Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **3**, August **2025**

Machine Learning-Based Physical Health Condition Monitoring in Smart Campus Platforms

Bo-Wen Zan[1], Xiao-Wen Zhang[2], Yu-Xin Tang[3], Sheng-Feng Zan[4,*]

[1]School of Art Management, Shandong University of Arts, Jinan 250399, P. R. China
360350228@qq.com

[2]Brussels School of Governance, Vrije Universiteit Brussel, Pleinlaan 2-1050, Brussel, Belgium
Xiaowen.zhang@vub.be

[3]College of Sports Science, South China Normal University, Guangzhou 510898, P. R. China
2023010095@m.scnu.edu.cn

[4]School of Physical Education, Shandong University, Jinan 250013, P. R. China
200993000022@sdu.edu.cn

*Corresponding author: Sheng-Feng Zan

Received May 13, 2024, revised October 11, 2024, accepted January 15, 2025.

Abstract. Traditional health monitoring systems in campus environments usually face the challenges of discontinuous data collection, insufficient real-time performance, and limited analysis capabilities. With the advancement of smart campus construction and the development of big data technologies, there is an increasingly urgent need for a platform that can monitor, efficiently store and deeply analyse health data in real time. This study proposes a big data platform for health monitoring on smart campuses based on Kafka and Attention Bi-GRU. First, the real-time collection and transmission of physiological data (e.g., heart rate, blood pressure, and body temperature) generated by health monitoring devices is achieved by using the Kafka message queuing system, which ensures the continuity of the data flow and high throughput, thus enabling timely discovery and response to health problems. Secondly, after uploading the data to the distributed storage system, the Bi-GRU model, which combines recurrent neural networks and attention mechanisms, is used to analyse the data in depth, automatically identify abnormal patterns and provide accurate health status assessment. In addition, a secondary indexing mechanism is introduced into the HBase external storage management to optimise the query efficiency of the platform. The experimental results show that compared with the existing models, the proposed attention Bi-GRU model has significantly improved the accuracy of health state monitoring and realised real-time monitoring and remote assessment of the health status of campus members.
Keywords: Health monitoring; Smart campus; Bi-GRU; Attention mechanism; Kafka; HBase

---

1. **Introduction.** As the construction of smart campuses continues to advance, health management on campuses has gradually become a focus of attention [1, 2, 3]. Traditional health monitoring methods often rely on regular physical examinations and manual records, which are not only inefficient, but also difficult to achieve real-time tracking and timely response to health conditions [4]. In addition, health data on campuses are scattered across isolated systems and lack effective integration and analysis, resulting in an inability to fully exploit the potential value of the data. To address these issues, there is an urgent need for a platform that can collect, store and analyse health data in real time to improve the efficiency and quality of campus health management.

At this stage, the main problems faced by smart campus health monitoring big data platforms include data silos, insufficient real-time performance, limited analytical capabilities, and insufficient privacy protection [5]. As health monitoring devices and systems on campus usually come from different manufacturers, inconsistent data standards and interfaces lead to difficult data integration and the formation of data silos. In addition, the platform's ability to collect and process health data in real time is limited, making it difficult to meet the demand for rapid response to emergencies. In terms of data analysis, traditional data processing methods often lack depth and accuracy, making it difficult to extract valuable health information from large amounts of data.

Cloud computing and machine learning technologies have great potential for application in the smart campus health monitoring big data platform. The elastic computing resources and large-scale data storage capabilities provided by cloud computing enable the platform to efficiently process and analyse massive health monitoring data from all corners of the campus, while supporting real-time data updating and access to ensure the continuity and reliability of monitoring services [6]. And machine learning technologies, especially deep learning algorithms, are able to automatically identify patterns and trends from complex health data [7, 8], provide accurate health status assessment and abnormal warning, and greatly enhance the intelligence of health monitoring. In addition, combined with the distributed computing advantages of cloud computing, machine learning models can be trained and optimised faster, thus better adapting to the changing health monitoring needs and providing scientific and effective decision support for campus health management. The combined application of these technologies can not only improve the efficiency and accuracy of health monitoring, but also create a safer and healthier learning and living environment for campus members. Therefore, this study proposes a smart campus health monitoring big data platform based on cloud computing and machine learning.

1.1. **Related work.** The current state of research in the field of smart health monitoring shows that although some progress has been made, such as real-time data collection using IoT technology, data processing and storage with the help of cloud computing platforms, and the application of machine learning algorithms for health state analysis and anomaly detection, it still faces many challenges. These challenges include the data silo problem, data privacy protection, real-time and accuracy of the monitoring system, and interpretability of the algorithms [9, 10]. In addition, how to effectively integrate heterogeneous data from multiple sources, improve the scalability and stability of the monitoring system, and ensure a user-friendly interaction experience are also issues that need to be focused on in current research.

Tuna et al. [11] proposed a health monitoring system based on wireless sensor networks, which uses sensors to collect physiological data and transmit them over the network to a central server for processing. The main contribution of this study is that it enables real-time monitoring of health conditions, but the scalability of the system and user privacy protection measures still need to be enhanced. Sha and Rahamathulla [12] investigated a cloud-based health data management system, which stores and analyses health data through a cloud platform and provides convenient data access services. Although the system improved the efficiency of data processing, there is room for improvement in data security and privacy protection. Muralitharan et al. [13] designed a health early warning system incorporating machine learning and data mining techniques, which is able to identify potential health risks and issue timely warnings. However, the system is computationally inefficient when dealing with large-scale data, and the accuracy of recognising abnormal patterns needs to be verified. Sujith et al. [14] explored a deep learning-based health monitoring method that uses convolutional neural networks for feature extraction and classification of health data. It was found that although the method demonstrated high diagnostic accuracy for certain health problems, there is still room for improvement in terms of model interpretability and real-time performance. Wei et al. [15] proposed a health monitoring platform based on IoT technology, which collects students' health information in real time through sensor devices and analyses the data. The main limitations of this study are the high cost of deployment and maintenance of sensors, and the stability and accuracy of data collection in different environments need to be further investigated. Jaiman and Urovi [16] investigated a health data sharing platform incorporating blockchain technology, which aims to ensure data security and non-tampering through blockchain. Although the platform has achieved some results in data security, the performance bottleneck of blockchain technology in handling highly concurrent transactions has not been effectively solved.

