IoT Network for Diagnosis of Parkinson's disease Using Neural Networks and OSTIS

Vishniakou Uladzimir
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
vish@bsuir.by

Yiwei Xia
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
xiayiwei4@gmail.com

***Abstract*—The aim of the work is to propose a model for the diagnosis of Parkinson's disease (PD) within the framework of the Internet of Things (IoT) using OSTIS. The report is devoted to the development of the Internet of Things for the diagnosis of Parkinson's disease (PD) using OSTIS technology. The structure of the ontology for describing the elements of PD disease is given. The construction of an IT diagnostic network of the PD is considered, which uses the semantic capabilities of the OSTIS platform for processing and analyzing medical data of the PD. The elements of the description of the knowledge base, solvers and user interfaces for PD using a component-based approach are presented.**

***Keywords*—Internet of Things network, IT diagnostics of Parkinson's disease (PD), PD ontology, neural networks, knowledge base, OSTIS**

I. DIAGNOSIS METHOD APPLIED TO PARKINSON'S DISEASE

The early diagnosis of Parkinson's disease has been a challenge for the medical community, and usually about 60 % of nigrostriatal neurons have degenerated and 80 % of striatal dopamine is depleted by the time the disease is diagnosed [1]. Currently, the diagnosis of the disease is based on medical history, clinical signs and symptoms, and response to antiparkinsonian drugs, but because the disease starts slowly and clinical symptoms appear only when the nigrostriatal dopamine neurons are depleted to a certain extent, patients are often at an advanced stage of the disease when they are diagnosed, missing the best time for treatment. Therefore, the development of new treatments in this field depends on two main aspects:

1) The early diagnosis of the disease.
2) The correct and constant evaluation of the effectiveness of the treatment.

In article [2] we proposed the method for complex recognition of Parkinson's disease using machine learning, based on markers of voice analysis and changes in patient movements on known data sets. The time-frequency function, (the wavelet function) and the Meyer kepstral coefficient function are used. The KNN algorithm and the algorithm of a two-layer neural network were used for training and testing on publicly available datasets on speech changes and motion retardation in Parkinson's disease. A Bayesian optimizer was also used to improve the hyperparameters of the KNN algorithm. The constructed models achieved an accuracy of 94.7 % and 96.2 % on a data set on speech changes in patients with Parkinson's disease and a data set on slowing down the movement of patients, respectively. The recognition results are close to the world level. The proposed technique is intended for use in the IoT network of IT diagnostics of PD.

II. KNOWLEDGE BASE AND ONTOLOGY

The construction of the Intelligent Parkinson's Disease Diagnosis System is based on the following three main components:

1) Knowledge Base: Collects and stores knowledge and data about Parkinson's Disease;
2) Problem Solver: Utilizes information from the knowledge base to solve specific problems;
3) Interaction Interface: Provides a way for doctors and patients to interact with the system.

A. Construction of the Knowledge Base for the Intelligent Parkinson's Disease OSTIS System

The ontology construction of the knowledge base of the Smart Parkinson's Disease Diagnostic OSTIS system is subdivided into the following areas.

Parkinson's Disease Ontology
$\subset$ *ontology*
$\subset$ *ostis-system*
$\subset$ *Familial neurodegenerative disease*
$\Rightarrow$ *Decomposition*:**:
{• *Basic Classification of Parkinson's Disease*
• *Clinical Features of Parkinson's Disease*
• *Parkinson-Plus Syndromes [3]*
• *Etiology of Parkinson's Disease*
• *Neuropathology of Parkinson's Disease*
• *Information Models of Parkinson's Disease*
  $\Rightarrow$ *Decomposition*:**:
  {• *Research Models [4]*
  • *Predictive Analytic Models*
  }
}

Predictive Analytic Models

⇒ *Decomposition*:*:*

{• *Medical Imaging Data [5]*

:= [Processing MRI, PET, SPECT, etc., imaging data for the diagnosis and research of Parkinson's Disease.]

• *Biomarkers [6]*

:= [Analyzing α-synuclein, inflammatory markers, oxidative stress markers, etc., to monitor disease progression.]

• *Genetic Information [7]*

:= [Analyzing Single Nucleotide Polymorphisms (SNPs) and gene mutations associated with an increased risk of Parkinson's Disease.]

• *Clinical Feature Data Processor*

⇒ *Decomposition*:*:*

{• *Motion Data Processor*

• *Voice Data Processor*

}

}

Voice Data Processor

:= [Changes in speech and language in patients with Parkinson's Disease, such as softer voice, slower speech, and reduced intonation variability, can be quantified through voice analysis technology, serving as a tool for assessing the disease and its progression.]

