Neural Computing and Applications (2023) 35:14667–14680
https://doi.org/10.1007/s00521-021-05988-7

S.I: ML4BD_SHS

Deep learning-enabled block scrambling algorithm for securing telemedicine data of table tennis players

Bo Yang[1] · Bojin Cheng[2] · Yixuan Liu[3] · Lijun Wang[4,5]

Received: 14 December 2020 / Accepted: 25 March 2021 / Published online: 26 April 2021

Abstract

In sports, advance sensing technologies generate massive amount of unstructured telemedicine data that need to be refined for accurate diagnosis of underlying diseases. For accurate prediction of diseases and classification of athletes' data, deep learning algorithms are frequently used at the cloud. However, the transmission of raw data of athletes to the cloud faces numerous challenges. Among them, security and privacy are a major challenge in view of the sensitive and personal information present within the unstructured data. In this paper, first we present a data block scrambling algorithm (without key management) for secured transmission and storage of ECG (electrocardiogram) data of table tennis players at the cloud. A small piece of original data stored at the cloud is used for scrambling the massive amount of remaining ECG data. The secured telemedicine data is then imported into Hadoop Distributed File System for data management, which is read by Spark framework to form Resilient Distributed Datasets. Finally, a deep learning approach is used that extracts useful features, learns the related information, and weights and sums the feature vectors at different layers for classification. Theoretical analysis proves that our proposed approach is highly robust and resilient to brute force attacks and at the same time has a much better accuracy, sensitivity, and specificity as compared to the existing approaches.

Keywords Telemedicine · Table tennis players · Big data · Security · Deep learning

1 Introduction

Internet of Things (IoT) is expanding on an unprecedented scale because of the idea to connect virtually any real-world physical device with the Internet [1]. This vision of IoT has attracted a significant amount of research from various sectors. Among them, sports sector has seen keen interest from the researchers in recent years. In sports, the sensor-embedded IoT devices allow the athletes to improve their performance by monitoring and providing feedback on their progress [2]. Gathering data from the wearable sensors of athletes allows the teams to improve their game strategy, understand the weaknesses of the opponents and make efficient draft selection of their players. In sports, connected footwear is used to intelligently track the speed and footwork of the athletes. Connected apparel, e.g., shirts and similar clothing items embedded with sensors, allows tracking the overall fitness of the athletes by monitoring their respiration rate, heart rate, and muscle usage. These embedded sensors are extremely beneficial for well-being of the athletes by maintaining a balanced routine for them [3].

In sports, the sensor-embedded IoT devices generate massive amount of unstructured telemedicine data that contain outliers and higher correlation. To minimize the redundancy in athletes' data, novel machine learning and deep learning algorithms need to be designed to extract useful features from the raw telemedicine data [4]. Storing and processing the redundant, lossy and noisy data at the

✉ Lijun Wang
wanglj432@ylu.edu.cn

[1] Department of Physical Education and Health, Zhaoqing University, Zhaoqing 526061, Guangdong, China

[2] Department of Physical Education, Guangzhou Sport University, Guangzhou 510500, Guangdong, China

[3] Zhaoqing Engineering Technical School, Youth League Committee, Zhaoqing 526061, Guangdong, China

[4] Institute of Physical Education and Health, Yulin Normal University, Yulin 537000, China

[5] Institute of Physical Education, Soochow University, SuZhou 215000, JiangSu, China

remote cloud data centers not only consume valuable resources but may also lead to catastrophic health-related circumstances. The role of machine learning and deep learning algorithms cannot be ignored because they have the potential to remove outliers and redundancy by ensuring that only highly refined telemedicine data is available at the decision support system for making critical decisions about the athletes' health [5, 6]. These algorithms ensure that useful features are extracted from massive raw data to take accurate decision for time-critical and delay-sensitive applications in sport informatics.

In sport informatics, the massive data of the athletes faces numerous challenges. Security is one such challenge that is constantly threatened by the presence of malevolent entities [7]. These entities have the ability to jeopardize the operation of underlying devices and are seen as a constant threat to the broadcast of telemedicine data containing confidential information about the athletes. Any malicious injection to the in-transit data may have catastrophic effect on the performance of machine learning and deep learning algorithms that are executed at the cloud [8]. Although most of telemedicine platforms are equipped with built-in security solutions, there are numerous shortcomings when it comes to the design of telemedicine health applications for athletes. For example, it is impossible to fully guarantee that professional medical staff follows conventional information security measures. Moreover, it is not possible to be fully assured that the privacy of athletes will not be leaked when they lack sufficient cyber-security knowledge and skills [9]. Therefore, security solutions for telemedicine healthcare systems of athletes need to be studied further. In sport informatics, the athletes' physiological data is transmitted toward the remote cloud for storage. Hence, they no longer have control of their data. If these athletes cannot set a safe and reliable key in accordance with the requirements of security regulations, the data will be vulnerable to a wide range of malevolent threats and adversarial attacks.

Machine learning [10–12], deep learning [13–15], and neural networks [3, 16, 17] are frequently used in sports informatics for medical image analysis, electronic health records, image interpretation, disease prediction and diagnosis, and injury risk prediction of the athletes. These approaches allow the extraction of useful features from the raw telemedicine data at the cloud [18]. Unlike machine learning and neural networks, deep learning uses many hidden neurons and layers that enable the extensive coverage and high-level feature extraction from the raw telemedicine data of athletes [19]. For example, in medical imaging of table tennis players, deep learning approaches are capable to generate features, which are highly sophisticated and are extremely difficult to elaborate by descriptive means [20]. The latest trends of integrating security and differential privacy approaches with deep learning algorithms have ensured that only highly reliable data of athletes is subject to feature extraction and data mining at the cloud [21]. The heterogeneous data generated in sports informatics is massive and difficult for traditional approaches to analyze and extract useful features. The sensitive nature of the telemedicine data coupled with their heterogeneous sources exposes the data and their origin to numerous vulnerabilities. Prompt actions and decisions need to be taken to ensure the safe transmission of data along with extracting features from reliable data only. Besides, a platform needs to be in place to host and categorize this unlabeled data for mining [22, 23]. The platform should be capable to handle batch processing as well as real-time processing of telemedicine data to avoid any realistic possibility of injury sustained by the athletes [24, 25].

To accomplish the aforementioned tasks, we propose a deep learning-enabled data block scrambling algorithm that serves two purposes: preserving the security and privacy of massive telemedicine data, and mining useful features from the secured data at the cloud. The proposed algorithm operates on the ECG data of table tennis players. These players are capable to complete the encryption and decryption of sensitive data without the user's setting and memorizing passwords. The major contributions of this paper are as follows.

1. Our proposed algorithm uses a small piece of original data to scramble the rest of the data, i.e., big data. This small piece of data is stored at a secured private cloud space as a key to decrypt the rest of data. The authorized medical center personnel can read the key pair at any time. The ECG data of the remote user/patient, i.e., the table tennis player, is decrypted and restored, and the rest of the scrambled and encrypted data is sent to the remote medical center for storage and backup.
2. Our proposed algorithm enables the data encryption and decryption without the athletes providing a key, thereby effectively avoiding security threats caused by weak keys. After theoretical and experimental analysis, the proposed data block and scrambling algorithm can effectively remove the statistical characteristics and outliers from the original data.
3. Based on the SparkDL memory computing framework, the k-means improved algorithm is used to optimally divide the scrambled ECG data into groups (histograms) and the average of each group is calculated. Laplace noise is added to each group to maintain their differential privacy. By combining the features of scrambled data, Big Data Analytics, CNN, and RNN, a 4-layered neural network architecture is used for the ECG signal recognition by extracting useful features,

learning the related information about the ECG data, and classifying the weighted and summed feature vectors.