1.2. **Motivation and contribution.** The difficulties in data integration between the different data sources and systems in the above studies resulted in information not being shared effectively. Meanwhile, there are delays in real-time data collection and processing, and the accuracy needs to be improved. To address the above problems, the main innovations and contributions of this work include:

(1) Real-time collection and transmission of health monitoring data using Kafka message queuing system [17, 18] enables efficient integration between different data sources. Through Kafka's powerful message queuing and stream processing capabilities [19], the real-time transmission and processing of data are ensured, thus alleviating the problem of data silos.

(2) Introducing a secondary index mechanism in HBase external storage management optimises the query efficiency of the Kafka-based health monitoring big data platform.

(3) By introducing the attention mechanism and the Bi-Directional Gated Recurrent Unit (Bi-GRU) model, the platform is able to carry out deep learning and analysis of the large amount of health data collected, and is able to more accurately and automatically identify anomalies in the health data, providing accurate health assessments and timely warnings to campus members.

2. Analysis of relevant principles.

2.1. **Health monitoring data ecosystem.** In the context of the smart campus, the health monitoring data ecosystem has become an important research area, which involves the whole process from data collection, transmission, storage to analysis and application. The core of this ecosystem lies in building an efficient and reliable data flow platform that can process and analyse physiological and health data from students and staff. This data may originate from a variety of sources such as wearable devices, campus medical facilities, physical activity records, etc., and is highly diverse and dynamic. The goal of the Health Surveillance Data Ecosystem is to provide better health services and early warning mechanisms for the campus through real-time monitoring and in-depth analysis of these data, thereby improving overall health and quality of life.

The health monitoring data ecosystem is a complex system covering multiple aspects of data generation, transmission, processing, storage and application, as shown in Figure 1. In this ecosystem, users are data generators and their health information is collected through various health monitoring devices. These devices may include smart bracelets, blood pressure monitors, blood glucose meters, etc., which constantly monitor and record individual physiological parameters. Healthcare providers are also an important source of data, and the data they provide from specialised health checks and consultations is essential for a complete picture of an individual's health. These data are transferred to the health monitoring data management platform through efficient transmission channels, such as wireless networks or the Internet.

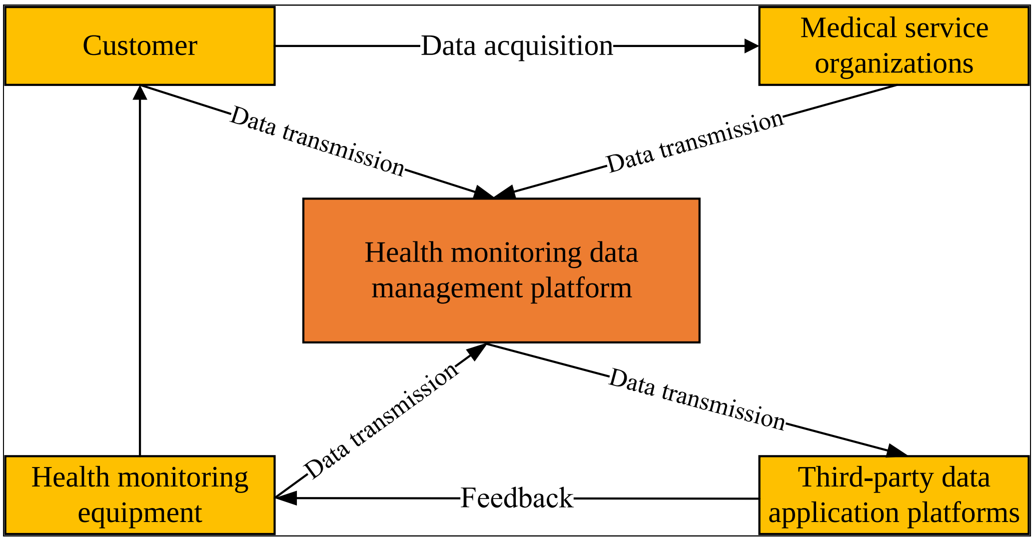

Figure 1. Health monitoring data ecosystem structure

The health monitoring data management platform is the core of the ecosystem, which is responsible for centrally storing and managing the collected health data and providing the capability of data analysis and processing. The platform extracts valuable information from massive data through advanced data processing technologies, such as big data analytics and machine learning, and provides support for further application of the data. The efficient operation of the entire ecosystem is of great significance in enhancing individual health management and promoting the development of smart healthcare.

2.2. **Introduction to Kafka and Hbase.** Kafka is a distributed stream processing platform developed by LinkedIn, which is primarily used to build real-time data streaming pipelines and streaming applications [20, 21]. Kafka offers high throughput, scalability, and fault-tolerance, making it ideal for processing large-scale data streams. Kafka uses a publish-subscribe model, which allows a producer to publish messages to topics, while consumers can subscribe to those topics to consume the messages. Its distributed architecture supports message storage and processing across multiple servers, ensuring high availability and persistence of data. Kafka also supports batch processing and compression of messages, further improving the efficiency of data transfer. Figure 2 shows the Kafka architecture with producers, consumers [22, 23], where ZooKeeper is responsible for coordinating load balancing and tracking the status of kafka cluster nodes, topics and consumers.

