Elderly Health Care Data Integration Framework: Design and Implementation

Zhean Zhong[1], Chenyu Liu[1], Jing Zhao[1], Jianmao Xiao[1,2(✉)], Qinghang Gao[1], Chao Gao[3], and Chuying Ouyang[4]

[1] School of Software, Jiangxi Normal University, Nanchang 330022, China
jm_xiao@jxnu.edu.cn
[2] Jiangxi Provincial Engineering Research Center of Blockchain Data Security and Governance, Nanchang 330022, China
[3] College of Intelligence and Computing, Tianjin University, Tianjin 300072, China
[4] Department of Physics, Jiangxi Normal University, Nanchang 330022, China

Abstract. The application of Internet of Things (IoT) detection devices in the field of elderly healthy generates a large amount of detection data, which faces problems such as diversified data sources and non-uniform parsing methods. The fragmentation and confusion of the process from data subscription to data analysis have become important challenges affecting the development of the healthy aging field. In this paper, we designed and proposed an elderly health care framework with a model-as-a-service orientation. The framework automates the integration of health data for the elderly, including subscription services. It handles Model Training Analysis through a workflow-based system. Users can configure workflows using a low-code method provided by the platform. First, we abstracted and formalized the integration problem and workflow and defined a workflow description language. Then, we built an healthy aging unified modeling framework based on the problem model and provided a general introduction to the framework architecture. The module for analyzing is responsible for model training, persistence, and data analysis, enabling users to customize the selection of machine learning models and corresponding parametric. Ultimately, we built a model library and an external RESTful interface based on the automation option of Automated machine learning (AutoML). The experiments demonstrate this paper's proposed framework's feasibility and effectiveness.

Keywords: Data curation · Elderly health care · Model-as-a-Service · Machine learning

Supported by the Natural Science Foundation of Jiangxi Province (Grant No. 20224BAB212015, 20224ACB202007), the National Natural Science Foundation of China (NSFC) (Grant No.51962010, 61962026, 62363015, 62067003), the Foundation of Jiangxi Educational Committee under grant no. GJJ210338, and the Jiangxi Province Science and Technology Project (03 Special 5G Project) (Grant No. 20224ABC03A13, 20232ABC03A26).

Y. Cao and X. Shao (Eds.): DIONE 2023, LNICST 515, pp. 193–208, 2025.
https://doi.org/10.1007/978-3-031-80713-8_14

1 Introduction

Elderly health care is an essential social need. As more and more IoT devices and technologies related to health detection enter the health care field, the related issues become more intelligent and data-oriented. For example, various testing devices and sensors can easily collect a large amount of health testing data [1]. These test data cover almost all aspects of the elderly's body metrics and can meet various needs for chronic disease management, daily medical check-ups, and early disease screening. In addition, those device can use the Message Queue Telemetry Transport (MQTT) protocol for data distribution with very low power consumption and data subscription by remote users, making it possible for the elderly's family members and community elderly services to facilitates the monitoring and analysis of the well-being of elderly individuals from off-site [2].

Currently, the application of machine learning methods to process data collected from IoT devices to enhances diagnostic efficiency and aids physicians in diagnosis has been widely used [3]. Combined with the idea of the collected test data can be analyzed and processed by machine learning and other methods to determine the current health status of the elderly and the diseases they may suffer from. This can not only provide convenient remote analysis services for the elderly's children in the field, but also facilitate various community elderly service organizations to maximizes the added value of businesses by driving further improvement.

However, in health care for the elderly, converting models to services requires careful consideration of various factors such as device data reception, data parsing, data storage based on IoT communication protocols respectively. Firstly, in the realm of IoT, different devices follow independent vertical IoT architectures. This implies that the data generated by each device is published to its designated MQTT server. Secondly, the subsequent process of Achieve thorough parsing, storage, and analysis of data using machine learning techniques. often combines multiple technologies and systems in concert, which usually means being familiar with mastering a huge technology stack or multiple business systems. Thus, One of the primary obstacles lies in the insufficient integration and coordination, posing significant challenges that hinder the conversion landing in individuals and various senior service organizations in the healthy aging scenario to provide remote health detection and data analysis services for the elderly.

By building an integrated framework for health care scenarios, the total process of IoT data collection and model service is automatically completed by the integration framework, freeing users from code development of IoT middleware, database and data analysis, minimizing user workload, further improving work efficiency, and creating conditions for personalized data collection. It creates conditions for personalized data collection and analysis services in the healthy aging scenario, and provides strong support for implementing the model in the healthy aging scene in the post-epidemic era. The main contributions of this paper are as follows.

