

RESEARCH

Open Access



# The design of an indirect method for the human presence monitoring in the intelligent building

Jan Vanus<sup>\*</sup> , Jaroslav Machac, Radek Martinek, Petr Bilik, Jan Zidek, Jan Nedoma and Michal Fajkus

\*Correspondence:

jan.vanus@vsb.cz  
Department of Cybernetics  
and Biomedical Engineering,  
Faculty of Electrical  
Engineering and Computer  
Science, VSB-Technical  
University Ostrava, 17,  
Listopadu 15, Poruba, 708  
33 Ostrava, Czech Republic

## Abstract

This article describes the design and verification of the indirect method of predicting the course of CO<sub>2</sub> concentration (ppm) from the measured temperature variables  $T_{\text{indoor}}$  (°C) and the relative humidity  $rH_{\text{indoor}}$  (%) and the temperature  $T_{\text{outdoor}}$  (°C) using the Artificial Neural Network (ANN) with the Bayesian Regulation Method (BRM) for monitoring the presence of people in the individual premises in the Intelligent Administrative Building (IAB) using the PI System SW Tool (PI-Plant Information enterprise information system). The CA (Correlation Analysis), the MSE (Root Mean Squared Error) and the DTW (Dynamic Time Warping) criteria were used to verify and classify the results obtained. Within the proposed method, the LMS adaptive filter algorithm was used to remove the noise of the resulting predicted course. In order to verify the method, two long-term experiments were performed, specifically from February 1 to February 28, 2015, from June 1 to June 28, 2015 and from February 8 to February 14, 2015. For the best results of the trained ANN BRM within the prediction of CO<sub>2</sub>, the correlation coefficient R for the proposed method was up to 92%. The verification of the proposed method confirmed the possibility to use the presence of people of the monitored IAB premises for monitoring. The designed indirect method of CO<sub>2</sub> prediction has potential for reducing the investment and operating costs of the IAB in relation to the reduction of the number of implemented sensors in the IAB within the process of management of operational and technical functions in the IAB. The article also describes the design and implementation of the FEIVISUAL visualization application for mobile devices, which monitors the technological processes in the IAB. This application is optimized for Android devices and is platform independent. The application requires implementation of an application server that communicates with the data server and the application developed. The data of the application developed is obtained from the data storage of the PI System via a PI Web REST API (Application Programming Integration) client.

**Keywords:** Monitoring, Intelligent Administrative Building (IAB), Presence of People, Bayesian Regulation Method, Artificial Neural Network, PI System, PI ProcessBook, JSON, Spring MVC, PI Web API, LMS, DTW

## Introduction

Intelligent or automated building systems play a crucial role in today's modern sophisticated buildings. Monitoring and automatic control systems are important for achieving the required level of building technology behaviour. From the perspective of the development of web/mobile applications, it is possible to use javascript applications for visualization of technological processes in an administrative building. Javascript has become a standard language for developing web browsers and hybrid applications today. Implementation of visualization and monitoring applications are becoming more and more useful in today's technological applications, e.g. in connection with the development of the Smart Cities (SC) technological concept. For the visualization and monitoring of the operational and technical functions in multiple buildings within the IAB (Smart Home (SH), Smart Home Care (SHC) [1]), it is possible to use more robust visualization and monitoring SW tools [2], which have already become widespread in larger administrative buildings [3]. The possibilities of developing visualization and monitoring applications using wireless components are described in detail in articles [4–7]. The actual monitoring of operational and technical variables may be used for optimizing the management of individual operational and technical functions of buildings, such as lighting [8], [9] or HVAC (Heating, Ventilation, and Air Conditioning), in order to reduce the operating costs of the building, for setting the optimal climate in individual rooms of the building [10] with respect to the requirements of the hygienic standards, or for classification [11], prediction [12] and foreseeing of building behaviour based on the variables [13] measured and processed.

To determine room occupancy in an IAB, a wide range of sensors can be used, for example rH sensors, wearable inertial sensors or wearable vital signs sensors [14], humidity sensors [15], RFID sensors [16], microphones [17], movement sensors [18], video surveillance [19], light sensor [20], presence sensors [21], CO<sub>2</sub> sensors [22], ambient sensors [23], video cameras [24], simple binary sensors or temperature sensors etc. Two approaches can be used for monitoring the occupancy of individual rooms in an administrative building [25]:

- Distributed Direct Sensing (DDS), which includes the option of installing special sensors for capturing the life activities of the users of the administrative premises. This approach does not respect the privacy and the personal zone of the people so much, it rather disrupts these needs.
- Infrastructure Mediated Sensing (IMS) which uses the existing sensors for managing the operational and technical functions in buildings. Temperature sensors for heating control, light intensity sensors for lighting control in individual rooms, humidity or CO<sub>2</sub> sensors for controlling forced ventilation in rooms, etc., may be used as examples. This approach rather respects the privacy and the personal zone of the people.

For recognition and classification of occupancy in an IAB, various mathematical methods, such as the Hidden Markov Model (HMM) [26], the Decision Trees method, the Bayesian network-based method, Support Vector Machines (SVM) [27],

[28], Artificial Neural Networks (ANN) [29], the Linear Discriminant Analysis (LDA) [30], the Adaptive systems [31], and other Soft Computing methods are used.

The information on the occupancy of an IAB is used for HVAC control [32], access systems, facility management of buildings [33], security systems [34], Energy saving control [35], maintenance staff of buildings [36], monitoring and visualization [37], human behaviour recognition [38], light control, building management systems and others.

Some of the above-described works share a common way of monitoring the occupancy of individual rooms of an IAB using CO<sub>2</sub> sensors. Carbon dioxide is the most common air pollutants inside residential buildings. Humans, their metabolism, breathing and thermoregulation processes are primarily the source of this gas inside buildings. The increasing concentration of carbon dioxide simultaneously increases the amount of water vapour in the air and, thus, also the relative humidity of the air. The number of people present in the room, the size of the room and insufficient ventilation are the main causes of the increase in carbon dioxide concentration. Higher concentrations of carbon dioxide adversely affect breathing—already at concentrations above 15,000 ppm. If its concentration in the air rises above 30,000 ppm, most people suffer from headaches, dizziness, and nausea [39]. The current efforts to maximize energy savings on heating and operation of buildings in the Czech Republic in the last 15 years most often include thermal insulation of facades and replacement of windows. In recent years, an increasing number of companies in the Czech Republic have become aware of the fact that these well-intentioned measures have a significant negative impact on human health [40]. CO<sub>2</sub> concentration in the air is a suitable indicator of stuffiness in an enclosed space and very well corresponds to the number of people staying in this enclosed space. By the breathing process, the inhaled oxygen is changed to carbon dioxide and the air exhaled by an adult human contains on average about 35,000–50,000 ppm of CO<sub>2</sub> (about 100 times higher concentration than in the outside air). Without the corresponding forced ventilation, then, logically, there is an increase in CO<sub>2</sub> concentration in the enclosed space [41].

The objective of the article is the design of a new method for monitoring the presence of people in an IAB office room using the SW Tool PI System (PI—Plant Information enterprise information system) within IAB automation from the measured temperature variables  $T_{\text{indoor}}$  (°C) and the relative humidity  $rH_{\text{indoor}}$  (%) and the temperature  $T_{\text{outdoor}}$  (°C) using the Artificial Neural Network (ANN) with the Bayesian Regulation Method (BRM).

The goal of the first part of the article is the design and the implementation of the FEVISUAL visualization monitoring application for the IAB of the new FEI at VŠB TU Ostrava using an Android mobile device that monitors the technological processes (HVAC) in the IAB. The application will be created using the PI System. In the IAB described, there is currently no processing and evaluation of room occupancy information for optimizing management of operational and technical functions in order to reduce the operational costs of the building. There is also no monitoring of CO<sub>2</sub> concentration in the air inside the individual rooms of the administrative building.

The second part of the article describes the design of a new indirect method and evaluation of the IMS approach for indirect monitoring of the presence of people of the individual IAB areas using prediction of the course of CO<sub>2</sub> concentration (ppm) from the measured temperature  $T_{\text{indoor}}$  (°C) indoors and the relative humidity  $rH_{\text{indoor}}$  (%) and the

temperature  $T_{\text{outdoor}}$  (°C) in order to determine the presence of people of the monitored areas using operational sensors installed in the building.

The aim of this part of the article is to verify the use of ANN for predicting the measured quantities for the purpose of monitoring the person presence in the interior of the IAB office room with the gradient algorithm of error backpropagation using the BRM. For the classification of the prediction quality, a Correlation Analysis (correlation coefficient R), the calculated MSE (mean squared error) were used. In addition, an adaptive filter with the LMS algorithm will be used for filtering the output of the predicted CO<sub>2</sub> course and, thus, for improving the reliability of the designed indirect method. The DTW criterion will be used for classifying and comparing the output course.

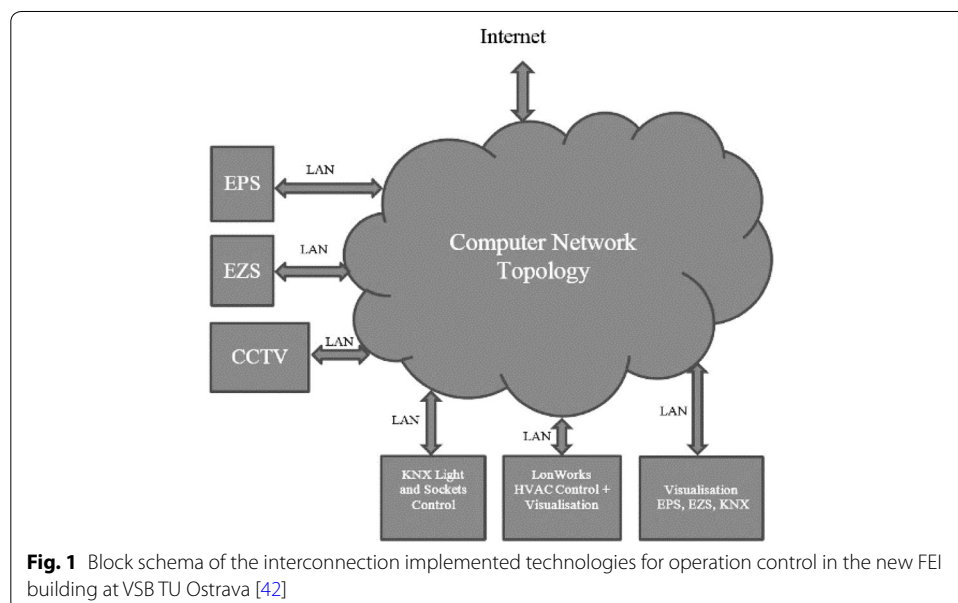
The main contribution of the article is to verify the assumption of improving the accuracy of the CO<sub>2</sub> prediction method using ANN BRM with the LMS adaptive filter algorithm to remove additive noise from the predicted CO<sub>2</sub> course.

Another goal is to verify the proposed method on the measured data for different seasons in a time interval between winter and spring (from February 1 to February 28, 2015) and for a time interval between spring and summer (from June 1 to June 28, 2015) in long experiments and for a time interval (from February 8 to February 14, 2015) in short experiment in winter.

**Part One—The design and the implementation of the visualization monitoring application FEVISUAL for the administrative building of the new FEI**

In the new FEI building at VSB—Technical University Ostrava, the HVAC (heating, ventilation and air conditioning) system is controlled using the LonWorks bus system. Other implemented technological systems in the described building include (Fig. 1):

- Control of lighting and outlets using the KNX technology.
- Electronic security system (ESS).



- Electronic fire alarm system (EPS).
- CCTV (closed circuit television) circuit.
- Visualization of EPS, ESS.
- Visualization of HVAC.
- Visualization of the condition of lighting and outlets.
- Wireless system for controlling blinds.

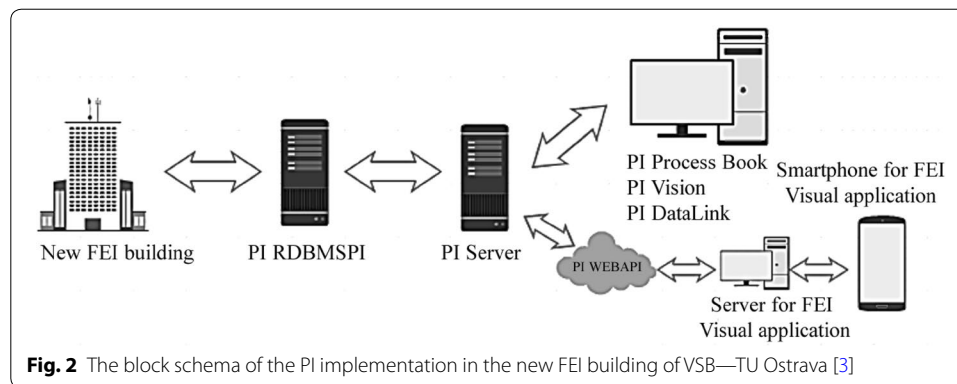
In terms of the execution of the management of operation and technical features of the individual structures and buildings located within the area of VSB TU Ostrava, it is possible to generally say that various types of technologies, standards, data protocols and bus systems for building automation (KNX, LonWorks, BACnet) have been used here. At present, the need to implement a unifying tool (BMS—building management system) for processing and analysis of the measured data from the implemented technology, not only in the new FEI building at the Technical University of Ostrava, but also at other buildings of the whole area of the Technical University of Ostrava, becomes obvious. A properly chosen BMS tool can serve for ensuring monitoring of the condition of the controlled devices, active alerting operators to the occurrence of atypical events—faults, exceeding threshold values with the monitored variables in the environment of the building (e.g. room temperature) or data archiving for the retrospective analysis and subsequent optimization of the building operation in order to reduce the operating costs, thus ensuring energy savings [43], for example with wireless sensor network [44].

### **PI System implementation**

To unify the monitoring and control of various automation and electronic systems into a single BMS (building management system) environment, it is necessary to use a software tool that allows you to consolidate several types of bus systems, standards and communication and data protocols into a single monitoring application. In our case, we took advantage of the opportunities offered by the PI System software tool (hereinafter referred to as PI) produced by OSIsoft. This is an industry standard designed for industrial enterprises and IAB for data reporting and real-time events. PI allows for easier retrieval, storage, distribution, analysis and visualization of data from various sources such as (database systems, SCADA systems, manual inputs, application software, actuators, sensors etc.). Individual PI programs or groups of PI programs can also exist separately. PI System can be roughly divided into a data, server and client part. The client part of the PI (PI client) system is suitable for providing BMS in the described building. PI clients are PI software applications, with which end users work and which present the data in a user-friendly environment with interesting graphics. These are desktop applications (PI DataLink, PI ProcessBook, PI Data Access) and web applications (PI WebParts, PI ActiveView, PI Vision).

### **PI System description**

The block schema of the implementation of the PI System to monitoring technical and operating functions (HVAC) in the new FEI building in the VSB Technical University Ostrava is shown below in Fig. 2. The acquired data measured by operating variables (HVAC) are stored in a database, from where they are stored on the



PI server (PIAF) via the RDBMSPI interface. This data is then used in desktop and web applications on a remote computer, which communicates with the PI server. The applications that connect to PI, use PI SDK (Software Development Kit), PI API (Application Programming Integration) libraries or a combination of both. PI SDK and PI API are dynamically linked libraries, through which the individual applications communicate with the PI server (Fig. 2).

Libraries are installed on a computer as a part of the PI SDK installation process.

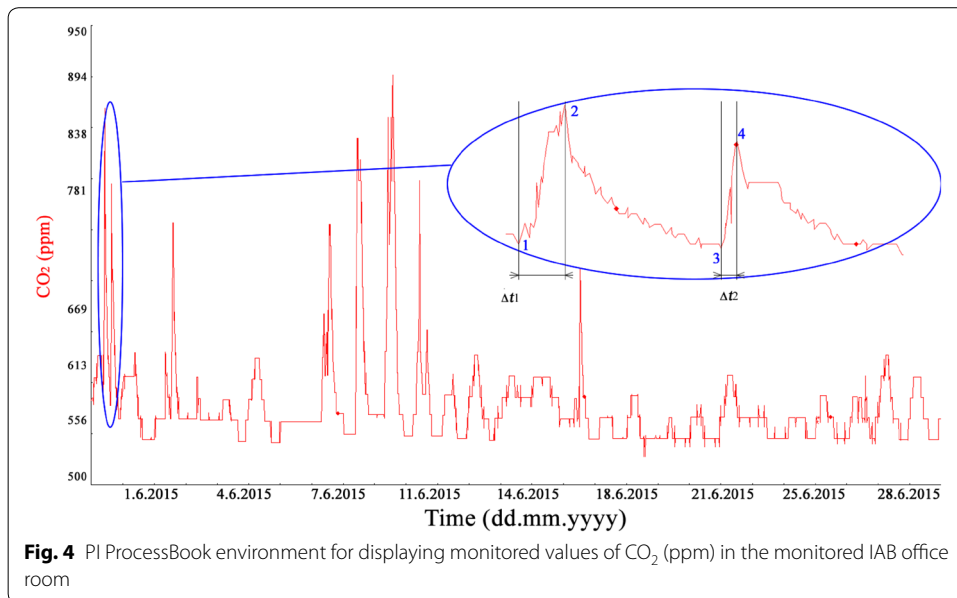
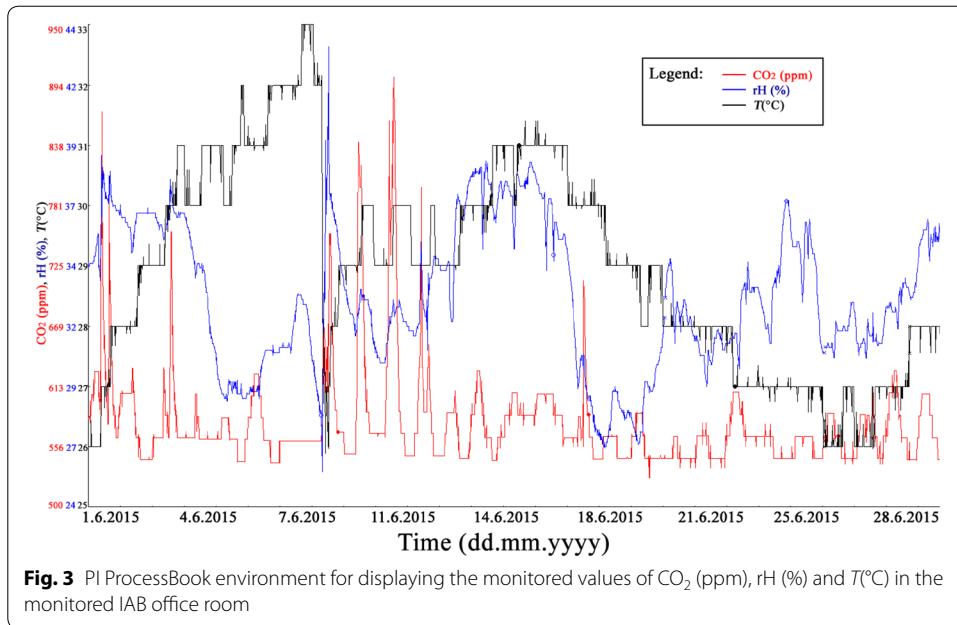
#### **PI DataLink**

It is a desktop application. It establishes a direct connection between the PI System and Microsoft Excel software to view data in real time (Fig. 15). PI DataLink can also process and analyze the historical data. PI DataLink performs such tasks as statements of values with their timestamps, reporting, modelling, analysis, aggregation function of time periods (hourly averages) and planning process [3].

