Lightweight Outsourced Privacy-Preserving Heart Failure Prediction Based on GRU

Zuobin Ying[1,2(✉)], Shuanglong Cao[2], Peng Zhou[2], Shun Zhang[2], and Ximeng Liu[3,4]

[1] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

[2] School of Computer Science and Technology, Anhui University, Hefei 230601, China
yingzb@ahu.edu.cn

[3] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

[4] Key Lab of Information Security of Network System (Fuzhou University), Fuzhou 350108, Fujian, China

Abstract. The medical service provider establishes a heart failure prediction model with deep learning technology to provide remote users with real-time and accurate heart failure prediction services. Remote users provide their health data to the health care provider for heart failure prediction through the network, thereby effectively avoiding the damage or death of vital organs of the patient due to the onset of acute heart failure. Obviously, sharing personal health data in the exposed data sharing environment would lead to serious privacy leakage. Therefore, in this paper, we propose a privacy-preserving heart failure prediction (PHFP) system based on Secure Multiparty Computation (SMC) and Gated Recurrent Unit (GRU). To meet the real-time requirements of the PHFP system, we designed a series of data interaction protocols based on additional secret sharing to achieve lightweight outsourcing computing. Through these protocols, we can protect the user's health data privacy while ensuring the efficiency of the heart failure prediction model. At the same time, to provide high-quality heart failure prediction services, we also use the new mathematical fitting method to directly construct the safety activation function, which reduces the number of calls to the security protocol and optimizes the accuracy and efficiency of the system. Besides, we built a security model and analyzed the security of the system. The experimental results show that PHFP takes into account the safety, accuracy, and efficiency in the application of heart failure prediction.

Keywords: Secure Multiparty Computation · Privacy-preserving · Heart failure prediction · Gated Recurrent Unit

S. Wen et al. (Eds.): ICA3PP 2019, LNCS 11945, pp. 521–536, 2020.
https://doi.org/10.1007/978-3-030-38961-1_45

1 Introduction

Heart Failure (HF) is a complex clinical symptom cluster and a severe stage of various heart diseases with high morbidity and mortality. According to the European Society of Cardiology (ESC), 26 million adults worldwide are diagnosed with heart failure, and 3.6 million people are newly diagnosed each year. About 20% heart failure patients die within one year after diagnosis, and about 50% decease in five years once have been diagnosed [1]. To effectively reduce the incidence and mortality of heart failure, early accurate prediction of heart attack episodes is indispensable. It is difficult for the traditional clinical methods to diagnose the occult acute heart failure at an early stage, so usually, the patient diagnosed after being admitted to the emergency department. If the essential organs of some patients not diagnosed in time, irreversible damage or death may occur [2]. Therefore, it is essential to provide an early and accurate heart failure prediction service. In recent years, with the development of deep learning, medical research institutions have trained high-precision heart failure prediction models by acquiring patient health data to provide users with high-quality heart failure prediction services. Among them, Edward Choi et al. [3] used a GRU neural network to establish a time series model the records related to EMR and realized much accurate prediction at an early stage of heart failure. Moreover, the Area Under the Curve (AUC) of the model reaches 0.777, compared to the traditional clinical diagnostic (correct rate of 0.513) has better accuracy. Using this result, healthcare providers can establish a heart failure pre-diagnosis model to provide real-time, accurate, and convenient heart failure prediction services to remote users. As a result, more and more users are providing their personal health data to medical service providers via the Internet for the purpose of obtaining real-time, accurate heart failure prediction services.

However, in the actual scenario, it is necessary to comprehensively consider the privacy of personal health data and the security of the heart failure prediction model provided by the medical service provider. In general, there are two ways to predict data sharing for heart failure. One way is that the user provides personal health data to the medical provider through the network, and then the medical service provider performs heart failure prediction and returns the result to the user locally. However, the medical service provider may disclose the user's health data during the process, which will lead to the leakage and abuse of the user's personal health data privacy. Another way is for the medical service provider to send the heart failure model to the user, and heart failure prediction is made by the user locally. Due to the high commercial value attached to this kind of prediction, the leakage of the model will bring economic losses to the medical service provider. Therefore, how to design a heart failure prediction system that can protect data privacy has become a vital issue to be solved. Furthermore, considering that the sudden onset of acute heart failure is often life-threatening and requires urgent rescue measures, this requires us to balance the timeliness and accuracy of the PHFP system.

In recent years, researchers have proposed a variety of technologies to protect medical privacy, such as anonymous technology and homomorphic encryption, in response to data privacy breaches in telemedicine scenarios. Nevertheless, anonymous technology [4] only protects the privacy of users to a certain extent, which makes it easy to lose valuable information, and then its prediction accuracy will be affected. And, studies have shown that anonymous techniques are not sufficient to resist re-identification attacks [5]. Moreover, current frameworks based on homomorphic encryption [6] are time-consuming and memory-intensive, and its computational overhead is enormous, which is not suitable for real-time heart failure prediction scenarios. None of the current work takes into account the balance between efficiency and precision. Therefore, when constructing a privacy-preventing heart failure prediction system, we must realize privacy protection in the premise of system accuracy and efficiency.