- We propose a workflow integration scheme for model-as-a-service in elderly health care. In order to realize and bridge all the processes of Model-as-a-Service from end to end, this paper abstracts and models the problem from the perspective of workflow, proposes a unified custom description language and uses the custom language to describe the integration workflow involving diverse task flows and tasks, each requiring specific dynamic configuration resources. At the same time, the corresponding integration scheme is given by combining the above model and the custom description language, and the workflow resolution processing flow for the custom description language is defined to align with the integration workflow's requirements, which provides a basis for the framework.
- We designed and implemented a framework for the elderly health care scenario. It provides functions such as data subscription, parsing, storage, machine learning, deployment, and analysis. The data subscription, parsing, and storage processes are accomplished through user-defined workflows that invoke the data subscription module to read dynamically configured resources. Dynamic updates to the above processes are accomplished by modifying the state identifiers of dynamically configured resources. By encapsulating the machine learning training steps into different components completes tasks such as model assembly, model training, dataset loading, data fetching, model invocation, etc.
- The integration framework's usability, ease of use, and security are validated with examples. The model-as-a-service capability is validated through workflow calls and interface calls, and demonstrates the potential of combining workflow description languages with automated machine learning components to further simplify workflows.

2 Related Work

For the past few years, there has been a significant rise in the utilization of IoT sensors and devices, particularly in the fields of disease surveillance. The requirements on IoT technologies is steadily growing [4]. IoT devices in healthcare applications involve major technical areas such as sensing, communication, data analysis and reasoning, and security [5,6].

An ECG monitoring system was proposed by Yang et al. [7] in which ECG signals are collected in an IoT equipment and transmitted using WiFi. Hypertext Transfer Protocol are used in their IoT cloud server to process data. The work conducted by Laport et al. [8] utilizes EEG sensors to categorize the eye conditions. The raw EEG signals are transmitted using the ESP8266 WiFi module, while the varied IoT agents is managed through the MQTT protocol. In the context of healthy aging, fall detection serves as a crucial application. Yacchirema et al. [9] introduced a 3D-axis accelerometer, integrated within an IPv6 device. They leveraged the MQTT protocol to transmit notifications to caregivers. Likewise, Kadarina et al. [10] focused on monitoring the health of mothers and infants. They employed blood oxygen sensors and other relevant sensors to

gather heart rate and blood oxygen data. The collected data was subsequently uploaded to their platform utilizing MQTT as the communication protocol. The literature [11] combined wearable devices with machine learning methods to provide a solution for real-time determination. The method automatically removed anomalies in the artificial signal from the blood pressure signal and extracted 11 features to analyze the relationship between diastolic and systolic blood pressure. The study was conducted only for the healthy population. The literature [12] designed a framework for a mobile health testing system based on a cloud computing platform, which was discussed in terms of a cloud data storage module, data analysis module, and resource allocation module. The literature [13] combined communication technologies, connected applications, and IoT devices to focus on health detection, signal enhancement, and analysis of ECG. Specifically, the literature mentions the need for watermarking of healthcare data before sending to the cloud, which has positive implications for protecting data privacy and security. The literature [14] designed a smart wearable equipment that implements temperature detection, heart rate detection, and one-touch rescue. The literature [15] considered the impact of the data transmission process on time delay in cloud computing and real-time data processing of patients through fog computing. The literature proposes a model based on BBN classifier by combining embedded data mining, distributed storage and notification services at the edge of the network. The literature [16] proposed a more comprehensive health detection scheme to regularly check fitness levels of the elderly by providing appropriate medical instruments and devices to obtain raw data and automating data analysis by uploading the data to the computer in the cloud for manipulation. The literature [17] combined a textile and clothing interface with dedicated software for data capture and warning flag generation to design a method that allows non-invasive measurement of set physiological parameters. Similarly, the literature [18] proposed to embed sensors into shoes to monitor the gait structure, blood pressure, and heart rate of aging parents so that children can keep an eye on the physical health status of the elderly. The literature [19] divided the home health monitoring system into two sub-networks and proposed cooperative game, decentralized non-cooperative game to allocate wireless channels and minimize the system-wide cost, respectively. In the literature [20], a system framework that can supervise the health condition of real-time data on seniors is proposed for elderly care centers to provide a more efficient and direct way for caregivers to keep a close watch over the state and activities of the elderly during the night.

The widespread adoption of transducers is the background underlying the task, and these devices support various data sources that provide the necessary data for monitoring analysis applying machine learning skills. In practice, the detection of health or diseases typically requires the consideration of numerous analytic metrics. It's common for a single sensor or device to be unable to capture all these metrics comprehensively. Therefore, remote data collection from multiple devices becomes necessary to gather a comprehensive dataset for further analysis.