HBase is an open-source non-relational distributed database (NoSQL) that is part of the Apache Hadoop ecosystem and is designed based on Google's Bigtable model [24]. HBase is designed to provide random, real-time read/write access to large-scale data sets. It uses the Hadoop Distributed File System (HDFS) as its storage layer, supports automatic partitioning and horizontal scaling, and is capable of handling very large-scale datasets. The HBase data model consists of row keys, column families, and timestamps, and it allows users to dynamically add columns to a column family, which makes it well suited for handling sparse datasets [25]. In addition, HBase provides a strong consistency model that ensures data integrity and consistency. HBase also supports advanced query operations such as scanning, filtering, and sorting, making it ideal for big data analytics and health monitoring data storage.

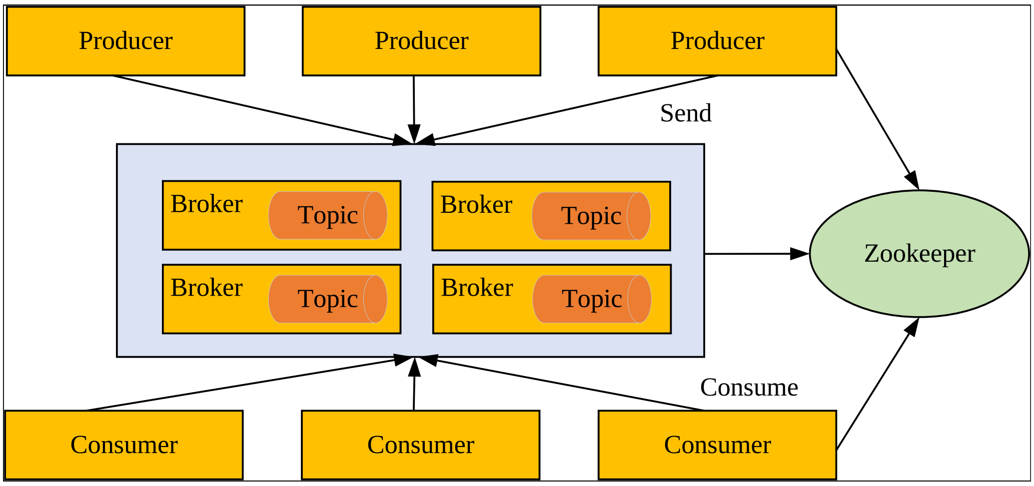

Figure 2

2.3. **Recurrent neural networks.** Unlike traditional feed-forward neural networks, Recurrent Neural Networks (RNN) are able to deal with the temporal dynamics between the input data, and are therefore well suited for processing continuous data such as time-series data, natural language text, and speech signals [26]. The state update of the hidden layer of the RNN is represented as follows:

$$h_t = f\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right) \tag{1}$$

where h_t is the hidden state at time step t; x_t is the input at time step t; W_{xh} and W_{hh} are weight matrices; b_h is the bias term; f is the activation function.

The output layer of the RNN can be computed from the state of the hidden layer, for each time step t, the output y_t is computed as follows.

$$y_t = g\left(W_{hy}h_t + b_y\right) \tag{2}$$

where y_t is the output at time step t; W_{hy} is the output weight matrix; b_y is the bias term; g is the output activation function.

However, standard RNNs are prone to the problem of gradient vanishing or gradient explosion during training, which limits their ability to learn long sequence dependencies. Long Short-Term Memory (LSTM) effectively alleviates the gradient problem by introducing a gating mechanism to control the flow of information.

The core of LSTM is the introduction of three gates: a forgetting gate f_t, an input gate i_t and an output gate o_t, and a cell state c_t. The cell state update method of LSTM is shown below:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{4}$$

$$\tilde{c}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{5}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{6}$$

where σ is the sigmoid activation function; tanh is the hyperbolic tangent activation function; and $*$ denotes element-by-element multiplication.

Gated Recurrent Unit (GRU) is another popular RNN variant [27] that simplifies the LSTM as shown in Figure 3. GRU could decide how to combine the new input information with the past information. The computation of update and reset gates for GRU is shown below:

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t] + b_z\right) \tag{7}$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t] + b_r\right) \tag{8}$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t] + b\right) \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{10}$$

where z_t is the update gate; r_t is the reset gate; $\tilde{h}_t$ is the candidate hidden state; h_t is the final hidden state; W_z, W_r, and W are the weight matrices; b_z and b are the bias terms.

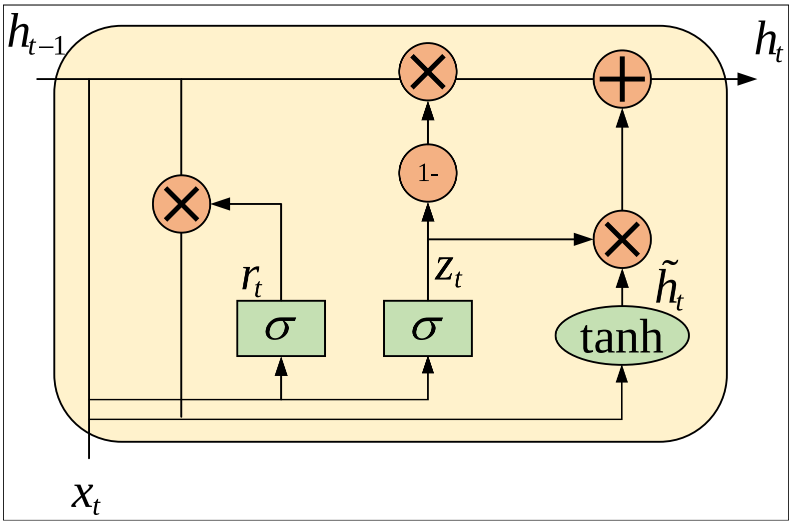

Figure 3. GRU architecture

GRU reduces the model parameters in this way, thus increasing the computational efficiency and showing comparable results to LSTM in many tasks.

