**DOI: https://doi.org/10.15276/aait.02.2021.1**
**UDC 004.93.1**

# DISTRIBUTED DEEP LEARNING FRAMEWORK FOR SMART BUILDING TRANSDUCER NETWORK

**Ivan M. Lobachev**[1)]
ORCID: https://orcid.org/0000-0002-4859-304X; lobachev.i.m@gmail.com. Scopus ID: 57192379296
**Svitlana G. Antoshchuk**[1)]
ORCID: https://orcid.org/0000-0002-9346-145X; asg@opu.ua. Scopus ID: 8393582500
**Mykola A. Hodovychenko**[1)]
ORCID: https://orcid.org/0000-0001-5422-3048; nick.godov@gmail.com. Scopus ID: 57188700773
[1)] Odessa National Polytechnic University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

## ABSTRACT

This work is devoted to the development of a distributed framework based on deep learning for processing data from various sensors that are generated by transducer networks that are used in the field of smart buildings. The proposed framework allows you to process data that comes from sensors of various types to solve classification and regression problems. The framework architecture consists of several subnets: particular convolutional net that handle input from the same type of sensors, a single convolutional fusion net that processes multiple outputs of particular convolutional nets. Further, the result of a single convolutional fusion net is fed to the input of a recurrent net, which allows extracting meaningful features from time sequences. The result of the recurrent net operation is fed to the output layer, which generates the framework output based on the type of problem being solved. For the experimental evaluation of the developed framework, two tasks were taken: the task of recognizing human actions and the task of identifying a person by movement. The dataset contained data from two sensors (accelerometer and gyroscope), which were collected from 9 users who performed 6 actions. A mobile device was used as the hardware platforms, as well as the Edison Compute Module hardware device. To compare the results of the work, variations of the proposed framework with different architectures were used, as well as third-party approaches based on various methods of machine learning, including support machines of vectors, a random forest, limited Boltzmann machines, and so on. As a result, the proposed framework, on average, surpassed other algorithms by about 8% in three metrics in the task of recognizing human actions and turned out to be about 13% more efficient in the task of identifying a person by movement. We also measured the power consumption and operating time of the proposed framework and its analogues. It was found that the proposed framework consumes a moderate amount of energy, and the operating time can be estimated as acceptable.

**Keywords**: Smart Building; Internet of Things; Deep Learning; Convolutional Neural Network; Gated Recurrent Unit; Recurrent Neural Network; Long Short-Term Memory

***For citation*: Lobachev I. M., Antoshchuk S. G., Hodovychenko M. A. Distributed Deep Learning Framework for Smart Building Transducer Network. *Applied Aspects of Information Technology*. 2021; Vol. 4 No. 2: 127–139. DOI: https://doi.org/10.15276/aait.02.2021.1**

## INTRODUCTION, FORMULATION OF THE PROBLEM

In recent years, the topic of smart buildings has attracted a lot of attention from scientists and engineers around the world. According to a report published by the Building Services Research and Information Association (BRSIA) [1], the market for the smart building industry was US $ 1,036 billion in 2020, creating ample opportunity for the development and application of advanced information and telecommunications technologies. For the first time, the term “Intelligent Building” was introduced into circulation by UTBS Corporation [2] in 1981 and, until now, among researchers there is no common understanding of the concept of smart buildings [3, 4], [5, 6].



Within the framework of this work, the term “smart building” should be understood as a building equipped with complex technological systems, communications and controls that ensure the safety, comfort and health of its inhabitants.

To maximize comfort, minimize costs, and adapt to the needs of their residents, smart buildings must rely on sophisticated tools to educate, predict, and make smart decisions.

These tools span a range of technologies including forecasting, decision making, robotics, smart materials, wireless sensor networks, multimedia, mobile computing, and cloud computing.

With these technologies, buildings can cognitively manage many services such as security, privacy, energy efficiency, lighting, maintenance, elderly care, and multimedia entertainment.