The rest of the paper is organized as follows. In Sect. 2, the architecture and design of our system model are discussed. Our proposed approach for data block scrambling and DL-enabled feature extraction is discussed in Sect. 3. In Sect. 4, the performance analysis and experimental results are provided. Finally, the paper is concluded and future research directions are provided in Sect. 5.

2 System model: architecture and design

In this section, we discuss the architecture and design of our proposed approach. The system model consists of three main components: data block scrambling, big data analytics, and deep learning. Initially, the ECG data of the table tennis player is divided into N blocks of equal size. The first block (A0) is scrambled using a scrambling function and stored at the third party cloud. At the cloud, bitwise inversion operation is performed. The remaining N-1 blocks of the ECG data are scrambled and stored at the remote cloud for healthcare services. Instead of Bitwise inversion, these blocks are subject to block inversion. Scrambling allows the ECG data to be converted into an encrypted form. The scrambled blocks are decrypted using scrambling functions that retrieve the original ECG data of the athletes. Upon retrieval, the secured data is stored in HDFS that are read using Spark framework to form RDD datasets. The datasets contain non-correlated and redundant data that needs to be refined properly. The useful features of the secured data need to be extracted. For this purpose, a simple DL approach is adopted. The system architecture and its design are shown in Fig. 1.

3 Proposed work

In this section, first we discuss the encryption and decryption operations performed by data block scrambling algorithm on telemedicine data in Sect. 3.1. Next, the extraction of useful features from secured telemedicine data using a deep learning approach is discussed in Sect. 3.2.

3.1 Encryption and decryption of Athlete's data

The operational behavior of data block and scrambling algorithm is shown in Fig. 2. In this figure, A_0 is the first data block of the original data, which will be stored in the third-party secure cloud space as a key, and the remaining data will be partitioned, scrambled, and synchronously transmitted to a remote medical center (cloud) for storage. The parameter k represents the current number of blocks and scrambles. For different data types, a reasonable configuration of the number of cycles can effectively eliminate the statistical characteristics of the original data.

First, we determine the size l_0 of the first data block A_0. In order to better adapt to different types of data, l_0 allows random selection between $[L_{\min}, L_{\max}]$, where $L_{\min}$ and $L_{\max}$ represent the minimum and maximum number of bytes of the data block, respectively. The values of $L_{\min}$ and $L_{\max}$ need to be configured in advance. The configuration method will be discussed later. Next, A_0 needs to be scrambled by the scrambling function S_1. The scrambling method is shown in Fig. 3, where a_i represents a single byte of the first data block A_0. Before the exclusive *OR* operation (XOR) of a_i and a_{i+1}, the Hamming weight needs to be analyzed. If the weight is odd, it should be inverted first. It can expand the range of data effectively and make the algorithm flexibly applied to any type of data.

Next, the rest of the data will be divided into blocks of different lengths using Eq. 1.

$$l_i = HW(A_{i-1}) \bmod (L_{\max} - L_{\min} + 1) + L_{\min}(i = 1, \ldots, n) \tag{1}$$

when $i \geq 1$, l_i represents the byte size of data block A_i, and $HW(A_{i-1})$ represents the Hamming weight value of binary sequence of data block A_{i-1}.

It can be observed from Eq. 1 that the length of data block has a certain degree of randomness. When the data block A_0^1 is unknown, the attacker cannot easily guess the size of the subsequent block, thereby effectively improving the security. It should be noted that the data block length A_{i-1} can be calculated by A_0, so there is no need to save length information. This feature makes this algorithm more practical and flexible. Finally, we use the function S_2 to perform a scrambling operation on the partitioned data, and the scrambling process is shown in Fig. 4. Among them, the result of the previous data block is used to scramble the next data block, i.e.,

$$\begin{cases} A_1^1 = A_0^1 \oplus A_1 \\ A_i^1 = A_{i-1} \oplus A_i \end{cases} \quad (i \geq 2). \tag{2}$$

A_i^1 represents the ith block of data that has been scrambled earlier. Similar to the scrambling function S1, if the Hamming weight of the data block is odd, it should be inverted bitwise first.

Since the size of data block is obtained randomly, hence the size of two adjacent data blocks is likely to be different. While performing an XOR operation on a data block, if the previous data block has fewer bytes, the block is recycled to match the size of next data block.

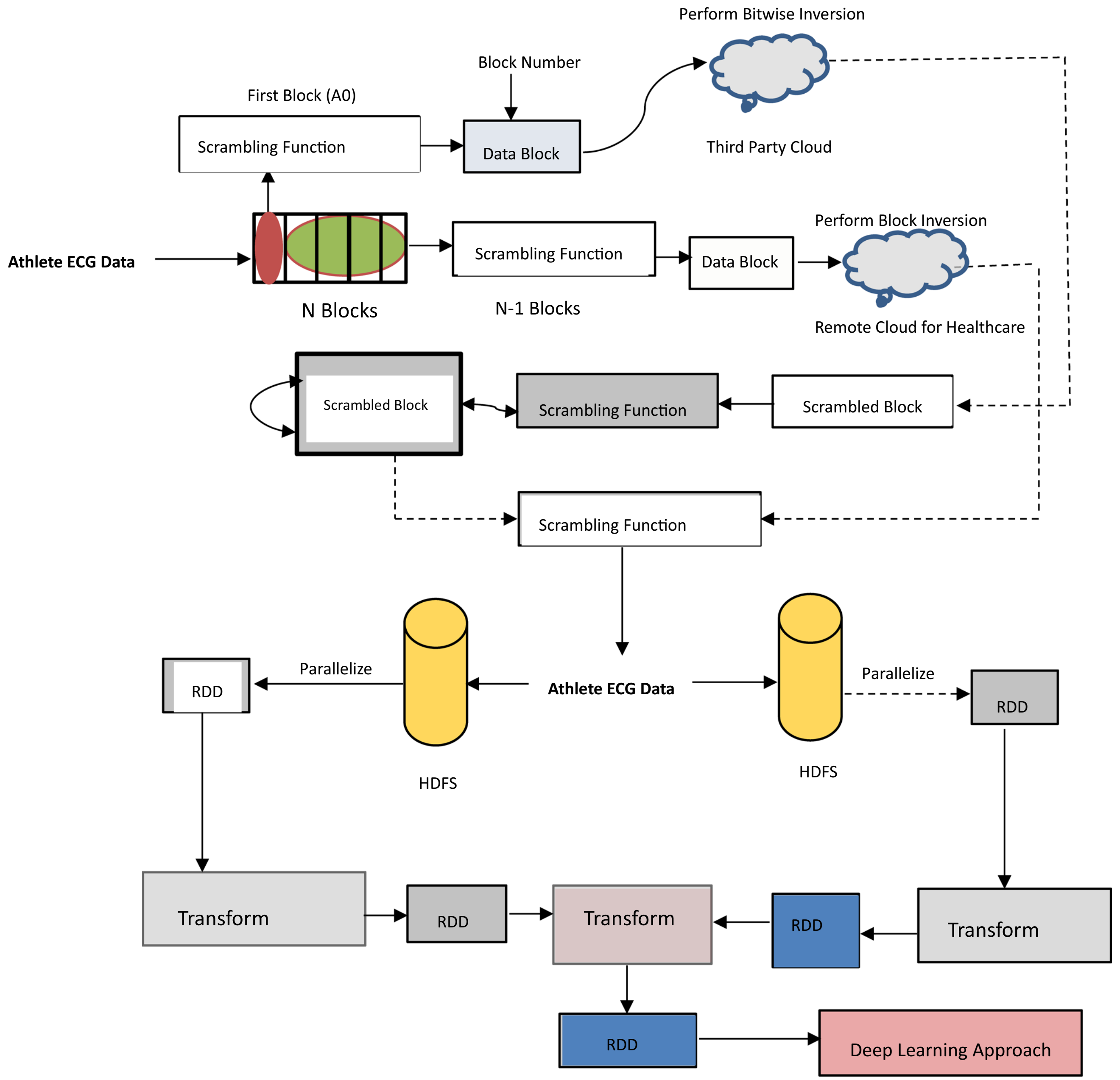


Fig. 1 System architecture and design

We denote the data block as $A_i = \{a_{k|i} | 0 \le k \le l_i\}$, where k is an integer, representing the kth byte of data block A_i, and the first round of XOR operation of two adjacent data blocks can be expressed as:

(a) If the current block is greater than or equal to the next block, i.e., when $l_{i-1} \ge l_i$, then

$$A_i^1 = \{a_{k|i} \oplus a_{k|i-1} | 0 \le k \le l_i\}. \tag{3}$$

(b) If the current block is less than or equal to the next block, i.e., when $l_{i-1} < l_i$, then

$$A_i^1 = \{a_{k|i} \oplus a_{k|i-1} | 0 < k < l_i\} | \{a_{k'|i} \oplus a_{(k'-I_{i-1})|i-1} | l_{i-1} \le k' < l_i\}. \tag{4}$$

Here, $1 \le i \le N$ and the symbol | indicates the connection operation. The XOR operation with A_1 produces a data block headed by A_0^1 of A_1^1, which cannot be XOR with the previous data block. In this special case, the scrambling operation is realized by cycling the XOR before and after the bytes in the block A_0^1.

If the number of block scrambling rounds is set to t ($t > 1$), then the cycle operation starts from the second

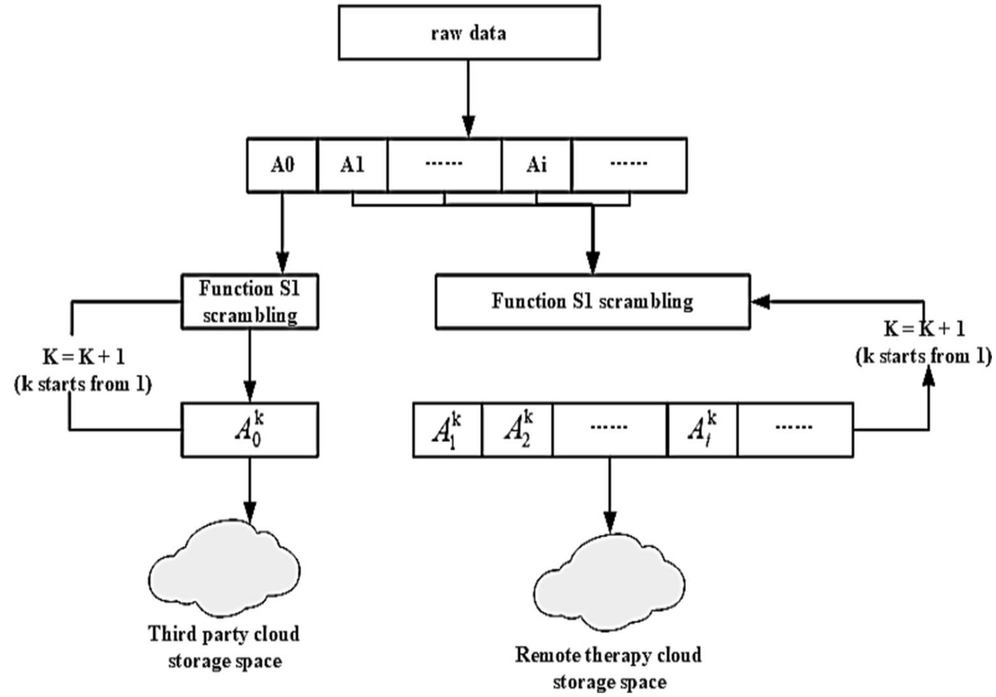
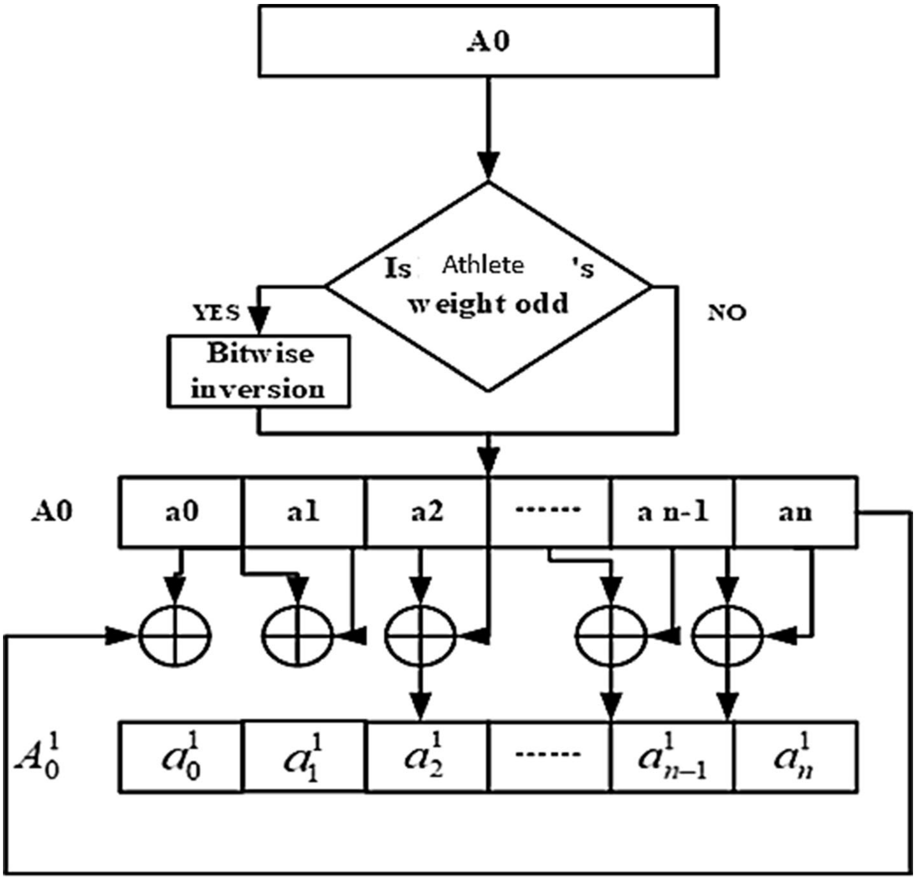