Overall, RNNs and their variants are uniquely suited for processing sequential data, they are capable of capturing long-term dependencies in time series, and they play an important role in tasks such as health surveillance data analysis, anomaly detection, and so on.

2.4. **Attention Mechanisms.** The attention mechanism is a technique that enables a model to focus on a specific part of the input sequence. In traditional RNN models, hidden states are used to store historical information about the sequence, but this mechanism encounters the problem of information loss. The attention mechanism makes the RNN model dynamically focus on the important information by assigning different weights to each element in the sequence. The computational process of this mechanism can be represented as:

$$\text{Attention}(q, K, V) = \text{softmax}\left(\frac{qK^T}{\sqrt{d_k}}\right) V \tag{11}$$

where q, K, and V stand for Query, Key, and Value, respectively; d_k is the dimension of the key vector. In this way, the model can compute a context vector that highlights the most relevant information to the query.

In the health monitoring big data platform, the attention mechanism can help the model identify key events or abnormal patterns in the health monitoring sequence. For example, when analysing heart rate monitoring data, the attention mechanism can help the model identify key time periods of abnormal heart rhythms, thus providing an important basis for subsequent abnormality detection and health assessment.

3. Design and optimisation of a Kafka-based big data platform for health monitoring.

3.1. **Overall architecture design.** The Kafka-based health monitoring big data platform proposed in this study aims to achieve efficient collection, transmission, storage and sharing of health monitoring data. The overall architecture design takes full consideration of the characteristics and needs of health monitoring data, adopts a layered service-oriented design, and constructs a scalable and highly reliable data processing channel with Kafka as the core.

The overall architecture based on Kafka mainly includes the processes of data collection, data storage, and data release and sharing. For data collection, data is imported into the Kafka message queue through the data transfer tool of Kafka Connect. For data storage, HBase is chosen to provide services for real-time query, statistical analysis and data publishing and sharing. Data publishing and sharing is mainly combined with Kafka's publish-subscribe theme pattern and Spark Streaming real-time processing framework to achieve health monitoring data query service, and at the same time, real-time data calculation is carried out to detect health status problems.

As shown in Figure 4, the health monitoring data collected by the health monitoring data collection process is passed into kafka for caching through kafka's producerAPI.

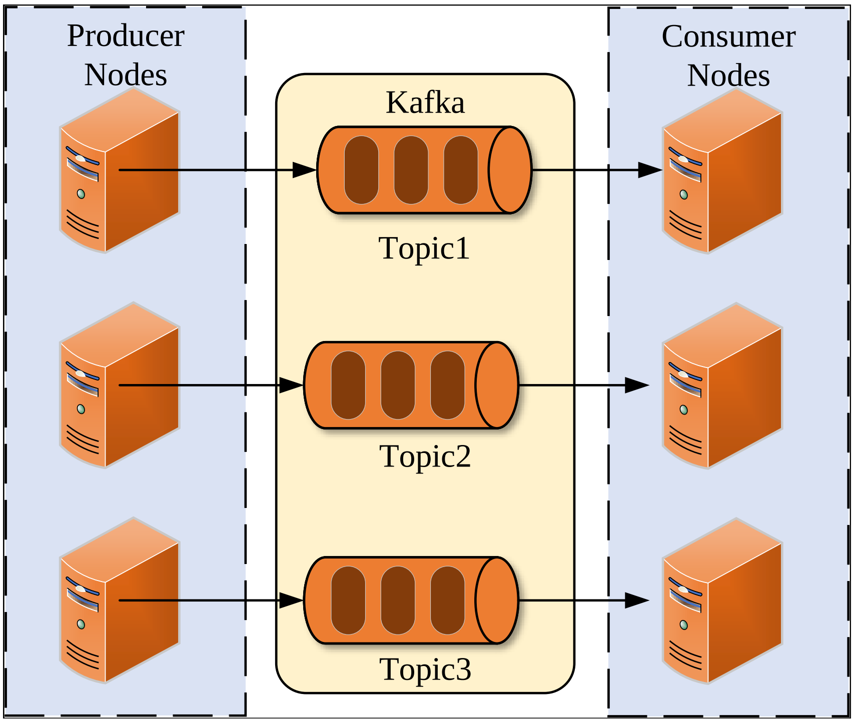

Figure 4. Kafka data collection and sharing architecture

(1) The data collection module is the foundation of the entire health monitoring big data platform, responsible for collecting data from various health monitoring devices and medical service providers. The module adopts Kafka Connect as the data transmission tool, and uses its Source Connector interface to import data into the Kafka message queue in real time. Kafka Connect supports collecting data from multiple data sources (such as databases, file systems, APIs, etc.), and has good scalability and high reliability. The design of the data collection module takes into account the compatibility and scalability of the system and ensures that the data can flow into the platform efficiently and stably.

(2) The data storage module is responsible for the persistent storage of the collected health monitoring data for subsequent data query and analysis. Considering the massive and diverse nature of health monitoring data, HBase is chosen as the main storage component in this study.

(3) The Data Publishing and Sharing Module is one of the core services of the platform that allows third-party data application platforms and users to access and utilise health monitoring data. The module is implemented based on the publish-subscribe model of Kafka and publishes data to subscribers through Kafka Topic. The platform provides a unified API interface, which enables third-party platforms to obtain the required data through the query interface, and also supports the platform to actively push the health status monitoring results to users. In addition, for enhance the query efficiency, this study also designs a secondary indexing mechanism based on the coprocessor, which significantly improves the query response speed.