To achieve the above objectives, we propose a PHFP system. Our main contributions can be summarized as follows:

- We are first design a lightweight system to protect privacy data and service provider model parameters for the medical user heart failure prediction. The system is based on the addition of secret sharing technology in secure multiparty computing, which transfers intricate work to the edge server, reducing the cost of medical users. Moreover, the system avoids the interaction between the medical user terminal and the server, with the results that the overall efficiency of the system is improved.
- PHFP use a new mathematical method to directly construct the secure *Sigmoid* function and the *Tanh* function, which avoids the time overhead caused by the system calling too many security components during the running process. At the same time, system solve the problem of low function fitting precision within a specific interval caused by the local fitting of the Taylor series. Compared with the existing additive secret sharing scheme, our system has significantly improved in terms of computational overhead and precision.
- We conduct a comprehensive experimental evaluation to measure the performance of our program. The experimental results show that the system is superior to the previous work in terms of computational overhead, communication overhead, and computational accuracy while protecting the privacy of heart failure prediction data.

The remaining part of this paper is organized as follows. We formulate the problem and present the system model and security model in Sect. 2. In Sect. 3, the primitives about GRU and secure multiparty computation are briefly introduced followed by problem analysis and model presentation. Then the building blocks that support efficient, secure computation based on secret sharing techniques are provided in Sect. 4. On the basis of that, we propose the details of our system in Sect. 5. And Sects. 6 and 7 covers the theoretical analysis and experimental results respectively. Finally, related conclusion is stated Sect. 8.

2 Problem Formulation

In this section, we formalize the system model, security model and identify our design goal.

2.1 System Model

In our system model, we focus on how users with sensitive medical data can obtain accurate and privacy-preserving real-time heart failure prediction services from cloud service providers. Precisely, the system consists of five parts: (1) smart wearable device (SWD); (2) the Medical User (MU); (3) the Edge Servers (ESs); (4) the Medical Service Provider (MSP); (5) Trusted Third Party (TTP). As shown in Fig. 1.

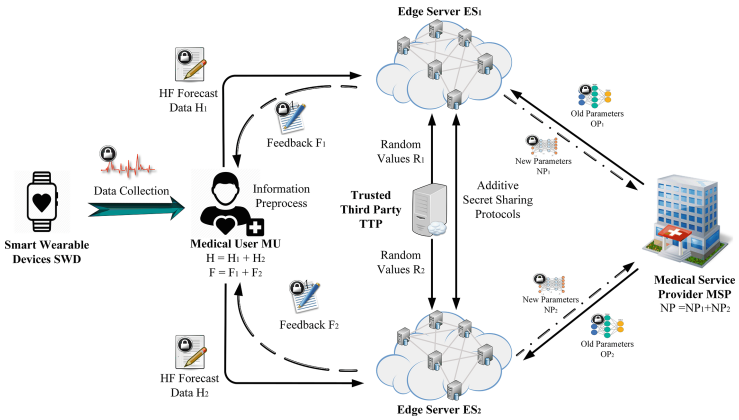


Fig. 1. System model under consideration

- SWD is used to collect various health data of healthy users. The collected data has a total of 279 feature dimensions, such as heart rate, blood pressure, body temperature, etc. And it sends the collected health data to the medical user.
- MU wants to know its future heart failure attack risk coefficient, and it will preprocess its heart failure data on the phone to form the heart failure eigenvector, which is randomly divided into the different secret values and sent to different ESs. Besides, MU was able to accept the feedback results from ESs and combines the feedback results to obtain the final correct prediction results of heart failure.

– ESs can be a cloud service provider that assists healthcare providers in collecting data related to heart failure prediction and training new data. At the same time, ESs can return the correct heart failure prediction result to the user, and promote the user to provide more data sets.
– MSP, such as pharmaceutical companies or hospitals, can provide real-time heart failure risk prediction services. Individually, with the help of ESs, MSP can obtain the latest training parameters of GRU recurrent neural network. Considering the benefits of ESs, MSP is also willing to commission ESs to effectively predict the risk of heart failure and return the results to MU.
– TTP is only responsible for generating random numbers, which means that TTP doesn't require a lot of computing power. It can be replaced by a light server or even a personal computer.

2.2 Security Model

In the security model, we use the standard semi-honest security model [7], which is also perceived as passive or honest-but-curious. In this security model, each edge server enforces the protocol as required by the contract. But out of curiosity, they can try to get as much information as they can from the data they receive and the data they process.

Also, we assume that the two edge servers ES_1 and ES_2 are independent of each other, and there is no collusion between them. This means that the data acquired by each of the edge servers will not be revealed. In this way, even if each edge server has durable computing power, they can only get some split medical or intermediate interaction data and model parameters. In other words, real raw medical data and model parameters cannot be recovered.

It is worth noting that TTP is merely responsible for generating random numbers, and it is honest and trustworthy. Last but not least, we also assume that medical users and service providers are honest and a secure channel for communication exists between the entities.

3 Preliminaries

3.1 Features of GRU

GRU neural network is a variant of Long Short Term Memory (LSTM), besides, GRU maintains the effect of LSTM while making the structure simpler. It's a very popular neural network. The mathematical expression is shown below:

$$
\begin{aligned}
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \\
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\
\tilde{h}_t &= \tanh(W_{\tilde{h}} \cdot [r_t \odot h_{t-1}, x_t] + b_{\tilde{h}}) \\
h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t
\end{aligned}
$$

It is worth noting that in GRU, the value of hidden layer h_{t-1} at time step $t-1$ and the input value at time step t doesnt directly change the value of h_t.