Fig. 2 Data block and scrambling algorithm

Fig. 3 Scrambling function S1 processes the telemedicine data



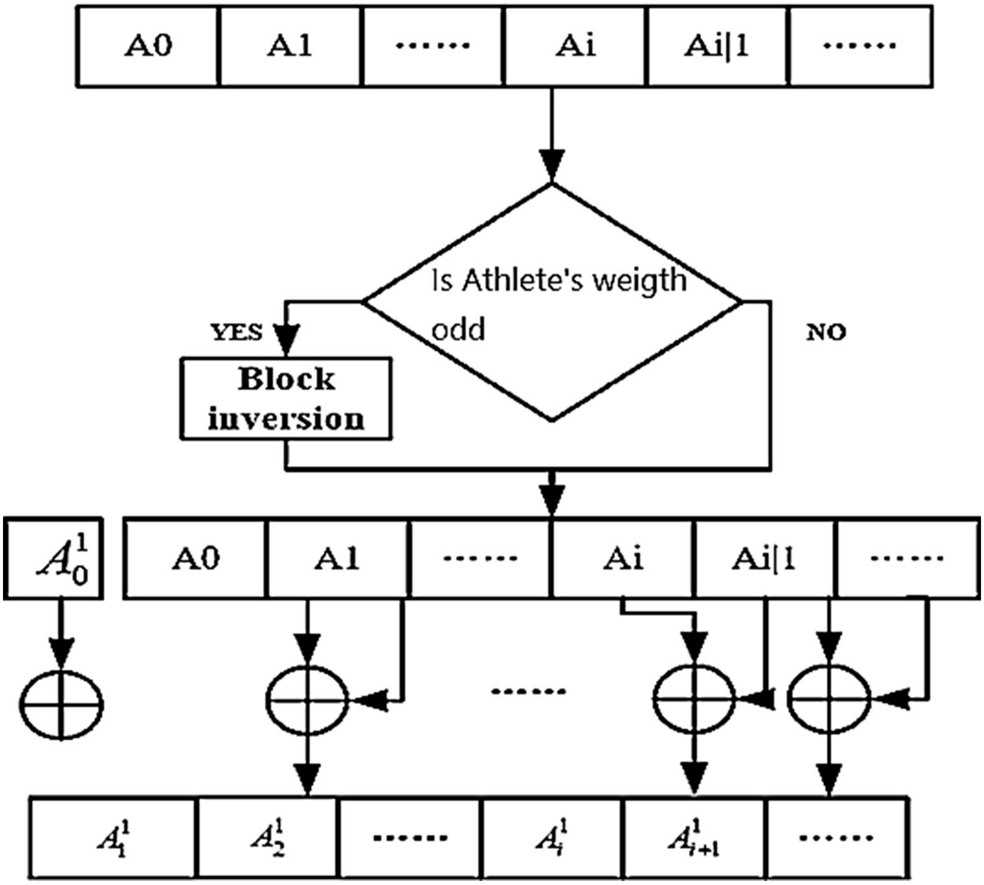


Fig. 4 Scrambling function S2 processes the telemedicine data

step. The final result after the *t*th round of blocking and scrambling can be expressed as $\{A_i^t|i = 1, 2, ...\}$. Except for the first data block A_0 of the original file that is stored offsite for decryption, the rest of the ciphertext with random characteristics will be sent to the cloud for storage.

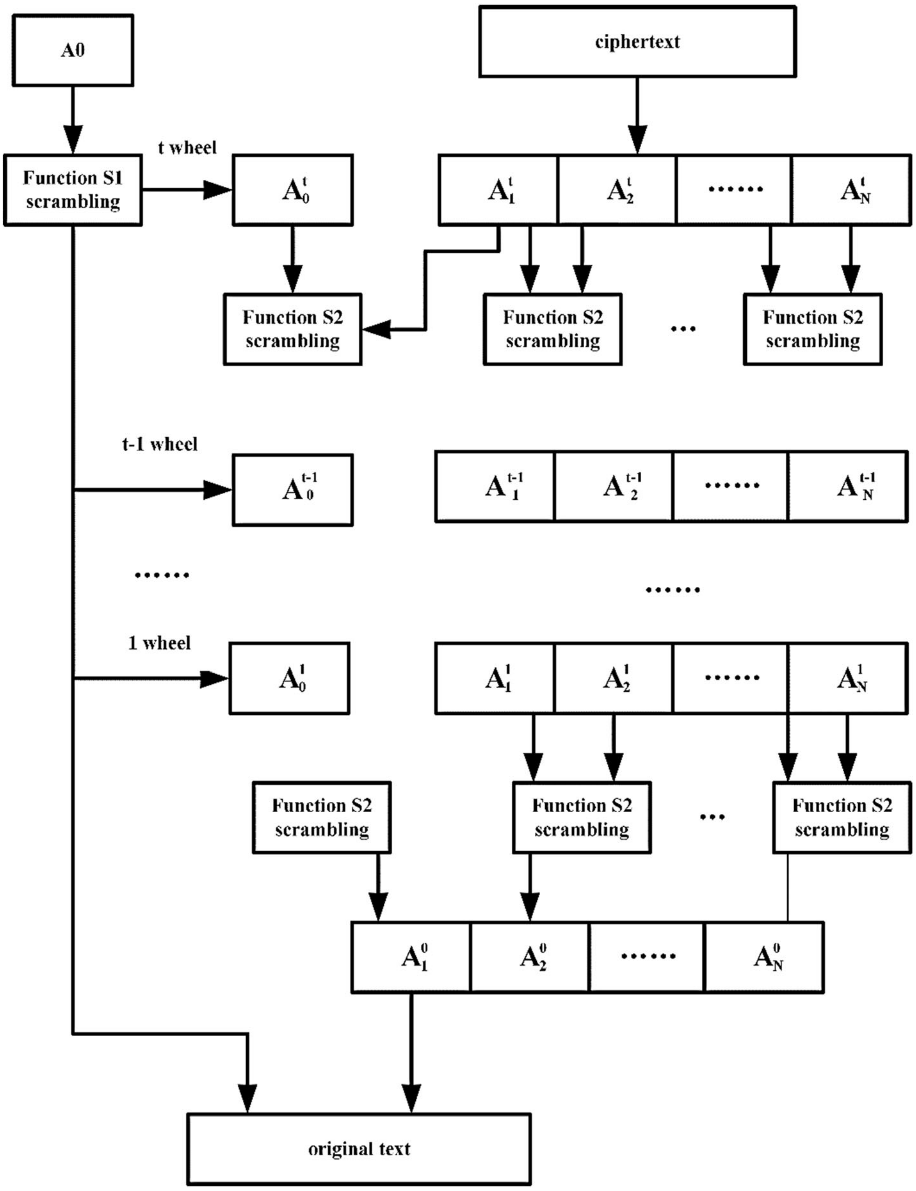
The decryption operation of $\{A_i^t|i = 1, 2, \ldots\}$ is the reverse process of the encryption operation, as shown in Fig. 5. If the data encryption process performs *t* rounds of scrambling, then the decryption process must complete the reverse operation with the same number of rounds. Prior to initiating the decryption operation, the *k* rounds of scrambling results are calculated. This calculation aims to determine the size of first block of decrypted data and also decrypt the block at the same time. Subsequently, the previous data block is used to calculate the size of current block, and the decryption result of the previous data block is used to decrypt the current block. For example, in the first round of decryption, $A_0^t(1 \leq k \leq t)$ is used to calculate the size of first block of our input telemedicine data. When the length of first data block A_1^t is determined, the decryption result can be generally be expressed using Eq. 5.

$$\begin{cases} A_1^{k-1} = A_0^k \oplus A_1^k \\ A_i^{k-1} = A_{i-1}^{k-1} \oplus A_i^k \end{cases} (i \geq 2). \tag{5}$$

Here, the XOR operation is similar to the one used for encryption of telemedicine data. When A_1^{t-1} is determined, it can be used to calculate the size of A_1^2, and then A_2^{t-1}, i.e., $A_2^{t-1} = A_1^{t-1} \oplus A_2^t$, can be calculated according to Eq. 5. After the blocking and decryption process is executed *t* times, the original data is restored.

In decryption, only two parameters, $L_{\min}$ and $L_{\max}$, need to be set in advance. These parameters represent the minimum byte length and the maximum byte length of the data block. When setting the value of $L_{\min}$, there is a trade-off between the required security level and the size of local data. The larger the value of $L_{\min}$ is, the more difficult it is for unauthorized parties to obtain A_0 information, thereby increasing the security. On the other hand, if the value of $L_{\min}$ is too large, the size of A_0 will be too large, which will increase the storage space consumption. Therefore, under the premise that the security level is acceptable, it is recommended to set $L_{\min}$ value to 32 bytes, i.e., the minimum length of the source data block is set to 32 bytes.