⇒ *Decomposition*:*:*

{• *Specific Voice Data Collection Methods*

⇒ *Decomposition*:*:*

{• *Model Selector*

⇒ *Decomposition*:*:*

{• *LSTM Model*

• *GRU Model*

• *KNN Model*

• *Random Forest Model*

}

• *Voice Data Feature Analyzer*

⇒ *Decomposition*:*:*

{• *Voice Feature Analyzer:*

• *Raw Voice Data Processor*

}

}

}

Motion Data Processor

:= [Involves patient's motor abilities and control, including tremors, muscle rigidity, bradykinesia, gait, and balance issues, analyzed through motion analysis technologies like wearable devices or motion capture systems.]

⇒ *Decomposition*:*:*

{• *Specific Motion Data Collection Methods*

⇒ *Decomposition*:*:*

{• *Model Selector*

⇒ *Decomposition*:*:*

{• *LSTM Model*

• *GRU Model*

• *KNN Model*

• *Random Forest Model*

}

• *Motion Data Feature Analyzer*

⇒ *Decomposition*:*:*

{• *db6 Wavelet Feature Analyzer:*

• *Raw Motion Data Processor*

}

}

}

B. Problem Solver for the Intelligent Parkinson's Disease Diagnosis System

The task resolver of the Intelligent Parkinson's Disease Diagnosis System is a collective of interacting sc-agents that facilitate the resolution of diagnostic and management issues related to Parkinson's disease. Herein is a fundamental decomposition of the task resolver for the Intelligent Parkinson's Disease Diagnosis System, based on the principal sc-agent classes designated for Parkinson's disease diagnostic purposes.

Intelligent Parkinson's Disease Diagnosis System Problem Solver

⇒ *Decomposition*:*

{• *Clinical Data Analysis abstract sc-agent*

⇒ *Decomposition*:*

{• *Motion Data Analysis abstract sc-agent*

• *Voice Data Analysis abstract sc-agent*

• *Genetic Information Analysis abstract sc-agent*

}

• *Symptom Severity Assessment abstract sc-agent*

• *Medical Database and Diagnostic Tool Interface abstract sc-agent*

⇒ *Decomposition*:*

{• *Medical Imaging Database Access abstract sc-agent*

• *Biomarker Analysis abstract sc-agent*

• *Time Series Analysis abstract sc-agent*

}

• *Symptom and Treatment Plan Correlation abstract sc-agent*

• *Diagnosis and Treatment Knowledge Base Verification abstract sc-agent*

⇒ *Decomposition*:*

{• *Treatment Plan Efficacy Verification abstract sc-agent*

• *Disease Diagnosis Accuracy Verification abstract sc-agent*

• *Patient Data Privacy and Security Protection abstract sc-agent*

}

• *Intelligent Question-Answering abstract sc-agent*

⇒ *Decomposition*:*

{• *User Interaction abstract sc-agent*

• *Medical Knowledge Graph*

• *Answer Generation abstract sc-agent*

}

}

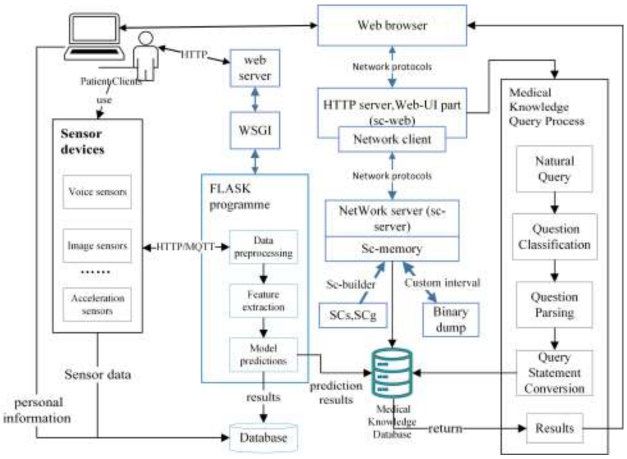

Figure 1. Framework Diagram for the Intelligent Parkinson's Disease OSTIS System's Automated Diagnostic and Question-Answering Tool.

III. System Architecture Overview

This section offers a comprehensive overview of the architectural underpinnings governing the deployment of our Parkinson's disease diagnosis model [2] within the IoT framework. It expounds upon the intricate interplay between system components designed to facilitate efficient data processing, storage, and presentation. Fig 1 illustrates an IoT system architecture enhanced with neural network capabilities using OSTIS technology.