At the edge computing level, many research scholars and industries have also proposed integrating the functions of various IoT devices through various middleware and frameworks. The Niagara framework [21], introduced by Honeywell, is widely used in areas such as building automation, allowing developers to build custom, web-enabled applications through which smart devices can be visited, automated, and controlled in real time over the Internet. niagara can be hosted on a variety of platforms ranging from small embedded-time systems to high-end servers. Niagara's sites communicate via a private Fox protocol, and data is presented in point form. As an IoT framework for the industrial sector, it is designed to integrate all networks and protocols and offers a complete solution from hardware to plug-in development. This means that although it can be used for data subscription and the related processing for remote health detection, developers need to obtain a commercial license in Niagara and spend a lot of time learning and developing the corresponding components. In the EAaaS proposed by Xu et al. [22], the Edge Analytics Agent is used as a component of the IBM Watson IoT platform to shift the analytics workload to the edge and provide managed services for edge analytics. However, the Edge Analytics Agent only subscribes to MQTT messages from the cloud for loading rule-based analytics models. Its device adapter only interfaces to IoT devices containing its Software Development Kit (SDK) through General-Purpose Input/Output (GPIO) and serial ports.

In summary, most of the IoT middleware or platforms do not utilize workflows to manage the whole business process, although there are some middleware under edge architecture [23,24] that involve the use of workflows, they mainly assign work tasks to one or more local edge functional modules for specific implementations through workflows. Finally, most of the IoT platforms segment the individual functions and do not connect the whole process of model-as-a-service end-to-end, so there is a need to design a new framework to implement the model-as-a-service landing in healthy aging scenarios.

3 System Model

In this paper, we modularize various functions to form a framework, use workflow to enables the invocation of each module component to carry out specific tasks, which provides a low-code approach, reducing the complexity for users and improving the scalability of the system. It allows users to easily configure the workflow based on distinctive task demands, enabling the consolidation of complicated operational processes.

3.1 System Framework

As shown in Fig. 1, the conceived framework is characterized by a parsing engine, a data access module, a data parsing module, a data analysis module, a data preservation module and a model-as-a-service module. The management and

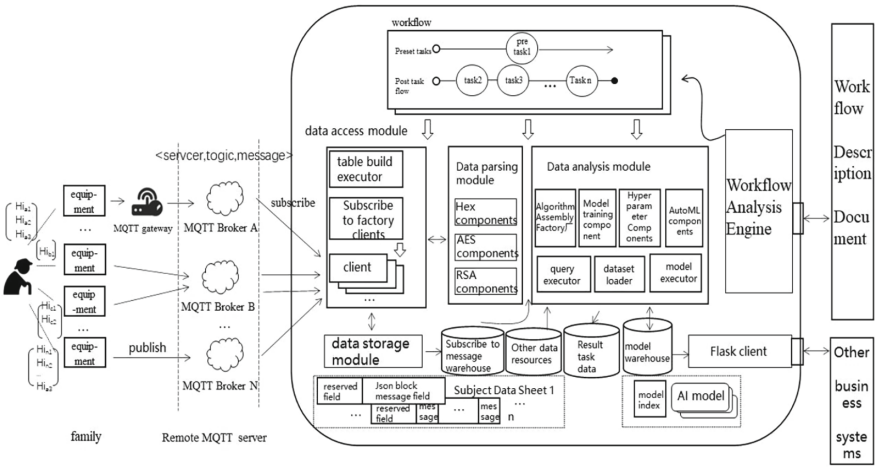


Fig. 1. IoT Data Integration Framework.

maintenance between individual feature modules is carried out via workflow. The whole framework has the following main functions.

The framework has a work process orchestration function. The framework through the workflow parsing engine for the framework of the components of the module process scheduling, management, operation monitoring and timing execution. The framework supports scheduling workflows based on custom workflow description documents to perform tasks such as data acquisition, data parsing and memory, and data analysis, as well as using custom description documents to update the workflow sequence during the timed execution.

The framework has IoT management capabilities for healthy aging devices. Given that the MQTT protocol has become a widely used docking protocol in the IoT platform, the framework manages the docking of healthy aging devices communicating based on the MQTT protocol by adding the topics of healthy aging devices to be subscribed to in the dynamic resource taskResource, and the data access module calls the subscription client factory to assemble the MQTT client. At the same time, the data access module achieves the closure of the device subscription client based upon the judgment of the subscription status flag of the device in taskResource, thus realizing the dynamic addition and deletion management of the healthy aging device by the user during the task execution.

The framework has the function of automatic database creation and data storage for healthy aging devices. When connected, the user can store the data persistently as needed. The framework performs table building and field building operations by calling the table builder executor. Users do not need to develop script code for the database, but only need to define the type of database and field name to be built in the rule file to complete the automatic creation of the data warehouse. It also takes advantage of the PostgreSQL database's support for Json format to design two different table creation forms for data with clear field

correspondence and complex document-based data, reducing the complexity of applying other nosql databases. After executing the data subscription task, the framework automatically stores the received device data into the corresponding data warehouse after matching calls to the data preservation module.

The framework has data source configuration and machine learning analysis functions. In the healthy aging scenario, the functions of various devices are often single, and in the process of data analysis and machine learning often need to use and combine the data collected by multiple devices, so the data needs to be assembled as needed, and the framework supports configuration and management of the data sources required for analysis through configuration files. Also based on automated machine learning techniques, the data is matched with algorithm libraries and parameters and automated to complete model generation and preservation.