#### **PI ProcessBook**

For monitoring the presence of people in the individual room of IAB SW TOOL PI, ProcessBook was used (Fig. 3). PI ProcessBook allows instant access and data visualization both in real-time and in the past, using interactive, graphical displays of the time. The data can be viewed through the trends (Fig. 4), dynamic symbols, tables, numerical values for subsequent processing and analysis of the measured data. For fast, basic statistical data analysis, it is possible to use the table (Fig. 5) with the basic statistical parameters of the measured data source (Average, Minimum, Maximum, Range, Standard Deviation, Time interval ...) [3]. Figure 3 shows the measured waveforms of non-electrical  $\text{CO}_2$  indoor values (ppm),  $r_{\text{H}}^{\text{indoor}}$  (%) and  $T_{\text{indoor}}$  ( $^{\circ}\text{C}$ ) in the monitored area using the PI ProcessBook in the period from June 1, 2015 to June 28, 2015.

Figure 4 shows the measured course of  $\text{CO}_2$  in the monitored IAB area. From the measured courses, the arrival at the room (point 1 and point 3), the departure from the room (point 2 and point 4) can be determined. It is also possible to determine the occupancy time of the monitored room at the selected time  $\Delta t_1$  and  $\Delta t_2$ . A more detailed description of the departures and arrivals of people in the monitored room within a given time is shown in Fig. 20.



**Technologies, tools and frameworks used for the design of the application for mobile devices running Android**

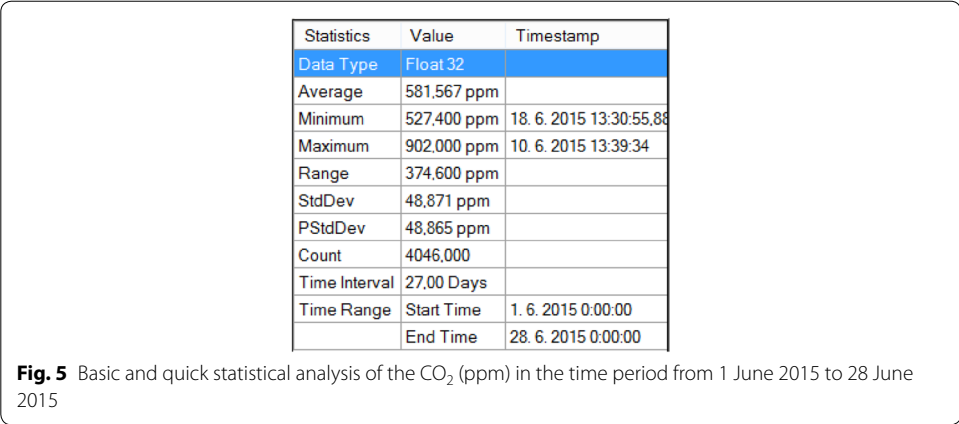
**Server part—Spring MVC**

Spring MVC is mainly used for creating web applications. The main advantage of Spring MVC is the simplification of the development of Java applications.

**Server part—Apache Tomcat**

Apache Tomcat is a web server and servlet container. It is being developed as an open source project. Apache Tomcat will be used as a container for testing and development of custom applications.





**Server part—PI Web API**

PI Web API is a member of the PI System software tools. RESTful is the PI System access layer that provides a cross-platform programming interface for PI System. The RESTful interface allows for the cross-platform development of web, desktop and mobile applications in various programming languages. PI Web API enables the retrieval and manipulation of data time-series from the PI data archive, and also loads and configures asset data and framework events from the PI AF server. It contains a set of REST services and APIs, which are being further developed. It is a web-based communication method.

It is a commonly used tool for bilateral communication and communication with clients, because it allows companies to communicate between different IT infrastructures. In essence this creates a universal language for communication across IT infrastructures. The most commonly used models of web services are SOAP and REST. The Web service client in this case sends an HTTP GET request to the URL <https://picoresight/PIVision/PB/#/PBDisplays/423> for the Web server services. The URL indicates the source location which the client wants to access. Designations such as GET, POST or PUT indicate the action with which the client wants to query the server. When the server receives this request, it processes it and sends a response back to the client as XML1 or JSON2. Web API has the following system requirements:

- Windows Server 2008 R2, 2012 or 2012R2 (2012 or newer).
- PI data archive 3.4.380 or newer (2012 or newer).
- PI Asset Server 2.1 or higher (2012 or newer).

**Server part—Eclipse IDE**

Another important tool is the development environment. The officially supported one is the Eclipse IDE, currently the most popular Java development environment.

**Client part—Hybrid development**

Hybrid development combines both native and web approaches. The application consists of a native frame in which the web view and the web application are located. It is part of the application or it can be downloaded remotely via the Internet. The pros



and cons of hybrid applications arise from the nature of the two approaches combined in the hybrid development. The advantage of hybrid applications compared to the web ones is the possibility of publication in the official platform store. The user installs and runs them alongside the classic native applications. Another key advantage compared to traditional web applications is the ability to use frameworks which link the web application running in WebView with the system.

#### ***Client part—Framework Sencha Touch***

The application programming is done using JavaScript and not the HTML5 markup. Although Sencha Touch takes full advantage of HTML5, the whole application code is placed in JavaScript files, and the HTML file contains only skeleton pages with the embedded application script. Sencha Touch has its own library named Ext. It is a robust library with various components. Sencha Touch makes writing the application code a little harder than it is in an ordinary JavaScript or e.g. jQuery framework, mainly due to its special syntax. In addition to writing the code, however, the Sencha Touch framework is much more robust than e.g. the mentioned jQuery Mobile that contains more functionality; while a full-fledged Web application with all the trimmings can be created only with the help of this framework. Sencha Touch has so much functionality and settings that the programmer is not prevented from doing almost anything.

#### ***Client part—PhoneGap***

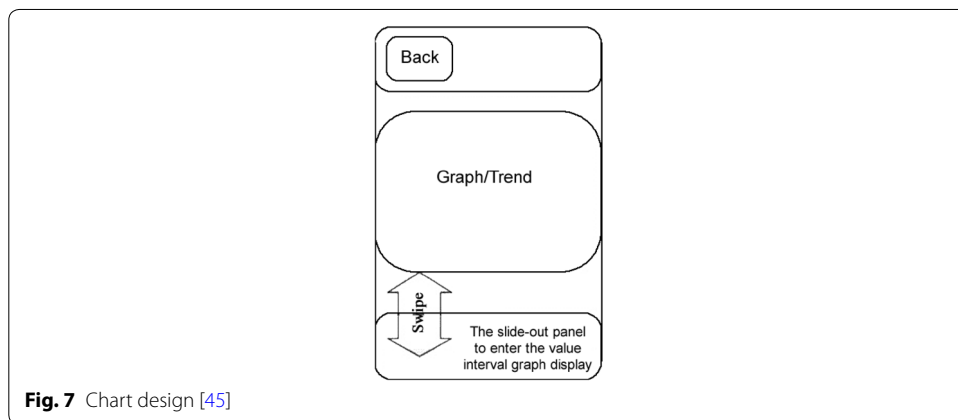
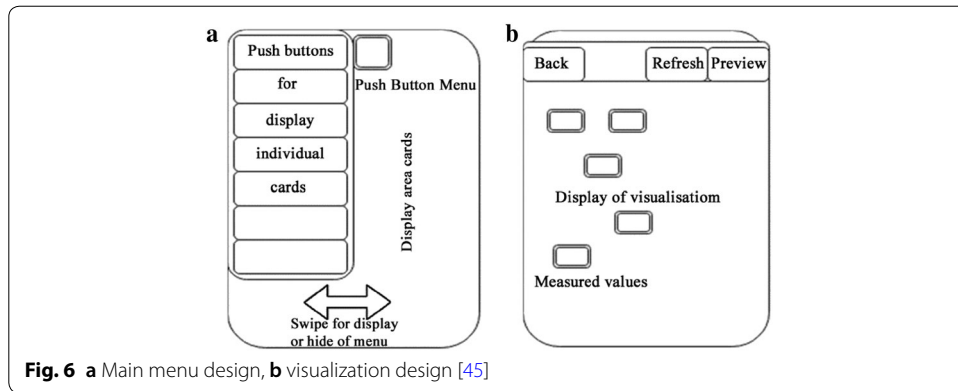
One of the most successful so called “wrappers” is the PhoneGap. It is an application with an inserted web code including HTML5, CSS3 and JavaScript, packaged for any operating system. This wrapper is often extended with the option of bridging some functionality that a website alone cannot handle. PhoneGap can bridge some of these. PhoneGap offers downloadable SDK on its website, through which application projects are created. Each project has its own special folder locations for native applications in different devices. Currently, PhoneGap version 4.2.0 supports the creation of applications for the following operating systems: iOS, Android, Windows Phone, Bada, WebOS and BlackBerry.

### **Design and description of the application and server architecture**

#### ***Design of the user interface***

Since the application is primarily used for mobile devices with a touch screen, it is necessary to adapt the controls for this purpose. The Sencha Touch Framework offers controls and user interface components directly adapted for touch control. The interface layout was selected so that each part of the application has its own tab and these tabs are switched using menu buttons that swipe to the right, or using a button in the upper left corner (Fig. 6a). Figure 6b) is the design of the visualization. It contains the view of real data and diagrams of the given room or system.

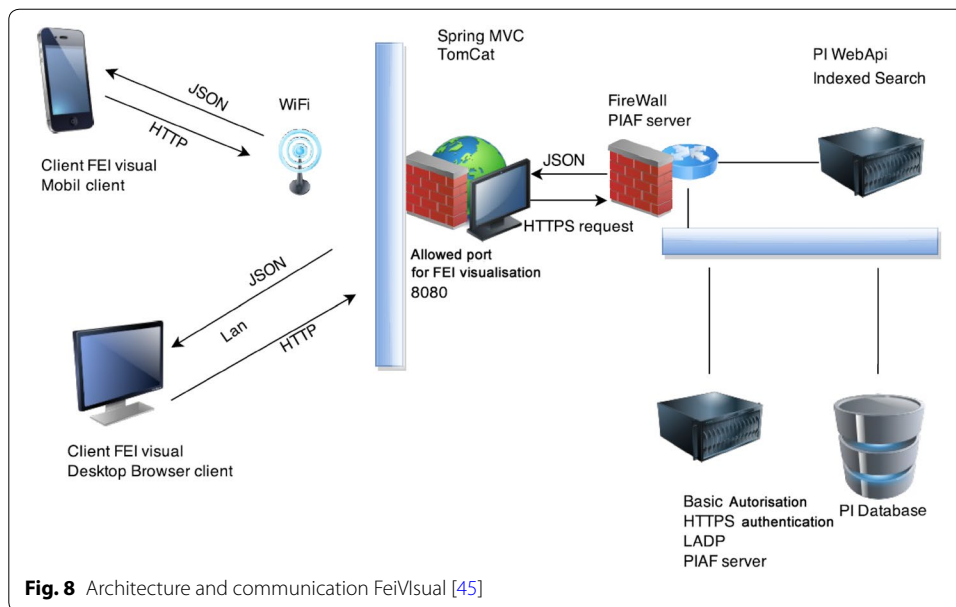
Figure 7 shows the chart design. These designs were discussed with the office building administrator.



**Architecture, application communication, UseCase diagram**

The Sencha Touch architecture is based on the MVC model. Model-View-Controller has its beginnings in the Smalltalk, where it is originally used to incorporate a traditional process: input, processing, output to the UI (user interface) programs. It is equally suitable, however, for use in multi-tier web applications. A description of the architecture and communication is shown in the following Fig. 8.

The server component and the web client of the application are deployed on the application server Apache Tomcat, which is located on a PC within the VSB domain. The applications on mobile devices must be connected via WiFi to the VSB domain to function; without it, they cannot receive any data necessary to create the visualization for the users. There is also the possibility of using the university VPN. A mobile device can also connect to the web client with a web browser after entering the proper URL, but they must also be located in the VSB domain or connected through the university VPN. With the connection to the web client, the initial startup is slow because the entire application must be downloaded into the mobile or computer browser, and it is a major problem if the connection to the VPN is established using mobile data, where one usually tries to be economical. After connecting and starting the application, the HTTP GET request with the relevant parameters is sent to Tomcat that verifies the parameters and connects to the PI Web API using a certificate and HTTPS. As parameters, the PI Web API receives



the relevant attribute (value) web id and another parameter, which is the required interval or what historical data shall be returned, for example. – 8 for historical data up to – 8 h. Web API processes the request and then sends the data as JSON to the client application. The client application receives the data. The model creates objects based on the received data which are then stored in the repository. From there, the client application processes them and calculates the minimum and the maximum as well as data for the waveform graph of various temperatures. Furthermore, they are processed for the visualization of equipment temperatures, such as ventilation, engineering and visualization of the heat pumps.

#### Diagram of major use cases

This UseCase diagram (Fig. 9) shows the major use cases as data acquisition, visualization of the given space and the display of trends of attributes of individual spaces which are:

- VZT 1 auditorium EC3.
- VZT 2 auditorium EC2.
- VZT 3 auditorium EC1.
- VZT 4 depository.
- VZT 5 boardroom.
- Cooling plant.
- Heat pumps.

#### Deployment diagram

The deployment diagram (Fig. 10) shows the deployment of hardware and software resources and their major interprocesses.

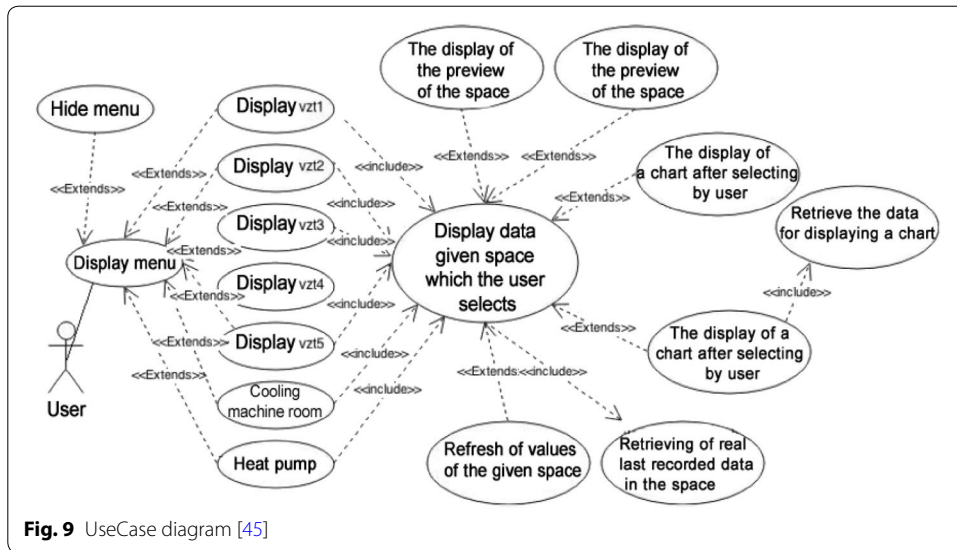


Fig. 9 UseCase diagram [45]

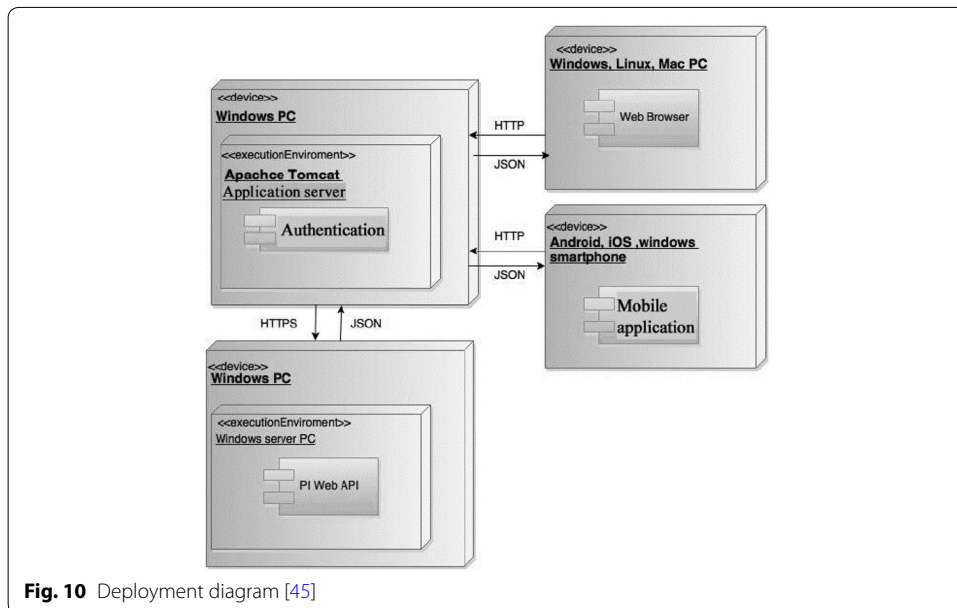


Fig. 10 Deployment diagram [45]

**Description of the development of the individual parts of the application and the server**

The following text will deal with the process of developing the individual parts of the new FeiVisual visualization system. The procedures described are also applicable to other projects.

**Creating a database in the application**

A database of individual attributes has been created for data persistence. The final number of these attributes is 91. It was necessary to create two models; one model that represents what the “store” for descriptions, webid and the variables of the

attributes will look like; the second model represents an object received with JSON. The following is a sample of this model:

```
Ext.define('FeiVisual2.model.value', {
    extend: 'Ext.data.Model',
    config: {
        fields: [
            {name: 'Timestamp', type: 'string'},
            {name: 'Value', type: 'double'},
        ]
    }
});
```

Using these models, 93 “stores”, where the received data is stored in the form of objects, were created and, from these “stores”, this data is further used for processing

```
Ext.define('FeiVisual2.store.GrafStore', {
    extend: 'Ext.data.Store',
    config: {
        model: 'FeiVisual2.model.value',
        autoLoad: true,
        sorters: [{
            property: 'Timestamp',
            direction: "ASC"
        }],
        proxy: {
            type: 'ajax',
            URL: 'API/hel?param=P0BLI0-dK5cUmei3ANsYU-zA2wQAAAUElBRlxDSExBWkVOSV9CVDA2XzA2XzQz&data=value&time=notime',
            reader: {
                type: 'json',
                rootProperty:
                function(data) {if (data.Items!=null) {data.Items.pop();return data.Items;}else {return data;}}
            }
        },
    }
});
```

and visualization. An example of a selected “store” is in the following code:

## View and controllers

### View

In this part of the application, there are entire UI applications. The files have a js extension, i.e. javascript, since the entire UI is written in JavaScript using Sencha Touch. There are individual visualized systems, but also the application menu that bears the name Main.js. There are also thumbnails to preview the measured system in your application. These files comprise implementation of the zoom function.