The value of h_t is determined by updating gate z_t, resetting gate r_t, and intermediate storage cell $\tilde{h_t}$. In short, reset gates allow the hidden layer to remove any information that is not useful for future prediction, while update gates determine how much information from the previous hidden layer should be retained by the current hidden layer.

3.2 Additive Secret Sharing Protocols

Secret sharing protocol is mainly used for secure multiply party computing (SMC) and privacy protection. The encryption protocol based on secret sharing has good performance and can be used to design an efficient privacy protection computing model [8]. The secret sharing protocol can be thought of as consisting of a large number of "components" through which we can build a larger and equally secure system.

Lemma 1. *If all the sub-protocols of a protocol are fully emulated, then the protocol is fully emulated* [9].

- *Random Bit Protocol.* The *RanBits*$(\cdot)$ protocol [10] can be thought of simply as a random number generator. It doesn't need any input to generate any bit sequence $(r_0, \cdots, r_l)$. At the same time, a hex random number r can be calculated by
$$r = \sum_{i=0}^{l} r_i \cdot 2^i.$$
- *Secure Addition and Subtraction Protocol.* The *SecAdd*$(\cdot)$ and *SecSub*$(\cdot)$ protocol [10] can calculate $f(u, v) = u \pm v$. Since $u \pm v = (u_1 + u_2) \pm (u_1 + v_2) = (u_1 \pm v_1) + (u_2 \pm v_2)$, it's easy to see that the protocol can perform secure additions and subtractions locally without the need for interaction between servers. After the computation, each participating party will output $f_i = u_i \pm v_i$. Obviously, we have $f_1 + f_2 = u \pm v$.
- *Secure Multiplication Protocol.* The *SecMul*$(\cdot)$ protocol [10] is based on the *Beaver's triplet* [11]. Given an input binary group (u, v), the protocol outputs another binary group (f_1, f_2) to the two participants, where $f = f_1 + f_2 = u \cdot v$. In this process, a trusted third party is required to generate a random triple (x, y, z) and $z = x \cdot y$. It is worth noting that the Participants will not be informed of each other's input.
- *Secure Comparison Protocol.* The *SecCmp*$(\cdot)$ protocol [10] can be achieved in the comparison of the size of two inputs u and v, the input of both sides will not be leaked. And, if $u < v$, *SecCmp*$(\cdot)$ outputs 1, otherwise outputs 0.
- *Secure Vector Concatenation Protocol.* The *SecCon*$(\cdot)$ protocol [12] is to connect two short vectors into one long vector. That is (Table 1),
$$\begin{aligned}[\boldsymbol{u}, \boldsymbol{v}] &= [(\boldsymbol{u_0}, \boldsymbol{u_1}, \boldsymbol{u_2}, ...), (\boldsymbol{v_0}, \boldsymbol{v_1}, \boldsymbol{v_2}, ...)] \\ &= (\boldsymbol{u_0}, \boldsymbol{u_1}, \boldsymbol{u_2}, ..., \boldsymbol{v_0}, \boldsymbol{v_1}, \boldsymbol{v_2}, ...).\end{aligned}$$

Table 1. Variables and their description

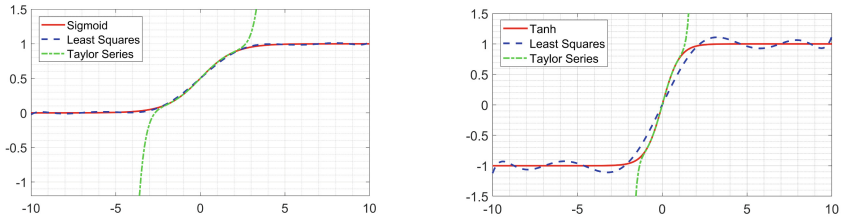
Variables	Description
z_t	The update gate at timestep t
r_t	The reset gate at timestep t
$\widetilde{h_t}$	The intermediate memory unit at timestep t
h_t	The hidden layer at timestep t
W	The weight matrix
b	The bias term
$\odot$	Hadamard product
$\sigma_{sec}(.)$	Secure sigmoid function
$\tanh_{sec}(.)$	Secure tanh function
W_{ih}, W_{ix}	The split matrixes of W_i
σ	The sigmoid function
δ_t	The error vector at time
∇	The symbol of gradient

4 Secret Sharing Based Functions

4.1 Nonlinear Function Fitting Method

The GRU has at least one activation function deployed in each gate. These activation functions are nonlinear functions such as the *Sigmoid* function and the *Tanh* function. According to Lemma 1, we can use some of the security protocols mentioned in Sect. 3 to build a secure nonlinear function protocol. However, nonlinear functions need to include not only addition and multiplication operations, but also complex operations, such as exponents and reciprocals. It is impossible to construct safe nonlinear functions directly with the security protocols mentioned before. Therefore, we need to fit the nonlinear numbers in the GRU gates with polynomials that only contain multiplication and addition.