The framework has model-as-a-service functionality. In order to make more effective use of the models trained by the framework, the framework treats the generated models as service resources. It integrates a flask server internally to provide a RESTfulAPI interface for the generated models. Other local business systems can call the models in the model repository according to the corresponding API interface document descriptions, providing conditions for the integration of the framework with other business systems.

3.2 Remote Data Access Based on MQTT Messages

In Fig. 2, we give the functional diagram of the IoT data integration framework system.

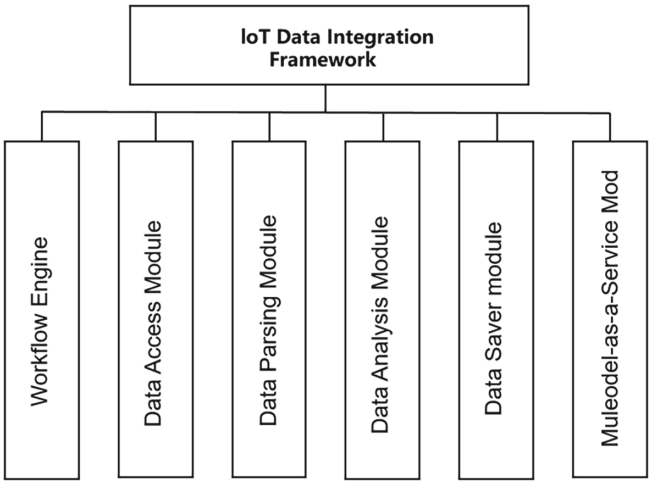


Fig. 2. IoT data integration framework system functional diagram.

The data subscription task requires subscribing that Data can be remotely transmitted using MQTT clients, where each MQTT client is capable of subscribing to a specific subscription topic. However, it should be noted that an client can only subscribe to a topic under a single server. When multiple servers or different topics under the same server have different message resolution and storage methods, a single MQTT client obviously cannot meet the requirements. Therefore, the framework periodically access the active allocations from the resource, incorporating the corresponding parameters for different server-side or parsing storage mechanisms. This information is then utilized to assemble specific MQTT clients, each of which is responsible for subscribing to subscription topics under the same MQTT server. Additionally, the framework allows for the configuration of a state identifier for each subscribing client. This state identifier provides a mechanism to dynamically manage message subscriptions.

The module attains the respective server-side IP address, port number, etc., by traversing all the equipment sub nodes, obtaining all the thematic dots according to the equipment sub node, and cross over them to obtain the list of subscribed topics. According to the db flag status, the subscription client plant is launched, or the client demonstration is closed. Eventual, the module updates the taskResource deployment to enable dynamical renewal of the tasks. The overall flowchart of the module is shown in Fig. 3.

3.3 Machine Learning Automated Data Analysis

The data analysis module primarily serves the purpose of supporting fitness aged data analytics tasks and fulfilling subscribers requirements for model training and utilization within the workflow. Machine learning is currently one of the key ways to perform artificial intelligence analysis on health data. In this framework, the algorithmic library module assumes that the user has possessed suitable data for testing. Consequently, the component focuses on the subsequent stages after functional programming and incorporates a range of assemblies to cater to various task requirements.

Data Loader. Firstly, the module includes an abstracted component called DataExtract, which serves as the data loader. The primary utility of the data loader is to mount data and provide the necessary data for algorithm learning and category. It aims to be matched with various document forms and data characterization approaches. Specifically, the sample code of the DataLoader component receives parameters such as the location of the dataset file and destination column delivered from the workflow. This method handle for executing and loading the data. To determine the file type, the component parses the file path. Based on the file type, the suitable methodology is invoked to extract the data from the file. The data loader supports multiple file formats, ensuring flexibility for users to work with different sources of data. For different data files, their target tags may have various forms of expressions, such as class, target, etc. Therefore, after reading the file, the workflow must give the target column