Fig. 5 Decryption procedure for telemedicine data



Another condition for us to choose A and B is that the data block size should be randomly selected between $L_{\min}$ and $L_{\max}$ according to its statistical significance, i.e., $HW(A_{i-1}) \bmod (L_{\max} - L_{\min} + 1)$ as depicted in Eq. 1, changes uniformly between [0, $L_{\max} - L_{\min}$]. Since the byte size of A_0 lies between [$L_{\max} - L_{\min}$], the statistically average Hamming weight of A_0 is

$$\overline{HW(A_0)} = \frac{L_{\max} + L_{\min}}{2} * 4 = 2(L_{\max} + L_{\min}). \tag{6}$$

Therefore, the statistical average result of the modulo operation of Eq. 1 is

$$\begin{aligned} &\overline{HW(A_0) \bmod (L_{\max} + L_{\min} + 1)} \\ &= 2(L_{\max} + L_{\min}) \bmod (L_{\max} - L_{\min} + 1)\,. \\ &= 4(L_{\max} - 2) \bmod (L_{\max} - L_{\min} + 1) \end{aligned} \tag{7}$$

In order to obtain a statistically uniform distribution, it should satisfy

$$\begin{aligned} &4L_{\max} - 2 \geq L_{\max} - L_{\min} + 1, \text{ i.e.} \\ &L_{\max} \leq 5L_{\min} + 1. \end{aligned} \tag{8}$$

3.2 Deep learning-enabled feature extraction from secured telemedicine data

Upon encryption and decryption of telemedicine data, the statistical characteristics of the data are taken as the starting point for data mining and feature extraction. We use a deep learning approach to extract useful features and at the same time privacy of the data is preserved using an integrated differential privacy mechanism. Based on the SparkDL memory computing framework [26, 27], the k-means improved algorithm is used to optimally divide the telemedicine data (in the shape of histogram group), and calculate the average value of each group. The Laplace noise is added to each group to maintain and preserve the differential privacy.

In this section, a non-interactive computing framework that meets the needs of differential privacy protection is proposed. The framework is mainly composed of three parts: original data collection and storage, data processing under the Spark framework, and privacy preservation. First, we import the secured telemedicine data of the athletes into Hadoop Distributed File System (HDFS) for data management. The data is read from HDFS into the Spark framework to form Resilient Distributed Datasets (RDD), map operations are performed, join operations and shuffle processes are executed, and finally, the RDD processing results are generated and saved to HDFS [28]. The privacy protection of this data (to be released) mainly uses the Spark parallel computing framework to implement calculation tasks such as classification and statistics, feature extraction, and clustering of the preprocessed data. After the clustering is completed, the original large dataset is divided into different small datasets. These datasets are sorted, their difference is calculated and the small datasets are grouped after clustering. After the grouped data is averaged within the group, Laplace noise is added to the average of each group to obtain the released dataset, and the histogram of the released data after differential privacy protection is released.

3.2.1 k-means optimization

The k-means clustering algorithm calculates the Euclidean distance between the samples and the cluster center. This algorithm selects the closest center and classifies the samples of telemedicine data (histograms). The use of this algorithm generates a large amount of calculation, especially for distributed computing, since the sample data are stored on different nodes. As a result, it will bring large communication overhead. Therefore, the distance optimization approach [29] adopts the optimization measures that associate the sample data with its two norms to avoid repeated calculations of distances, thereby reducing the computational cost of the k-means clustering process. That is, the coordinates (x, y) of the data point are associated with its second normal form to form $<x, y, \|(x, y)\|^2>$ a key-value pair form, and the square value of the difference between the second norm (*bound Distance*) and the nearest center point (*best Distance*) are calculated and compared. The main steps of the improved k-means algorithm are as follows:

- Initialize k data as initial cluster centers to form sample clusters.
- Traverse the data sample, if $boundDistance < bestDistance$, then perform real Euclidean distance calculation (real-Distance); if $realDistance < bestDistance$, classify the smallest distance to the cluster center to form k clusters.
- Calculate the mean value of the data in each cluster and update the cluster center.
- Loop Step1 $\sim$ Step3 until the specified number of iterations is reached or the clustering converges and the cluster center does not change.
- Output the results of clustering processing.

3.2.2 SPDL-GS algorithm description

The stochastic propositional dynamic logic-Gale Shapley (SPDL-GS) algorithm described in this section mainly focuses on the realization of data type statistics, k-means clustering, grouping averaging, and differential privacy. Since classification statistics is out of the scope of this article, we limit our discussion to the specific implementation for group division and noise addition as shown in Algorithm 1.

Algorithm 1: k-means clustering and grouping algorithm

```
Input: $D\{x_1, x_2, \ldots, x_n\}$ //D is the dataset after statistical classification using Hash_map algorithm.
Output: $C = \{c_1, c_2, \ldots, c_k\}$ // C is the clustering group of k clusters, where the mean of the group is $u_{c_j}$ and the number of data in the group is $num_{c_j}$
Begin
  KMeansCluster(hashmapResult)
  { Kmeans.setMax (k)
    sdata=kmeans.loadData(hashmapresult)；   //Read the statistical classification result dataset
    for i=1:1: n
      { for j=1:1: k {
          if ((boundDistance<bestDistance) & (realDistance<bestDistance)) then
            bestDistance<realDistance;
          else $realDistance = \|x_i - u_j\|_2$
          $c_j = \arg\min(realDistance_{ij})$
          $u_{c_j} = \frac{1}{|c_j|} \sum_{c_j} x$
          $num_{cj} = num_{cj} + 1$ }}
  result:RDD[(int, $C_k$)]; // The clustering results are stored in DD
End
```

3.2.3 Convolutional loop network architecture

By combining the features of encrypted telemedicine data of athletes (ECG signals), Big Data Analytics, CNN, and RNN, we propose a novel neural network structure for the ECG signal recognition task. The network structure is shown in Fig. 6, which consists of four parts from bottom to top, i.e., the convolutional layer, the RNN layer, the attention layer, and the classification layer. The convolutional layer is also called the feature extraction layer. It mainly uses the powerful feature extraction capabilities of the convolutional neural network to automatically extract features from the input ECG signal, and then send the extracted feature sequence to the RNN layer. The main function of the RNN layer is to learn the related information of the ECG signal through the feature sequence and then send the feature vector output to the attention layer. The attention layer automatically learns the weights assigned by the feature vectors at RNN layer. This layer weight and sum these feature vectors to obtain the weighted and summed feature vectors and send them to the classification layer for classification. The network has the following characteristics: (1) there is no need to manually design features, and the useful information can be learned directly from the original ECG signals, (2) it can process variable-length ECG signals and make full use of the related information before and after the ECG signals, (3) end-to-end learning from the ECG signal is possible.

4 Performance analysis of the proposed work

Our data block scrambling algorithm operating on the ECG data of table tennis players is highly resilient to brute force attacks and at the same time operates without key management. In this section, first we present the safety analysis of our proposed algorithm in Sect. 4.1 followed by experimental analysis and results in Sect. 4.2.