The temperature.js file contains a UI section where the charts for tracking the progress of the values measured are created. Next, the maximums and minimums for a specific interval are also implemented. The files of the individual visualized systems are the most important ones:

- vzt1.js is the air-conditioning of EC3 lecture hall,
- vzt2.js is the air-conditioning of EC2 lecture hall,
- vzt3.js is the air-conditioning of EC1 lecture hall,1,
- vzt\_board.js is the air-conditioning of the office,
- vzt\_depo.js is the air conditioning of the depository,
- t\_c.js is a room with the heat pumps,
- strojovna.js is the engine room for cooling.

In these seven UIs, there are individual measured values which are, as a result, logically arranged into usable and transparent visualizations of these systems.

A sample of these UIs can be seen in Figs. 11, 12 and 13. The views are divided by room types.

Figure 11 shows the examples of heat pump system applications.

Figure 12 shows the examples of applications displaying the cooling plant schematics in the IAB (Fig. 12a) and the graph of the measurement of the selected variable (temperature) within the selected interval (Fig. 12b).

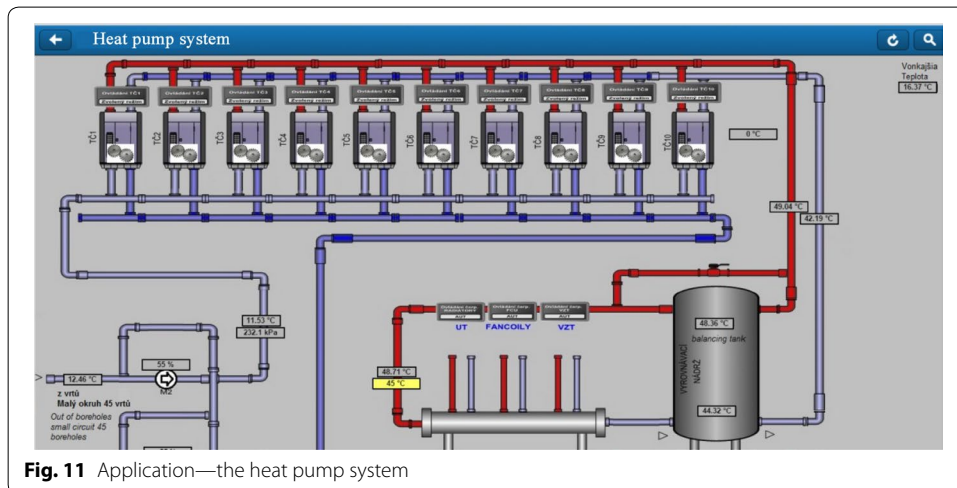
**Controllers**

These are basically the view drivers, so for each view there is a Controller with the same designation as view but is located in the Controllers folder. Controllers control the UI elements and perform the required actions. For example, when pressing the menu button in the UI or swiping from the left to the right, a drop-down menu is displayed. If necessary, the calculations required for displaying the minimum and maximum charts, etc., are performed.

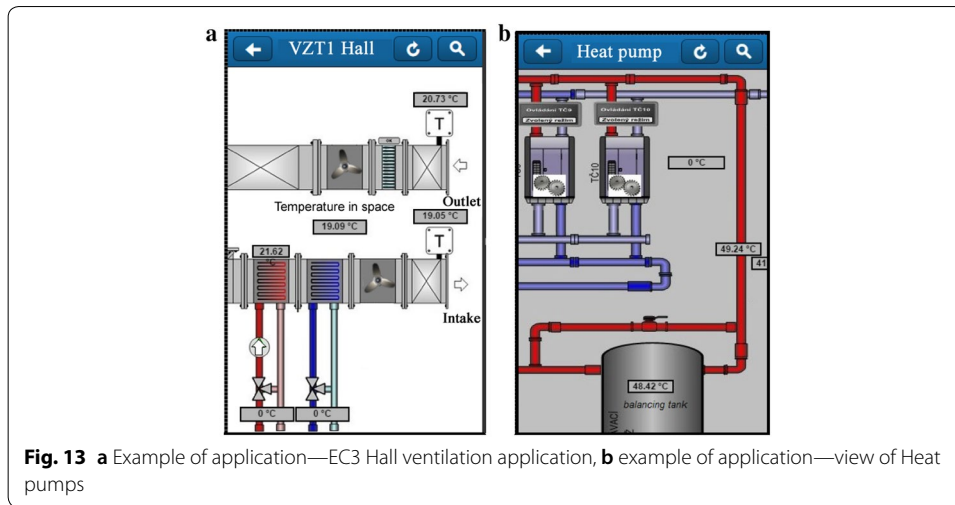
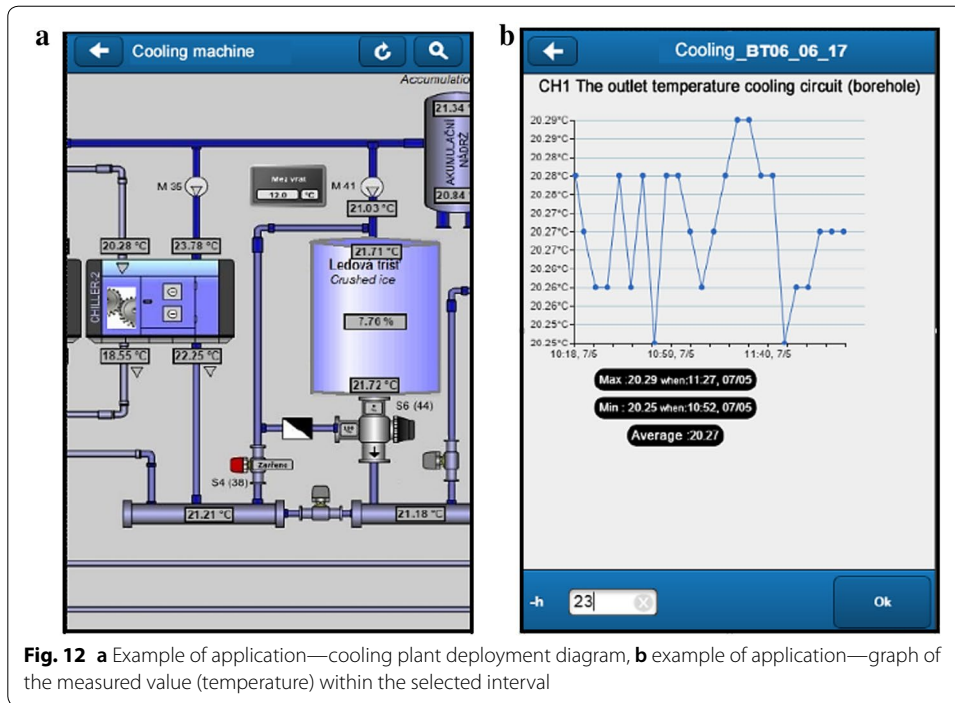
Figure 13 shows the examples of application—(a) EC3 hall ventilation application and (b) view of heat pumps.

**File app.js**

This file is one of the most important ones in the application. It contains initializations of individual views, controllers, stores, models, and also initializations of third-party packages, which, in this case, chart. Next, there is also the actual start of the application,



**Fig. 11** Application—the heat pump system



which shows what is to be run on the start screen or immediately after the application starts.

**The server part of the application**

The server part of the application is created by the application server that performs authorization. Subsequently, the mobile application sends a request for the data parameters to the application server (AS). The AS processes this request and passes this request with the appropriate parameters to the PI Web API, which, however, requires basic authorization along with the need to show the HTTPS secure communication



certificate. In the future, the server can also be used for accessing data, or for more complicated calculations with which a mobile phone might have problems.

### **Implementation and placing on the server**

#### ***Node JS***

Application development is performed mainly on the command line and in your favourite code management editor. PhoneGap and Sencha Touch are dependent on Node.js, which needs to be installed.

#### ***Sencha Touch CMD***

Using this application, a basic Sencha Touch 2.4.2 application will be created, in which you can choose the application's own name, in this case FeiVisual2. Once the application has been created, it can be further developed and modified, whether in a conventional notepad or in another editor; in this case, Eclipse has been selected. The version used for this application is Sencha CMD 5.1.2.52. This is a tool that not only creates the basic application framework, but also inserts additional packages along with the framework itself.

#### ***Sencha Touch CMD and PhoneGap***

Sencha CMD is a very useful tool that can integrate the PhoneGap wrapper directly into the application developed.

#### ***Build and installation of FeiVisual in a mobile device***

After finishing the development, completing the previous and initializing steps, you can switch to the actual application using Sencha Touch CMD.

#### ***FeiVisual application test***

The application was tested mainly during the development phase, after consultation with the caretaker. During the testing stage, emphasis was placed on the relevance of the data and its proper display in the visualizations. Testing was performed on a mobile device running Android version 4.4.2. The individual UI components and their proper functionality in individual applications were tested.

#### ***Authorization description for PI Web API***

The PI Web API can be connected via secure HTTPS communication, but it requires a server certificate where the API is implemented. Before exporting the certificate, the IIS administrator must be running on the server where the API is located in order to enable export of the required certificate.

#### ***Export and placing on the server***

After completing the above-stated procedure, it was possible to place the application and the server part in the VSB PC. To do this it, was necessary to export the application and the server part. The export of the server part was implemented using the Eclipse development environment. In order to place the server part to the Apache

Tomcat application server, the option 'export to a war file' was selected because this file type supports the aforementioned application server.

#### ***The test of the server part***

As with the application testing, the correctness of the data as well as the proper authentication were verified. The web application has been created as an add-on and is not fully customized for viewing on a large computer screen, but, despite this fact, the visualization is transparent and works reliably. The server handled a larger number of connected applications using a mobile application or a browser on a computer.

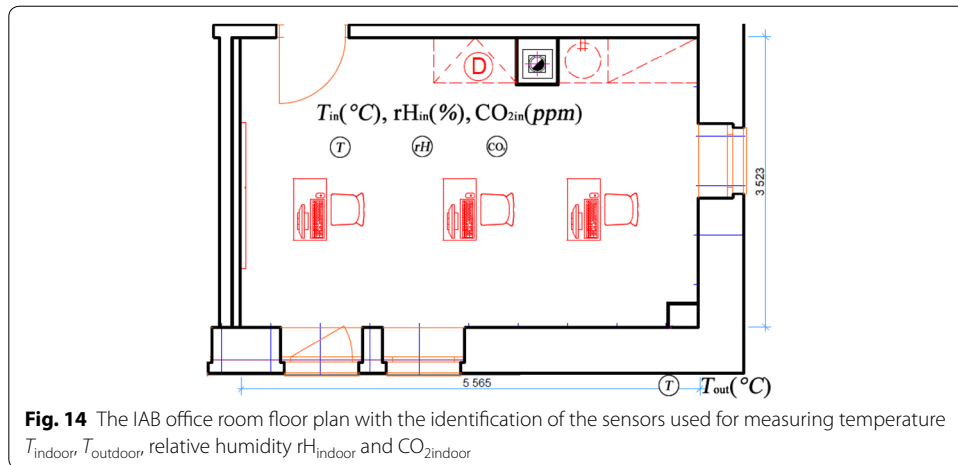
#### **Part II—designing a method for monitoring the presence of people in an IAB office room using prediction of the course of CO<sub>2</sub> concentration**

The designed application for monitoring technological processes in the IAB can be used for obtaining information about the current state of the measured variables from common operating sensors (temperature, rH, CO<sub>2</sub>, light intensity  $E$ ) for further practical applications in the context of operational and technical functions in the IAB. As an example, the article presents a design of the method for obtaining information on the quality of the indoor environment of the IAB office rooms of the building and, subsequently, information on the occupancy of the IAB office rooms of the building. The basic information on the quality of the indoor environment in the IAB is provided by the sensors of temperature (indoors)  $T_i$  (°C) and air humidity rH (%). However, it is much more important to know the concentration of carbon dioxide CO<sub>2</sub> (ppm). This is important for the comfort and the overall habitability of the individual IAB offices, or for HVAC management.

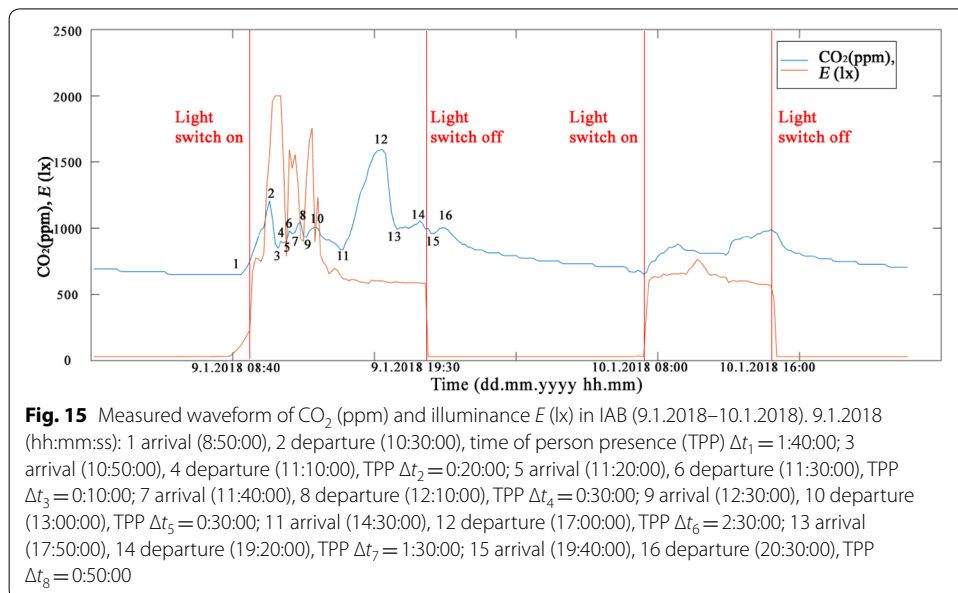
The price of high-quality temperature and air humidity sensors is in the order of tens of Euros. The price of sensors for measuring CO<sub>2</sub> concentration moves the price of a sensor to the order of several tens of Euros.

In the Czech Republic, the emphasis is placed on securing energy savings of buildings by thermal insulation of the facade within the requirements of the legislation approved in the European Union, which is implemented in large administrative buildings, schools, hospitals, residential complexes, family homes and panel houses. However, the indoor environment of the reconstructed insulated buildings is deteriorated by the increase in the concentration of CO<sub>2</sub> and humidity rH. Nonetheless, many investors in the Czech Republic do not take this into account. To reduce CO<sub>2</sub> concentration in a room, it is possible to simply open the window and the door or to use forced ventilation as part of a comprehensive solution provided by the technology (HVAC). In case of using the HVAC technology, it is necessary to measure CO<sub>2</sub> concentration. In order to reduce the investment costs concerning the acquisition of the missing sensors for monitoring CO<sub>2</sub> concentration in the IAB or large residential complexes, it is possible to use the method of predicting CO<sub>2</sub> from the measured courses of the temperature  $T$  (°C) (indoors and outdoors) and air humidity rH (%) (indoors), which is often standard IAB equipment.

As an example of possible electric energy savings, depending on the presence of people in the monitored area, a typical day of an IAB office room having an area of 19.2 m<sup>2</sup> is stated here (Fig. 14).



**Fig. 14** The IAB office room floor plan with the identification of the sensors used for measuring temperature  $T_{indoor}$ ,  $T_{outdoor}$ , relative humidity  $rH_{indoor}$  and  $CO_{2indoor}$



**Fig. 15** Measured waveform of CO<sub>2</sub> (ppm) and illuminance  $E$  (lx) in IAB (9.1.2018–10.1.2018). 9.1.2018 (hh:mm:ss): 1 arrival (8:50:00), 2 departure (10:30:00), time of person presence (TPP)  $\Delta t_1 = 1:40:00$ ; 3 arrival (10:50:00), 4 departure (11:10:00), TPP  $\Delta t_2 = 0:20:00$ ; 5 arrival (11:20:00), 6 departure (11:30:00), TPP  $\Delta t_3 = 0:10:00$ ; 7 arrival (11:40:00), 8 departure (12:10:00), TPP  $\Delta t_4 = 0:30:00$ ; 9 arrival (12:30:00), 10 departure (13:00:00), TPP  $\Delta t_5 = 0:30:00$ ; 11 arrival (14:30:00), 12 departure (17:00:00), TPP  $\Delta t_6 = 2:30:00$ ; 13 arrival (17:50:00), 14 departure (19:20:00), TPP  $\Delta t_7 = 1:30:00$ ; 15 arrival (19:40:00), 16 departure (20:30:00), TPP  $\Delta t_8 = 0:50:00$

After entering the office, the employee usually turns on the light (Fig. 15) and quite often leaves the light on for the entire working time in the room. In the meantime, they leave the room several times and the light is inefficiently on (Fig. 15). In order to obtain an overview of room occupancy (arrival time, departure time) in the monitored IBA room, indirect CO<sub>2</sub> concentration (ppm) measurement, by means of the QPA 2062 operating sensor [range 0 and 2000 (ppm), accuracy  $\pm 50$  (ppm)], is used in this article (Figs. 4, 20, 21). Production of CO<sub>2</sub> concentration is directly proportional to the physical activity of the people in the IAB monitored room. Carbon dioxide concentration is indicated in ppm (parts per million). Based on the course of the measured values of CO<sub>2</sub> concentration (Fig. 15) in an IAB office room, it is possible to specify the time of arrival and departure or presence and the time spent of the person in the monitored area. It assumes that a person is present in the IAB office room (a CO<sub>2</sub> source), if CO<sub>2</sub>

concentration increases. When the person leaves the monitored area, CO<sub>2</sub> concentration decreases. This can be used for switching the lights in the IAB office room on and off in case of absence of people.

The text further describes the procedure for determining the appropriate method for predicting the course of CO<sub>2</sub> concentration from the measured values taken by the indoor temperature sensor  $T_i$  (°C) (QPA 2062) in an IAB office room (range 0 to 50 °C/−35 to 35 °C, accuracy ±1 K) and relative humidity rH (%) (QPA 2062) (range 0 and 100%, accuracy ±5%), outdoor temperature  $T_o$  (°C) (AP 257/22) (range −30... +80 °C, resolution: 0.1 °C) using a gradient algorithm of backpropagation for adapting the multilayered forward Artificial Neural Network (ANN) using the Bayesian Regulation Method (BRM) (Fig. 16).