3.2. **Platform performance optimisation.** In HBase database, the query of non-primary key fields is often inefficient, because the query needs to traverse the whole table. In order to improve the query performance, auxiliary indexing technology can be used. Specifically, you can select a group of columns according to business requirements, and then create a mapping relationship, and associate the values of these columns with Rowkey and store them in a user-defined data structure. When you need to query the data of a specific column, you can quickly locate the corresponding row key directly through this mapping structure, and then visit the HBase cluster to retrieve the detailed data of the record. The core function of auxiliary index is to build a direct connection between column values and row keys.

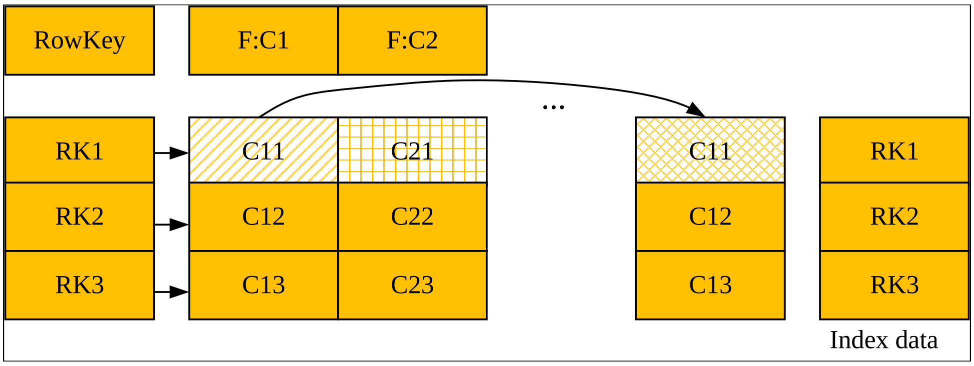

Figure 5. HBase secondary index design ideas

Take the personal health checkup physiological data as an example to introduce the secondary index table building and probing data master table process, as shown in Table 1. The secondary index table constructed by querying the Id_number column from the data in the above table is shown in Table 2. The client sends out a request to query the above secondary index table. First, the related Rowkey is retrieved from the auxiliary index table. Then, use this row key to locate and obtain the corresponding data record in the main table.

The secondary index technique may be used to greatly increase HBase's query performance, making the health monitoring big data platform more efficient in data distribution and sharing.

4. **Attention Bi-GRU based health status monitoring.**

Table 1. Personal health examination data main table

RowKey	Id_number
135427234241-20230920	141823197311472135
125682994533-20231020	134450726236721261
135427234241-20231120	141823197311472135

Table 2. Constructed secondary index table

RowKey	Id_number
141823197311472135—135427234241-20230920	141823197311472135
134450726236721261—125682994533-20231020	134450726236721261
141823197311472135—135427234241-20231120	141823197311472135

4.1. **Feature selection.** Feature selection is a crucial step when constructing a health status monitoring model, which directly affects the performance and accuracy of the model. Correct features can provide more useful information about an individual's health status, while irrelevant or redundant features may lead to degradation of model performance. Therefore, in this study, the feature selection phase begins with an in-depth analysis of health surveillance data to identify the feature set with the most predictive value.

In the feature selection process, we used a strategy that combines statistically based methods and domain expertise. First, we considered physiological data from health monitoring devices, such as heart rate, blood pressure, blood glucose levels, and sleep quality, which directly reflect an individual's health status. Second, we also incorporated basic information about the individual, such as age, gender, and Body Mass Index (BMI), which are known to have a significant impact on the risk of multiple diseases. In addition, we looked at individual lifestyle factors, such as frequency of exercise and dietary habits, as these are also strongly associated with health status.

The selected feature set aims to provide comprehensive health information for the model to more accurately predict an individual's health status and potential risks. By taking the above factors into consideration, we constructed a feature set containing key health indicators. The specifics of the feature set include, but are not limited to, the following:

(1) Physiological indicators: heart rate, blood pressure (systolic and diastolic), blood glucose level, oxygen saturation, etc.

(2) Basic information: age, gender, BMI, disease history, etc.

(3) Lifestyle: frequency of exercise, amount of daily activity, eating habits, etc.

(4) Sleep quality: length of sleep, number of sleep interruptions, percentage of deep sleep, etc.

4.2. **Data preprocessing.** After feature selection is completed, data preprocessing is the next key step in constructing a health state monitoring model. The data preprocessing in this study includes the following main aspects:

(1) Data cleaning: the collected health monitoring data were first cleaned to remove incomplete or erroneous records. For missing values, KNN Imputation interpolation method was used.

(2) Feature scaling: since health surveillance data from different sources may have different scales and value ranges, features need to be standardised or normalised. In this study, methods such as Min-Max Normalization and Z-Score Standardization are used to

ensure that all features are on the same scale, thus avoiding certain features from taking up too much weight in model training.

(3) Data transformation: In some cases, raw data need to be transformed to better represent underlying health patterns. For example, converting categorical data to numerical data, or using polynomial feature expansion to increase model complexity and fit.

(4) Time series processing: Since health monitoring data usually have time series characteristics, special processing of time series data is required. In this study, the method of time window division was adopted to split continuous health monitoring data into time periods of fixed length and to aggregate or statistically analyse the data within each time period in order to extract representative features.

(5) Feature coding: For features that contain categorical information, such as gender or disease state, the One-Hot Encoding technique is used to convert them into numerical data that can be processed by the model.