4.1 Safety analysis

The main idea of the algorithm proposed in this paper is to randomly partition the source data of the players, and use the first data block to scramble the rest of A_0. Because of the ciphertext stored at the cloud, the attacker has to guess the first data block or directly attack the potential statistical characteristics of the ciphertext. Based on the random nature of ciphertext, if a brute force attack is performed on the first data block A_0^1, the average number of searches will reach 2^{l_0*8-1}. If a brute force attack is performed on the first block A_0^t after round t scrambling, the actual number of searches will reach

$$M = 2^t \times 2^{l_0*8-1}. \tag{9}$$

Considering that the scrambling process uses the cyclic XOR operation of Eq. 2, when the actual number of searches reaches M, the time overhead will reach $M = 2^t \times 2^{l_0*8-1} * 0(S_2)$. If the attacker directly attacks the first data block A_0, regardless of other computing overhead, when using brute force search to attack the encrypted ciphertext after t rounds, if the length of A_0 is taken as the aforementioned recommended value of 32, the average number of attacks will reach B. When the value of t is 6, and assuming a time of $1\mu s$(microsecond) is used to test A_0, it takes about 1.175*1065 years. However, considering that the first data block has significant statistical characteristics, the actual anti-attack strength will be greatly reduced, so

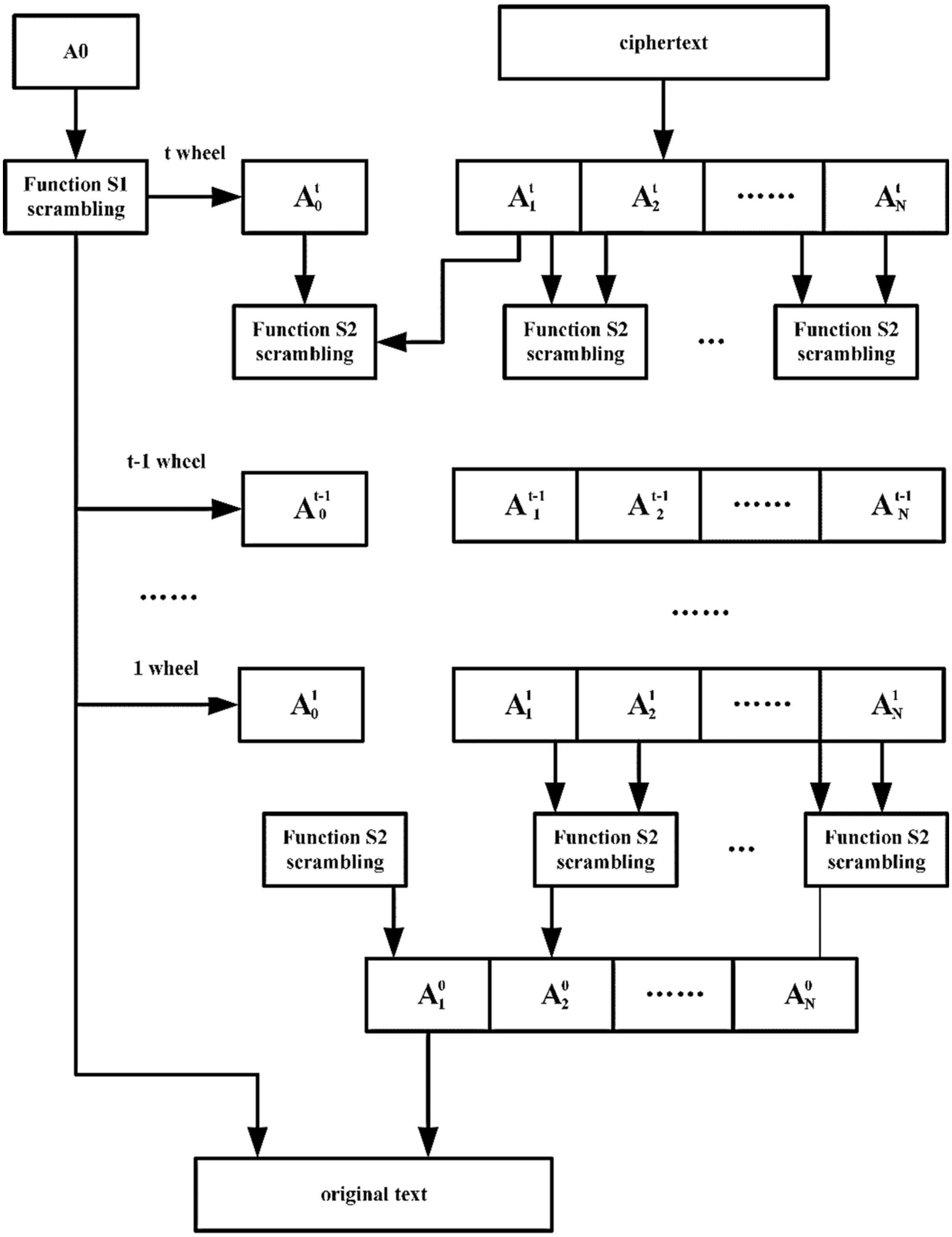


Fig. 6 Convolutional loop network structure diagram

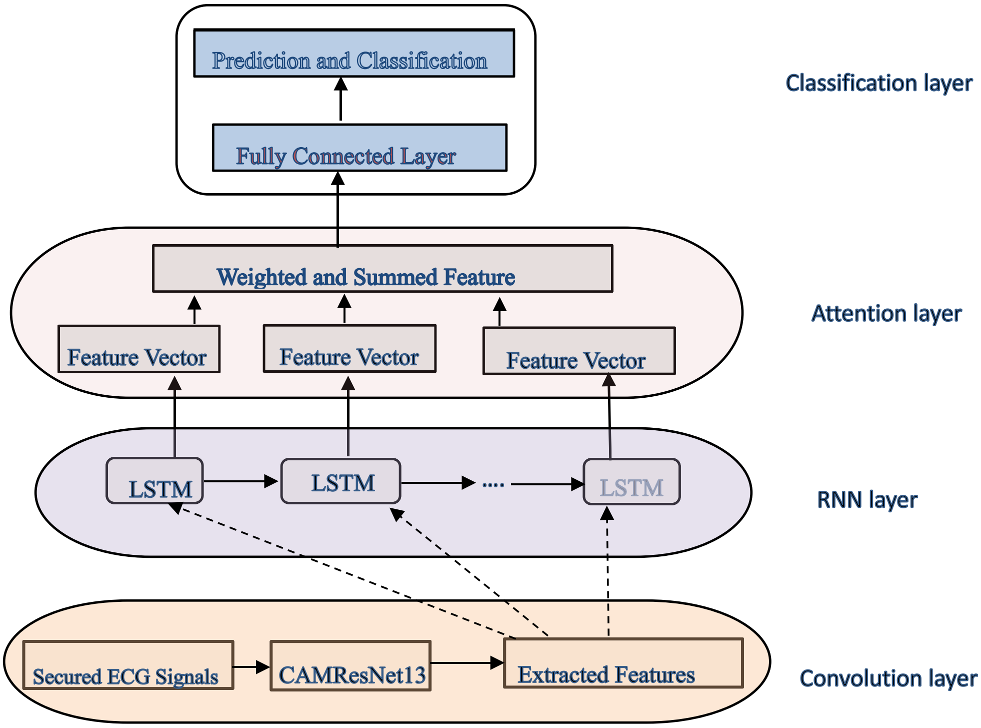


Fig. 7 Average encryption time for different file sizes of ECG data

the value of its length needs to be weighed between security performance and storage consumption. On the other hand, it is more difficult to directly attack the scrambled first data block, i.e., A_0^t. The unique encryption method of the proposed algorithm can completely eliminate its statistical characteristics. For the analysis of random experimental results of A_0^t and scrambled data, please see Sect. 3. More importantly, since the size of each data block changes randomly, it can effectively deal with parallel computing attacks and increases the difficulty of launching various attacks. This algorithm also has the advantages of being simple and requires low computation. It should be noted that this research does not include other attacks such as man-in-the-middle attacks and denial of service attacks.