For the classification of prediction quality, Correlation Analysis (CA) and the calculated MSE (mean squared error) parameters were used.

**Description of the technologies used**

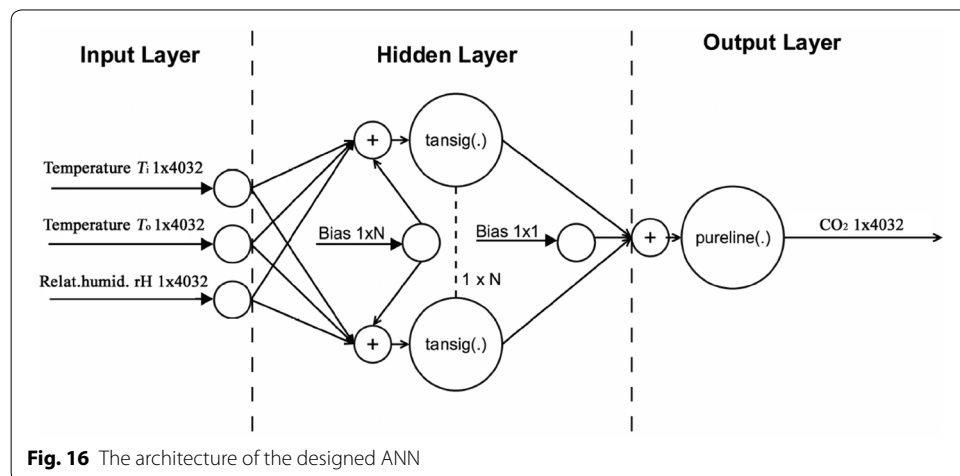
The visualization, monitoring, and processing of the measured values of non-electric variables, such as measurement of temperature, humidity, CO<sub>2</sub> for monitoring the quality of the indoor environment of the selected room in the building described, are implemented using the PI System software application. Figure 17 shows the process of measuring and processing data in connection with the use of the PI System SW tool (PI ProcessBook) in the IAB.

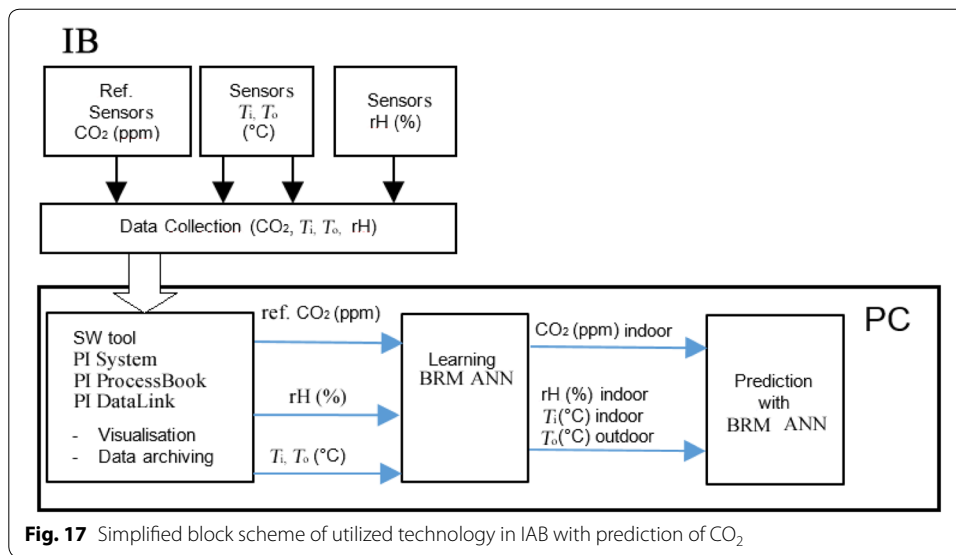
**Implementation of the practical part experiments**

The practical implementation and ANN learning process together with the verification of the ANN BRM learned on the test data is carried out in the following steps:

Step 1. *Measurement of non-electrical variables  $T_i$ ,  $T_o$ , rH a CO<sub>2</sub> in the office IAB room from February 1 to February 28, 2015 (transition from winter to spring), from June 1 to June 28, 2015 (transition from spring to summer) using operating sensors in a selected IAB room.*

Step 2. *Data preprocessing–normalization with the min–max method.*





**Table 1 Comparison of learning quality ANN (BRM), [1.6.2015–28.6.2015 (data normalized)]**

Number of neurons (–)	Time t (hh:mm:ss)	MSE training (ppm)	R training (–)
10	0:00:11	8.39·10 <sup>-3</sup>	0.629
50	0:02:20	3.495·10 <sup>-3</sup>	0.872
100	0:05:44	3.287·10 <sup>-3</sup>	0.869
150	0:10:22	2.194·10 <sup>-3</sup>	0.914
200	0:15:45	2.151·10 <sup>-3</sup>	0.916
250	0:23:12	1.938·10 <sup>-3</sup>	0.927
300	0:30:37	1.279·10 <sup>-3</sup>	0.953
350	0:45:44	1.546·10 <sup>-3</sup>	0.943
400	0:54:49	1.472·10 <sup>-3</sup>	0.945
450	1:15:44	1.436·10 <sup>-3</sup>	0.945
500	1:29:56	1.303·10 <sup>-3</sup>	0.954
550	1:53:55	1.259·10 <sup>-3</sup>	0.955
600	2:23:11	1.105·10 <sup>-3</sup>	0.959
650	2:46:25	1.148·10 <sup>-3</sup>	0.958
700	3:11:44	1.201·10 <sup>-3</sup>	0.954
750	3:47:46	1.186·10 <sup>-3</sup>	0.954
800	4:47:55	1.025·10 <sup>-3</sup>	0.961

Step 3. *Creating an ANN with the BRM structure (Fig. 16).*

Step 4. *Training ANN, BRM for neuron counts in an interval of 10-800, measuring the time t(s) of learning, the MSE calculation, and the correlation coefficient R for data:*

- a. from June 1 to June 28, 2015 (Fig. 20, Table 1). The best results were recorded for ANN BRM with 600 neurons, MSE = 1.105·10<sup>-3</sup> (ppm) and R = 0.959 (Table 1)

- b. from February 1 to February 28, 2015 (Fig. 21, Table 5). The best results were recorded for ANN BRM with 600 neurons,  $MSE = 9.239 \cdot 10^{-4}$  (ppm) and  $R = 0.932$  (Table 5)
- c. from February 8 to February 14, 2015 (Table 9). The best results were recorded for ANN BRM with 600 neurons,  $MSE = 7.726 \cdot 10^{-4}$  (ppm) and  $R = 0.987$  (Table 9).

Step 5. *Prediction CO<sub>2</sub> of the test data generalization—the MSE calculation, and the correlation coefficient R for:*

- a. ANN (BRM) (from June 1 to June 28, 2015) with the tested data (from June 1 to June 28, 2015) (Table 2). The best results were recorded with the predicted course of CO<sub>2</sub> for ANN BRM with 500 neurons,  $MSE = 1.640 \cdot 10^{-3}$  (ppm) and  $R = 0.94$  (Table 2)
- b. ANN (BRM) (from February 1 to February 28, 2015) with the tested data (from February 1 to February 28, 2015) (Table 6). The best results were recorded with the predicted course of CO<sub>2</sub> for ANN BRM with 600 neurons,  $MSE = 1.017 \cdot 10^{-3}$  (ppm) and  $R = 0.93$  (Table 6)
- c. ANN (BRM) (from February 8 to February 14, 2015) with the tested data (from February 8 to February 14, 2015) (Table 10). The best results were recorded with the predicted course of CO<sub>2</sub> for ANN BRM with 400 neurons,  $MSE = 2.021 \cdot 10^{-3}$  (ppm) and  $R = 0.97$  (Table 10).

Step 6. *Cross-validation data prediction CO<sub>2</sub> of the results achieved the MSE calculation, and the correlation coefficient R for:*

**Table 2 Comparison of prediction quality ANN (BRM) [1.6.2015–28.6.2015 (data normalized)], with the tested data from the interval [1.6.2015–28.6.2015 (data normalized)]**

Number of neurons (–)	MSE training (ppm)	R testing (–)
10	$8.691 \cdot 10^{-3}$	0.61
50	$3.516 \cdot 10^{-3}$	0.86
100	$2.935 \cdot 10^{-3}$	0.89
150	$1.989 \cdot 10^{-3}$	0.93
200	$2.467 \cdot 10^{-3}$	0.91
250	$2.072 \cdot 10^{-3}$	0.92
300	$4.300 \cdot 10^{-3}$	0.85
350	$1.894 \cdot 10^{-3}$	0.93
400	$2.692 \cdot 10^{-3}$	0.90
450	$1.693 \cdot 10^{-3}$	0.94
500	$1.640 \cdot 10^{-3}$	0.94
550	$1.893 \cdot 10^{-3}$	0.93
600	$3.178 \cdot 10^{-3}$	0.89
650	$2.181 \cdot 10^{-3}$	0.92
700	$3.397 \cdot 10^{-3}$	0.88
750	$1.990 \cdot 10^{-3}$	0.93
800	$1.798 \cdot 10^{-3}$	0.93

**Table 3 Comparison of prediction quality ANN (BRM) [1.6.2015–28.6.2015 (data normalized)], with the tested data from the interval [1.2.2015–28.2.2015 (data normalized)] (cross validation)**

Number of neurons (–)	MSE training (ppm)	R testing (–)
10	$5.168 \cdot 10^{-3}$	0.52
50	$2.002 \cdot 10^{-3}$	0.85
100	$1.590 \cdot 10^{-3}$	0.88
150	$1.515 \cdot 10^{-3}$	0.89
200	$1.418 \cdot 10^{-3}$	0.89
250	$1.552 \cdot 10^{-3}$	0.88
300	$2.297 \cdot 10^{-3}$	0.82
350	$1.532 \cdot 10^{-3}$	0.88
400	$1.414 \cdot 10^{-3}$	0.89
450	$1.469 \cdot 10^{-3}$	0.89
500	$2.870 \cdot 10^{-3}$	0.77
550	$1.776 \cdot 10^{-3}$	0.86
600	$2.098 \cdot 10^{-3}$	0.84
650	$2.592 \cdot 10^{-3}$	0.80
700	$2.726 \cdot 10^{-3}$	0.78
750	$2.780 \cdot 10^{-3}$	0.76
800	$2.451 \cdot 10^{-3}$	0.81

- a. ANN (BRM) (from June 1 to June 28, 2015) with the tested data (from February 1 to February 28, 2015) (Table 3). The best results were recorded with the cross-validation data prediction CO<sub>2</sub> for ANN BRM with 400 neurons, MSE =  $1.414 \cdot 10^{-3}$  (ppm) and R = 0.89 (Table 3)
- b. ANN (BRM) (from February 1 to February 28, 2015) with the tested data (from June 1 to June 28, 2015) (Table 7). The best results were recorded with the cross-validation data prediction CO<sub>2</sub> for ANN BRM with 250 neurons, MSE =  $2.042 \cdot 10^{-3}$  (ppm) and R = 0.92 (Table 7)
- c. ANN (BRM) (from February 8 to February 14, 2015) with the tested data (from Jun 8 to Jun 14, 2015) (Table 11). The best results were recorded with the predicted course of CO<sub>2</sub> for ANN BRM with 400 neurons, MSE =  $1.414 \cdot 10^{-3}$  (ppm) and R = 0.89 (Table 11).

Step 7. *Graphic view of the cross-validation data prediction CO<sub>2</sub> results for:*

- a. ANN (BRM) with 400 neurons (Table 3), Step 6a (from June 1 to June 28, 2015) with the tested data (from February 1 to February 28, 2015) (Fig. 22)
- b. ANN (BRM) with 250 neurons (Table 7), Step 6b (from February 1 to February 28, 2015) with the tested data (from June 1 to June 28, 2015) (Fig. 24)
- c. ANN (BRM) with 400 neurons (Table 11), Step 6c (from February 8 to February 14, 2015) with the tested data (from June 8 to June 14, 2015) (Fig. 26).

Step 8. *The use of the LMS adaptive filter algorithm for noise elimination in the predicted signal using the ANN BRM*

- a. ANN (BRM) with 400 neurons (Table 3), Step 6a (from June 1 to June 28, 2015) with the tested data (from February 1 to February 28, 2015) (Fig. 23)



- b. ANN (BRM) with 250 neurons (Table 7), Step 6b (from February 1 to February 28, 2015) with the tested data (from June 1 to June 28, 2015) (Fig. 25)
- c. ANN (BRM) with 400 neurons (Table 11), Step 6c (from February 8 to February 14, 2015) with the tested data (from June 8 to June 14, 2015) (Fig. 27).

Step 9. *Comparison* of the prediction results of the ANN BRM without filtering and with LMS filtering using the DTW criteria.

Step 10. *Evaluation* of the results achieved.

**A training and test set of data**

The goal of the *test* is to verify whether the results achieved on the training data can be used in the future for other input data files. The training set is made directly from the measured values of temperature  $T_i$  (°C), relative humidity rH (%) in the selected IAB office room (Fig. 14) and outdoor temperature  $T_{out}$  (°C). The output (target) set of data consists of the measured CO<sub>2</sub> values (ppm). The values measured in February 2015 and in June 2015 were selected.

The data is sampled in an interval of 10 min. Before the actual learning, the training set of data was normalized using the min–max method.

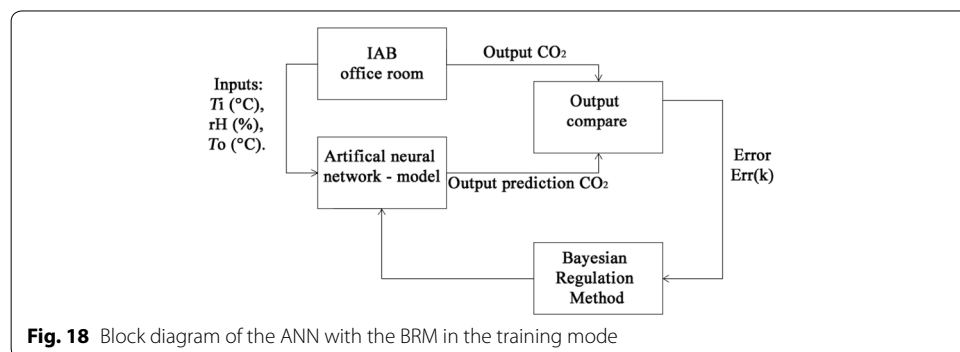
$$x'_i = \frac{x_i - \min(x_1 \dots x_i)}{\max(x_1 \dots x_i) - \min(x_1 \dots x_i)}, \tag{1}$$

where  $\min(x_1 \dots x_i)$  is the smallest value of the set  $(x_1 \dots x_i)$  and  $\max(x_1 \dots x_i)$  is the biggest value of the set  $(x_1 \dots x_i)$ .

The min–max method was used for the reasons of eliminating potential errors in training the ANN BRM. After training the neural network and at simulating the training or test data, it was necessary to denormalize the output from the neural network according to the original range of the training input values. The ANN training process is shown in Fig. 18.

**Selection of the diagnostic tools, their application and evaluation**

The quality of the diagnostic tools that are presented by artificial intelligence methods can be evaluated using various methods, such as mean square error (MSE), mean absolute percent error (MAPE) or Root Mean Squared Error (RMSE). In this article, the



**Fig. 18** Block diagram of the ANN with the BRM in the training mode

quality of performing the prediction using the BRM ANN is assessed using the mean square error (MSE), which is given by Eq. (2):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^*)^2, \quad (2)$$

where  $n$  is the number of measured values,  $y_i$  represents the measured data (response) of the real system and  $y_i^*$  is the model response.

The force of (linear) dependence between two measured non-electrical quantities has been evaluated by means of the value of the Pearson's correlation coefficient

$$\mathbf{R}_{x,y} = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}} \quad (3)$$

The correlation coefficient  $\mathbf{R}_{x,y}$  can take values from the closed interval  $\langle -1, +1 \rangle$ . The more the absolute value of the correlation coefficient approaches 1, the stronger the dependence of the random quantities is.

### Bayesian regularization backpropagation algorithm

#### *Backpropagation algorithm*

The backpropagation algorithm is a method for minimizing the average square error value between the actual output value and the pattern that is input into the neural network in order to give a desired target against each unit of the output layer when a neural network is learned [46]. In this article, the Bayesian regularization backpropagation algorithm [47] was chosen for its accuracy and high quality. The backpropagation algorithm is a gradient algorithm by means of which multilayer forward networks are adapted. Backpropagation is conducted as learning supervised by a teacher where the calculated network output is compared to the desired output. Subsequently, using the backpropagation of the signal, the scales are adjusted, so that the network responded to the pattern with the desired output. It has been proved that such networks are able to approximate any continuous function with the required accuracy, and therefore are broadly applicable, for example, for regression analysis [48]. In general, the backpropagation principle is the most widely used algorithm in the field of neural networks. It is used in about 80–90% of applications [49]. Backpropagation optimization is used for forward neural networks with hidden layers and is designed for data classification that is generally not linearly separable [50]. The learning process of backpropagation can be divided into four main parts [15]:

1. Initialization.
2. Pattern introduction.
3. Comparison.
4. Error backpropagation and weight modification (in hidden layers)

**Bayesian regularization backpropagation algorithm**

The Bayesian evidence framework proposed by MacKay (1992) provides a unified theoretical treatment of learning in neural networks. In the Bayesian evidence framework, the weights and biases of the network are assumed to be random variables with specified distributions [51]. BRM approach works similar to the Levenberg–Marquardt optimization, in a sense that it minimises the squared errors and the weights and finds the optimal combination so that the network performs well. Bayesian based NN training is more robust than the standard back propagation nets [52]. When the parameters (weights and biases) increase, the network loses its ability to generalize. The error on the training set is driven to a very small value, but when new data is presented to the network, the error is great. The predictor has memorized the training examples, but it has not learned to generalize to new situations; the network overfits the data [53]. Typically, training aims to reduce the sum of squared errors in Eq. 4:

$$F(\omega) = \alpha E_w + \beta E_D, \tag{4}$$

where  $w$  is the sum of the squared network weights and  $w$  is the sum of network errors. Both  $\alpha$  and  $\beta$  are the objective function parameters. In the BRM framework, the weights  $w$  of the network are viewed as random variables, and then the distribution of the network weights and training set are considered as Gaussian distribution [54]. The advantage of the BRM ANN is the fact that classical data division into training, validation and testing is not necessary. As the Bayes’ formula indicates, the data and the model will correct each other. The task is to evaluate the models—thus obtaining the control parameters (the scale, or possibly the bias, the vector). Subsequently, the models are compared to each other, which results in optimal hyperparametric gains. The inference task is specified within the model using the Bayes’ formula (5) [55]:

$$P(w|D, \alpha, \beta, M) = \frac{P(D|w, \beta, M) \cdot P(w|\alpha, M)}{P(D|\alpha, \beta, M)} \tag{5}$$

In Eq. (5),  $D$  is the training sample and the prior distribution of weights is defined as  $P(w|\alpha, M) = (\alpha 2\pi)^m / 2 \exp\{-\alpha 2w'w\}$ .  $M$  is the particular ANN used and  $w$  is the vector of networks weights.  $P(w|\alpha, M)$  states our knowledge of weights before any data is collected,  $P(D|w, \beta, M)$  is the likelihood function which is the probability of the occurrence, giving the network weights [56].