IoT Device (Client): Data collection and preprocessing occur on the client-side. This could be a smart device such as a sensor or a smartphone, responsible for data acquisition and preliminary data preprocessing and feature extraction.

Local Flask Server [8]: The Flask server is located locally and serves as the receiver for data from the IoT device. It acts as an intermediary for data transmission, forwarding the received feature data to the OSTIS server.

OSTIS [9] Server: The OSTIS server is a knowledge graph platform that receives and processes data from the local Flask server. Running on this server is a Neural Network Predictor Agent responsible for loading internal neural network model files.

Neural Network Predictor Agent: This agent is responsible for loading and executing neural network models, processing incoming feature data, and making predictions. Predictions can be associated with knowledge within the OSTIS system and ultimately saved to a local database.

Local Database: The local database is used to store the prediction results returned from the OSTIS server, along with other relevant information.

The workflow of the entire system is as follows:

1) IoT device collects and preprocesses data.
2) Preprocessed data is sent to the local Flask server.
3) The local Flask server forwards the data to the OSTIS server.
4) The Neural Network Predictor Agent on the OSTIS server loads the neural network model and processes the data.
5) Determining the subclass, type, and subtype based on the Parkinson's disease diagnostic classifier, i. e., the types of diagnostic entities in medical ontology;
6) Establishing the inherent attributes and characteristics of the diagnostic category;
7) Determining the values of features for that diagnostic category;
8) Resolving polysemy in the diagnostic process;
9) Establishing the corresponding connections between diagnostic entities and concepts with medical semantic features in the knowledge base;
10) Establishing relationships between diagnostic entities belonging to a specific category of Parkinson's disease.
11) The processed results are associated with the knowledge graph and finally saved to the local database.

This system allows real-time data processing and complex object recognition in an IoT environment, combining data with a knowledge graph to support advanced analysis and decision-making. Proper configuration and management of each component are required to ensure the efficient operation of the system.

IV. IoT Device and Data Collection

Within this subsection, we embark on an in-depth exploration of the IoT device deployed for data collection. We elucidate its pivotal role in the acquisition of two critical data modalities: movement data and audio data. We delve into the intricacies of data preprocessing and transmission to the local server, underscoring the cardinal importance of real-time data acquisition capabilities. As shown in Figure 2, the IoT system architecture based on neural network with OSTIS technology is illustrated.

First, let us introduce the IoT device used. The device plays a key role in data acquisition by capturing two important data types: motion data and audio data. Motion data records the movement characteristics of Parkinson's patients, while audio data captures their sound characteristics.

To capture the acceleration data of the cell phone:

1) use a third-party library in Python (PySensors) to access the phone's acceleration sensor.
2) set up the sensor parameters, setting the sampling rate to 64hz and the sensor precision to 16 bits.
3) create a data storage structure to hold the acquired acceleration data.
4) In a loop, periodically read the acceleration sensor data and store it in the data structure.
5) Store the data in a local file for further processing and analysis.

Collect the voice data from the cell phone:

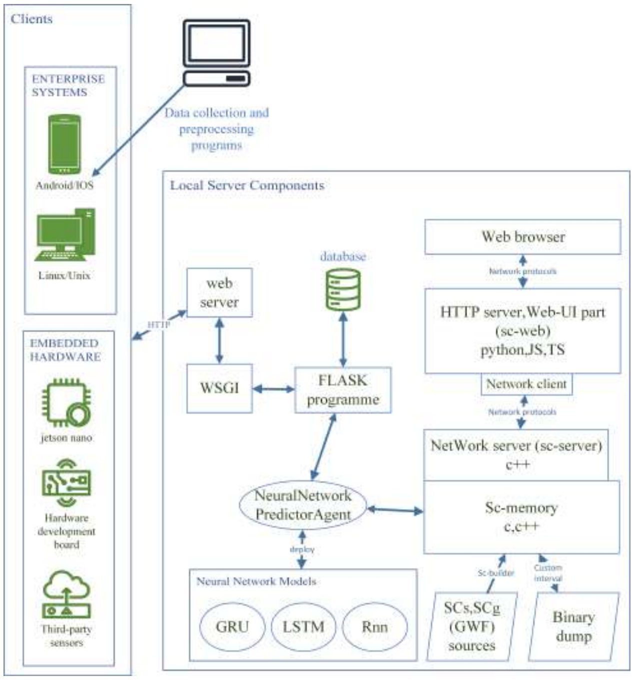

Figure 2. IoT system architecture based on neural network with OSTIS technology.