4.2 Experimental analysis and results

This paper uses the correlation coefficient approach to measure the data scrambling procedure, which is calculated as follows:

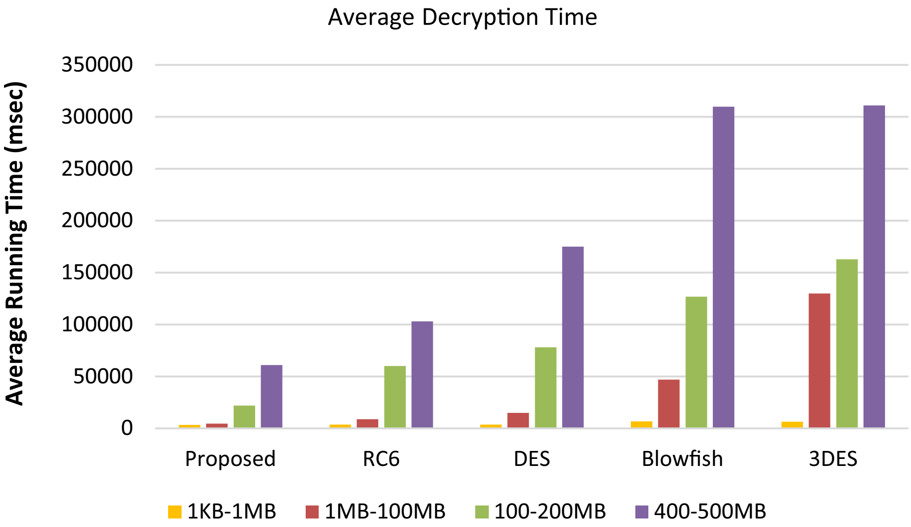


Fig. 8 Average decryption time for different file sizes of ECG data

Table 1 Results of Correlation testing after six rounds of scrambling

First Round	Second Round	Third Round	Fourth Round	Fifth Round	Sixth Round
0.1145	0.0752	0.0318	0.0413	0.0757	0.0394

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{10}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x)) \tag{11}$$

$$\operatorname{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \tag{12}$$

$$\rho_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)\cdot\sqrt{D(y)}}}. \tag{13}$$

To compute the efficiency and resilient of our proposed DL-enabled scrambling algorithm, we calculated a number of performance metrics. We calculated the average encryption and decryption time to measure the average running time. For DL-related results, we calculated accuracy, sensitivity, and specificity of our proposed CRNN (CAMResNet13) model, which is used to extract features from the data of athletes.

4.2.1 Average encryption and decryption time of DL-enabled scrambling algorithm

In Fig. 7, we evaluated our proposed block scrambling algorithm against the existing algorithms to compute the average running time for encrypting the telemedicine data. We executed our algorithm for different file sizes ranging from 1 KB to 500 MB that contain the sensitivity ECG data of the athletes. Irrespective of the file size, our proposed algorithm outperforms the existing algorithms. For a file size of 1 kb–1 Mb, our proposed algorithm has an average running time of 1800 ms to encrypt it. For a file containing much larger ECG data, i.e., 400–500 Mb, the average running time to encrypt this file is 27000 ms. In comparison, all the existing state-of-the-art algorithms have a much higher encryption time for different file sizes due to complex cryptographic encryption performed by them.

Table 2 Comparison of recognition results at the feature layer

Method	Rate of accuracy (%)
CRNN + no downsampling	90.98
CRNN + 2 times downsampling	90.05
CRNN + 4 times downsampling	89.93
CRNN + 5 times downsampling	88.74

In Fig. 8, we evaluated our proposed block scrambling algorithm against the existing algorithms to compute the average running time for decrypting the scrambled telemedicine data. We executed our algorithm for different file sizes ranging from 1 kb to 500 Mb that contain the sensitivity ECG data of the athletes in scrambled form. Irrespective of the file size, our proposed algorithm outperforms the existing algorithms. For a scrambled file size of 1 kb–1 Mb, our proposed algorithm has an average running time of 3400 ms to decrypt it. For a file containing much larger ECG data, i.e., 400–500 Mb, the average running time to decrypt this file is 61000 ms. In comparison, all the existing state-of-the-art algorithms have a much higher decryption time for different file sizes due to complex cryptographic operations performed by them.

The experiment carried out the correlation test between the original ECG signal and the ciphertext after six rounds of scrambling encryption, and obtained certain test results as shown in Table 1. It can be seen from the table that the randomness of the third round of scrambling results has reached the best. After six rounds of scrambling, the randomness of the data has reached an ideal level.

4.2.2 Experimental results of CRNN model

To analyze the influence of downsampling of input data on the ECG signal recognition at the feature layer, we have performed the following comparative experiments: no downsampling, and 2×, 4×, and 8× downsampling of the input data. Specifically, for 2× downsampling, we set the 4th residual channel attention unit of CAMResNet13 to 2× downsampling the input, i.e., set the strides of the first convolutional layer of the residual unit to 2. For 4× downsampling, we set CAMResNet13 to downs-ample the input by 2× for every 3 residual channel attention units starting from the first residual channel attention unit, and perform 2× downsampling twice, i.e., the data is downsampled 4 times. For 8× downsampling, every 2 residual channel attention units are set to downsample the input 2 times, and the input data is downsampled 8 times in total. For other parameters of CAMResNet13, we set them according to the best experimental results. For example, the size of the convolution kernel is set to 5, and the compression rate is set to 8. The experimental results are shown in Table 2. Based on the experimental results, the input

Table 3 Comparison of recognition results of feature layers using different network structures

Method	Rate of accuracy (%)
CRNN + 2 times downsampling(CAMResNet13)	93.41
CRNN + 2 times downsampling(ResNet13)	91.68

Table 4 Confusion matrix of CRNN (4 times sampling) recognition

	Predicted label		
True label		HF	IO
	HF	0.941	0.086
	IO	0.0931	0.942

data is downsampled by 2 times, which has the best effect on the ECG recognition task.

In this paper, we also compare the effect of different network structures (ResNet13, CAMResNet13) at the feature layer on the recognition results of ECG signals. Based on the optimal results of the first set of experiments (CRNN + 2 times downsampling), the network feature layer structure CAMResNer13 was replaced by ResNet13 for experiment. The experimental results are shown in Table 3. From the experimental results, it can be seen that the network structure using CAMResNer13 as the feature layer is much better than ResNet13, which again proves that the channel attention module and network structure CAMResNet13 proposed in this paper are very useful for ECG signal recognition.

In order to analyze the recognition effect of the network structure designed for each category, we use the confusion matrix as an indicator to measure the quality of system model. Table 4 shows the confusion matrix of our proposed CRNN model using the ECG signal recognition results for all the categories on the test set. We have used CRNN model with 4 times sampling. From this table, it can be observed that CRNN recognition results in each category have been significantly improved with CAMResNetl3 as compared to ResNet13. In general, the network structure is for each category, and it achieved good recognition results.

Finally, we calculate the accuracy, sensitivity and specificity for our proposed CRNN (CAMResNet13) against the existing CRNN (ResNet13), as shown in Fig. 9. It can be observed from this figure that our proposed model achieves much better results in comparison to CRNN (ResNet13).