Scheme I: Taylor Series [13]**.** At present, a useful tool for solving nonlinear problems is the Taylor series. By using Taylor expansion multi-order approxima-



(a) (b)

Fig. 2. Taylor series and least squares approximations for Sigmoid and Tanh function

tion, nonlinear problems can be linearised, which makes calculation and understanding more conveniently. The literature [12] uses the Taylor series to construct a secure exponential function with base e and Newton iteration method to build a secure reciprocal function. Then the security sigmoid and the secure tanh function are further built by invoking the security exponential and the reciprocal function, but this will make the overhead of the two edge servers more massive. Also, each invoke to the security function will result in a loss of precision, and too many invokes to the safety function will result in more loss of accuracy. It is not suitable for high-precision and high-efficiency scenarios like heart failure prediction. Hence, in our scheme I, we borrowed homomorphic encryption [14] to direct fit *Sigmoid* and *Tanh* using Taylor series directly and then build the addition secret sharing protocol.

Scheme II: Least Square Method. Although in scheme I, direct fitting of the sigmoid and tanh functions using Taylor series can reduce the overhead and precision loss of the edge server, this method still has a defect. As shown in the Fig. 2(a)–(b), the basic idea of the Taylor series is to approximate a function in the neighborhood of a point. For points that are not included in the neighborhood, the approximation error is much larger than the point contained within the area. To avoid the problem of the local fitting function in scheme I, we additionally consider the method of fitting the function by least squares [15], which finds the best function matching of the activation function by minimizing the sum of the squares of the errors. Its expression is as follows,

$$E_{min} = \sum_{i=1}^{n} (p(x_i) - y_i)^2.$$

Where y_i is the function value of the activation function to be fitted, and $p(x_i)$ is the function value of the polynomial to be constructed. Next, we will give the process of fitting the *Sigmoid* function $\sigma(x)$ by the least squares method, as shown below.

1. Let the least squares fit the polynomial as follows,

$$p(x) = a_0 + a_1 x + , \cdots , + a_m x^m. \tag{1}$$

2. The expression of the sum of squares of deviations is as follows,

$$E = \sum_{i=1}^{n} (a_0 + a_1 x_i + , \cdots , + a_m x_i^m - \sigma(x_i))^2. \tag{2}$$

3. To find the a_j value satisfying the minimum value of E, it is necessary to derive the partial derivative of Eq. (2) on the right side of a_j.

$$\frac{\partial E}{\partial a_j} = \sum_{i=1}^{n} 2 \cdot (a_0, a_1 x_i + , \cdots , a_m x_i^m - \sigma(x_i)) x_i^j. \tag{3}$$

4. By sorting out, we can get the following equations.

$$\begin{cases} na_0 + (\sum_{i=1}^{n} x_i)a_1 +, \cdots, +(\sum_{i=1}^{n} x_i^m)a_m = \sum_{i=1}^{n} y_i \\ (\sum_{i=1}^{n} x_i)a_0 + (\sum_{i=1}^{n} x_i^2)a_1 +, \cdots, +(\sum_{i=1}^{n} x_i^{m+1})a_m \\ = \sum_{i=1}^{n} x_i\sigma(x_i) \\ \cdots\cdots \\ (\sum_{i=1}^{n} x_i^m)a_0 + (\sum_{i=1}^{n} x_i^{m+1})a_1 +, \cdots, +(\sum_{i=1}^{n} x_i^{2m})a_m \\ = \sum_{i=1}^{n} x_i^m\sigma(x_i). \end{cases}$$

Finally, by solving the equations, we can get the values of $(a_1, a_2, \cdots, a_m)$ and get the least squares fit polynomial $p(x)$ of $\sigma(x)$. Therefore, $\sigma(x) \approx p(x)$.

4.2 Secure Sigmoid Function.

We use the least squares method to fit the Sigmoid function, Let x denote the input, the polynomial of the least squares fitting of the sigmoid function is expressed as follows,

$$f(x) = \sigma(x) = \frac{1}{1 + e^{(-x)}} \approx \sum_{i=0}^{\infty} C_i x^i.$$

Where C_i represents the coefficient of the least squares polynomial and i represents the order of the least squares polynomial.

Initialization. ES_1 and ES_2 respectively get random values x_1 and x_2, satisfying $x = x_1 + x_2$. In the process of initialization, ES_1 need to compute $f_0' \leftarrow C_0 + C_1 x_1$, ES_2 calculation $f_0'' \leftarrow C_1 x_2$. According to the polynomial exponent value, the iterative process shown below.

Iteration. In the process of iteration, we are mainly implemented by alternatively invoking $SecAdd(\cdot)$ and $SecMul(\cdot)$. First of all, ES_1 and ES_2 common computing $(g_0', g_0'') \leftarrow SecMul(C_2 x_1, C_2 x_2, x_1, x_2)$ and $f_1 \leftarrow SecAdd(f_0, g_0)$. Subsequently, g_i can be calculated iteratively by similar calculation methods. And invoke the secure comparison function $SecCmp(i, n)$ to determine whether to achieve the required polynomial index. If the required polynomial order is reached, terminate the iteration and output f_i' and f_i''. Otherwise, invoke the secure addition function to compute $SecAdd(f_{i-1}, g_{i-1})$.