The network was built by using MATLAB’s trainbr function. The logistic sigmoid functions are used for the activation function in each neuron and a linear transfer function, which is used to calculate the network output [57].

**LMS algorithm of the adaptive filter**

The stability and convergence of the LMS algorithm (Fig. 19) are ensured by having the optimum settings of a step size parameter  $\mu$  and the length  $M$  of the adaptive filter with the LMS algorithm. The least mean-square (LMS) algorithm was developed by Widrow and Hoff in 1960. This algorithm is a member of the stochastic gradient algorithms [58]. The LMS algorithm is a linear adaptive filtering algorithm, which, in general, consists of two basic processes:

a. *Filtering process*, which involves, computing the output  $y(n)$  of the adaptive filter in response to the vector input signal  $\mathbf{x}(n)$  (6),

Generating an estimation error  $e(n)$  (Fig. 19) by comparing this output  $y(n)$  with the desired response  $d(n)$  (7),

b. *An adaptive process* (8), which involves the automatic adjustment of the parameters  $\mathbf{w}(n+1)$  of the filter in accordance with the estimation error  $e(n)$  [59], [60]

$$y(n) = \mathbf{w}^T(n) \mathbf{x}(n), \tag{6}$$

where  $\mathbf{w}(n)$  is tap—weight vector.

$$e(n) = d(n) - y(n), \tag{7}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n), \tag{8}$$

where  $\mathbf{w}(n+1)$  is tap—weight vector update,  $\mu$  is step size parameter.

**Settings of a step size parameter  $\mu$**

a. Calculating of a step size parameter  $\mu$  to ensure the stability of the adaptive filter with the LMS algorithm.

For determination, when the LMS algorithm remains stable, it is necessary to find the upper bound of  $\mu_{\max}$ , that guarantees stability of the LMS algorithm (9) [58]

$$\mu_{stab} < \frac{1}{3\text{tr}[\mathbf{R}]}, \tag{9}$$

where  $\text{tr}[\mathbf{R}]$  is trace of  $\mathbf{R}$ , which is the mean of the sum of the diagonal elements of  $\mathbf{R}$  (Toeplitz autocorrelation matrix calculated from the vector of input signal  $\mathbf{x}(n)$ , size  $\mathbf{R}$  is  $M \times M$ .)

b. Calculating of a step size parameter  $\mu$  to ensure the convergence of an adaptive filter with the LMS algorithm

Convergence behaviour of the LMS algorithm is directly linked to the eigenvalue spread of the autocorrelation matrix  $\mathbf{R}$  and the power spectrum of  $\mathbf{x}(n)$ . Convergence of the LMS algorithm is directly related to the flatness in the spectral content of the underlying input process.  $E[\mathbf{v}(n)]$  converges to zero when  $\mu$  remains within the range of formula (10).  $E[\mathbf{v}(n)]$  is expectation of weight—error vector  $\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}_0$ .

$$\mu_{conv} \leq \frac{1}{\lambda_{\max}}, \tag{10}$$

where  $\lambda_{\max}$  is maximum eigenvalue of the autocorrelation matrix  $\mathbf{R}$  of the input vector  $\mathbf{x}(n)$ .

c. Calculating of the optimal value of a step size parameter  $\mu_{\text{opt}}$  of an adaptive filter with the LMS algorithm

As a result, a small value  $\mu$  slows down the convergence of the LMS algorithm and, consequently, increases the inaccuracies in the filtration of non-stationary signals [60]. The following equation is used for the optimal value of the parameter  $\mu_{opt}$  (7) [58]

$$\mu_{opt} = \frac{M}{(1 + M) \cdot \text{tr}[\mathbf{R}]}, \tag{11}$$

where  $\text{tr}[\mathbf{R}]$  is trace of  $\mathbf{R}$ , which is the mean sum of the diagonal elements of  $\mathbf{R}$ ,  $M$  is parameter misadjustment.

Parameter misadjustment  $M$  is defined as the ratio of the steady-state value of the excess mean-square error (MSE)  $\xi_{excess}$  to the minimum mean square (MSE) error  $\xi_{min}$ .

$$M = \frac{\xi_{excess}}{\xi_{min}} = \mu \text{tr}[\mathbf{R}]. \tag{12}$$

The misadjustment  $M$  is a dimensionless parameter that provides a measure of how close the LMS algorithm is to optimality in the mean-square-sense. The closer adaptive filter length  $M$  is to 0, the more accurate is the adaptive filtering action being performed by the LMS algorithm.

**DTW criterion for comparing two waveforms**

The Dynamic Time Warping (DTW) criterion is used to compare the two sequences of compared vectors: reference vector  $\mathbf{P} = [p(1), \dots, p(P)]$  of length  $P$  and test vector  $\mathbf{O} = [o(1), \dots, o(T)]$  of length  $T$  [61]. The distance  $d$  between the sequences  $\mathbf{O}$  and  $\mathbf{P}$  is given as minimum distance over the set of all possible paths (all possible lengths, all possible courses) [62]. Minimum distance computation

$$D(\mathbf{O}, \mathbf{P}) = \min_{\{C\}} D_c(\mathbf{O}, \mathbf{P}) \tag{13}$$

is simple, when normalization factor  $N_c$  is not a function of the path and it is possible to write  $N_c = N$  for  $\forall_c$

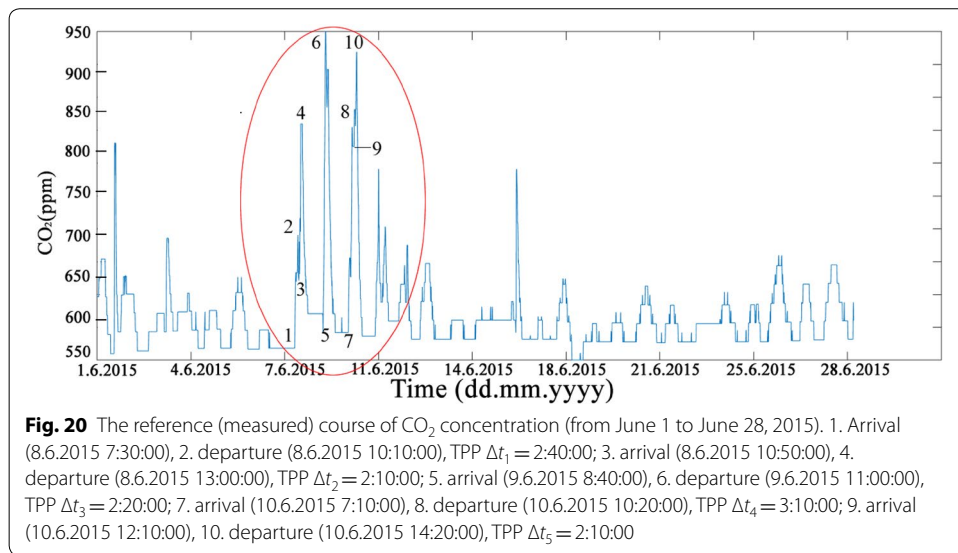
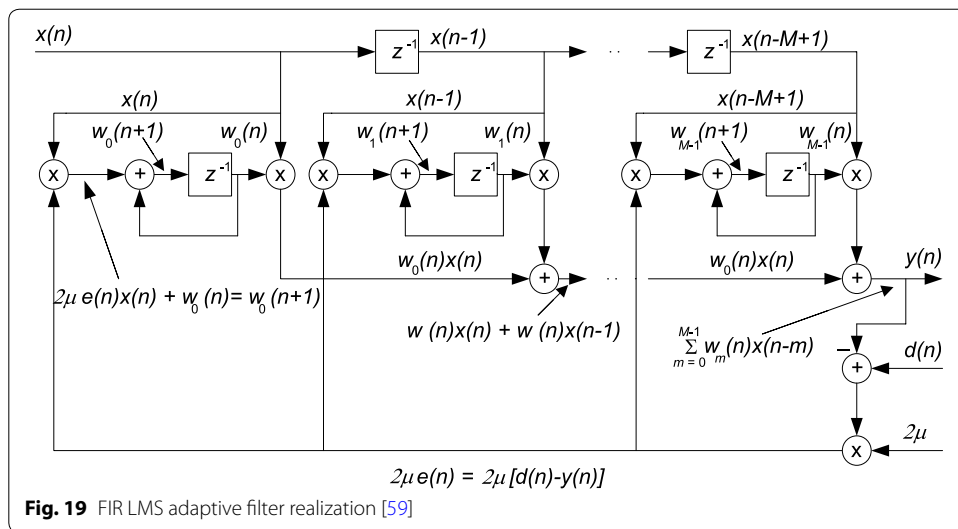
$$D(\mathbf{O}, \mathbf{P}) = \frac{1}{N} \min_{\{C\}} \sum_{k=1}^{Kc} d[\mathbf{o}(t_c(k)), \mathbf{p}(r_c(k))] W_c(k) \tag{14}$$

If the calculated distance  $D(\mathbf{O}, \mathbf{P}) = 0$  is between the two compared waveforms, the comparison curves are identical. If the distance  $D(\mathbf{O}, \mathbf{P}) \neq 0$  is not the same waveforms compared.

**Data preparation for the experiments**

**The experimental practical part**

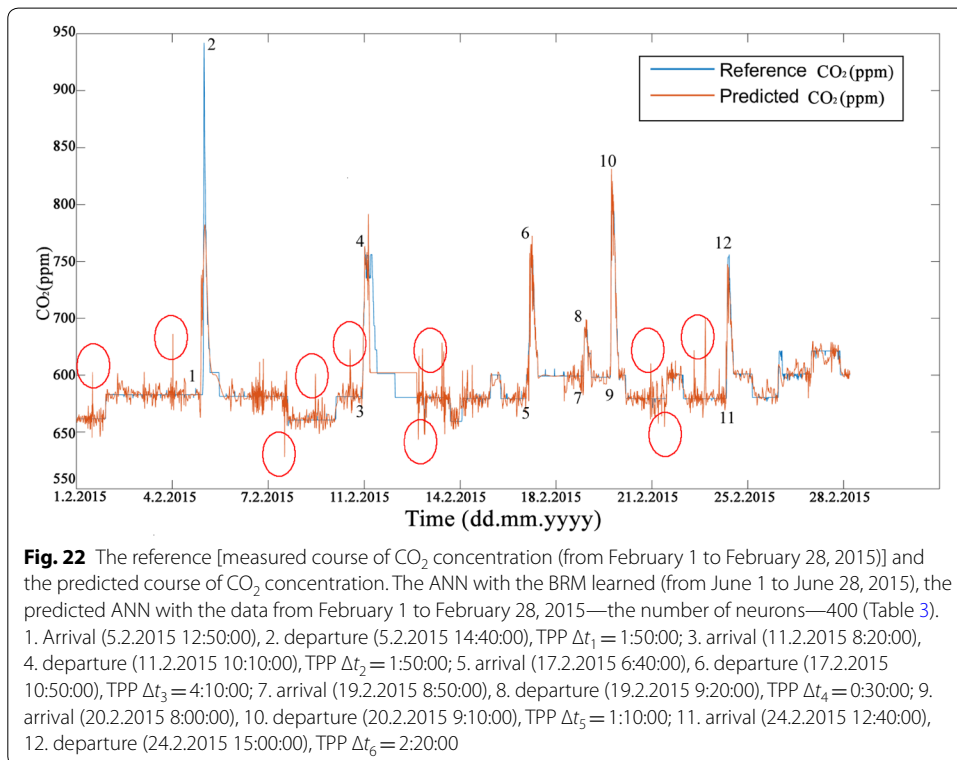
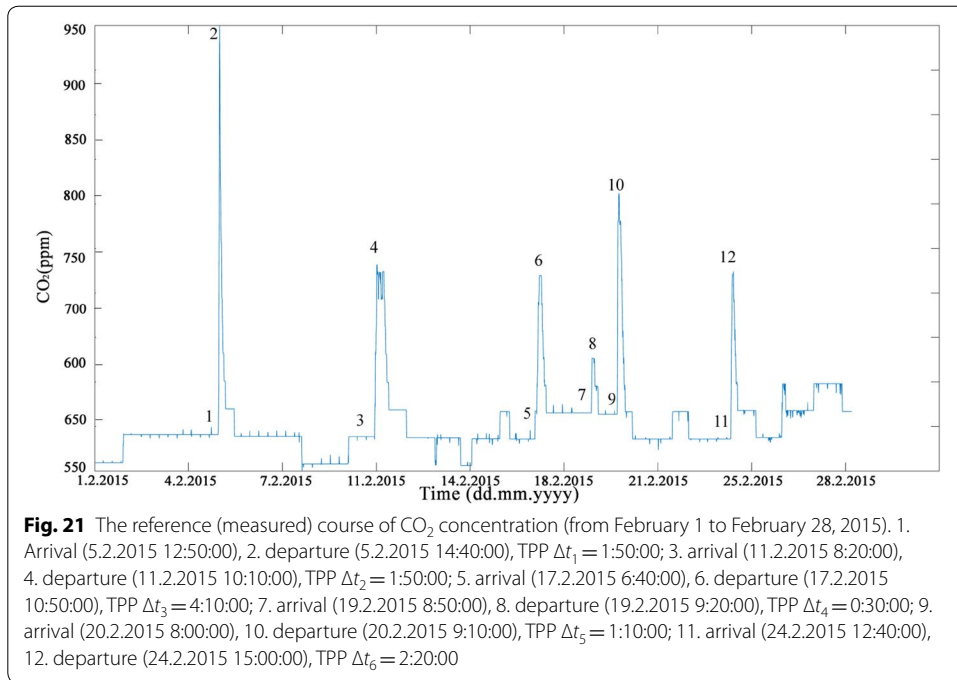
This section contains the description of the experiments performed, the setting of the individual parameters, the calculated and measured results. In the first long-term experiment,  $\text{CO}_2$ , temperature ( $T$ ) and relative humidity (rH) indoor measurement were



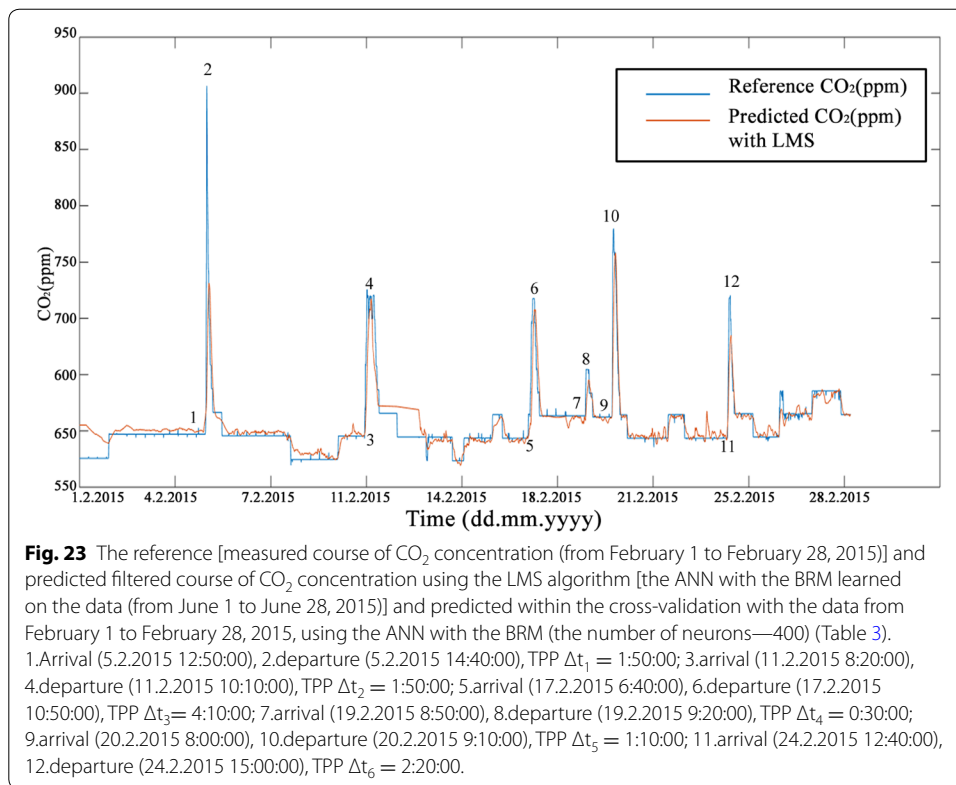
performed in a summer month, from Jun 1 to Jun 28, 2015. The measured course of CO<sub>2</sub> concentration in Fig. 20 indicates the occurrence of people in the monitored area.

In the second long-term experiment, CO<sub>2</sub>, temperature (*T*) and relative humidity (rH) indoor measurement were performed in a winter month, from February 1 to February 28, 2015. Figure 21 shows the arrival and departure of people in the monitored area (see the legend) in the selected time interval.

The design of the experiments is based on the requirement for verifying the quality of the prediction execution for the summer months [selected in a time interval of spring/summer (from June 1 to June 28, 2015)] as well as for the winter months [winter/spring (from February 1 to February 28, 2015)].







Next, it was necessary to experimentally verify the quality of the prediction by means of cross-validation for the values measured in 28 days.

The values measured from June 1 to June 28, 2015, and from February 1 to February 28, 2015, within the individual experiments served for the cross-validation and testing of the trained ANN with the BRM on the data obtained by measuring from June 1 to June 28, 2015, and from February 1 to February 28, 2015.

The actual experiments were carried out as follows:

I. The long-term experiment—28 days. Training the ANN with the BRM was implemented on the data (from June 1 to June 28, 2015) (Table 1).