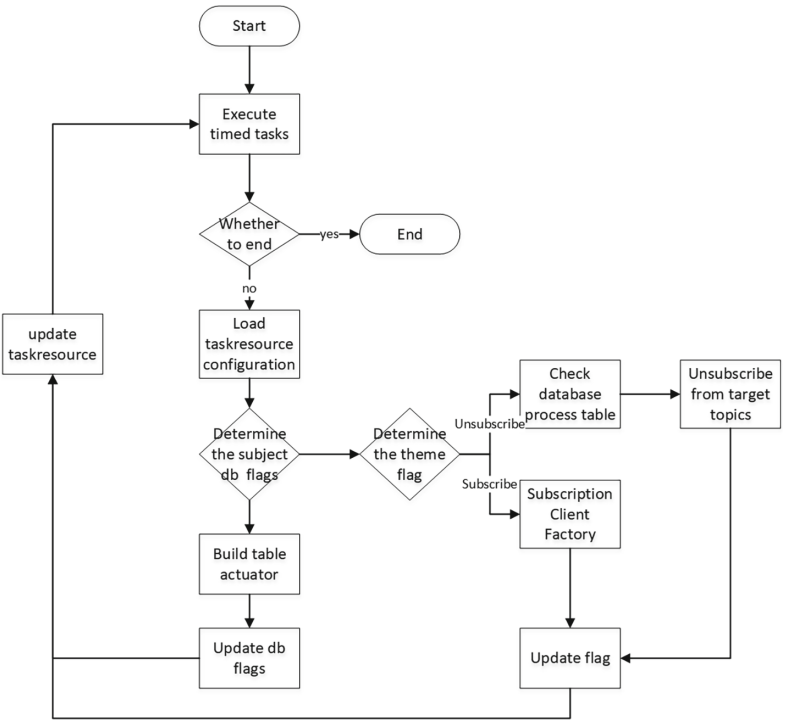


Fig. 3. Data access module flow description.

parameter to separate the data column from the tag column. Finally, the tag and data columns are divided into training and test data, which are returned to the next components.

Algorithm Assembly Plant. Usually, machine learning is generally hard-coded to import the algorithm modules required for training at the beginning of the code in a fixed form. However, such loading of modules and algorithms requires direct code writing operations by the user. In the case that the user may use the workflow to select any kind of algorithm, the implementation of an algorithm operation of the user requires static loading of all algorithm modules and classes, which is undoubtedly very rigid. At the same time This is undoubtedly very rigid and causes a huge waste of resources. Therefore, the unit of construction offers the ModelSet arithmetic setting plant, which utilizes the reflection organization in Python to attain the customized arithmetic components. This feature enables the ingredients for the framework to use the algorithm library in addition to scikitlearn library provides a very strong extension capability. Specifically, to facilitate the dynamic loading and assembly of user-selected algorithm classes, the methodology of instances accepts two string parameters via the workflow. It utilizes the import functions from the Python comprehensive library to

dynamically import the required algorithm module using the provided module name. Once the module is imported, the method employs the hasattr() function to check if the module contains the specified class name. If the class exists within the module, the getattr() function is used to retrieve the specific class object. When the user calls the module to load the component are illustrated in Fig. 4, this module will call this component to dynamically complete the loading of the tree module and return a DecisionTreeClassifier object.

```
<task taskName="t2" taskType="ModuleSet">
<parm parmName="module">sklearn.tree</parm>
<parm parmName="class">DecisionTreeClassifier</parm>
</task>
```

Fig. 4. User workflow example.

Model Training Components. Once the dynamic loading of the data for training and the assigned arithmetic module is completed, the next step is training. The training module receives several parameters passed through the workflow, including the training data, algorithm assembly, hyperparameters, and model preservation names. However, due to the different background knowledge of users, training the data via workflow needs to consider the user's mastery of machine learning algorithms, so the data analysis module is designed with three separate components to suit different users' needs. For users with sufficient knowledge and mastery of the analyzed data and machine learning algorithms, the MlTrain model training component is invoked for direct training. The instance method of the MlTrain training component needs to receive the algorithm name, parameters, model saving name and assembled algorithm module objects, pre-divided training data, and test data passed by the workflow. The instance method of this component first assembles the received hyperparameter strings into a dictionary format, then initializes and trains the algorithm model, and finally outputs the accuracy of its test set and store the model in a designated path.

AutoML Component. Although the above workflow has simplified the complexity of leveraging machine learning, a complete workflow for machine learning requires at least one call to either a data loading class component, a model loading class component, and a model training class component or a model parameter finding class component, which is still difficult if a subscribers has limitation in machine learning expertise, the purpose of the framework is to provide support and assistance to overcome this challenge. Therefore, it is necessary to further simplify the framework by providing a component that can automatically determine the algorithm and hyperparameters for the user based on the

dataset, assuming that the user does not know the algorithm and machine learning parameters. The ATM framework utilizes scikit-learn as the foundation for its algorithms, ensuring seamless integration and compatibility. It supports a wide range of popular classification algorithms, enabling users to leverage these algorithms within the framework effectively. The framework can call ATM through user-defined workflows by integrating the workflow engine. When the user provides a workflow configuration as depicted in Fig. 5, the approach of the AutoML component class expropriates parameters. It then invokes the ATM framework by passing these parameters to initiate the model and hyperparameter search process.

```
<task taskType="AutoMLTrain" taskName="step2">
<parm parmName="url">D: \heart_disease\heart.csv</parm>
<parm parmName="joblib">heart_aTM1.pkl</parm>
</task>
```

Fig. 5. User workflow example.