4.3. **Vectorised coding of feature sequences.** After the data preprocessing is complete, the next key step is to vectorise the feature sequences for encoding. Feature sequence vectorisation is the process of converting the features at each moment in the time series data into numerical vectors to capture the dynamic changes and patterns in the sequence.

In this study, the CBOW model in Word2Vec is used to encode textual features (e.g., symptom descriptions, doctor's recommendations, etc.) in health monitoring data to generate vector representations with semantic information. Feature vectors from different sources and types of health monitoring data are spliced to form a comprehensive feature vector. For example, text vectors encoded in Word2Vec are spliced with numeric vectors extracted from time window features to form a complete feature sequence representation.

In order to deal with time-series data, we use a time window approach to segment continuous health monitoring data. For each time window W, we can compute the statistical features within the window in the following way:

$$Mean(W) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{12}$$

$$Std(W) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - Mean(W))^2} \tag{13}$$

$$Max(W) = \max_{1\leq i\leq n} \{x_i\} \tag{14}$$

$$Min(W) = \min_{1\leq i\leq n} \{x_i\} \tag{15}$$

where x_i denotes the data points within the time window; n is the length of the time window.

With these statistics, we can extract a feature vector $\mathbf{v}_w$ from each time window.

$$\mathbf{v}_w = [Mean(W), Std(W), Max(W), Min(W)]^T \tag{16}$$

Finally, to enable the model to better learn the patterns of the sequence data, we normalised the feature vectors. For each feature vector $\mathbf{v}$, we performed L2 paradigm normalisation using the following formula.

$$\mathbf{v}_{\text{norm}} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2} \tag{17}$$

Through these steps, we transform the raw health monitoring data into a set of normalised feature vector sequences that can be converted into a format suitable for neural network processing, providing structured and numerical inputs for subsequent health anomaly detection models.

4.4. **Anomaly detection model based on attention mechanism and Bi-GRU.** In this study, we propose an anomaly monitoring model that combines the attention mechanism and Bi-GRU for health monitoring in a smart campus big data platform. The model aims to improve the accuracy of anomaly detection by capturing long-term dependencies in health monitoring data and focusing on key information using the attention mechanism.

The model architecture consists of a Bi-GRU layer and an attention layer. The Bi-GRU layer contains two directional GRUs [28, 29], one dealing with forward sequences and the other with reverse sequences to obtain complete timing information. The attention layer, on the other hand, is used to enhance the model's attention to the key parts of the sequence.

The forward and reverse hidden states h_t^f and h_t^b of the Bi-GRU layer are computed as shown below.

$$h_t^f = GRU_f(x_t, h_{t-1}^f) \tag{18}$$

$$h_t^b = GRU_b(x_t, h_{t-1}^b) \tag{19}$$

where x_t is the input feature vector at time step t; GRU_f and GRU_b are the forward and backward GRU units, respectively; h_{t-1}^f and h_{t-1}^b are the hidden states from the previous time step.

The purpose of the attention layer is to assign different weights to the hidden states at each time step that reflect the importance of the state to the current prediction. The attention weights α_t are calculated as shown below.

$$e_t = \mathbf{v}^T \tanh\left(W\left[h_t^f, h_t^b\right]\right) \tag{20}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{T}\exp(e_i)} \tag{21}$$

where W and v are learnable parameters; e_t is the energy fraction at time step t; α_t is the corresponding attention weight.

Combining the Bi-GRU layer and the attention layer, the final output of the model y_t is as follows.

$$y_t = \text{softmax}\left(W_y \sum_{t=1}^{T} \alpha_t \left[h_t^f, h_t^b\right]\right) \tag{22}$$

where W_y is the learnable weight matrix; the softmax function is used to output the anomaly probability distribution at each time step.

Training of the model is accomplished by minimising a loss function for anomaly monitoring, usually using cross-entropy loss:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{23}$$

where N is the number of training samples; y_i is the true label; $\hat{y}_i$ is the anomaly probability predicted by the model.

5. Platform construction and testing.

5.1. Experimental environment. In this paper, a server is used to divide into 6 virtual machine nodes to build the system running environment. Each virtual node is configured with a CPU of 2.40GHz, 4.0G RAM, and a hard disc of 200GB. The distribution is shown in Table 3.

Table 3. Cluster node distribution

Hosts	Kafka	Zookeeper	HADOOP	Spark	HBase	Redis
Virtual-node01	Broker		Name Node	Master	HMaster	Master
Virtual-node02	Broker		StandbyNameNode	StandbyMaster	StandbyHMaster	Master
Virtual-node03	Broker		Data Node	Worker	HRegionServer	Master
Virtual-node04		HQuorumPeer	Data Node	Worker	HRegionServer	Slave
Virtual-node05		HQuorumPeer	Data Node	Worker	HRegionServer	Slave
Virtual-node06		HQuorumPeer	Data Node	Worker	HRegionServer	Slave

The software environment and version of the system are shown in Table 4.

Table 4. System software environment and version

Hardware	Releases
Operating system	Centos-6.7
Jdk	Javaversion "1.8.0_141"
Hadoop	Hadoop-2.6.0
Zookeeper	Zookeeper-3.4.10
Hbase	Hbase-1.2.0
Kafka	Kafka2.11-0.10.2.1
Spark	Spark-2.2.0
Redis	Redis-3.2.4

5.2. Kafka cluster performance testing. Kafka producer and consumer throughput rates are tested using the `kafka-producer-perf-test.sh` and `kafka-consumer-perf-test.sh` scripts provided by Kafka. In the experiments, `batch.size` is set to 100kb and `linger.ms` is set to 50ms, so that messages are sent when they reach 100kb or 50ms after the first message arrives. The throughput rate of the consumer is tested before and after the introduction of the secondary indexing mechanism and the results are shown in Figure 6.