4.3 Secure Tanh Function

Tanh is a hyperbolic tangent function, and the curves of the *Tanh* function and the *Sigmoid* function are relatively similar. The only difference is the output interval. The *Tanh* output interval is between (−1, 1) and the full function center at 0. Therefore, we can also fit the *Tanh* function by least squares. Let x be a function input, and the polynomial of the least squares fit of the $\tanh_{sec}(x)$ function is expressed as follows.

$$f(x) = \tanh(x) = \frac{e^x - e^{(-x)}}{e^x + e^{(-x)}} \approx \sum_{i=0}^{\infty} H_i x^i.$$

Also, since the *Initialization* and *Iteration* process of $\tanh_{sec}$ is similar to that of σ_{sec}, these processes are not repeated.

5 Lightweight Privacy-Preserving GRU for Encrypted HF Data

5.1 Secure Forward Propagation of GRU

Due to all the necessary security "components" have been constructed, the following work for the secure forward propagation of GRU is simply combining these security "components" appropriately to design a secure interactive subprotocol between the two edge servers ES_1 and ES_2. Note that in the following sections $[\![i]\!]$ stands for '′' and '″'.

Reset Gate. The reset gate allows the hidden layer to delete any information that is not useful for future prediction. To achieve this, the input vector x_i and information about the previous timestep h_{t-1} are put into the sigmoid function after a series of linear operations. And, the final output will be between 0 and 1. Because matrixed weight W_r and bias b_r for the reset gate is not publicly known, at timestep t, ES_1 and ES_2 compute separately,

$$r_t^{[\![i]\!]} \leftarrow \sigma_{sec}(W_r^{[\![i]\!]} \cdot [h_{t-1}^{[\![i]\!]}, x_t^{[\![i]\!]}] + b_r^{[\![i]\!]}).$$
$$\tilde{h}_t^{[\![i]\!]} \leftarrow \tanh_{sec}^{[\![i]\!]}(W_{\tilde{h}}^{[\![i]\!]} \cdot [r_t^{[\![i]\!]} \odot h_{t-1}^{[\![i]\!]}, x_t^{[\![i]\!]}] + b_{\tilde{h}}^{[\![i]\!]}).$$

Update Gate. The update gate determines how much information from the previous time step and the current timestep needs to be transmitted. Given the input weight matrix W_z, input bias b_z and timestep t, ES_1 and ES_2 consociation calculations,

$$z_t^{[\![i]\!]} \leftarrow \sigma_{sec}(W_z^{[\![i]\!]} \cdot [h_{t-1}^{[\![i]\!]}, x_t^{[\![i]\!]}] + b_z^{[\![i]\!]}).$$

The final output by invoking the secure multiplication function and the secure addition function. We let ES_1 and ES_2 respectively compute,

$$h_t^{[\![i]\!]} \leftarrow z_t^{[\![i]\!]} \odot h_{t-1}^{[\![i]\!]} + (1 - z_t^{[\![i]\!]}) \odot \tilde{h}_t^{[\![i]\!]}.$$

Both h_t' and h_t'' are then sent to MU as feedback. And MU can decrypt the ciphertext by simply adding them together, $h_t = h_t' + h_t''$.

5.2 Back Propagation Based Training of GRU

It is assumed that the iterative forward propagation of the privacy protection GRU has been completed. Let δ_{t-1} represents the error term at time $t-1$. It can be calculated by the partial derivative function of the output h_t at the timestep t. ES_1 and ES_2 combine calculates,

$$\delta_{t-1}^{[\![i]\!]} \leftarrow \delta_{r,t}^{[\![i]\!]} \cdot W_{rh}^{[\![i]\!]} + \delta_{z,t}^{[\![i]\!]} \cdot W_{zh}^{[\![i]\!]} + \delta_{\tilde{h},t}^{[\![i]\!]} \cdot W_{\tilde{h}h}^{[\![i]\!]} \odot r_t^{[\![i]\!]} + \delta_{h,t}^{[\![i]\!]} \odot (1 - z_t^{[\![i]\!]}).$$

Respectively, $\delta_{r,t}$, $\delta_{z,t}$, $\delta_{\tilde{h},t}$ and $\delta_{h,t}$ denote the derivative with respect to h_{t-1}. Here, we present a calculation formula based on the addition secret sharing protocol. During this time, the values of r_t, z_t, and $\tilde{h}_t$ can be obtained by forwarding propagation.

$$\begin{aligned}
\delta_{r,t}^{[\![i]\!]} &\leftarrow \delta_t^{[\![i]\!]} \odot z_t^{[\![i]\!]} \odot [1 - (\tilde{h}_t^{[\![i]\!]})^2] \odot W_{\tilde{h}h}^{[\![i]\!]} \odot h_{t-1}^{[\![i]\!]} \odot r_t^{[\![i]\!]} \odot (1 - r_t^{[\![i]\!]}), \\
\delta_{z,t}^{[\![i]\!]} &\leftarrow \delta_t^{[\![i]\!]} \odot (\tilde{h}_t^{[\![i]\!]} - h_{t-1}^{[\![i]\!]}) \odot z_t^{[\![i]\!]} \odot (1 - z_t^{[\![i]\!]}), \\
\delta_{\tilde{h},t}^{[\![i]\!]} &\leftarrow \delta_t^{[\![i]\!]} \odot z_t^{[\![i]\!]} \odot [1 - (\tilde{h}_t^{[\![i]\!]})^2], \\
\delta_{h,t}^{[\![i]\!]} &\leftarrow \delta_t^{[\![i]\!]}.
\end{aligned}$$