1) Use a third party library (PyAudio) in Python to access the microphone of the phone.
2) Initialize the microphone and set the audio parameters . The sample rate is 44100hz, the number of channels is 1 and the bit depth is 16.
3) Create an audio stream for receiving and recording voice data.
4) Start recording voice data and store it on a local file.

The next step after collecting and recording accelerometer and voice data from Parkinson's disease patients is to preprocess and extract features from these data. This process is aimed at preparing the data for further analysis and model training.

Data Preprocessing and Feature Extraction Process:

1) Data Cleaning and Calibration: Firstly, the collected data undergoes initial cleaning and calibration to remove potential noise and outliers. For accelerometer data, sensor calibration is performed to ensure data consistency within a common reference frame.
2) Time and Frequency Domain Analysis: Time and frequency domain analyses are conducted on voice data to obtain fundamental signal characteristics. Time domain analysis includes waveform shape, energy, duration, and more. Frequency domain analysis encompasses spectral distribution, frequency components, and related features.
3) Feature Extraction: Features are extracted from accelerometer and voice data, representing informative and meaningful attributes that aid in subsequent model training and analysis. For accelerometer data, statistical features of motion patterns such as mean, standard deviation, energy, etc., can be extracted. For voice data, sound features like fundamental frequency, pitch, spectral features, etc., can be extracted.
4) Feature Normalization: Extracted features are normalized to ensure they share similar scales, preventing certain features from disproportionately influencing model training.
5) Feature Selection: With a large number of extracted features, feature selection can be performed to choose the most relevant and useful ones, reducing dimensionality and enhancing model efficiency and performance.
6) Data Storage: The feature data is saved as a Comma-Separated Values(CSV) file for local storage.

The specific process of transferring data from a mobile device to the Flask server on a computer is as follows:

1) The mobile device collects accelerometer data or voice data and performs preprocessing and feature extraction.
2) The feature data is packaged in JSON format. Below is a JSON example:

```
{
  "data_type": "acceleration",
  // or "voice"
  "features":
    [
      { "feature_name":
      "feature_1",
        "value": 0.123},
      { "feature_name":
      "feature_2",
        "value": 0.456}
      // More features...
    ]
  }
```

3) Send the JSON data to the Flask server using the HTTP protocol with a POST request. Target URL: http://192.168.100.14:5000/getdata. The request header includes data type information and can be set as 'Content-Type: application/json'.
4) The Flask server receives and handles the request at the 'getdata' route.
5) Upon reception of the data on the server side, a range of distinct processing strategies can be applied contingent on the data type. These strategies encompass data storage within local files, conducting more in-depth analytical procedures, or the utilization of predictive models for decision-making.

In this process, data is transferred from the mobile device to the Flask server, which receives the data via the specified route and performs the necessary processing.

V. LOCAL SERVER SETUP WITH OSTIS SYSTEM

The OSTIS Web Platform [10] is a web-oriented software platform of the OSTIS Project. It serves as a robust framework for deploying existing OSTIS systems and creating new ones.

The OSTIS Web Platform includes the following components:

1) Knowledge Base [11]: It contains top-level ontologies to assist in developing various information models.
2) Knowledge Processing Machine [12]: This component features semantic network storage and agent-based knowledge processing.
3) Web-Oriented Semantic Interface [13]: It allows users to interact with the intelligent system.

Below are the steps for a quick start using Docker Compose, suitable for windows:

1) Clone the repository and navigate to the directory.
2) Download images from Docker Hub:

```
docker compose pull
```

3) Build the knowledge base:

```
docker compose run machine
build
```

4) Launch the web platform:

```
docker compose up
```

Using Docker Compose:

1) Build the Knowledge Base (required before the first startup or if you've made updates to KB sources):

```
docker compose run machine
build
```

2) Start platform services and access the web interface locally (address: localhost:8000):

```
docker compose up
```

3) Launch the knowledge processing machine:

```
./scripts/run_sc_server.sh
```

4) In another terminal, launch the semantic web interface (address: localhost:8000):

```
./scripts/run_sc_web.sh
```

When the Flask server receives data, it forwards it to the agent within the OSTIS server, namely the neural network predictor agent. The primary role of this agent is to load the appropriate neural network model and process the received data. During the processing, the model makes predictions and generates results.

