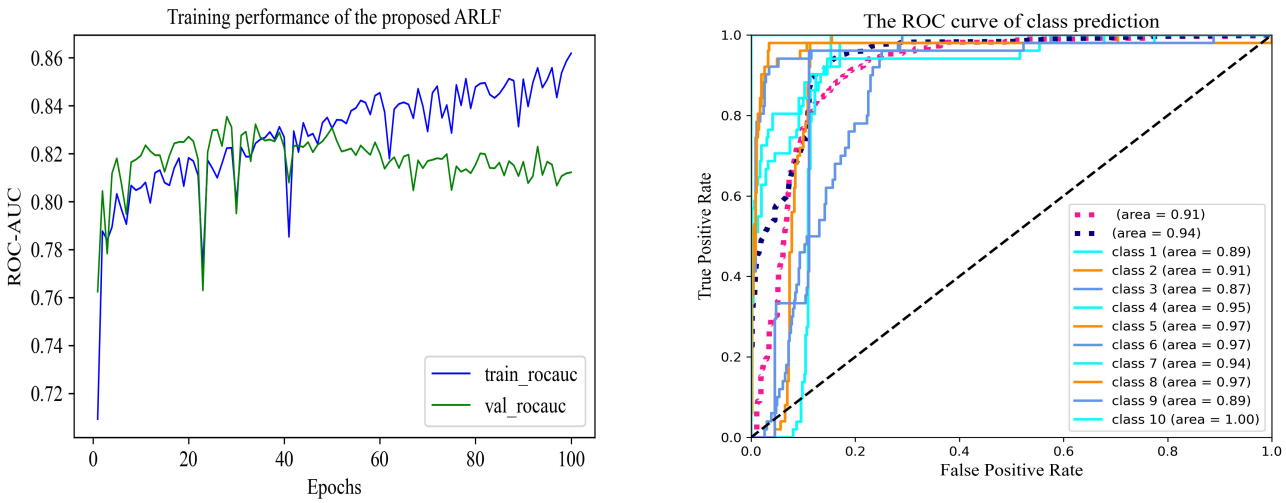
To demonstrate the robustness of the proposed approach, six popular methods, including extreme gradient boosting (XGBoost), light gradient boosting machine (lightGBM), LSTM, Gate Recurrent Unit (GRU), ClinicNet [37], and Transformer [38] are selected for comparative analysis. These methods are widely used in various fields, such as time series prediction tasks, and they have shown SOTA performance in inpatients' LoS and mortality prediction [39]. To ensure fairness in comparison, the core hyperparameters of the compared models are set to the same as that of the proposed approach. Concretely, the adaptive moment estimation (Adam) [40] is utilized as a training optimizer of the models, with a mini-batch size of 16, a learning rate of 1×10^{-3}, and 100 epochs of training. The dataset is randomly split into the training, validation, and testing sets in a 4:2:1 ratio. Table 2 summarizes the LoS prediction performance of the different methods on the training and validation sets, and the test results of different methods are presented in Table 3. Fig. 4(a) and Fig. 5 portray the training performance of the proposed approach and the tested confusion matrices of different methods, respectively.

Table 2 shows that the proposed ARLF has attained a competitive training and validation performance compared to other methods. After training for 100 epochs, the proposed method has achieved a validation accuracy and F1-Score of 0.401 and 0.411, respectively, which are superior to that of other compared methods. Besides, the ROC-AUC and PR-AUC of the proposed method also reach the best values except that of the tree algorithms like XGBoost and LightGBM. Nevertheless, these tree algorithms are ensemble learning models comprising multiple decision trees, which can easily lead to overfitting risks. Tables 2 and 3 show that the training performance of the tree algorithms is good, but it drops significantly on the validation and testing sets. By contrast, the proposed method shows robustness and effectiveness, and it has realized the best prediction results compared to other SOTA methods on the testing dataset, e.g., the proposed ARLF achieves the highest score with ROC-AUC of 0.936. In addition, the proposed ARLF has a relatively small volume (around 0.42M, see Table 1) and the running time for LoS prediction is 89.038 seconds, which is a competitive time, compared with other methods, as shown in Table 3.