For the entire sample, its error is the sum of the errors at all times, and the gradient of the weights associated with the previous moment is equal to the amount of the gradients at all times, and the other weights do not have to be accumulated. Let $\gamma \in \{r, z, \tilde{h}\}$. We have,

$$\begin{aligned}
\nabla W_{\gamma,h}^{[\![i]\!]} &\leftarrow \sum_{t=1}^{T} SecMul(\delta_{\gamma,t}^{[\![i]\!]}, h_{t-1}^{[\![i]\!]}), \\
\nabla W_{\gamma,x}^{[\![i]\!]} &\leftarrow SecMul(\delta_{\gamma,t}^{[\![i]\!]}, x_t^{[\![i]\!]}), \\
\nabla b_{\gamma}^{[\![i]\!]} &\leftarrow \sum_{t=1}^{T} \delta_{\gamma,t}^{[\![i]\!]}.
\end{aligned}$$

Let α denote the learning rate of the gradient drop, and α is public. Then we can use the following formula to update the weight matrix and bias.

$$\begin{aligned}
W_{new,\gamma}^{[\![i]\!]} &\leftarrow W_{old,\gamma}^{[\![i]\!]} - \alpha \odot \nabla W_{\gamma}^{[\![i]\!]}, \\
b_{new,\gamma}^{[\![i]\!]} &\leftarrow b_{old,\gamma}^{[\![i]\!]} - \alpha \odot \nabla b_{\gamma}^{[\![i]\!]}.
\end{aligned}$$

Unlike forward propagation, after the backpropagation training complete, all updated encryption parameters are sent to the MSP instead of the MU. And MSP can decrypt the ciphertext by simply adding them together, $W_{new,\gamma} = W'_{new,\gamma} + W''_{new,\gamma}$, and $.b_{new,\gamma} = b'_{new,\gamma} + b''_{new,\gamma}$.

6 Theoretical Analysis

6.1 Correctness

Before the medical user uploads the heart failure specific data, the feature data H is divided into $H = H_1 + H_2$. Then, under our security protocol built on the addition of secret sharing, a large number of linear and nonlinear operations are performed on H. Strictly speaking, the final output prediction result F and the

model parameter NP may not be equal to the value of the original unencrypted algorithm. Here, we demonstrate through the theoretical derivation that the value of the output of our system is highly close to the original value.

First, some of the protocols [10] mentioned in Sect. 3.2 have been proven, and their output is still accurate no matter how many times they are invoked. Secondly, the security functions we construct are all approximated by polynomial. The operations used in these functions are only addition and multiplication. Therefore, in theory, as long as the edge server computing power is strong enough, we can achieve arbitrary calculation accuracy. As long as the accuracy reaches the precision required by GRU, it can be said that our proposed function is additive and correct as of the original function. In addition, since the activation function is composed of a combination of polynomials containing only addition and multiplication. This means that their output ξ satisfies $\xi = \xi_1 + \xi_2$. Finally, we can draw some conclusions and give an arbitrary function F. We have $F = F_1 + F_2$ if and only if $F = f(\zeta_1, \zeta_2, \cdots)$, where $\zeta_i (i = 1, 2, \cdots)$ is a random linear mapping function and xi can be any of the security functions in this paper. Thus, based on the inference, we can ensure that $F = F_1 + F_2$ and $NP = NP_1 + NP_2$, because both forward and backward propagation can be considered as F.

6.2 Security

In this section, we analyse the safety of the proposed PHFP system. To prove the security of the system in this paper, we first need to define what is semi-honest security [9] formally.

Definition 1. *We say that a protocol s secure if there exists a probabilistic polynomial-time simulator $\mathcal{S}$ that can generate a view for the adversary $\mathcal{A}$ in the real world and the view is computationally indistinguishable from its rear view.*

In addition to the Lemma 1 mentioned in Sect. 3, also need the following lemmas.

Lemma 2 [9]**.** *If a random element r is uniformly distributed on $\mathbb{Z}_n$ and independent from any variable $x \in \mathbb{Z}_n$, then $r \pm x$ is also uniformly random and independent from x.*

Lemma 3 [10,12]**.** *The protocols SecAdd, SecMul, SecCmp and SecCon are secure in the semi-honest model.*

According to Lemma 3, we only need to verify the safety of other protocols. *The protocols σ_{sec} and $\tanh_{sec}$ are secure in the semi-honest model.*

Proof. In σ_{sec}, given the order of the polynomial n, what ES_1 holds is receiver $Rec_1 = (u_1, G_1', F_1', \alpha')$, where $G_1' = g_0', g_1', \cdots, g_n'$ and $F_1' = f_0', f_1', \cdots, f_{n-1}'$. And g_i' and f_i' are respectively the outputs of *SecMul* and *SecAdd*. In the meantime, with u_1, they also compose the inputs of the next iteration. According to Lemma 3, it is guaranteed that G_1' and F_1' are sets of uniformly random values.

So they can all be correctly simulated by simulator ES_1, and are unable to distinguish by the adversary $\mathcal{A}$ in polynomial time. Similarly, ES_2 can also hold Rec_2 which is simulatable and distinguishable. In addition, $\tanh_{sec}$ protocols are implemented by a similar polynomial composed of protocols and can be proved to be secure.