5 Conclusions

In this paper, we proposed a deep learning (DL)-enabled block scrambling algorithm for a smart healthcare application of table tennis players. Our proposed algorithm trains on a small piece of ECG data of table tennis players to decide whether or not to store the massive amount of remaining data at the remote cloud. The secured telemedicine data is then loaded into Hadoop Distributed File System (HDFS) for data management. The data is read from HDFS into the Spark framework to form Resilient Distributed Datasets (RDD). Various map operations are performed, join operations and shuffle processes are

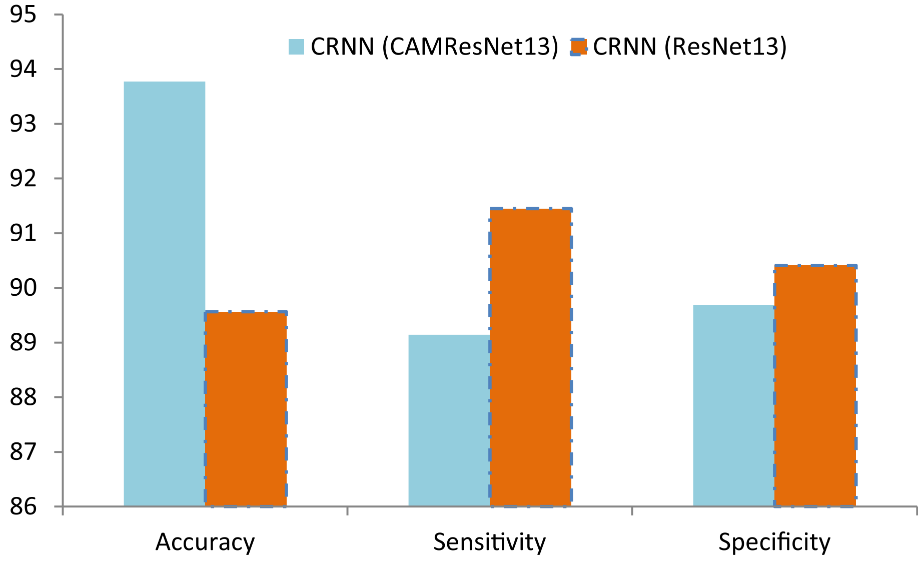


Fig. 9 CRNN (CAMResNet13) vs. CRNN (ResNet13) for 2-time downsampling

executed, and finally, the RDD processing results are generated and saved to HDFS. We used the k-means improved algorithm to optimally divide the scrambled table tennis player telemedicine data into histograms and their average is calculated. We use a 4-layered neural network architecture to recognize the ECG signals from scrambled data by extracting feature vectors. The experimental results showed that randomness of scrambling reaches an optimal level in the third round and an ideal level in the sixth round. The use of CRNN (CAMResNet13) ensures that our proposed approach achieves much better results in terms of accuracy, specificity, and sensitivity. In the future, we aim to employ a software-defined network (SDN) at the network edge to regulate the traffic flow and achieve a balanced load along with enhanced security and feature extraction.

Acknowledgement This work was supported by the Scientific Research Start Fund for high-level talents of Yulin Normal University under Grant G2020SK18.

Declarations

Conflict of interest The authors have no conflict of interest for publication of this paper.

References

1. Hou R, Kong Y, Cai B, Liu H (2020) Unstructured big data analysis algorithm and simulation of Internet of Things based on machine learning. Neural Comput Appl 32(10):5399–5407
2. Mainetti L, Patrono L, Stefanizzi ML (2016) An Internet of sport architecture based on emerging enabling technologies. In: International Multidisciplinary Conference on Computer and Energy Science (SpliTech). IEEE, pp 1–6
3. Liang H (2021) Evaluation of fitness state of sports training based on self-organizing neural network. Neural Comput Appl 1–13
4. Kallipolitis A, Galliakis M, Menychtas A, Maglogiannis I (2020) Affective analysis of patients in homecare video-assisted telemedicine using computational intelligence. Neural Comput Appl 32(23):17125–17136
5. Ma H, Pang X (2019) Research and analysis of sport medical data processing algorithms based on deep learning and Internet of Things. IEEE Access 7:118839–118849
6. Wang H, Dong C, Fu Y (2020) Optimization analysis of sport pattern driven by machine learning and multi-agent. Neural Comput Appl 1–11
7. Usman M, Jolfaei A, Jan MA (2020) RaSEC: an intelligent framework for reliable and secure multilevel edge computing in industrial environments. IEEE Trans Ind Appl 56(4):4543–4551
8. Qiu H, Qiu M, Lu Z (2020) Selective encryption on ECG data in body sensor network based on supervised machine learning. Inform Fus 55:59–67
9. Devriendt T, Chokoshvili D, Favaretto M, Borry P (2018) Do athletes have a right to access data in their Athlete Biological Passport? Drug Test Anal 10(5):802–806
10. Rathore H, Mohamed A, Guizani M, Rathore S (2021) Neuro-fuzzy analytics in athlete development (NueroFATH): a machine learning approach. Neural Comput Appl 1–14
11. De Leeuw AW, van der Zwaard S, van Baar R, Knobbe A (2021) Personalized machine learning approach to injury monitoring in elite volleyball players. Eur J Sport Sci 1–14
12. Oliver JL, Ayala F, Croix MBDS, Lloyd RS, Myer GD, Read PJ (2020) Using machine learning to improve our understanding of injury risk and prediction in elite male youth football players. J Sci Med Sport 23(11):1044–1048
13. Tang D (2020) Hybridized hierarchical deep convolutional neural network for sports rehabilitation exercises. IEEE Access 8:118969–118977
14. Yu H (2020) Research and optimization of sports injury medical system under the background of Internet of things. Trans Emerg Telecommun Technol 31(12):e3929
15. Hatamzadeh M, Hassannejad R, Sharifnezhad A (2020) A new method of diagnosing athlete's anterior cruciate ligament health status using surface electromyography and deep convolutional neural network. Biocybern Biomed Eng 40(1):65–76
16. Yuan C, Yang Y, Liu Y (2020) Sports decision-making model based on data mining and neural network. Neural Comput Appl 1–14
17. Chen H, Liu C (2020) Research on knee injuries in college football training based on artificial neural network. In: IEEE conference on telecommunications, optics and computer science (TOCS). IEEE, pp 35–37
18. Saheb T, Izadi L (2019) Paradigm of IoT big data analytics in the healthcare industry: A review of scientific literature and mapping of research trends. Telemat Inform 41:70–85
19. Rezaeibagha F, Mu Y (2018) Practical and secure telemedicine systems for user mobility. J Biomed Inform 78:24–32
20. Aiyegbusi A, Oduntan M (2020) The relationship between grip styles and musculoskeletal injuries in table tennis players in Lagos, Nigeria: a cross-sectional study. J Clin Sci 17(3):52–61
21. Rahardja U, Hardini M, Al Nasir AL, Aini Q (2020) Taekwondo sports test and training data management using blockchain. In: 5th International conference on informatics and computing (ICIC). IEEE, pp 1–6
22. Liu J, Tang H, Sun R, Du X, Guizani M (2019) Lightweight and Privacy-Preserving Medical Services Access for Healthcare Cloud. IEEE Access 7:106951–106961
23. Yu X, Jiang F, Du J, Gong D (2019) A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains. Pattern Recogn 94:96–109
24. Ning X, Gong K, Li W, Zhang L (2020) JWSAA: joint weak saliency and attention aware for person re-identification. Neurocomputing
25. Yu X, Chu Y, Jiang F, Guo Y, Gong D (2018) SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features. Knowl-Based Syst 141:80–91
26. Cai W, Liu B, Wei Z, Li M, Kan J (2021) TARDB-Net: triple-attention guided residual dense and BiLSTM networks for hyperspectral image classification. Multimed Tools Appl 1–22
27. Wang Z, Zou C, Cai W (2020) Small sample classification of hyperspectral remote sensing images based on sequential joint deeping learning model. IEEE Access 8:71353–71363
28. Yu X, Yang J, Xie Z (2014) Training SVMs on a bound vectors set based on Fisher projection. Front Comput Sci 8(5):793–806
29. Huang L, Xie G, Blenkinsopp J, Huang R, Bin H (2020) Crowdsourcing for sustainable urban logistics: Exploring the factors influencing crowd Workers' participative behavior. Sustainability 12(8):3091