Moreover, to validate the superiority of the proposed model, we performed the statistical test to give a detailed analysis of diverse algorithms. We have used the Friedman statistical test to compare the average ranking of different methods. Let R_i^j be the sorting of the j-th method on the i-th metrics, and the average ranking can be computed by

$$\bar{R}_j = \frac{1}{N} \sum_i R_i^j \tag{33}$$

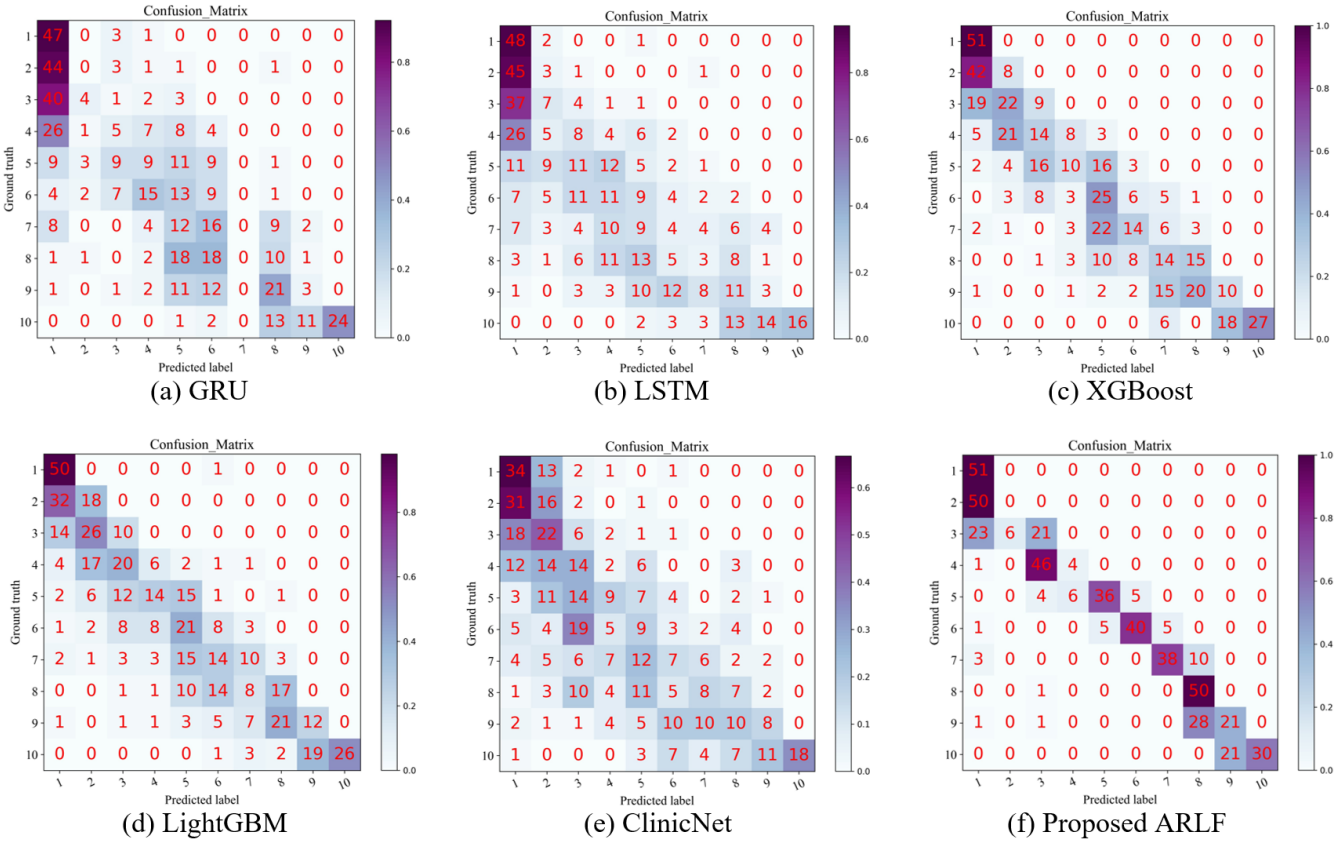
where N indicates the number of measurement metrics. The null hypothesis of Friedman's test is that there is no difference among these methods, i.e., the performance differences of these comparative methods are not significant. Mathemati-



(a) ROC-AUC on the training set

(b) ROC curve on the testing set

FIGURE 4. **ROC-AUC and ROC curves of the proposed method for LoS prediction.**



(a) GRU (b) LSTM (c) XGBoost

(d) LightGBM (e) ClinicNet (f) Proposed ARLF

FIGURE 5. **The LoS test confusion matrices of different methods.**

cally, the Friedman's test statistic can be expressed by

$$F_F = \frac{(N-1)x_F^2}{N(k-1)-x_F^2} \tag{34}$$

Here k denotes the number of methods, F_F follows the F-distribution with the freedom degree of k-1 and (k-1)(N-1), and x_F^2 is computed by

$$x_F^2 = \frac{12N}{k(k+1)}\left[\sum_j \bar{R}_j^2 - \frac{k(k+1)^2}{4}\right] \tag{35}$$