It can be seen that with the growth of data size, the data processing time of standard Kafka is increased compared to the other two external storage methods, which will cause some impact on the system throughput rate. After the introduction of the secondary index mechanism in HBase external storage management, Kafka is more advantageous in the case of large-scale data. This indicates that the establishment of secondary indexes can significantly reduce the query response time of non-indexed data, which verifies the effectiveness of the proposed query optimisation strategy.

5.3. Comparison of anomaly monitoring models. For the validation of the proposed anomaly monitoring model based on the attention mechanism with Bi-GRU, a series of comparison experiments were conducted. In the experiments, the model was compared with several other popular anomaly detection methods. The experimental dataset consisted of simulated health monitoring logs with both normal and abnormal categories. Precision, Recall and F1-score were used as evaluation metrics [**?**] to comprehensively measure the abnormality detection effectiveness of the model, and the results are shown in Table 5.

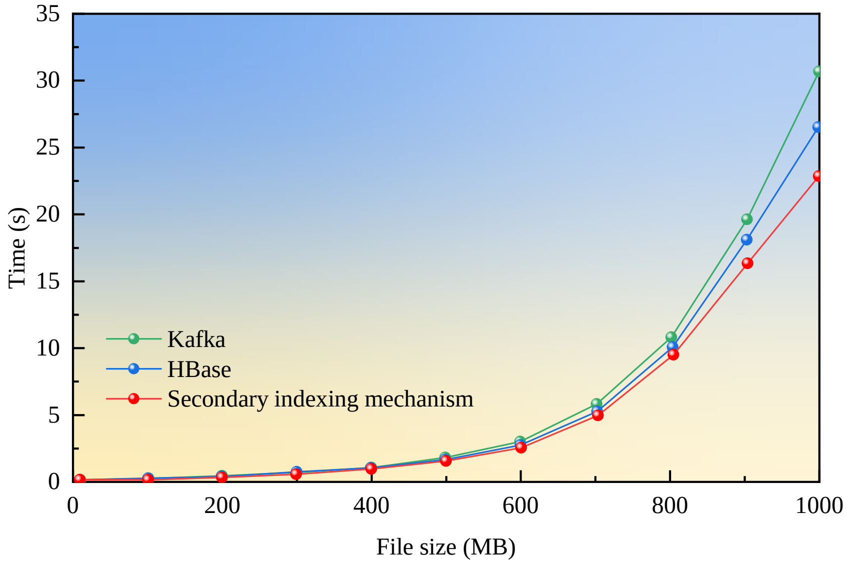

Figure 6. Relationship between data size and cluster processing time

Table 5. Comparison of Anomaly Monitoring Models

Methodological model	**Precision**	**Recall**	**F1-score**
Gradient boosted tree	0.962	0.973	0.965
RNN	0.891	0.948	0.918
LSTM	0.956	0.964	0.960
GRU	0.970	0.966	0.968
Ours	0.974	0.981	0.977

The experimental results show that our model outperforms the comparison methods in all evaluation metrics. Specifically, the Attention Mechanism Bi-GRU model outperforms LSTM and GRU in terms of precision rate, which indicates that the model is able to identify anomalous patterns more accurately and reduce false alarms. In terms of recall, our model also performs well, effectively identifying more real anomalies and reducing underreporting. The F1-score combines precision and recall, and our model also achieves the best result in this metric, at 0.977, suggesting that the model maintains a high precision rate while also having a high coverage rate.

6. **Conclusions.** This work proposes a smart campus health monitoring big data platform based on Kafka and Attention Bi-GRU. The real-time collection and transmission of health monitoring data is achieved using the Kafka message queuing system, which is able to achieve efficient integration between different data sources. Through Kafka's powerful message queuing and stream processing capabilities, it ensures real-time data transmission and processing, thus alleviating the problem of data silos. The introduction of the secondary index mechanism in HBase external storage management optimises the query efficiency of the Kafka-based health monitoring big data platform. By introducing the attention mechanism and the Bi-GRU model, the platform is capable of deep learning and analysing a large amount of collected health data, and is able to more accurately and automatically identify anomalies in the health data, providing campus members with accurate health assessments and timely warnings. The experimental results show that the F1-score metric of the proposed platform reaches 0.977, indicating that the model has a high coverage rate while maintaining a high accuracy rate. Subsequent research plans

will explore more deeply in safeguarding data security and privacy, as well as optimising resource management.

Acknowledgment. This work is supported by the Research project on graduate education and teaching reform at Shandong University, "Research on the generation logic and physical-mental intervention strategies of academic burnout among direct graduate students in first-class universities" (No. XYJG2023080).

REFERENCES

[1] M. Gao, "Smart campus teaching system based on ZigBee wireless sensor network," Alexandria Engineering Journal, vol. 61, no. 4, pp. 2625-2635, 2022.

[2] Y. P. Wang, "The research of wisdom campus construction and development based on internet of things," Advanced Materials Research, vol. 989, pp. 5439-5443, 2014.

[3] S. C. Brown, "Learning across the campus: How college facilitates the development of wisdom," Journal of College Student Development, vol. 45, no. 2, pp. 134-148, 2004.

[4] S. Lu, "Exploration and Application of Wisdom Garden in University Campus," Journal of Smart Cities, vol. 6, no. 1, pp. 9-13, 2021.

[5] Q. Cao, "The Influence of Wisdom Platform Teaching on Students' Learning Effectiveness in Higher Vocational Colleges," Turkish Journal of Computer and Mathematics Education, vol. 12, no. 11, pp. 4270-4278, 2021.