7 Performance Evaluation

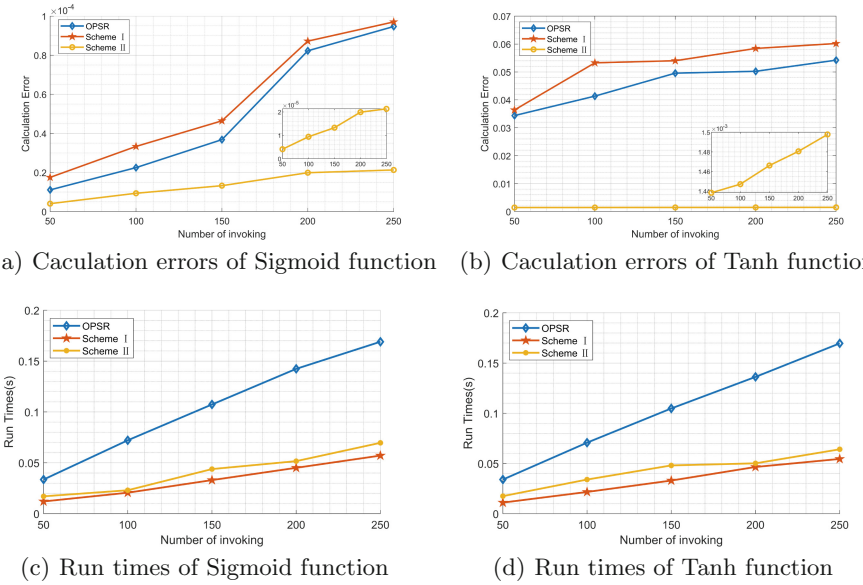
To implement our framework, we utilise NumPy for parallel computation of matrixes in Python 3. All the data is encrypted on a personal computer with an Intel(R) Core (TM) i7-6700 CPU @3.40 GHz and 8.00 GB of RAM. Then, the ciphertexts respectively sent to two edge servers for privacy-preserving GRU training and pre-trained heart failure prediction. Each server is equipped with an Intel(R) Core (TM) i7-7700HQ CPU @2.80 GHz and 8.00 GB of RAM. Also, to obtain the correct pre-diagnosis results in the above evaluation environment, we considered a real data set from the UCI machine learning library called Arrhythmia to evaluate the accuracy and efficiency of our solution. The selected Arrhythmia dataset contains 452 instances, each of which includes 279 attributes (such as age, weight, gender, heart rate, QRS duration, P-R interval, Q-T interval, T interval, P interval, etc.)

7.1 Performance of Secure Sigmoid and Tanh Function

In the PHFP system, we tried two approaches to approximate the activation function in the GRU neural network. To avoid the local fitting problem of Taylor series, we finally use the least squares method to construct high-order polynomials to approximate the *Sigmoid* and *Tanh* functions. Since each hidden unit of the GRU contains two *Sigmoid* functions and one *Tanh* function, when our PHFP system has multiple hidden units, the secure *Sigmoid* and *Tanh* functions are invoked multiple times. Therefore, we evaluated the performance of the scheme II security function under different number of calls, and we also compared it with scheme I and OPSR scheme [12], as shown in Fig. 3(a)–(d). From the figure, we can see that scheme II is both accurate and efficient. It is better than the other two programs. The reasons summarise as follows: Firstly, since scheme II uses the least squares method to fit the activation function, the local fitting problem of the Taylor series is avoided, and the accuracy is improved to some extent. Secondly, scheme II adopts a scheme of directly constructing a security function, which prevents the time overhead caused by multiple invokes of security components. To sum up, scheme II is more suitable for our heart failure prediction system in terms of accuracy and efficiency.

7.2 Performance of PHFP

Accuracy Evaluation. To further evaluate the performance of the PHFP, we deployed the constructed safety components to our system to assess the accuracy of the system's forward propagation calculations. At the same time, we also



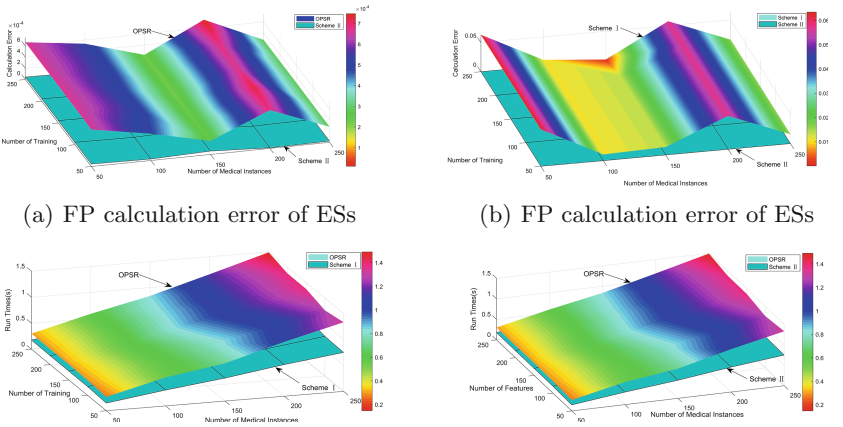
(a) Caculation errors of Sigmoid function (b) Caculation errors of Tanh function

(c) Run times of Sigmoid function (d) Run times of Tanh function

Fig. 3. Performance of secure Sigmoid and Tanh function

deployed the components built by [12] into our GRU neural network and used the same data set to evaluate the computational error of forwarding propagation. As shown in Fig. 4(a)–(b), since the numerical range of our dataset is not entirely concentrated on a certain point, the error of the scheme II we constructed in forwarding propagation is significantly better than the other two schemes. This benefit from the nature of the global fit of the least squares method. It is noteworthy that when medical users predict heart failure, only the process of forwarding propagation is needed, while the calculation error of forwarding propagation is controlled within 10^{-5}, which can be neglected in actual heart failure prediction.