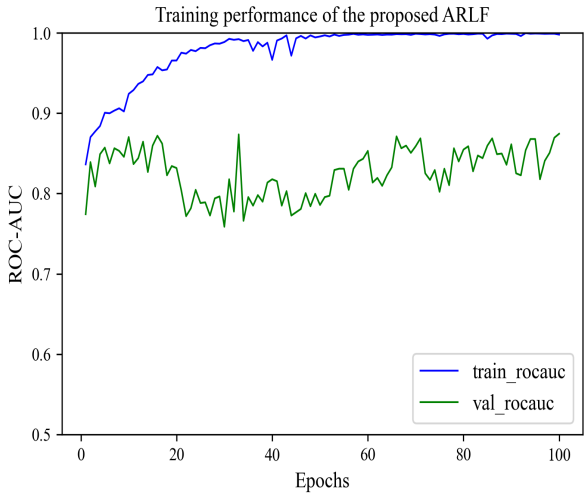
In the experiments of LoS prediction, there are 7 methods evaluated by 6 different measurement metrics, and thus the k and N are separately assigned as 7 and 6. Referring to Eqs. (34, 35), the x_F^2 and F_F are calculated as 32.86 and 52.25, respectively. Whilst, referring to the probability distribution table, the critical value of $F(k\text{-}1,(k\text{-}1)(N\text{-}1))$ is obtained as $F(6,30) = 2.42$. Since the observed $F_F = 52.25$ is much larger than 2.42, the null hypothesis of Friedman's test is rejected, indicating that there are significant statistical differences among the algorithms. Therefore, based on the statistical comparative analysis, it can be concluded that the proposed approach has outperformed other SOTA methods and shows significant advantages for LoS prediction. Besides, it can be visualized from Fig. 4(b) that the proposed method has exhibited superior performance with the ROC curves of all classes close to the top-left corner of this figure, which is also reflected in the confusion matrix of Fig. 5(f). The proposed method has successfully predicted the LoS categories of most test samples in different lengths of stay.

Other than that, we have investigated the model performance of the proposed method for the inpatients' mortality prediction. Similarly, the dataset is split into the training and validation sets with a ratio of 2:1, and more test samples including 2,720 survival and 330 mortality samples, a total of 3,050 samples are drawn from the original time series data as the testing set to evaluate the models. Tables 4 and 5 present the training and test performance of mortality prediction for different methods. Fig. 6 depicts the area under the ROC Curve (ROC-AUC) and ROC Curve of the proposed method for model training and test, and the test confusion matrices of different methods are portrayed in Fig. 7. From Table 5 it can be seen that the proposed approach has realized a test *Accuracy* of 90.30%, and the test *Precision*, *Recall*, and F_1-*Score* also attain no less than 88.70%, 90.30%, and 89.10%, respectively. According to the results reported in Table 5, the x_F^2 and F_F of different methods for mortality prediction are computed as 21.08 and 7.07, respectively. Since the computed $F_F = 7.07$ is larger than the critical value $F(6,30)$ of 2.42 referring to the probability distribution table (test size α = 0.05), the null hypothesis of Friedman's test is rejected suggesting a statistically significant difference between the algorithms. Therefore, the statistical comparative analysis demonstrates the competitive advantages of the proposed approach for predicting patients' mortality in hospitals. Also, from Fig. 6(a) it can be visualized that the training ROC-AUC is close to 100% and the validation ROC-AUC exceeds 80%, which exhibits the outstanding performance of the proposed method. Besides, as shown in Fig. 6(b), the proposed method has revealed superior operating characteristics, with the ROC curves close to the top-left corner of this figure. This positioning signifies the effectiveness and feasibility of the proposed approach for mortality prediction. Additionally, from the confusion matrices of Fig. 7, it can be observed that the ARLF has properly recognized most of the samples. 177 mortality samples and 2,468 survival samples have been correctly recognized by the proposed approach.