Subsequently, the model stores these results in a database for further analysis and persistent storage. Once the results are successfully stored, the Flask server constructs a response, encapsulating the prediction results in JSON format, and sends it back to the client for users to access real-time analysis results.

This process ensures data integrity and persistence while allowing users to interact with the OSTIS server through the Flask server to access the information they need.

VI. DISPLAYING DATA IN THE OSTIS SYSTEM

Once the prediction results are stored in the OSTIS system's database, they can be accessed and displayed within the system's knowledge base. This allows users to interact with and visualize the results for further analysis or decision-making. The process of displaying data in the OSTIS system involves the following steps:

1) Knowledge Base Integration: The prediction results, which are stored in the database, need to be integrated into the OSTIS knowledge base. This integration typically involves creating or updating knowledge structures within the system to represent the newly acquired data.
2) Semantic Representation: The data should be represented in a semantically meaningful way using the OSTIS Semantic Computer Code (SC-code) [14] to ensure compatibility with the system's knowledge processing capabilities. This may involve defining new semantic relationships or entities to represent the data.
3) User Interface: OSTIS provides a web-oriented semantic interface that allows users to interact with the intelligent system. This interface can be customized to display the prediction results in a user-friendly and informative manner. Users can query the system to access specific data and visualize it through the interface. Figure 3 displays SCn-nodes featured within the OSTIS web interface's knowledge base.
4) Visualization and Analysis: Depending on the nature of the prediction results, the OSTIS system may offer various visualization tools and analysis capabilities. Users can explore the data, generate reports, or perform further analysis within the system. Figure 4 presents SCg-nodes displayed within the knowledge base of the OSTIS web interface.

VII. Conclusion

The report presents the architecture of the IoT network for the diagnosis of Parkinson's by voice and movement of patients. The IoT device collects and preprocesses patient data using a smartphone to collect data and extract the necessary functions from them. This recognition data is transmitted through the local Flask server, which acts as the main intermediary. The heart of the system, the OSTIS server, serves as a knowledge graph platform hosting a neural network prediction agent that downloads, executes and links predictions with existing knowledge. The advantage of this system lies in its ability to facilitate real-time data processing and recognition of complex objects in the IoT network. Combining data and

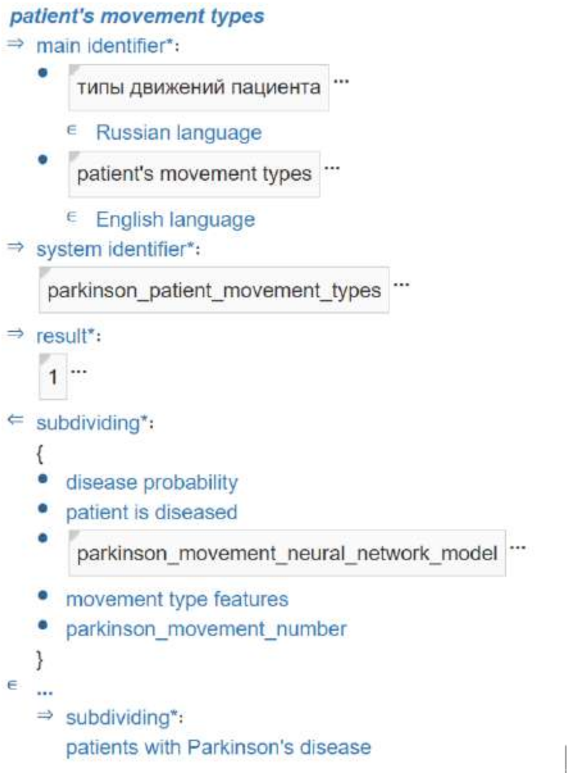

Figure 3. SCn-nodes showcased in the OSTIS web interface's knowledge base.

knowledge base allows for in-depth analysis and informed decisions in the field of diagnosis of Parkinson's disease. This work highlights the potential of the IoT network and deep learning technologies for rapid remote IT diagnostics. An innovative combination of data processing, neural networks and knowledge base integration promises a more accurate and timely diagnosis of Parkinson's disease, which will ultimately improve the further treatment of patients.