Training ANN, BRM (Step 4a) for neuron counts in an interval of 10-800, measuring the time  $t(s)$  of learning, the MSE calculation, and the correlation coefficient R for data from June 1 to June 28, 2015 (Fig. 20) (Table 1). The total number of samples is 4032. Training—70%, 2822 samples, which are presented to the network during the training, and the network is adjusted according to its error). Validation—15%, 605 samples, which were used to measure the network generalization, and to halt the training when the generalization stopped improving. Testing—15%, 605 samples, which have no effect on the training and, therefore, they provide an independent measure of the network performance during and after the training. The prediction (Step 5a) of CO<sub>2</sub> concentration using the ANN with the BRM was implemented by, firstly, using the measured values of rH (%) indoors, temperature indoors  $T_{in}$  (°C),

temperature outdoors  $T_{\text{out}}$  ( $^{\circ}\text{C}$ ) on the training data from June 1 to June 28, 2015 (Fig. 16, Table 2). Further, the cross-validation (Step 6a) used the training data [rH (%) indoors, temperature indoors  $T_{\text{in}}$  ( $^{\circ}\text{C}$ ), temperature outdoors  $T_{\text{out}}$  ( $^{\circ}\text{C}$ )] that was measured from February 1 to February 28, 2015 (Table 3). Table 1 shows the calculated and measured parameters for training the ANN BRM (Step 4a) (a number of neurons set, the measured time  $t$  of training the ANN with the BRM and the calculated R and MSE values for the trained ANN with the BRM) on the data measured in the interval from June 1 to June 28, 2015.

The minimum value of  $\text{MSE} = 1.105 \cdot 10^{-3}$  parameter and maximum value of correlation coefficient  $R = 0.959$  were calculated for ANN with the BRM where the configured number of neurons was 600 (Table 1). The learning time of ANN BRM is not directly proportional to the number of configured neurons.

Table 2 shows the calculated values of R and MSE parameters for the number of neurons set within the testing of the ANN with the BRM (from June 1 to June 28, 2015) for predicting (Step 5a) the course of  $\text{CO}_2$  using the ANN with the BRM on the data measured in the interval from June 1 to June 28, 2015. The best calculated minimum MSE parameter ( $\text{MSE} = 1.640 \cdot 10^{-3}$ ) and maximum R correlation coefficient ( $R = 0.94$ ) (Table 2) for predicted courses of  $\text{CO}_2$  are for trained ANN BRM with 500 neurons.

Table 3 shows the calculated values of R and MSE parameters for the number of neurons (10–800) set within the testing of the ANN with the BRM (from June 1 to June 28, 2015) for predicting (Step 6a) the course of  $\text{CO}_2$  using the ANN with the BRM on the data measured in the interval from February 1 to February 28, 2015. The best calculated minimum value of MSE parameter ( $\text{MSE} = 1.414 \cdot 10^{-3}$ ) and maximum value of R correlation coefficient ( $R = 0.89$ ) (Table 3) for predicted courses of  $\text{CO}_2$  (Fig. 22) are for trained ANN BRM with 400 neurons (cross validation).

Figure 22 illustrates the reference measured course of  $\text{CO}_2$  concentration from February 1 to February 28, 2015, and the predicted course of  $\text{CO}_2$  concentration using the ANN with the BRM learned (Step 4a) on the data (from June 1 to June 28, 2015) and predicted within the cross-validation (Step 6a) (with the data from February 1 to February 28, 2015) using the ANN with the BRM (number of neurons 400) (Table 3). The red circles indicate the predicted  $\text{CO}_2$  values that were incorrectly calculated within the prediction error using the ANN with the BRM.

Figure 23 illustrates the reference measured course of  $\text{CO}_2$  concentration from February 1 to February 28, 2015. Figure 23 further illustrates the predicted filtered course of  $\text{CO}_2$  concentration using the LMS algorithm [the ANN with the BRM learned on the data (from June 1 to June 28, 2015) and predicted within the cross-validation (Step 6a) with the data from February 1 to February 28, 2015, using the ANN with the BRM (the number of neurons—400) (Table 3)]. Based on relations (6) to (11), the step size parameter  $\mu$  values (Table 4) for setting the structure of the LMS algorithm adaptive filter ( $M = 48$ ,  $\mu = 0.0062$ ) for noise elimination in the calculated predicted course of  $\text{CO}_2$  (Fig. 22).

Comparison of the prediction results of the ANN BRM without filtering and with LMS filtering using the DTW criteria for the long-term experiment—28 days (from June 1 to June 28, 2015) (Table 1). Using relations (6)–(8), the resulting predicted course of  $\text{CO}_2$  concentration was calculated using the ANN BRM (400 neurons) (Table 3), with

suppression of the additive noise using the LMS algorithm (Fig. 23). For the comparison and classification of the measured course of CO<sub>2</sub> (Fig. 21) and the predicted course of CO<sub>2</sub> without filtering (Fig. 22) using the ANN BRM with cross-validation (400 neurons) (Table 3) was distance  $D$  calculated using relations (13), (14)  $D=0.239$ . For the comparison and classification of the measured course of CO<sub>2</sub> (Fig. 21) and the predicted course of CO<sub>2</sub> using the ANN BRM with cross-validation (400 neurons) (Table 3), using the LMS algorithm for suppression of the additive noise (Fig. 23) distance  $D$  was calculated using relations (13), (14)  $D=0.176$  (M—Misadjustment).

II. The long-term experiment—28 days. Training the ANN with the BRM was implemented on the data (from February 1 to February 28, 2015) (Table 5).

Training ANN, BRM (Step 4b) for neuron counts in an interval of 10–800, measuring the time  $t$ (s) of learning, the MSE calculation, and the correlation coefficient  $R$  for the data from February 1 to February 28, 2015 (Fig. 21, Table 5). The total number of samples is 4032. Training—70%, 2822 samples, which are presented to the network during the training, and the network is adjusted according to its error. Validation—15%, 605 samples, which were used to measure the network generalization, and to halt the training when the generalization stopped improving. Testing—15%, 605 samples, which have no effect on training and, therefore, they provide an independent measure of the network performance during and after the training).

The prediction (Step 5b) of CO<sub>2</sub> concentration using the ANN with the BRM was implemented by, firstly, using the measured values of rH (%) indoors, temperature indoors  $T_{in}$  (°C), temperature outdoors  $T_{out}$  (°C) on the on the training data from February 1 to February 28, 2015 (Fig. 16, Table 6).

Further, the cross-validation (Step 6b) used the training data [rH (%) indoors, temperature indoors  $T_{in}$  (°C), temperature outdoors  $T_{out}$  (°C)] that was measured from June 1 to June 28, 2015 (Table 7).

Table 5 shows the calculated and measured parameters for training the ANN BRM (Step 4b), a number of neurons set, the measured time  $t$  of training the ANN with the BRM and the calculated  $R$  and MSE values for the trained ANN with the BRM) on the data measured in the interval from February 1 to February 28, 2015. The minimum value of MSE parameter is  $MSE=9.239 \cdot 10^{-4}$  and maximum value of correlation coefficient  $R=0.932$  were calculated for ANN with the BRM where the configured number of neurons was 700 (Table 5).

Table 6 shows the calculated values of  $R$  and MSE parameters for the number of neurons set within the testing of the ANN with the BRM (Step 4b) (from February 1 to February 28, 2015) for predicting the course (Step 5b) of CO<sub>2</sub> using the ANN with the BRM on the data measured in the interval from February 1 to February 28, 2015. The best calculated minimum value of MSE parameter ( $MSE=1.017 \cdot 10^{-3}$ ) and maximum value of  $R$  correlation coefficient parameter ( $R=0.93$ ) (Table 6), for predicted courses of CO<sub>2</sub> are for trained ANN BRM with 600 neurons.

Table 7 shows the calculated values of  $R$  and MSE parameters for the number of neurons set within the testing of the ANN with the BRM (from February 1 to February 28, 2015) for predicting the course of CO<sub>2</sub> using the ANN with the BRM on the data

**Table 4 The calculated optimal step size parameter  $\mu$  with securing the stability and convergence of the LMS algorithm**

---

$\mu_{conv} = 0.8614^{(M=10\%)}$ , calculation according to Eq. (9)  
 $\mu_{stab} = 0.0227^{(M=10\%)}$ , calculation according to Eq. (10)  
 $\mu_{opt1} = 0.0062^{(M=10\%)}$ , calculation according to Eq. (11)  
 $\mu_{opt2} = 0.0113^{(M=20\%)}$ , calculation according to Eq. (11)  
 $\mu_{opt3} = 0.0157^{(M=30\%)}$ , calculation according to Eq. (11)

---

*M* Misadjustment

**Table 5 Comparison of learning quality ANN (BRM), [1.2.2015–28.2.2015 (data normalized)]**

Number of neurons (–)	Time t (hh:mm:ss)	MSE training (ppm)	R training (–)
10	0:00:19	$4.735 \cdot 10^{-3}$	0.549
50	0:02:52	$2.258 \cdot 10^{-3}$	0.827
100	0:05:42	$2.319 \cdot 10^{-3}$	0.817
150	0:10:38	$2.443 \cdot 10^{-3}$	0.813
200	0:15:14	$1.466 \cdot 10^{-3}$	0.889
250	0:19:11	$1.457 \cdot 10^{-3}$	0.89
300	0:32:27	$1.447 \cdot 10^{-3}$	0.893
350	0:43:05	$1.041 \cdot 10^{-3}$	0.917
400	0:55:55	$1.256 \cdot 10^{-3}$	0.908
450	1:13:59	$1.059 \cdot 10^{-3}$	0.923
500	1:29:05	$9.580 \cdot 10^{-4}$	0.931
550	1:57:27	$1.185 \cdot 10^{-3}$	0.909
600	2:13:16	$1.006 \cdot 10^{-3}$	0.926
650	2:50:10	$1.110 \cdot 10^{-3}$	0.919
700	3:20:34	$9.239 \cdot 10^{-4}$	0.932
750	3:49:26	$1.074 \cdot 10^{-3}$	0.920
800	4:40:42	$9.378 \cdot 10^{-4}$	0.934

measured in the interval from June 1 to June 28, 2015. The best calculated minimum MSE parameter ( $MSE = 2.042 \cdot 10^{-3}$ ) and maximum R correlation coefficient ( $R = 0.92$ ) (Table 7) for predicted courses of CO<sub>2</sub> (Fig. 24) are for trained ANN BRM with 250 neurons (cross validation).

Figure 24 illustrates the reference measured course of CO<sub>2</sub> concentration from June 1 to June 28, 2015, and the predicted course of CO<sub>2</sub> concentration using the ANN with the BRM learned on the data (from February 1 to February 28, 2015) and predicted within the cross-validation (with the data from June 1 to June 28, 2015) using the ANN with the BRM (number of neurons 250) (Table 7).

Figure 25 illustrates the reference measured course of CO<sub>2</sub> concentration from June 1 to June 28, 2015, and the predicted filtered course of CO<sub>2</sub> concentration using the LMS algorithm (the ANN with the BRM learned on the data from February 1 to February 28, 2015) and predicted within the cross-validation (with the data from June 1 to June 28, 2015) using the ANN with the BRM (number of neurons 250) (Table 7). Based on relations (6) to (11), the step size parameter  $\mu$  values (Table 8) for setting the structure of the LMS adaptive filter algorithm ( $M = 34, \mu = 0.013$ ) for noise elimination in the calculated predicted course of CO<sub>2</sub> (Fig. 25).

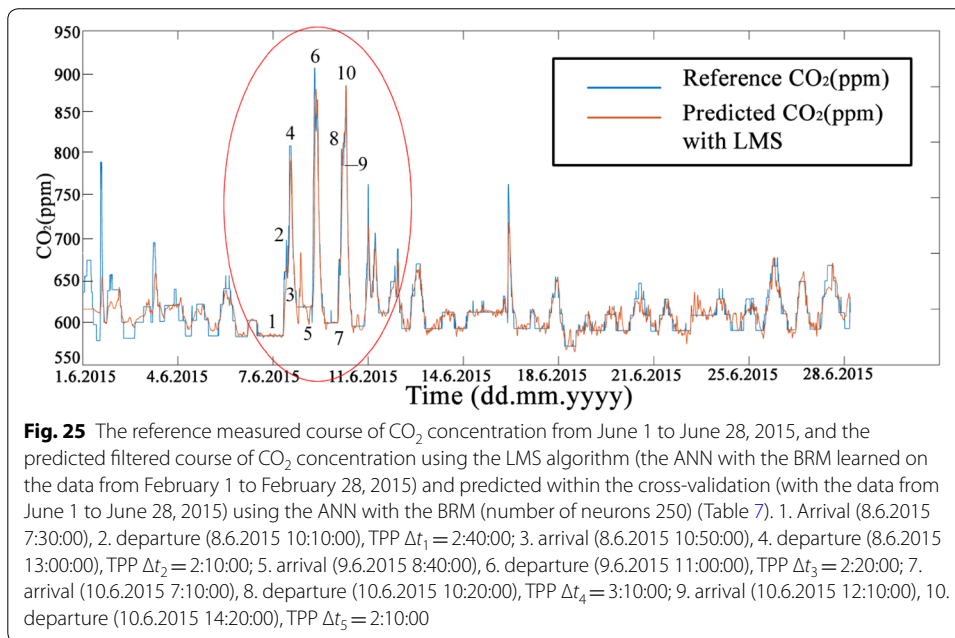
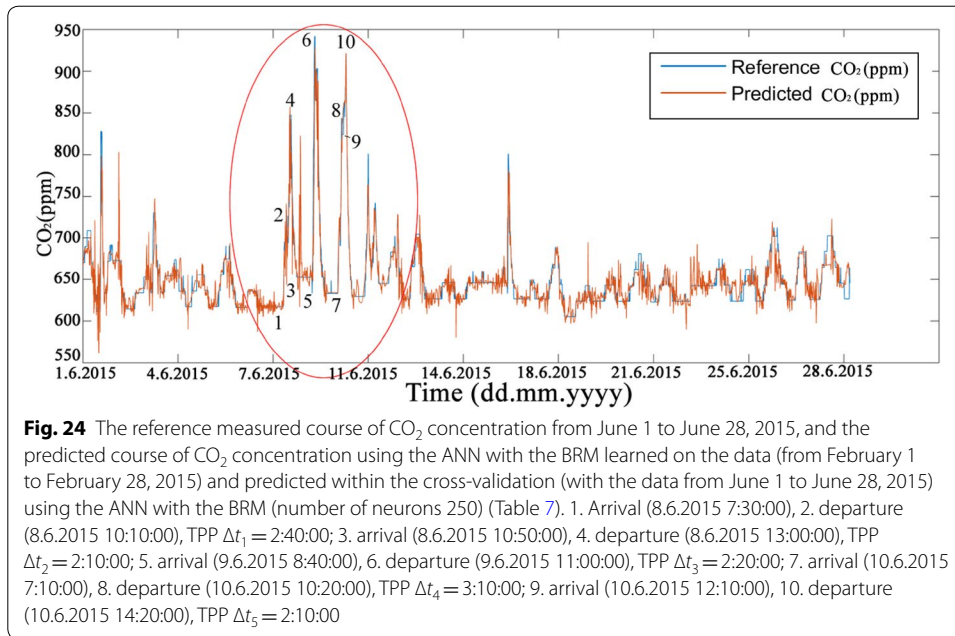
**Table 6 Comparison of prediction quality ANN (BRM) [1.2.2015–28.2.2015 (data normalized)], with the tested data from the interval [1.2.2015–28.2.2015 (data normalized)]**

Number of neurons (–)	MSE training (ppm)	R testing (–)
10	4.946·10 <sup>-3</sup>	0.55
50	2.179·10 <sup>-3</sup>	0.83
100	1.706·10 <sup>-3</sup>	0.87
150	1.708·10 <sup>-3</sup>	0.87
200	1.426·10 <sup>-3</sup>	0.89
250	1.316·10 <sup>-3</sup>	0.90
300	1.410·10 <sup>-3</sup>	0.89
350	1.209·10 <sup>-3</sup>	0.91
400	1.240·10 <sup>-3</sup>	0.91
450	1.147·10 <sup>-3</sup>	0.91
500	1.204·10 <sup>-3</sup>	0.91
550	1.287·10 <sup>-3</sup>	0.90
600	1.017·10 <sup>-3</sup>	0.93
650	1.137·10 <sup>-3</sup>	0.92
700	1.269·10 <sup>-3</sup>	0.91
750	1.134·10 <sup>-3</sup>	0.92
800	1.555·10 <sup>-3</sup>	0.89

**Table 7 Comparison of prediction quality ANN (BRM) [1.2.2015–28.2.2015 (data normalized)], with the tested data from the interval [1.6.2015–28.6.2015 (data normalized)] (cross validation)**

Number of neurons (–)	MSE training (ppm)	R testing (–)
10	8.374·10 <sup>-3</sup>	0.63
50	3.691·10 <sup>-3</sup>	0.86
100	3.012·10 <sup>-3</sup>	0.89
150	2.874·10 <sup>-3</sup>	0.89
200	2.573·10 <sup>-3</sup>	0.90
250	2.042·10 <sup>-3</sup>	0.92
300	3.324·10 <sup>-3</sup>	0.87
350	3.604·10 <sup>-3</sup>	0.86
400	3.376·10 <sup>-3</sup>	0.87
450	3.758·10 <sup>-3</sup>	0.85
500	3.251·10 <sup>-3</sup>	0.87
550	3.105·10 <sup>-3</sup>	0.88
600	3.223·10 <sup>-3</sup>	0.88
650	3.161·10 <sup>-3</sup>	0.88
700	3.787·10 <sup>-3</sup>	0.85
750	4.753·10 <sup>-3</sup>	0.81
800	2.677·10 <sup>-3</sup>	0.90

Comparison of the prediction results of the ANN BRM without filtering and with LMS filtering using the DTW criteria for the long-term experiment—28 days (from February 1 to February 28, 2015) (Table 1). Using relations (6)–(8), the resulting predicted course of CO<sub>2</sub> concentration was calculated using the ANN BRM (250



neurons) (Table 7), with suppression of the additive noise using the LMS adaptive filter algorithm (Fig. 25). For the comparison and classification of the measured course of CO<sub>2</sub> (Fig. 20) and the predicted course of CO<sub>2</sub> without filtering (Fig. 24) using the ANN BRM with cross-validation (250 neurons) (Table 7) distance  $D$  was calculated using relations (13), (14)  $D = 0.288$ . For the comparison and classification of the measured course of CO<sub>2</sub> (Fig. 20) and the predicted course of CO<sub>2</sub> using the ANN BRM with cross-validation (250 neurons) (Table 7), using the LMS adaptive filter for

suppression of the additive noise (Fig. 25), distance  $D$  was calculated using relations (13), (14)  $D = 0.143$  (M—Misadjustment).

III. The short-term experiment—7 days. Training the ANN with the BRM was implemented on the data (from February 8 to February 14, 2015) (Table 3). 1 week (08/02/2015–14/02/2015).