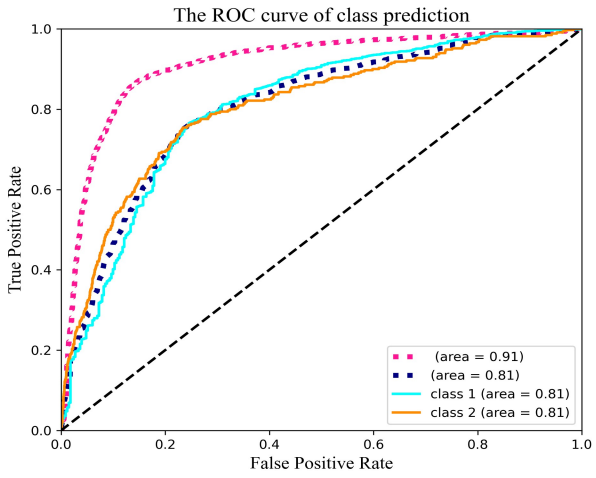
Moreover, we have implemented a performance investigation of the proposed approach compared to the results presented in previous literature concerning the predictions of both LoS and mortality, as shown in Table 6. From Table 6 it can be seen that the proposed approach has delivered a competitive result and outperformed most of the existing methods. To sum up, the comparative analysis results demonstrate the superiority and effectiveness of the proposed approach for the predictions of inpatients' LoS and mortality. The key explanation for the substantial performance of the proposed approach is that the fully-connected residual convolutional networks coupled with LSTM unit are incorporated into the model, which maximizes information transfer and extracts the spatial and temporal features for the class prediction tasks. Besides, a self-attention mechanism is embedded into the networks to highlight favorable information while filtering out unnecessary noises, thereby improving the accuracy of the model. Moreover, the customized L_{ES} loss function used in the network also alleviates the imbalanced sample problem. By comparison, the other methods are single network structures or frequently-used EL methods. Since only unimodal feature such as temporal feature or spatial feature is extracted, these methods do not achieve the optimal performance. Consequently, the proposed approach attained a competitive performance in comparative experiments.

D. ABLATION STUDY

We perform the ablation study on our model, where we analyze the behavior of the residual block, self-attention mechanism, and the customized loss function on the MIMIC-III v1.4 dataset for the LoS and mortality predictions. First, we separately remove the residual block and the module of the self-attention mechanism in the network to investigate the performance. Then, we evaluate the effect of the optimized loss function by substituting the customized loss function with the traditional CE loss function. Table 7 summarizes the comparison results of ablation experiments. From Table 7, we note an evident decreased performance in the results of the ablated models. The ROC-AUC of removing the residual block and self-attention mechanism for LoS prediction drops to 0.820 (decrease by 0.116) and 0.891 (decrease by 0.045). Also, the ROC-AUC for mortality prediction drops to 0.755 (decrease by 0.077) and 0.759 (decrease by 0.073), respectively. These ablation experiments indicate that both the residual block and self-attention mechanism contribute to



(a) ROC-AUC on the training set



(b) ROC curve on the testing set

FIGURE 6. **ROC-AUC and ROC curve of the proposed method for mortality prediction.**