3.4 Integrated Model-as-a-Service Module

The Model-as-a-Service module contains a model repository and a model service backend. The model repository stores model files in the form of local files, and models are stored in the repository after being serialized by the above components. The model repository index table provides index information of model file locations, which enables other components to invoke the corresponding models in the repository by model file names. Using the previously designed components, the user can call them through the workflow description language, which completes the model service. However, the model invocation through the workflow language is limited to the users of this framework, and other external business systems cannot invoke the primed model, which generate a large waste of computing sources. Therefore, the model-as-a-service module opens the RESTfulAPI interface by constructing a flask service backend so that other systems can call the machine learning models generated by the framework through API extensions. The interface supports the post request method, and the raw type body parameters include string type ml parameters for the model index and array type data for model data parameters. After calling the API interface, the flask backend searches for the designated algorithm model in the model repository according to the corresponding parameters and calls the corresponding component of sklearn to load the model and output the model results. The return parameters include the String type resultInfo, which indicates the output model result, and the String type resultMsg, which indicates whether the API call is successful.

Based on this, this work also develops a corresponding front-end page based on this back-end, which invokes the model-as-a-service back-end to realize the workflow and generate the model's monitoring function. It provides the function of executing the workflow from the web page.

4 Model-as-a-Service Business Cases

4.1 Case Needs Analysis

Some elderly people may suffer from various diseases such as cognitive impairment and Alzheimer's disease. Detecting abnormal activities of the elderly at home may not only identify the risk of sudden illnesses, falls or coma, but also provide insight into the progress of chronic diseases of the elderly, so it is important to detect abnormal behavior points. In this case, the community elderly service center installs activity detection sensors in an elderly home environment, records activity data, including activity time points and durations. It classifies this data into normal and abnormal activity patterns by assigning appropriate labels. The labeled data is then utilized to train a model. During the training process, the framework learns from the labeled data, extracting patterns and characteristics that differentiate normal and abnormal activities in the elderly population. After the model is trained, it is saved for future use. When a request is made by an elderly user, the central system invokes the saved model.

4.2 Experimental Procedure and Results

The process of registering a data subscription operation is basically the same as in the previous case. First, the user needs to define a workflow consisted of regular interval execution and includes a task flow. The work is responsible for progressively subscribing to mount data by accessing the relevant resources from the resource configuration. In this scenario, the device data does not require any particular analysis, so it is unnecessary to declare a separate parsing component. The data subscription task flow directly accesses and retrieves the device data without any additional processing. The dynamic resource configuration for human activity detection is given in Fig. 6. After the data subscription task is executed, the framework calls the data access module to subscribe and store the device data.

In Fig. 7, the data analysis task flow is illustrated. It begins by executing a data loading task on lines 3–6, which loads the dataset for analysis. Next, an algorithm assembly task is defined on lines 7–10. This task calls the components and the sklearn.neighbors module and related classes. This allows for the incorporation of the K-nearest neighbors (KNN) algorithm into the analysis workflow. Subsequently, a machine learning training task is defined on lines 11–16. This task involves manually determining the parameters and training the KNN model. A data query task is defined on lines 17–20. This task is designed to query the pir table for specific data, focusing on records dated after May 5.

```
<IotWF>
    <taskResourse resourceName="RN_0002">
        <equipmentsub>
            <server>101.43.***.**</server>
            <port>18**</port>
            <subject db="0" flag="0">t</subject>
            <username></username>
            <password></password>
        </equipmentsub>
        <equipmentdb>
            <sub>t</sub>
            <type>customField</type>
            <table>pir</table>
            <field type="VARCHAR(255)">start_time</field>
            <field type="VARCHAR(255)">duration</field>
            <field type="DATE">create_date</field>
        </equipmentdb>
    </taskResourse>
</IotWF>
```

Fig. 6. Dynamic resourcing of human activity detection sensors.