Training ANN, BRM (Step 4c) for neuron counts in an interval of 10–800, measuring the time  $t$ (s) of learning, the MSE calculation, and the correlation coefficient  $R$  for data from June 8 to June 14, 2015 (Fig. 26, Table 9). The total number of samples—1152. Training—70%, 806 samples. Validation—15%, 173 samples. Testing—15%, 173 samples. The prediction (Step 5c) of  $\text{CO}_2$  concentration using the ANN with the BRM was implemented by, firstly, using the measured values of rH (%) indoors, temperature indoors  $T_{\text{in}}$  (°C), temperature outdoors  $T_{\text{out}}$  (°C) on the training data from June 8 to June 14, 2015 (Table 10). Further, the cross-validation (Step 6c) used the training data [rH (%) indoors, temperature indoors  $T_{\text{in}}$  (°C), temperature outdoors  $T_{\text{out}}$  (°C)] that was measured from February 1 to February 28, 2015 (Table 11). Table 9 shows the calculated and measured parameters for training the ANN BRM (Step 4c) (a number of neurons set, the measured time  $t$  of training the ANN with the BRM and the calculated  $R$  and MSE values for the trained ANN with the BRM) on the data measured in the interval from June 8 to June 14, 2015. The minimum value of  $\text{MSE} = 7.726 \cdot 10^{-4}$  and maximum value of correlation coefficients  $R = 0.987$  were calculated for ANN with the BRM where the configured number of neurons was 600 (Table 9).

Table 10 shows the calculated values of  $R$  and MSE parameters for the number of neurons set within the testing of the ANN with the BRM (from June 8 to June 14, 2015) for predicting (Step 5c) the course of  $\text{CO}_2$  using the ANN with the BRM on the data measured in the interval from June 8 to June 14, 2015.

The best calculated minimum MSE parameter ( $\text{MSE} = 2.021 \cdot 10^{-3}$ ) and maximum  $R$  correlation coefficient ( $R = 0.97$ ) (Table 10) for predicted courses of  $\text{CO}_2$  are for trained ANN BRM with 400 neurons.

Table 11 shows the calculated values of  $R$  and MSE parameters for the number of neurons (10–800) set within the testing of the ANN with the BRM (from June 8 to June 14, 2015) for predicting (Step 6c) the course of  $\text{CO}_2$  using the ANN with the BRM on the data measured in the interval from February 8 to February 14, 2015. The best calculated minimum MSE parameter ( $\text{MSE} = 1.414 \cdot 10^{-3}$ ) and maximum  $R$  correlation coefficient ( $R = 0.89$ ) (Table 11) for predicted courses of  $\text{CO}_2$  (Fig. 26) are for trained ANN BRM with 400 neurons (cross validation).

Figure 26 illustrates the reference measured course of  $\text{CO}_2$  concentration from February 8 to February 14, 2015, and the predicted course of  $\text{CO}_2$  concentration using the ANN with the BRM learned (Step 4c) on the data (from June 8 to June 14, 2015) and predicted within the cross-validation (Step 6c) (with the data from February 8 to February 14, 2015) using the ANN with the BRM (number of neurons 400) (Table 11). The red circles indicate the predicted  $\text{CO}_2$  values that were incorrectly calculated within the prediction error using the ANN with the BRM.



Figure 27 illustrates the reference measured course of CO<sub>2</sub> concentration from February 8 to February 14, 2015. Figure 27 further illustrates the predicted filtered course of CO<sub>2</sub> concentration using the LMS algorithm [the ANN with the BRM learned on the data (from June 8 to June 14, 2015) and predicted within the cross-validation (Step 6c) with the data from February 8 to February 14, 2015, using the ANN with the BRM (the number of neurons—400) (Table 11)]. Based on relations (6) to (11), the step size parameter  $\mu$  values (Table 12) for setting the structure of the LMS algorithm adaptive filter ( $M=44$ ,  $\mu=5.63 \cdot 10^{-2}$ ) for noise elimination in the calculated predicted course of CO<sub>2</sub> (Fig. 26).

Comparison of the prediction results of the ANN BRM without filtering and with LMS filtering using the DTW criteria for the short-term experiment—7 days (from June 8 to June 14, 2015) (Table 9). Using relations (6)–(8), the resulting predicted course of CO<sub>2</sub> concentration was calculated using the ANN BRM (400 neurons) (Table 11), with suppression of the additive noise using the LMS algorithm (Fig. 27). For the comparison and classification of the measured course of CO<sub>2</sub> and the predicted course of CO<sub>2</sub> without filtering (Fig. 26) using the ANN BRM with cross-validation (400 neurons) (Table 11) was distance  $D$  calculated using relations (13), (14)  $D=0.256$ . For the comparison and classification of the measured course of CO<sub>2</sub> and the predicted course of CO<sub>2</sub> using the ANN BRM with cross-validation (400 neurons) (Table 11), using the LMS algorithm for suppression of the additive noise (Fig. 27) distance  $D$  was calculated using relations (13), (14)  $D=0.107$  (M—Misadjustment).

## Discussion

### Experiment I

For the training ANN BRM (Step 4a) learned on the data from June 1 to June 28, 2015, for the set number of neurons (10–800), R values ranging from 63 to 96% were calculated (Table 1). Using the trained ANN BRM (Step 4a) learned on the data from June 1 to June 28, 2015, the course of CO<sub>2</sub> concentration, with the input measured values rH and  $T_p$ ,  $T_o$  from June 1 to June 28, 2015, with the calculated R values ranging from 61 to 94% (Table 2) for the set number of neurons from 10 to 800, was predicted (Step 5a). For the prediction (Step 6a) of the course of CO<sub>2</sub> on the ANN learned from June 1 to June 28, 2015, within the cross-validation for the data measured from February 1 to February 28, 2015, R values ranging from 52 to 89% (Table 3) were calculated. Distance  $D$  calculated using the DTW criteria (3) and (4) was bigger,  $D=0.239$ , for the predicted course of CO<sub>2</sub> concentration (Fig. 22) without using the LMS algorithm. When the LMS algorithm was used (Fig. 23), the distance between the compared courses was smaller,  $D=0.176$ . From the above-stated results, it can be stated that the LMS algorithm has brought more

**Table 8 The calculated optimal step size parameter  $\mu$  with securing the stability and convergence of the LMS algorithm**

$$\mu_{\text{stab}} = 0.0476 \cdot 10^{-1} (M=10\%), \text{ calculation according to Eq. (9)}$$

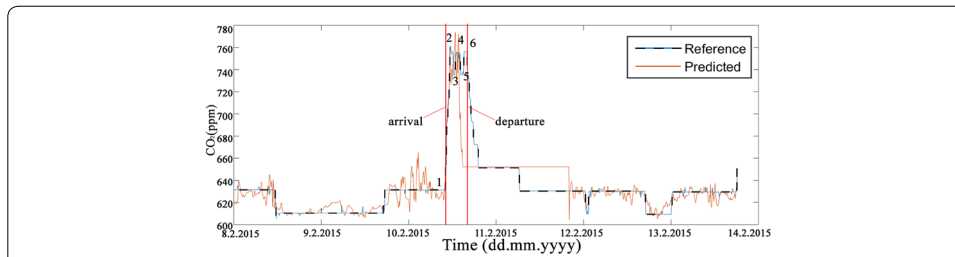
$$\mu_{\text{conv}} = 1.2233 (M=10\%), \text{ calculation according to Eq. (10)}$$

$$\mu_{\text{opt1}} = 0.013 (M=10\%), \text{ calculation according to Eq. (11)}$$

$$\mu_{\text{opt2}} = 0.0238 (M=20\%), \text{ calculation according to Eq. (11)}$$

$$\mu_{\text{opt3}} = 0.329 (M=30\%), \text{ calculation according to Eq. (11)}$$

$M$  Misadjustment



**Fig. 26** The reference [measured course of CO<sub>2</sub> concentration (from February 8 to February 14, 2015)] and the predicted course of CO<sub>2</sub> concentration. The ANN with the BRM learned (from June 8 to June 14, 2015), the predicted ANN with the data from February 8 to February 14, 2015—the number of neurons—400 (Table 11). 1. Arrival (8:50:00), 2. departure (10:30:00), TPP Δt<sub>1</sub> = 1:40:00; 3. arrival (10:50:00), 4. departure (11:10:00), TPP Δt<sub>2</sub> = 0:20:00; 5. arrival (11:20:00) and 6. departure (11:30:00), TPP Δt<sub>3</sub> = 0:20:00

**Table 9 Comparison of learning quality ANN (BRM), [8.6.2015–14.6.2015 (data normalized)]**

Number of neurons (–)	Time t (hh:mm:ss)	MSE training (ppm)	R training (–)
10	0:00:07	5.74·10 <sup>-3</sup>	0.902
50	0:02:37	9.904·10 <sup>-4</sup>	0.984
100	0:04:00	2.000·10 <sup>-3</sup>	0.967
150	0:06:42	1.766·10 <sup>-3</sup>	0.969
200	0:09:08	8.501·10 <sup>-4</sup>	0.985
250	0:13:48	1.121·10 <sup>-3</sup>	0.971
300	0:20:16	1.232·10 <sup>-3</sup>	0.979
350	0:30:36	9.423·10 <sup>-4</sup>	0.984
400	0:43:02	1.220·10 <sup>-3</sup>	0.980
450	0:56:39	8.365·10 <sup>-4</sup>	0.986
500	1:12:08	7.153·10 <sup>-4</sup>	0.987
550	1:28:24	1.072·10 <sup>-3</sup>	0.983
600	1:51:25	7.726·10 <sup>-4</sup>	0.987
650	2:09:47	1.133·10 <sup>-3</sup>	0.98
700	2:41:31	7.869·10 <sup>-4</sup>	0.987
750	3:23:09	1.123·10 <sup>-3</sup>	0.982
800	3:55:14	9.414·10 <sup>-4</sup>	0.984

accuracy to the proposed method in the long-term experiment for the training ANN BRM (Step 4a) learned on the data from June 1 to June 28, 2015, for the set number of neurons (400) (Table 3) within cross-validation data prediction CO<sub>2</sub> (Step 6a) with the tested data (from February 1 to February 28, 2015) (Table 3).

**Experiment II**

For the training ANN BRM (Step 4b) learned on the data from February 1 to February 28, 2015, for the set number of neurons (10–800), R values ranging from 55 to 93% were calculated (Table 5). Using the trained ANN BRM (Step 4b) learned on the data from February 1 to February 28, 2015, the course of CO<sub>2</sub> concentration, with the input measured values rH and T from February 1 to February 28, 2015, with the calculated R values ranging from 55 to 93% (Table 6) for the set number of neurons from 10 to 800, was predicted (Step 5b). For the prediction (Step 6b) of the course of CO<sub>2</sub> on the trained ANN

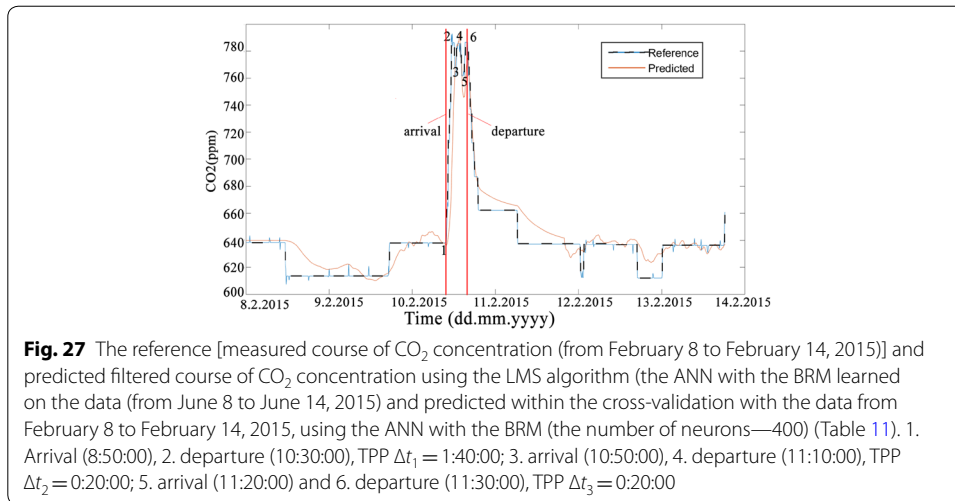
**Table 10 Comparison of prediction quality ANN (BRM) [8.6.2015–14.6.2015 (data normalized)], with the tested data from the interval [8.6.2015–14.6.2015 (data normalized)]**

Number of neurons (–)	MSE training (ppm)	R testing (–)
10	4.739·10 <sup>-3</sup>	0.92
50	3.550·10 <sup>-3</sup>	0.94
100	2.117·10 <sup>-3</sup>	0.96
150	2.923·10 <sup>-3</sup>	0.95
200	2.819·10 <sup>-3</sup>	0.95
250	3.417·10 <sup>-3</sup>	0.94
300	5.311·10 <sup>-3</sup>	0.91
350	6.680·10 <sup>-3</sup>	0.90
400	2.021·10 <sup>-3</sup>	0.97
450	4.247·10 <sup>-3</sup>	0.93
500	4.651·10 <sup>-2</sup>	0.59
550	3.580·10 <sup>-3</sup>	0.94
600	2.249·10 <sup>-3</sup>	0.96
650	2.934·10 <sup>-3</sup>	0.95
700	1.649·10 <sup>-2</sup>	0.79
750	2.383·10 <sup>-3</sup>	0.96
800	2.289·10 <sup>-3</sup>	0.96

**Table 11 Comparison of prediction quality ANN (BRM) [8.6.2015–14.6.2015 (data normalized)], with the tested data from the interval [8.2.2015–14.2.2015 (data normalized)] (cross validation)**

Number of neurons (–)	MSE training (ppm)	R testing (–)
10	5.168·10 <sup>-3</sup>	0.13
50	2.002·10 <sup>-3</sup>	0.38
100	1.590·10 <sup>-3</sup>	0.40
150	1.515·10 <sup>-3</sup>	0.41
200	1.418·10 <sup>-3</sup>	0.42
250	1.552·10 <sup>-3</sup>	0.41
300	2.297·10 <sup>-3</sup>	0.37
350	1.532·10 <sup>-3</sup>	0.42
400	1.414·10 <sup>-3</sup>	0.89
450	1.469·10 <sup>-3</sup>	0.82
500	2.870·10 <sup>-3</sup>	0.81
550	1.776·10 <sup>-3</sup>	0.87
600	2.098·10 <sup>-3</sup>	0.86
650	2.592·10 <sup>-3</sup>	0.83
700	2.726·10 <sup>-3</sup>	0.82
750	2.780·10 <sup>-3</sup>	0.50
800	2.451·10 <sup>-3</sup>	0.83

BRM (Step 4b) learned from February 1 to February 28, 2015, within the cross-validation for the data measured from June 1 to June 28, 2015, R values ranging from 63 to 92% (Table 7) were calculated. The results obtained (Table 7) confirmed the improvement of the predicted course of CO<sub>2</sub> by means of the ANN BRM using the LMS algorithm.



**Table 12** The calculated optimal step size parameter  $\mu$  with securing the stability and convergence of the LMS algorithm

$$\mu_{\text{conv}} = 1.881^{(M=10\%)}, \text{ calculation according to Eq. (9)}$$

$$\mu_{\text{stab}} = 0.0227^{(M=10\%)}, \text{ calculation according to Eq. (10)}$$

$$\mu_{\text{opt1}} = 5.63 \cdot 10^{-2(M=10\%)}, \text{ calculation according to Eq. (11)}$$

$$\mu_{\text{opt2}} = 1.24 \cdot 10^{-1(M=20\%)}, \text{ calculation according to Eq. (11)}$$

$$\mu_{\text{opt3}} = 1.859 \cdot 10^{-1(M=30\%)}, \text{ calculation according to Eq. (11)}$$

*M* Misadjustment

Distance  $D$  calculated using the DTW criteria (3) and (4) was bigger,  $D = 0.288$ , for the predicted course of CO<sub>2</sub> concentration (Fig. 20) without using the LMS algorithm. When the LMS algorithm was used (Fig. 21), the distance between the compared courses was smaller,  $D = 0.143$ .

From the above-stated results, it can be stated that the LMS algorithm has brought more accuracy to the proposed method in the long-term experiment for the training ANN BRM (Step 4a) learned on the data from February 1 to February 28, 2015, for the set number of neurons (250) (Table 7) within cross-validation data prediction CO<sub>2</sub> (Step 6b) with the tested data from June 1 to June 28, 2015) (Table 7).

### Experiment III

The achieved results confirmed the improvement in the predicted course of CO<sub>2</sub> by means of ANN BRM using the LMS algorithm. Distance  $D$  calculated by means of the DTW criterion (13) and (14)  $D = 0.256$  was higher for the predicted course of CO<sub>2</sub> (Fig. 26) not using the LMS algorithm. Similarly, the correlation coefficient  $R = 0.89$  was lower. When the LMS algorithm was used (Fig. 27), the distance between the compared courses was smaller,  $D = 0.107$ . Similarly, the calculated correlation coefficient  $R = 0.93$  was higher. The advantage of the proposed method consists in the use of common operating sensors to obtain information on the state of the operational-technical functions in the IAB for the purpose of their optimum control on the basis of predictable needs of the persons using the IAB areas. The disadvantage of the proposed method with the

LMS algorithm is the slow adaptation of the LMS algorithm to real conditions depending on the precision of calculation of the step size parameter  $\mu$  and inaccurate determination of a person's arrival and departure to and from the monitored area (Fig. 27) (between points 2 and 4). This is, however, a small decline in the CO<sub>2</sub> concentration (ppm) depending on the quick departure and arrival of the person. The proposed method is not able to record the quick change due to the small or big changes in the CO<sub>2</sub> concentration. On the basis of the above-indicated results, however, one can state that the LMS algorithm has brought improved precision of the proposed method.

Due to the complexity of the ANN BRM mathematical model, the learning process is a non-trivial optimization problem. Iterative algorithms often encounter the local minima in which the learning process is terminated. The ANN instability is a consequence of such behaviour. Especially the fact that two learning processes run on the same data (but from different starting points) end in two different local solutions is often criticized. Partial elimination of this disadvantage is ensured by using a corresponding adaptive filter. The advantage of the method designed is the possible use of common operational sensors for the detection of information on the state of operational and technical functions in the IAB. The disadvantage of the method designed with the LMS algorithm is the slow adaptation of the LMS algorithm to real conditions, depending on the accuracy of the step size parameter  $\mu$  calculation and inaccurate determination of the arrival and departure time of the person from the monitored area.

## Conclusion

The article describes the design and verification of the indirect method of predicting the course of CO<sub>2</sub> concentration (ppm) using the ANN with the BRM for monitoring the presence of people in the IAB office room using the SW Tool PI System (PI—Plant Information enterprise information system).

The first part of this article covers the individual tools and the actual design of the user interface for monitoring the technological processes in an administrative building (IAB). The goal of the first part of the article—the design and the implementation of the FEVISUAL visualization monitoring application for the administrative building of the new FEI at VŠB TU Ostrava using an Android mobile device that monitors the technological processes (HVAC) in the IAB was solved using the PI System. The development and implementation of a mobile hybrid application are characterized by the fact that it is a platform-independent application. The application was tested in real-time operation of the building in conjunction with the PI System SW tool used for monitoring, archiving and processing the data measured in the new FEI administrative building at VŠB TU Ostrava (IAB).