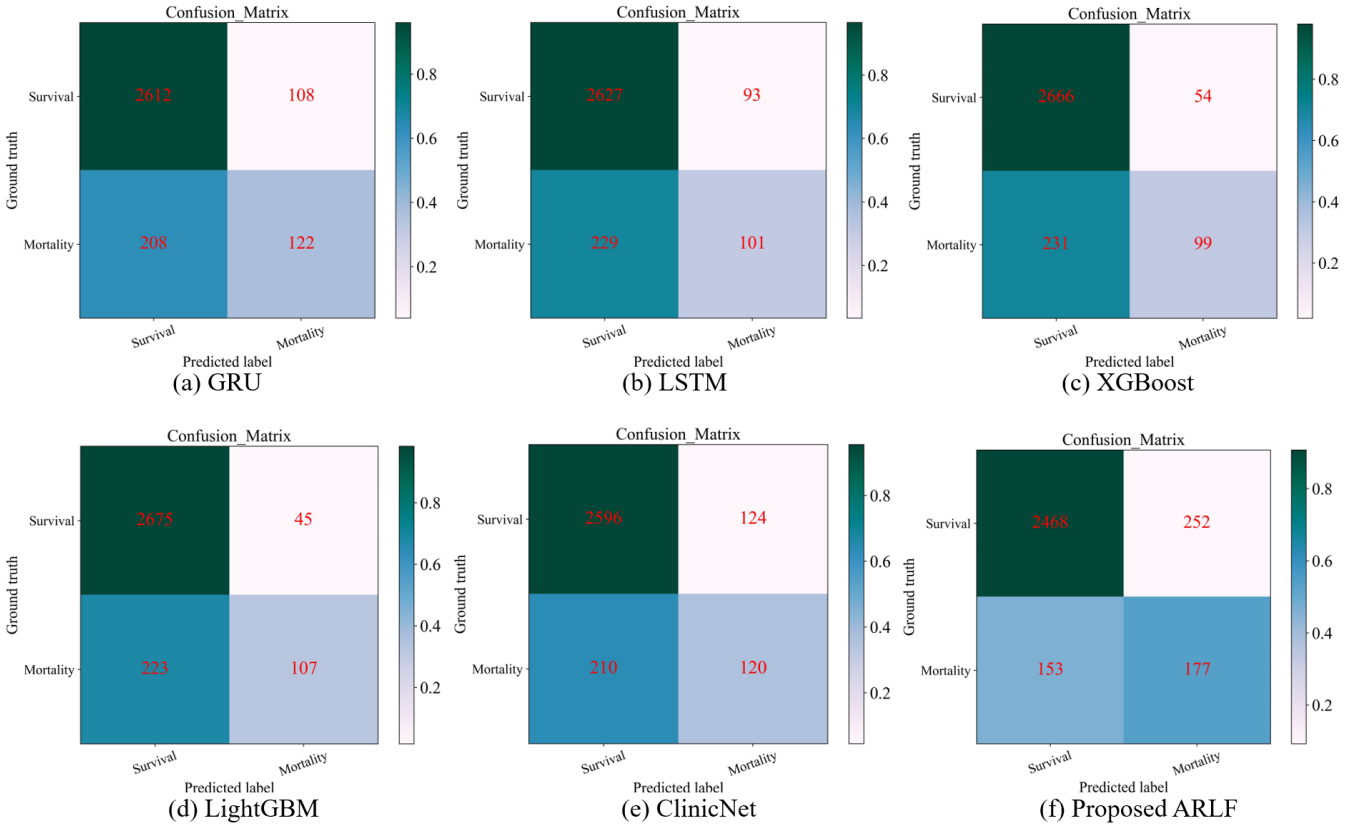


FIGURE 7. **The test confusion matrices of different methods.**

TABLE 4. The mortality prediction performance on training and validation sets.

No.	Methods	Training set						Validation set					
		ROC-AUC	PR-AUC	Acc	Pre	Rec	F1	ROC-AUC	PR-AUC	Acc	Pre	Rec	F1
1	XGBoost	0.889	0.903	0.970	0.971	0.970	0.969	0.679	0.723	0.910	0.919	0.910	0.891
2	LightGBM	0.960	0.859	0.932	0.934	0.932	0.923	0.856	0.688	0.891	0.878	0.891	0.875
3	LSTM	0.970	0.859	0.941	0.939	0.941	0.937	0.824	0.541	0.891	0.882	0.891	0.868
4	GRU	0.921	0.680	0.909	0.901	0.909	0.900	0.820	0.640	0.891	0.877	0.891	0.875
5	ClinicNet	0.985	0.934	0.963	0.962	0.963	0.962	0.865	0.541	0.881	0.864	0.881	0.867
6	Transformer	0.877	0.873	0.796	0.818	0.841	0.829	0.769	0.521	0.782	0.945	0.793	0.862
7	Proposed ARLF	0.937	0.942	0.933	0.936	0.933	0.933	0.759	0.533	0.881	0.864	0.881	0.867

TABLE 5. The test results of different methods for mortality prediction, (.) denote the ranking.

No.	Methods	Testing set						Average rank	Time (s)
		ROC-AUC	PR-AUC	Acc	Pre	Rec	F1		
1	XGBoost	0.640 (7)	0.511 (2)	0.907 (2)	0.891 (3)	0.906 (2)	0.891 (2.5)	3.083	498.961
2	LightGBM	0.868 (1)	0.525 (1)	0.912 (1)	0.899 (2)	0.912 (1)	0.897 (1)	1.167	493.879
3	LSTM	0.828 (4)	0.433 (7)	0.894 (5)	0.876 (7)	0.894 (5)	0.882 (6)	5.667	82.737
4	GRU	0.838 (2)	0.477 (5)	0.896 (4)	0.883 (5)	0.896 (4)	0.888 (4)	4.000	77.179
5	ClinicNet	0.825 (5)	0.451 (6)	0.890 (6)	0.878 (6)	0.890 (6)	0.883 (5)	5.667	97.373
6	Transformer	0.784 (6)	0.504 (3)	0.663 (7)	0.949 (1)	0.786 (7)	0.859 (7)	5.167	157.286
7	Proposed ARLF	0.832 (3)	0.492 (4)	0.903 (3)	0.887 (4)	0.903 (3)	0.891 (2.5)	3.250	133.561

TABLE 6. Comparison with previous literature.