[6] W. Chen, Z. H. Zhang, and Z. J. Liu, "The Framework Design of Wisdom Campus Based on the Network," Applied Mechanics and Materials, vol. 644, pp. 2719-2722, 2014.

[7] M. I. Jordan, and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science, vol. 349, no. 6245, pp. 255-260, 2015.

[8] B. Mahesh, "Machine learning algorithms-a review," International Journal of Science and Research, vol. 9, no. 1, pp. 381-386, 2020.

[9] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences," Reviews of Modern Physics, vol. 91, no. 4, 045002, 2019.

[10] J. G. Greener, S. M. Kandathil, L. Moffat, and D. T. Jones, "A guide to machine learning for biologists," Nature Reviews Molecular Cell Biology, vol. 23, no. 1, pp. 40-55, 2022.

[11] G. Tuna, R. Das, and A. Tuna, "Wireless sensor network-based health monitoring system for the elderly and disabled," International Journal of Computer Networks and Applications, vol. 2, no. 6, pp. 247-253, 2015.

[12] M. Sha M, and M. P. Rahamathulla, "Cloud-based Healthcare data management Framework," KSII Transactions on Internet and Information Systems, vol. 14, no. 3, pp. 1014-1025, 2020.

[13] S. Muralitharan, W. Nelson, S. Di, M. McGillion, P. Devereaux, N. G. Barr, and J. Petch, "Machine learning–based early warning systems for clinical deterioration: systematic scoping review," Journal of Medical Internet Research, vol. 23, no. 2, e25187, 2021.

[14] A. Sujith, G. S. Sajja, V. Mahalakshmi, S. Nuhmani, and B. Prasanalakshmi, "Systematic review of smart health monitoring using deep learning and Artificial intelligence," Neuroscience Informatics, vol. 2, no. 3, 100028, 2022.

[15] K. Wei, L. Zhang, Y. Guo, and X. Jiang, "Health monitoring based on internet of medical things: architecture, enabling technologies, and applications," IEEE Access, vol. 8, pp. 27468-27478, 2020.

[16] V. Jaiman, and V. Urovi, "A consent model for blockchain-based health data sharing platforms," IEEE Access, vol. 8, pp. 143734-143745, 2020.

[17] T.-Y. Wu, L. Wang, and C.-M. Chen, "Enhancing the Security: A Lightweight Authentication and Key Agreement Protocol for Smart Medical Services in the IoHT," Mathematics, vol. 11, no. 17, 3701, 2023.

[18] T.-Y. Wu, Q. Meng, L. Yang, S. Kumari, and M. Pirouz, "Amassing the Security: An Enhanced Authentication and Key Agreement Protocol for Remote Surgery in Healthcare Environment," Computer Modeling in Engineering & Sciences, vol. 134, no. 1, pp. 317-341, 2023.

[19] T.-Y. Wu, F. Kong, Q. Meng, S. Kumari, and C.-M. Chen, "Rotating behind security: an enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture," EURASIP Journal on Wireless Communications and Networking, vol. 2023, 36, 2023.

[20] D. R. Torres, C. Martin, B. Rubio, and M. Diaz, "An open source framework based on Kafka-ML for Distributed DNN inference over the Cloud-to-Things continuum," Journal of Systems Architecture, vol. 118, 102214, 2021.

[21] N. Pećina-Šlaus, A. Kafka, I. Salamon, and A. Bukovac, "Mismatch repair pathway, genome stability and cancer," Frontiers in Molecular Biosciences, vol. 7, 122, 2020.

[22] N. V. Patil, C. R. Krishna, and K. Kumar, "KS-DDoS: Kafka streams-based classification approach for DDoS attacks," The Journal of Supercomputing, vol. 78, no. 6, pp. 8946-8976, 2022.

[23] B. Vyas, "Integrating Kafka Connect with Machine Learning Platforms for Seamless Data Movement," International Journal of New Media Studies: International Peer Reviewed Scholarly Indexed Journal, vol. 9, no. 1, pp. 13-17, 2022.

[24] M. U. Hassan, I. Yaqoob, S. Zulfiqar, and I. A. Hameed, "A comprehensive study of hbase storage architecture—a systematic literature review," Symmetry, vol. 13, no. 1, pp. 109, 2021.

[25] H. Matallah, G. Belalem, and K. Bouamrane, "Evaluation of nosql databases: Mongodb, cassandra, hbase, redis, couchbase, orientdb," International Journal of Software Science and Computational Intelligence, vol. 12, no. 4, pp. 71-91, 2020.

[26] M. Abumohsen, A. Y. Owda, and M. Owda, "Electrical load forecasting using LSTM, GRU, and RNN algorithms," Energies, vol. 16, no. 5, 2283, 2023.

[27] L. Tong, H. Ma, Q. Lin, J. He, and L. Peng, "A novel deep learning Bi-GRU-I model for real-time human activity recognition using inertial sensors," IEEE Sensors Journal, vol. 22, no. 6, pp. 6164-6174, 2022.

[28] Z. Zhu, W. Dai, Y. Hu, and J. Li, "Speech emotion recognition model based on Bi-GRU and Focal Loss," Pattern Recognition Letters, vol. 140, pp. 358-365, 2020.

[29] A. Kumar, and N. Sachdeva, "A Bi-GRU with attention and CapsNet hybrid model for cyberbullying detection on social media," World Wide Web, vol. 25, no. 4, pp. 1537-1550, 2022.

[30] R.-H. Huan, J. Shu, S.-L. Bao, R.-H. Liang, P. Chen, and K.-K. Chi, "Video multimodal emotion recognition based on Bi-GRU and attention fusion," Multimedia Tools and Applications, vol. 80, pp. 8213-8240, 2021.