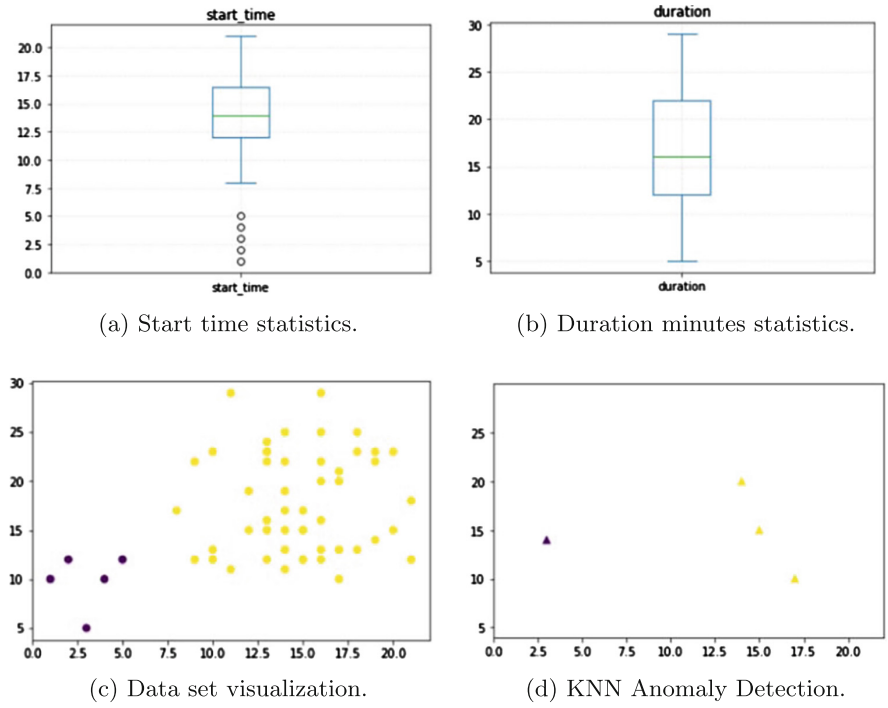
This enables the data analysis workflow to focus on the relevant and recent data for further examination. Finally, a model execution task is defined. This task utilizes assigned model in the model library to analyze and sense the queried data.

```
<IotWF>
    <TaskFlow taskFlowName="FN_0003" taskFlowType="AnalysisFlow" scheduleType="Once" scheduleValue="0">
        <task taskName="t1" taskType="SQLExtract">
            <parm parmName="SQL">D:\ecgPytorch\heart_disease\pir.csv</parm>
            <parm parmName="targetCol">3</parm>
        </task>
        <task taskName="t2" taskType="ModuleSet">
            <parm parmName="module">sklearn.neighbors</parm>
            <parm parmName="class">KNeighborsClassifier</parm>
        </task>
        <task taskName="t3" taskType="Ml_TRAIN">
            <parm parmName="alg">KNeighborsClassifier</parm>
            <parm parmName="p">{n_neighbors = 5, weights='distance'}
            </parm>
            <parm parmName="joblib">knn01.joblib</parm>
        </task>
        <task taskName="t4" taskType="SQL">
            <parm parmName="SQL">SELECT start_time,duration FROM pir where create_date > '2022-05-05'</parm>
            <parm parmName="outgoing">t5</parm>
        </task>
        <task taskName="t5" taskType="MlPredict">
            <parm parmName="joblib">knn01.joblib</parm>
        </task>
    </TaskFlow>
</IotWF>
```

Fig. 7. Definition of data analysis tasks below the behavioral anomaly detection scenario.

As depicted in Fig. 8(a) and 8(b), the visualization of the elderly behavior data reveals certain patterns. The regular schedule of activities for older persons is predominantly centralized between 13:00 and 16:00. The duration of these normal activities typically spans 10 to 20 min. However, there are four instances of abnormal activities during the timeframe of 0:00 to 5:00. These anomalies may be linked to the behavior of the elderly who need to find medicine urgently because of the health condition, so the data are labeled in advance before training, and the data set is formed to save. Therefore, before training, the data undergoes pre-labeling and is stored as a dataset. As shown in Fig. 8(c) and

8(d), the visualization of pre-labeled data obtained from the data subscription task clearly classifies whether the activity of the elderly is abnormal or not. When the task stream loads the data obtained from the data subscription task stream in rows 18–24 and invokes the model, the model determines that one of the data points is an abnormal point. This may indicate a change from the usual physical condition of the elderly and requires timely follow-up and understanding by the center's service staff.



(a) Start time statistics.

(b) Duration minutes statistics.

(c) Data set visualization.

(d) KNN Anomaly Detection.

Fig. 8. Behavioral anomaly detection data and model visualization.

5 Conclusion

Elderly health care has become more important in the context of the aging population. Using the MQTT protocol to collect and analyze various test data of the elderly remotely is tangible to understand the health condition of the elderly and also providing health guidance. However, existing studies face problems such as inconsistent message services and inconsistent data parsing. To address the above problems, we devise and implement a model-as-a-service oriented data integration framework for healthy aging, which automates the integration of processes such as data collection, parsing and storage, model building, and data

analysis for older people's IoT testing based on workflow which allows users to easily configure and customize their workflows without the need for extensive coding knowledge or expertise. Firstly, this paper abstracts and formalizes the ICH integration problem and workflow and defines a domain-driven language to describe the workflow. Secondly, we construct a health care IoT integration framework based on the problem model and provide a general introduction to the framework architecture. The framework implements dynamically managing access data and subscription. It fully takes into account different device message parsing and storage. The data analysis module is responsible for model training and analysis and provides customization and AutoML-based automation options for user selection of machine learning models and parameters. In addition, a model library and an external RESTful interface are built in the context of "model as a service". Finally, experiments are conducted to demonstrate the utilization of this frame to automate the process of dynamic subscription, parsing, storing, model training, and data convergence analysis of multi-device information using workflow.

In future research, in order to tackle complex tasks in various scenarios, techniques like streaming data analysis and deep learning automation can be effectively combined. This fusion enables the system to perform harnessing live data analytics for the promotion of healthy aging and the improvement of medical diagnostics.