ID	References	Year	Description	ROC-AUC (LoS)	ROC-AUC (mortality)
1	Harutyunyan et al. [1]	2019	LSTM	0.840	0.870
2	Rajkomar et al. [41]	2020	ConCare	0.860	0.770
3	Hu et al. [2]	2020	LightGBM (2x2 filter)	-	0.850
4	Catling et al. [42]	2020	TCN	0.895	0.795
5	Harerimana et al. [43]	2022	MHT	0.908	0.867
6	This study	2023	ARLF	0.936	0.832

the performance gains of the proposed method. Furthermore, we replace the customized loss function with the CE loss function to evaluate the effect of the model optimization. We also note a significant decrease, where the ROC-AUC and PR-AUC of LoS prediction separately drops to 0.838 (decrease by 0.098) and 0.382 (decrease by 0.035). The ROC-

TABLE 7. The results of ablation experiments.

Ablation approach	Test accuracy of LoS prediction		Test accuracy of mortality prediction		Time for LoS task (s)
	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	
Delete residual block	0.820	0.372	0.755	0.522	42.632
Delete self-attention	0.891	0.415	0.759	0.527	76.267
Replace the customized loss function with CE	0.838	0.382	0.763	0.540	57.357
This study	0.936	0.417	0.832	0.492	89.038

AUC of mortality prediction also drops to 0.763 (decrease by 0.069). The fundamental explanation for the effect of the loss function is that the CE loss function does not consider the class imbalance problem, resulting in decreased accuracy. This ablation experiment demonstrates that the optimized loss function delivers better performance than that of the CE loss function used in our model for inpatients' LoS and mortality predictions.

V. CONCLUSION

Enhancing the excellence of patient care and predicting future outcomes is the most crucial goal in intensive care research. Meanwhile, LoS and mortality predictions are critical topics for predicting possible outcomes. In this paper, by deploying clinical time series data obtained from the EHR database MIMIC-III v1.4, we develop an efficient attention embedded residual LSTM FCN model to perform the LoS and mortality predictions for patients using clinical time series data. The proposed ARLF is primarily composed of a CNN layer, three residual blocks, an LSTM unit, a FCN module, and a self-attention module. After the CNN layer and residual blocks extract the clinical data, the FCN and LSTM are used to extract the spatial and temporal features. Then a self-attention mechanism is embedded into the network to highlight favorable information while suppressing needless noises. Finally, the weighted features are input into a DC layer with the Softmax activation function for the final prediction. In experiments, the best performance of the proposed ARLF is proved by comparison with other SOTA methods. Ablation studies are then conducted to verify the performance gains of the residual block, self-attention mechanism, and the optimized loss function. The experimental findings reveal that the proposed ARLF architecture outperforms compared methods, or the ablation models that remove the relevant modules, demonstrating the validity and feasibility of the proposed approach. The novel integration of these techniques showcases significant advancements in predictive modeling, providing a powerful tool for improving patient care. However, it exhibits several potential limitations. Firstly, the computational complexity inherent in the integration of CNN, residual blocks, LSTM, FCN, and self-attention mechanisms may pose challenges in terms of training time and resource requirements. To mitigate this, model pruning algorithms can be added to simplify the model in the future work. Secondly, despite promising performance, treating LoS prediction as a classification problem still needs more investigation on the issue of LoS data skewness. A potential solution is to treat LoS prediction as a regression rather than a classification problem. In future work, we will apply the proposed model in more clinical tasks such as phenotype classification, risk assessment, patient flow prediction, and mining signatures from event sequences.

References

[1] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Scientific Data*, vol. 6, no. 1, p. 96, 2019.

[2] Y. Hu, R. Subramanian, W. An, N. Zhao, and W. Wu, "Faster healthcare time series classification for boosting mortality early warning system," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 8976–8981.

[3] F. Wang, N. Lee, J. Hu, J. Sun, S. Ebadollahi, and A. F. Laine, "A framework for mining signatures from event sequences and its applications in healthcare data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 272–285, 2012.