Efficiency Evaluation. In PHFP, the primary function of ESs is to calculate the user's data and train the model provided by the medical service provider. Both secure forward propagation and secure backpropagation are involved in training the model. However, the number of features of medical data, the number of medical instances and the number of GRU hidden layers have an essential impact on the computing cost of ESs. Accordingly, we first tested the computational overhead of ESs with a different number of features and a different number of medical cases. Here, we default the number of hidden layers of GRU to 20, and we compare scheme I and scheme II with OPSR. As shown in Fig. 4(c)–(d), since we adopted the idea of directly constructing sigmoid and tanh functions, and avoiding the overhead caused by repeated calls to multiple components, our two schemes are significantly better than the OPSR scheme in terms of computational cost. Besides, we noticed that in the scheme II adopted by the PHFP



(a) FP calculation error of ESs (b) FP calculation error of ESs

(c) FP calculation overhead for ES (d) FP calculation overhead for ES

Fig. 4. ESs efficiency evaluation

system, although we let the ES perform the forward propagation calculation of 250 medical cases with 250 features, its calculation time is less than one second. At the same time, we also evaluated the computational overhead of backpropagation ESs.

8 Conclusion

In this paper, we proposed a privacy-preserving heart failure prediction system based on Secure Multiparty Computation and Gated Recurrent Unit, named PHFP. The PHFP system was adopted to protect the privacy of users' heart failure prediction data and the security of neural network parameters of medical service providers with high accuracy and low computing cost. Accurately, the program randomly split the heart failure prediction data and neural network parameters into secret sharing, and the edge server calculated the user data in the state of ciphertext. Therefore, the medical service provider cannot obtain the user's private data, and the user cannot receive any neural network parameter information of the medical service provider. Finally, we use a large number of experiments to prove the effectiveness of the system.

Acknowledgment. This research is supported by the key project of Anhui provincial department of education (Grant No. KJ2018A0031), the National Natural Science Foundation of China under Grant Nos. U1804263 and 61702105.

References

1. Tripoliti, E.E., Papadopoulos, T.G., Karanasiou, G.S., Naka, K.K., Fotiadis, D.I.: Heart failure: diagnosis, severity estimation and prediction of adverse events through machine learning techniques. Comput. Struct. Biotechnol. J. **15**, 26–47 (2017)
2. Shoaib, A., et al.: Mode of presentation and mortality amongst patients hospitalized with heart failure? A report from the first euro heart failure survey. Clin. Res. Cardiol. **108**(5), 510–519 (2019)
3. Choi, E., Schuetz, A., Stewart, W.F., Sun, J.: Using recurrent neural network models for early detection of heart failure onset. J. Am. Med. Inform. Assoc. **24**(2), 361–370 (2016)
4. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: privacy beyond k-anonymity. In: 22nd International Conference on Data Engineering (ICDE 2006), pp. 24–24. IEEE (2006)
5. Narayanan, A., Shmatikov, V.: Myths and fallacies of "personally identifiable information". Commun. ACM **53**(6), 24–26 (2010)
6. Liu, X., Zhu, H., Lu, R., Li, H.: Efficient privacy-preserving online medical primary diagnosis scheme on Naive Bayesian classification. Peer-to-Peer Network. Appl. **11**(2), 334–347 (2018)
7. Ning, J., Xu, J., Liang, K., Zhang, F., Chang, E.-C.: Passive attacks against searchable encryption. IEEE Trans. Inf. Forensics Secur. **14**(3), 789–802 (2018)
8. Pullonen, P., Matulevičius, R., Bogdanov, D.: PE-BPMN: privacy-enhanced business process model and notation. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 40–56. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_3
9. Bogdanov, D., Laur, S., Willemson, J.: Sharemind: a framework for fast privacy-preserving computations. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 192–206. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88313-5_13
10. Huang, K., Liu, X., Fu, S., Guo, D., Xu, M.: A lightweight privacy-preserving CNN feature extraction framework for mobile sensing. IEEE Trans. Dependable Secure Comput. (2019)
11. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 420–432. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_34
12. Ma, Z., Liu, Y., Liu, X., Ma, J., Li, F.: Privacy-preserving outsourced speech recognition for smart IoT devices. IEEE Internet Things J. **6**, 8406–8420 (2019)
13. Greenspan, D.: Numerical Analysis. CRC Press, Boca Raton (2018)
14. Bos, J.W., Lauter, K., Naehrig, M.: Private predictive analysis on encrypted medical data. J. Biomed. Inform. **50**, 234–243 (2014)
15. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis, vol. 12. Springer, Heidelberg (2013)