References

1. Mutlag, A.A., Ghani, M., Arunkumar, N., et al.: Enabling technologies for fog computing in healthcare IoT systems. Futur. Gener. Comput. Syst. **90**, 62–78 (2018)
2. Hasan, H.M., Jawad, S.: IoT protocols for health care systems: a comparative study. Int. J. Comput. Sci. Mob. Comput. **7**, 38–45 (2018)
3. Sworna, N.S., Islam, A.M., Shatabda, S., et al.: Towards development of IoTML driven healthcare systems: A survey. J. Netw. Comput. Appl. **196**, 103–244 (2021)
4. Kulkarni, A., Sathe, S.: Healthcare applications of the internet of things: a review. Int. J. Comput. Sci. Inf. Technol. **5**, 6229–6232 (2014)
5. Habibzadeh, H., Dinesh, K., Shishvan, O.R., Boggio-Dandry, A., Sharma, G., Soyata, T.: A survey of healthcare internet of things (HIoT): a clinical perspective. IEEE Internet Things J. **7**, 53–71 (2020)
6. Kashani, M.H., Madanipour, M., Nikravan, M., Asghari, P., Mahdipour, E.: A systematic review of IoT in healthcare: applications, techniques, and trends. J. Netw. Comput. Appl. **192**, 103164 (2021)
7. Yang, Z., Zhou, Q., Lei, L., et al.: An IoTcloud based wearable ECG monitoring system for smart healthcare. J. Med. Syst. **40**, 1–11 (2016)
8. Laport, F., Dapena, A., Castro, P.M., et al.: A Prototype of EEG System for IoT. Int. J. Neural Syst. **30**, 1123–1134 (2020)
9. Yacchirema, D., de Puga, J.S., Palau, C., et al.: Fall detection system for elderly people using IoT and big data. Procedia Comput. Sci. **130**, 603–610 (2018)
10. Kadarina, T., Priambodo, R.: Monitoring heart rate and SpO2 using Thingsboard IoT platform for mother and child preventive healthcare. In: IOP Conference Series: Materials Science and Engineering, vol. 453, pp. 12–28 (2018)

11. Alazzam, M.B., Alassery, F., Almulihi, A.: A novel smart healthcare monitoring system using machine learning and the internet of things. Wirel. Commun. Mob. Comput. **2021** (2021)
12. Xu, B., Xu, L., Cai, H., Jiang, L.: Architecture of M-health monitoring system based on cloud computing for elderly homes application. In: 2014 Enterprise Systems Conference, Shanghai, China, pp. 45–50 (2014)
13. Hossain, M.S., Muhammad, G.: Cloud-assisted industrial internet of things (IIoT) - enabled framework for health monitoring. Comput. Netw. **101**, 192–202 (2016)
14. Gu, Y., Wang, P., Duan, H., Chen, H., Ren, Y.: Design of rescue and health monitoring bracelet for the elderly based on STM32. In: 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, pp. 1259–1262 (2019)
15. Verma, P., Sood, S.K.: Fog assisted-IoT enabled patient health monitoring in smart homes. IEEE Internet Things J. **5**, 1789–1796 (2018)
16. Pang, G.K.H.: Health monitoring of elderly in independent and assisted living. In: 2012 International Conference on Biomedical Engineering (ICoBE), Penang, Malaysia, pp. 553–556 (2012)
17. Frydrysiak, M., Tesiorowski, L.: Health monitoring system for protecting elderly people. In: 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, Croatia, pp. 1–6 (2016)
18. Reena, K.J., Parameswari, R.: IOT based health tracking shoe for elderly people using gait monitoring system. In: 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1701–1705 (2021)
19. Ning, Z., et al.: Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach. IEEE J. Sel. Areas Commun. **39**, 463–478 (2021)
20. Aburukba, R., Sagahyroon, A., Kamel, L.T., Al-Shamsi, A.M., Surti, H., Sajwani, E.: Remote monitoring framework for elderly care home centers in UAE. In: 2020 IEEE International Conference on E-health Networking, Application & Services (HEALTHCOM), Shenzhen, China, pp. 1–6 (2021)
21. Yuan, L.: Niagara framework technology for cloud edge collaborative intelligent building management system. In: Atiquzzaman, M., Yen, N., Xu, Z. (eds.) BDCPS 2021. LNDECT, vol. 102, pp. 1277–1283. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-7466-2_141
22. Xu, X., Huang, S., Feagan, L., et al.: Eaaas: edge analytics as a service. In: 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, pp. 349–356 (2017)
23. Eidenbenz, R., Locher, T.: Task allocation for distributed stream processing. In: IEEE INFOCOM 2016The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, pp. 1–9 (2016)
24. Elgamal, T., Sandur, A., Nguyen, P., et al.: Droplet: distributed operator placement for IoT applications spanning edge and cloud resources. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, pp. 1–8 (2018)
25. Dua, D., Graff, C.: UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml/datasets/Heart+Disease