[4] Y. Cheng, F. Wang, P. Zhang, and J. Hu, "Risk prediction with electronic health records: A deep learning approach," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, SIAM, 2016, pp. 432–440.

[5] A. Al-Dailami, H. Kuang, and J. Wang, "Predicting length of stay in icu and mortality with temporal dilated separable convolution and context-aware feature fusion," *Computers in Biology and Medicine*, vol. 151, p. 106 278, 2022.

[6] A. M. Mudge, P. McRae, M. Banks, *et al.*, "Effect of a ward-based program on hospital-associated complications and length of stay for older inpatients: The cluster randomized cherish trial," *JAMA Internal Medicine*, vol. 182, no. 3, pp. 274–282, 2022.

[7] T.-H. Cheng and P. J.-H. Hu, "A data-driven approach to manage the length of stay for appendectomy patients," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 6, pp. 1339–1347, 2009.

[8] B. Alsinglawi, F. Alnajjar, O. Mubin, *et al.*, "Predicting length of stay for cardiovascular hospitalizations in the intensive care unit: Machine learning approach," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2020, pp. 5442–5445.

[9] C Bao, F Deng, and S Zhao, "Machine-learning models for prediction of sepsis patients mortality," *Medicina Intensiva (English Edition)*, vol. 47, no. 6, pp. 315–325, 2023.

[10] K. Lin, Y. Hu, and G. Kong, "Predicting in-hospital mortality of patients with acute kidney injury in the icu using random forest model," *International Journal of Medical Informatics*, vol. 125, pp. 55–61, 2019.

[11] H. M. Zolbanin, B. Davazdahemami, D. Delen, and A. H. Zadeh, "Data analytics for the sustainable use of resources in hospitals: Predicting the length of stay for patients with chronic diseases," *Information & Management*, vol. 59, no. 5, p. 103 282, 2022.

[12] S. Ali, S. El-Sappagh, F. Ali, M. Imran, and T. Abuhmed, "Multitask deep learning for cost-effective prediction of patient's length of stay and readmission state using multimodal physical activity sensory data,"

IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 12, pp. 5793–5804, 2022.

[13] W. Caicedo-Torres and J. Gutierrez, "Iseeu2: Visually interpretable mortality prediction inside the icu using deep learning and free-text medical notes," *Expert Systems with Applications*, vol. 202, p. 117 190, 2022.

[14] D. E. Roopa, R Rajadevi, R Shanthakumari, E Praveen, S SethuRaj, and A. Shyam, "Mortality prediction of lung cancer from ct images using deep learning techniques," in *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, IEEE, 2022, pp. 13–18.

[15] F. Viton, M. Elbattah, J.-L. Guérin, and G. Dequen, "Multi-channel convnet approach to predict the risk of in-hospital mortality for icu patients.," in *DeLTA*, 2020, pp. 98–102.

[16] J. Theis, W. L. Galanter, A. D. Boyd, and H. Darabi, "Improving the in-hospital mortality prediction of diabetes icu patients using a process mining/deep learning architecture," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 1, pp. 388–399, 2021.

[17] X. Li, P. Ge, J. Zhu, *et al.*, "Deep learning prediction of likelihood of icu admission and mortality in covid-19 patients using clinical variables," *PeerJ*, vol. 8, e10337, 2020.

[18] J. Chen, W. Chen, A. Zeb, and D. Zhang, "Segmentation of medical images using an attention embedded lightweight network," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105 416, 2022.

[19] D. Thakur and S. Biswas, "Attention-based deep learning framework for hemiplegic gait prediction with smartphone sensors," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 11 979–11 988, 2022.

[20] D. Thakur, A. Guzzo, and G. Fortino, "Attention-based multihead deep learning framework for online activity monitoring with smartwatch sensors," *IEEE Internet of Things Journal*, 2023.

[21] Y. Hu, *Healthcare Information Platform in AI Era*. The University of Texas at Dallas, 2021.

[22] G. Clermont, D. C. Angus, S. M. DiRusso, M. Griffin, and W. T. Linde-Zwirble, "Predicting hospital mortality for patients in the intensive care unit: A comparison of artificial neural networks with logistic regression models," *Critical Care Medicine*, vol. 29, no. 2, pp. 291–296, 2001.

[23] L. A. Celi, S. Galvin, G. Davidzon, J. Lee, D. Scott, and R. Mark, "A database-driven decision support system: Customized mortality prediction," *Journal of Personalized Medicine*, vol. 2, no. 4, pp. 138–148, 2012.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] L. S.-T. Memory, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 2010.