The article further describes the indirect method of Bayesian Regularization Method (BRM) for predicting the course of CO<sub>2</sub> from the values measured by the temperature and humidity sensors using a gradient algorithm of backpropagation for adapting the multilayered forward Neural Network (ANN). Based on the long-term measurements and the data collected in relation to the above-mentioned variables, the BRM method can be used, with a sufficient precision rate, for estimating CO<sub>2</sub> concentration in the air in order to obtain information on the occupancy of individual rooms (arrival time, departure time, number of people) in an IAB.

The article has verified using the ANN with BRM to predict CO<sub>2</sub> for the purposes of monitoring the presence of people in the selected IAB office. In addition, the article verified the improvement of the reliability and accuracy of the proposed indirect method using the LMS adaptive filter for filtering the predicted CO<sub>2</sub> concentration course for seasons in a time interval between winter and spring (from February 1 to February 28, 2015), for a time interval between spring and summer (from June 1 to June 28, 2015) and for short experiment in winter (from February 8 to February 14, 2015).

It is obvious that before the method is put into practice, it is necessary to calibrate the methods designed in real conditions. Implementation of the designed indirect method of CO<sub>2</sub> concentration prediction has potential for reducing the investment and operating costs by reducing the total number of sensors used in the IAB. In the next work, the authors will focus on improving real-time implementation of the method designed using IoT platforms for determining the occupancy of the monitored IAB areas.

#### Authors' contributions

JV proposed the system idea, performed the experiments and wrote the manuscript. JM made the creation of visualization monitoring application for mobile devices, which should monitor processes in an administrative building, specifically VŠB FEI building. JV, RM, JN and MF edited the manuscript, designed and implemented the classification methods to adequately evaluate the proposed experiments. JV, RM, JN and MF critically evaluated the scientific validity of the proposed system and acquired the vital data and performed its final edits. All authors read and approved the final manuscript.

#### Acknowledgements

This article was supported by the Ministry of Education of the Czech Republic (Project No. SP2018/170). This work was supported by the European Regional Development Fund in the Research Centre of Advanced Mechatronic Systems project, Project Number CZ.02.1.01/0.0/0.0/16\_019/0000867 within the Operational Programme Research, Development and Education.

#### Competing interests

The authors declare that they have no competing interests.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 26 February 2018 Accepted: 31 August 2018

Published online: 19 September 2018

#### References

1. Vanus J, Cerny M and Koziorek J (2015) The proposal of the smart home care solution with KNX components. In: 2015 38th international conference on telecommunications and signal processing, TSP 2015. <https://doi.org/10.1109/TSP.2015.7296410>
2. Vanus J, Martinek R, Bilik P, Koziorek J et al (2015) Smart Home remote monitoring using PI System management tools. In: Proceedings of the 8th international scientific symposium on electrical power engineering, ELEKTROENERGETIKA 2015. pp 372–375. ISBN 978-80-553-2187-5
3. Vanus J, Martinek R, Bilik P, Zidek J et al (2015) Energy management strategies in intelligent office building using PI system. In: Proceedings of the 8th international scientific symposium on electrical power engineering, ELEKTROENERGETIKA 2015. pp 416–419. ISBN 978-80-553-2187-5
4. Vanus J, Kucera P, Koziorek J (2014) The software analysis used for visualization of technical functions control in smart home care. In: Advances in intelligent systems and computing. [https://doi.org/10.1007/978-3-319-06740-7\\_47](https://doi.org/10.1007/978-3-319-06740-7_47)
5. Vanus J, Kucera P, Koziorek J (2014) Visualization software designed to control operational and technical functions in smart homes. In: Advances in intelligent systems and computing. [https://doi.org/10.1007/978-3-319-06740-7\\_48](https://doi.org/10.1007/978-3-319-06740-7_48)
6. Vanus J, Kucera P, Koziorek J, Machacek Z et al (2015) Development of a visualisation software, implemented with comfort smart home wireless control system. In: Lecture Notes in Electrical Engineering. [https://doi.org/10.1007/978-3-662-45402-2\\_84](https://doi.org/10.1007/978-3-662-45402-2_84)
7. Vanus J, Kucera P, Martinek R, Koziorek J (2014) Development and testing of a visualization application software, implemented with wireless control system in smart home care. *Hum-Centr Comput Inf Sci.* 4(1):8. <https://doi.org/10.1186/s13673-014-0019-5>
8. Vanus J, Novak T, Koziorek J, Konecny J et al (2013) The proposal model of energy savings of lighting systems in the smart home care. In: IFAC proceedings volumes (IFAC-PapersOnline). vol 12. pp 411–415. <https://doi.org/10.3182/20130925-3-CZ-3023.00014>

9. Novak T, Vanus J et al (2013) Possibility to achieve the energy savings by the light control in smart home. In: Proceedings of the 7th international scientific symposium on electrical power engineering, ELEKTROENERGETIKA 2013, pp 260–263. ISBN 978-80-553-1442-6
10. Vanus J, Martinek R, Bilik P, Zidek J et al (2016) Evaluation of thermal comfort of the internal environment in smart home using objective and subjective factors. In: Proceedings 17th international scientific conference on electric power engineering, EPE 2016. <https://doi.org/10.1109/EPE.2016.7521768>
11. Vanus J and Koziorek J (2014) Employment of dynamic time warping method for monitoring states in smart home. In: 4th International workshop on computer science and engineering-winter, WCSE 2014
12. Vanus J, Martinek R, Bilik P, Zidek J et al (2016) New method for accurate prediction of CO<sub>2</sub> in the smart home. IEEE, New York, pp 1333–1337. <https://doi.org/10.1109/I2MTC.2016.7520562>
13. Mahmoud S, Lofti A, Langensiepen C (2013) Behavioural pattern identification and prediction in intelligent environments. *Appl Soft Comput* 13(4):1813–1822. <https://doi.org/10.1016/j.asoc.2012.12.012>
14. Ni Q, Garcia AB, de la Hernando IP, Cruz P (2015) The Elderly's independent living in smart homes: a characterization of activities and sensing infrastructure survey to facilitate services development. *Sensors* 15(5):11312–11362. <https://doi.org/10.3390/s150511312>
15. Vanus J, Belesova J, Martinek R, Nedoma J et al (2017) Monitoring of the daily living activities in smart home care. *Human-Centr Comput Inf Sci.* 7:30. <https://doi.org/10.1186/s13673-017-0113-6>
16. Alshahrany F, Abbod M, Moualek I (2015) WSN and RFID integration to support intelligent monitoring in smart buildings using hybrid intelligent decision support systems. *Acta Phys Pol* 128:B152–B159. <https://doi.org/10.12693/APhysPolA.128.B-152>
17. Hailemariam E, Goldstein R, Attar R, Khan A (2011) Real-time occupancy detection using decision trees with multiple sensor types. In: Proceeding SimAUD'11 proceedings of the 2011 symposium on simulation for architecture and urban design, pp 141–148. ISBN 978-1-61782-842-3
18. Le Ny J, Mohammady M, IEEE (2014) Differentially private MIMO filtering for event streams and spatio-temporal monitoring. In: 53rd IEEE annual conference on decision and control (CDC). Los Angeles, IEEE, pp 2148–2153. <https://doi.org/10.1109/CDC.2014.7039716>
19. Liu DX, Guan XH, Du YT, Zhao QC (2013) Measuring indoor occupancy in intelligent buildings using the fusion of vision sensors. *Meas Sci Technol* 24:7. <https://doi.org/10.1088/0957-0233/24/7/074023>
20. Xu YC, Stojanovic N, Stojanovic L, Anicic D, Studer R (2011) An approach for more efficient energy consumption based on real-time situational awareness. *Sem Web* 2011(6644):270–284. [https://doi.org/10.1007/978-3-642-21064-8\\_19](https://doi.org/10.1007/978-3-642-21064-8_19)
21. Silvestre-Blanes J, Perez-Llorens R (2011) Energy efficiency improvements through surveillance applications in industrial buildings. *Energy Build* 2011(43):1334–1340. <https://doi.org/10.1016/j.enbuild.2011.01.017>
22. Sun ZW, Wang SW, Ma ZJ (2011) In-situ implementation and validation of a CO<sub>2</sub>-based adaptive demand-controlled ventilation strategy in a multi-zone office building. *Build Environ* 2011(46):124–133. <https://doi.org/10.1016/j.buildenv.2010.07.008>
23. Yang Z, Li N, Becerik-Gerber B, Orosz M (2014) A systematic approach to occupancy modeling in ambient sensor-rich buildings. *Simul Trans Soc Model Simul Int* 2014(90):960–977. <https://doi.org/10.1177/0037549713489918>
24. Zou JH, Zhao QC, Yang W, Wang FL (2017) Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation. *Energy Build* 2017(152):385–398. <https://doi.org/10.1016/j.enbuild.2017.07.064>
25. Demiris G, IEEE (2009) Privacy and social implications of distinct sensing approaches to implementing smart homes for older adults. <https://doi.org/10.1109/IEMBS.2009.5333800>
26. Dong B, Lam KP (2011) Building energy and comfort management through occupant behaviour pattern detection based on a large-scale environmental sensor network. *J Build Perform Simul* 4(4):359–369. <https://doi.org/10.1080/19401493.2011.577810>
27. Liu Z, Zhang J, Geng L, IEEE (2017) An intelligent building occupancy detection system based on sparse auto-encoder. In: IEEE winter conference on applications of computer vision workshops (WACVW). Santa Rosa, IEEE, pp 17–22. <https://doi.org/10.1109/WACVW.2017.1031>
28. Shih HC (2014) A robust occupancy detection and tracking algorithm for the automatic monitoring and commissioning of a building. *Energy Build* 77:270–280. <https://doi.org/10.1016/j.enbuild.2014.03.069>
29. Mahmoud S, Lofti A, Langensiepen C (2013) Behavioural pattern identification and prediction in intelligent environments. *Appl Soft Comput* 2013(13):1813–1822. <https://doi.org/10.1016/j.asoc.2012.12.012>
30. Zhao ZB, Xu WS, Chen DZ, IEEE (2014) EM-LDA model of user behavior detection for energy efficiency. In: IEEE international conference on system science and engineering (ICSSE). Shanghai, IEEE, pp 295–300. <https://doi.org/10.1109/ICSSE.2014.6887952>
31. Machacek Z, Hercik R, Slaby R (2011) Smart user adaptive system for intelligent object recognizing. *Stud Comput Intell* 351:197–206. [https://doi.org/10.1007/978-3-642-19953-0\\_20](https://doi.org/10.1007/978-3-642-19953-0_20)
32. Aftab M, Chen C, Chau CK, Rahwan T (2017) Automatic HVAC control with real-time occupancy recognition and simulation-guided model predictive control in low-cost embedded system. *Energy Build* 2017(154):141–156. <https://doi.org/10.1016/j.enbuild.2017.07.077>
33. Cooper PB, Maraslis K, Tryfonas T, Oikonomou G (2017) An intelligent hot-desking model harnessing the power of occupancy sensing data. *Facilities* 35(13/14):766–786. <https://doi.org/10.1108/F-01-2016-0014>
34. Dong B, Andrews B, Lam KP, Höynck M, Zhang R, Chiou Y-S, Benitez D (2010) An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network. *Energy Build* 42(7):1038–1046. <https://doi.org/10.1016/j.enbuild.2010.01.016>
35. Khajenasiri I, Virgone J, Gielen G (2015) A presence-based control strategy solution for HVAC systems. *Int Conf Consumer Elect (Icce)* 2015:620–622. <https://doi.org/10.1109/ICCE.2015.7066553>
36. Onofrejova D, Onofrej P, Simsik D (2014) Model of production environment controlled with intelligent systems. *Model Mech Mech Syst* 2014(96):330–337



37. Qidwai U, Al-Sulaiti S, Ahmed G, Hegazy A, Ilyas SK, IEEE (2016) Intelligent integrated instrumentation platform for monitoring long-term bedridden patients. In: IEEE EMBS conference on biomedical engineering and sciences (IECBES). Kuala Lumpur, IEEE, pp 561–564. <https://doi.org/10.1109/IECBES.2016.7843512>
38. Shih HC, IEEE (2016) Automatic building monitoring and commissioning via human behavior recognition. In: IEEE 5th global conference on consumer electronics, p 2. <https://doi.org/10.1109/GCCE.2016.7800417>
39. Doležilková H. Bytové větrání ve vztahu k produkci CO<sub>2</sub>, vlhkosti a škodlivin (II). (<http://vetrani.tzb-info.cz/vnitri-prostredi/3042-bytove-vetrani-ve-vztahu-k-produkci-co2-vlhkosti-a-skodlivin-ii>). Accessed 28 Nov 2017
40. Zikan Z. Oxid uhličitý-utajený nepřítel. <http://vetrani.tzb-info.cz/vnitri-prostredi/7132-oxid-uhlicity-utajeny-nepritel>. Accessed 28 Nov 2017
41. Pracujete ve zdravém prostředí? Měření oxidu uhličitého v budovách. <http://vetrani.tzb-info.cz/vnitri-prostredi/5827-pracujete-ve-zdravem-prostredi>. Accessed 28 Nov 2017
42. Vanus J, Machac J, Sykora J, Martinek R, Bilik P, Zidek J (2017) Návrh aplikace pro monitorování technologických procesů v administrativní budově. In: Journal Elektro, 4 ISSN 1210-0889. [http://www.odbornecasopisy.cz/flipviewer/Elektro/2017/04/Elektro\\_04\\_2017/index.html#p=6](http://www.odbornecasopisy.cz/flipviewer/Elektro/2017/04/Elektro_04_2017/index.html#p=6). Accessed 24 Feb 2017
43. Vanus J, Machac J, Martinek R, Bilik P et al (2017) Design of an application for the monitoring and visualization of technological processes with PI system in an intelligent building for mobile devices. In: Proceedings of the 9th international scientific symposium on electrical power engineering, ELEKTROENERGETIKA 2017. pp 518–522. ISBN 978-805533195-9
44. Pies M, Hajovsky R, Ozana S, Haska J (2014) Wireless sensory network based on IQRF technology. In 4th international workshop on computer science and engineering-winter, WCSE 2014; Empire Hotel Hong Kong-Wan ChaiHong Kong; 26 December 2014 through 28 December 2014; Code 110946
45. Machac J (2015) Nástroje pro monitorování technologických procesů v inteligentních budovách. VŠB TU Ostrava. Diploma Thesis. <https://dspace.vsb.cz/handle/10084/110828>. Accessed 11 Jan 2017
46. Ahn N, Jang H, Park DK (2007) Presumption of shear strength of steel fiber reinforced concrete beam using artificial neural network model. *J Appl Polym Sci* 2007(103):2351–2358. <https://doi.org/10.1002/app.25121>
47. Aqil M, Kita I, Yano A, Nishiyama S (2007) Neural networks for real time catchment flow modeling and prediction. *Water Resour Manage* 2007(21):1781–1796. <https://doi.org/10.1007/s11269-006-9127-y>
48. Hwang Y, Deng S (2008) Applying neural networks to the solution of the inverse heat conduction problem in a gun barrel. *J Press Vessel Technol Trans Asme* 2008(130):8. <https://doi.org/10.1115/1.2937763>
49. Jazayeri K, Jazayeri M, Uysal S (2016) Comparative analysis of Levenberg-Marquardt and Bayesian regularization backpropagation algorithms in photovoltaic power estimation using artificial neural network. *Adv Data Mining* 2016(9728):80–95. [https://doi.org/10.1007/978-3-319-41561-1\\_7](https://doi.org/10.1007/978-3-319-41561-1_7)
50. Kumar P, Merchant SN, Desai UB (2004) Improving performance in pulse radar detection using Bayesian regularization for neural network training. *Dig Signal Process* 2004(14):438–448. <https://doi.org/10.1016/j.dsp.2004.06.002>
51. Aqil M, Kita I, Yano A, Nishiyama S (2007) A comparative study of artificial neural networks and neuro-fuzzy in continuous modeling of the daily and hourly behaviour of runoff. *J Hydrol* 2007(337):22–34. <https://doi.org/10.1016/j.jhydrol.2007.01.013>
52. Arora I, Saha A (2016) Comparison of back propagation training algorithms for software defect prediction. In: Proceedings of the 2016 2nd international conference on contemporary computing and informatics (Ic3i). pp 51–58. <https://doi.org/10.1109/IC3I.2016.7917934>
53. Caballero J, Fernandez M (2006) Linear and nonlinear modeling of antifungal activity of some heterocyclic ring derivatives using multiple linear regression and Bayesian-regularized neural networks. *J Mol Model* 12:168. <https://doi.org/10.1007/s00894-005-0014-x>
54. Baghirli O (2015) Comparison of Lavenberg-Marquardt, scaled conjugate gradient and Bayesian regularization backpropagation algorithms for multistep ahead wind speed forecasting using multilayer perceptron feedforward neural network. In: Dissertation in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE WITH A MAJOR IN ENERGY TECHNOLOGY WITH FOCUS ON WIND POWER. Uppsala University, Department of Earth Sciences, Campus Gotland. <https://www.diva-portal.org/smash/get/diva2:828170/FULLTEXT01.pdf>. Accessed 4 Jan 2018
55. Hložek B (2017) Bayesovské a neuronové sítě. VUT Brno. Diploma Thesis. <https://dspace.vutbr.cz/bitstream/handle/11012/69530/7526.pdf?sequence=1&isAllowed=y>. Accessed 4 Jan 2018
56. Kayri M (2016) Predictive abilities of Bayesian regularization and Levenberg-Marquardt algorithms in artificial neural networks: a comparative empirical study on social data. *Math Comput Appl* 21(2):20. <https://doi.org/10.3390/mca21020020>
57. Chae YT, Horesh R, Hwang Y, Lee YM (2016) Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings. *Energy Build* 2016(111):184–194. <https://doi.org/10.1016/j.enbuild.2015.11.045>
58. Farhang-Boroujeny B (2005) Adaptive filters: theory and applications. Wiley, Chichester, pp 139–168. ISBN 0-471-98337-3
59. Poularikas AD, Ramadan ZM (2006) Adaptive filtering primer with MATLAB. CRC/Taylor & Francis, Boca Raton, pp 101–122. ISBN 0-8493-7043-4
60. Haykin SS (2014) Adaptive filter theory. Pearson, Upper Saddle River, pp 231–319. ISBN 013267145X
61. Uhlir J (2007) Technologie hlasových komunikací. Nakladatelství ČVUT, Praha, pp 161–165. ISBN 978-80-01-03888-8
62. Vanus J, Styskala V (2010) Application of optimal settings of the LMS adaptive filter for speech signal processing. In: Proceedings of the international multiconference on computer science and information technology. pp 767–774. <https://doi.org/10.1109/IMCSIT.2010.5679739>