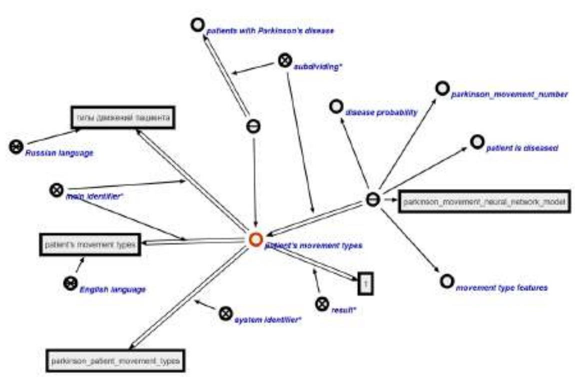

Figure 4. SCg-nodes showcased in the OSTIS web interface's knowledge base.

References

[1] W. Dauer and S. Przedborski, "Parkinson's disease: mechanisms and models," *Neuron*, vol. 39, no. 6, pp. 889–909, 2003.

[2] U. Vishnyakou and X. Yiwei, "Recognition of signs of parkinson's disease based on voice markers and motor activity," *Informatics*, no. 3, pp. 106–114, 2023.

[3] A. C. Vertes, M. R. Beato, J. Sonne, and M. Z. K. Suheb, "Parkinson-plus syndrome," in *StatPearls [Internet]*. StatPearls Publishing, 2023.

[4] S. J. Chia, E.-K. Tan, and Y.-X. Chao, "Historical perspective: models of parkinson's disease," *International journal of molecular sciences*, vol. 21, no. 7, p. 2464, 2020.

[5] J. Zhang, "Mining imaging and clinical data with machine learning approaches for the diagnosis and early detection of parkinson's disease," *NPJ Parkinson's disease*, vol. 8, no. 1, p. 13, 2022.

[6] U. Ganguly, S. Singh, S. Pal, S. Prasad, B. K. Agrawal, R. V. Saini, and S. Chakrabarti, "Alpha-synuclein as a biomarker of parkinson's disease: good, but not good enough," *Frontiers in aging neuroscience*, vol. 13, p. 702639, 2021.

[7] H. Pan, Z. Liu, J. Ma, Y. Li, Y. Zhao, X. Zhou, Y. Xiang, Y. Wang, X. Zhou, R. He *et al.*, "Genome-wide association study using whole-genome sequencing identifies risk loci for parkinson's disease in chinese population," *npj Parkinson's Disease*, vol. 9, no. 1, p. 22, 2023.

[8] V. R. Vyshnavi and A. Malik, "Efficient way of web development using python and flask," *Int. J. Recent Res. Asp*, vol. 6, no. 2, pp. 16–19, 2019.

[9] N. Zotov, "Implementation of information retrieval subsystem in the software platform of ostis-systems," *Open Semantic Technologies for Intelligent Systems (OSTIS)*, vol. 7, pp. 77–94, 2023.

[10] M. Sadouski, "User interface of the ostis ecosystem," *Open Semantic Technologies for Intelligent Systems (OSTIS)*, vol. 7, pp. 153–158, 2023.

[11] I. Davydenko, "Ontology-based knowledge base design," *Open Semantic Technologies for Intelligent Systems (OSTIS)*, pp. 57–72, 2017.

[12] D. Shunkevich, "Ontology-based design of knowledge processing machines," *Open Semantic Technologies for Intelligent Systems (OSTIS)*, pp. 73–94, 2017.

[13] M. Sadouski, "Ontological approach to the building of semantic models of user interfaces," *Open Semantic Technologies for Intelligent Systems (OSTIS)*, vol. 5, pp. 105–116, 2021.

[14] K. Bantsevich, "Metasystem of the ostis technology and the standard of the ostis technology," *Open Semantic Technologies for Intelligent Systems (OSTIS)*, vol. 6, pp. 357–368, 2022.

СЕТЬ IOT ДЛЯ ДИАГНОСТИКИ БОЛЕЗНИ ПАРКИНСОНА С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ И OSTIS

Вишняков В.А., Ивей С.

Цель работы состоит в том, чтобы предложить модель диагностики болезни Паркинсона (БП) в рамках Интернета вещей (IoT), используя ОСТИС. Доклад посвящен разработке сети Интернет вещей для ИТ-диагностики болезни Паркинсона (БП) с использованием технологии OSTIS. Приведена структура онтологии для описания элементов заболевания БП. Рассматривается построение ИТ-диагностической сети БП, которая использует семантические возможности платформы OSTIS для обработки и анализа медицинских данных БП. Приведены элементы описания базы знаний, решателей и пользовательских интерфейсов для БП с использованием подхода, основанного на компонентном прое.

Ключевые слова: сеть Интернет вещей, ИТ-диагностика болезни Паркинсона (БП), онтология БП, нейронные сети, база знаний, ОСТИС.

Received 31.03.2024