[26] J. Chen, Y. Wen, Y. Nanehkaran, M. Suzauddola, W. Chen, and D. Zhang, "Machine learning techniques for stock price prediction and graphic signal recognition," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 106 038, 2023.

[27] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[29] A. Katrompas and V. Metsis, "Enhancing lstm models with self-attention and stateful training," in *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 1*, Springer, 2022, pp. 217–235.

[30] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 606–615.

[31] J. Zeng, X. Ma, and K. Zhou, "Enhancing attention-based lstm with position context for aspect-level sentiment classification," *IEEE Access*, vol. 7, pp. 20 462–20 471, 2019.

[32] H. Gao, R. Li, J. Wang, H. Zhao, S. Yan, and L. Ma, "Research on bidirectional recurrent imputation of multivariate time series for clinical outcomes prediction," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2022, pp. 954–960.

[33] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[35] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 322–330.

[36] A. E. Johnson, T. J. Pollard, L. Shen, *et al.*, "Mimic-iii, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, pp. 1–9, 2016.

[37] J. X. Wang, D. K. Sullivan, A. C. Wells, and J. H. Chen, "Clinicnet: Machine learning for personalized clinical order set recommendations," *JAMIA Open*, vol. 3, no. 2, pp. 216–224, 2020.

[38] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[39] L. R. Soenksen, Y. Ma, C. Zeng, *et al.*, "Integrated multimodal artificial intelligence framework for healthcare applications," *NPJ digital medicine*, vol. 5, no. 1, p. 149, 2022.

[40] D Kingma and J Ba, "Adam: A method for stochastic optimization in: Proceedings of the 3rd international conference for learning representations (iclr'15)," *San Diego*, vol. 500, 2015.
[41] A. Rajkomar, E. Oren, K. Chen, *et al.*, "Scalable and accurate deep learning with electronic health records," *NPJ Digital Medicine*, vol. 1, no. 1, p. 18, 2018.
[42] F. J. Catling and A. H. Wolff, "Temporal convolutional networks allow early prediction of events in critical care," *Journal of the American Medical Informatics Association*, vol. 27, no. 3, pp. 355–365, 2020.
[43] G. Harerimana, J. W. Kim, and B. Jang, "A multi-headed transformer approach for predicting the patient's clinical time-series variables from charted vital signs," *IEEE Access*, vol. 10, pp. 105 993–106 004, 2022.

JUNDE CHEN received his master's degree from Sichuan University, China, in 2010, and the Ph.D. degree in Computer Science from the school of informatics, Xiamen University, China, in 2022. Currently, he is doing Postdoctoral research in the Schmid College of Science and Technology, Chapman University. His research interests include the aspects of Data Mining, Image Processing, Big data and Decision Support System, etc.

MASON LI is pursuing a BS degree in Computer Science from Chapman University, Orange, CA, USA. He is an undergraduate research assistant in the Fowler School of Engineering at Chapman University. His research focuses on multi-modal data analysis and machine learning, with applications in healthcare and scientific computing.

MILES MILOSEVICH received a BS degree in Software Engineering from Chapman University, Orange, USA, in 2024. He is currently a Research Assistant in the Fowler School of Engineering at Chapman University. His research interests focus on statistical modeling, symbolic representation, reinforcement learning, and predictive analytics.

TIFFANY LE is pursuing a BS degree in Data Science and MS degree in Electrical Engineering and Computer Science from Chapman University, Orange, CA, USA. She is currently a research assistant in the Fowler School of Engineering at Chapman University. Her research interests are focused on machine learning, sentiment analysis, and prognostics with the application in healthcare and natural language processing.

ANDREW BAHSOUN is pursuing a BS degree in the Fowler School of Engineering from Chapman University, Orange, CA, USA. He is currently a research assistant in the Fowler School of Engineering at Chapman University. His research interests involve AI and fairness.

YUXIN WEN received a BS degree in Medical Informatics and Engineering from Sichuan University, Chengdu, China, in 2011, a MS degree in Biomedical Engineering from Zhejiang University, Hangzhou, China, in 2014, and a PhD degree in Electrical and Computer Engineering from the University of Texas at El Paso (UTEP), El Paso, TX, USA, in 2020. She is currently an Assistant Professor in the Fowler School of Engineering at Chapman University, Orange, CA, USA. Her research interests are focused on statistical modeling, prognostics, and reliability analysis with the application in manufacturing and healthcare.