The development of smart building is directly

related to the concept of the Internet of Things (IoT) since the Internet of Things allows all communications and devices of a smart building to be linked into a single interacting network.

The term IoT is usually understood as a distributed network, the nodes of which are a set of transducers (sensors and actuators) controlled by embedded microprocessor systems. Transducers are used to collect data and interact with the external environment, and microprocessor systems provide network node control and communication between nodes.

The conventional approach to the development of IoT systems implies sending the entire array of data generated by transducers to a single point of aggregation and analysis, which is usually represented by a cloud server that applies machine learning methods and algorithms to obtain new knowledge, predict and generate control action on network nodes [7].

The exact opposite of centralized IoT systems is the approach in which the IoT system is a peer-to-peer distributed mesh network, in which a centralized server is either completely absent or acts as a client interface for monitoring the network status or for manual operator control. This approach does not allow full use of machine learning technologies to analyze the collected data. This is due to the low computing power of the nodes, as well as restrictions on the consumption of electricity by the nodes, which are often imposed on IoT systems [8].

These contradictions can be resolved by constructing a hierarchical model of the IoT network based on the principle of fog computing, in which data aggregation and analysis is performed in the immediate vicinity of the place of their generation [9]. To improve the efficiency of such networks, it is necessary to develop methods for adapting deep learning technologies for their use in the context of edge computing.

Thus, **the goal of this paper** is to develop a distributed framework based on deep learning that would allow aggregating time series of data from a variety of sensors of different types in the context of an fog computing model.

## 1. LITERATURE REVIEW

Fog computing is a new paradigm whereby the cloud computing model is expanded by a network node to be used as a computing node [10]. Like the cloud paradigm, fog computing also enables storage functions and application services [11].

The paper [12] presents the main characteristics of fog computing systems:

– heterogeneity – computing nodes are located at the border of a network with diverse and heterogeneous end devices;

– functionality – it is possible to use in a large number of industrial tasks due to instant response;

– storage and services – has its own network and computing services, as well as data warehouses;

– work area – works locally (one “jump” from the device to the fog node);

– additional capabilities – offers cheap, flexible, and portable deployments in terms of hardware and software.

Fog computing should be distinguished from the cloud computing model. The works [13, 14], [15] summarize the main differences between the two models:

– resources – fog nodes have significantly less resources (memory, processing power and storage) than cloud servers, but resources can be increased upon request;

– functionality – both models process data that was generated by a different set of devices;

– distribution strategies – can be densely or sparsely distributed geographically;

– connectivity – both models support wireless communication and machine-to-machine interaction;

– flexibility – the peripheral system can be deployed on low-end devices such as internet routers and ip cameras.

The fog computing model should be distinguished from the edge computing model, although they serve the same purpose of reducing network load and end-to-end latency. The main difference between edge computing and fog computing is the way data is processed and the computing power found.

The edge computing model is that computing power is concentrated around data sources such as mobile devices, sensors, and actuators. In the model of fog computing, a node makes a decision on local processing of data or on sending them to a server for centralized processing [16, 17], [18].

Consider several works that are related to the problem of network resource management and the problem of decision making in systems using the edge computing model.

In [19], a method for balancing computing on


ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

peripheral devices using a decentralized machine learning algorithm is presented, which makes it possible not to send data to a centralized server.

A similar solution was proposed in [20], where local machine learning algorithms are used to automate the management of the components of a remote medical surveillance system.

In [21], an algorithm is used for the efficient allocation of computing resources in the problem of recognizing and understanding music.

To solve the big data problem, in [22] the authors propose a solution for transferring the computational load from the central server to the fog nodes.

In [23] an intelligent surveillance stream data balancing solution was proposed. The main idea of the proposed solution is the use of an intelligent system for classifying images periodically received from cameras of a video surveillance system.

A similar approach was used in [24], which is devoted to the analysis of vehicle traffic to predict road congestion. The proposed distribution of the computational load, according to the authors, makes the traffic congestion prediction algorithm immune to the problem of losing the connection with the central server. The design of the network allows the use of a distributed network of conditionally limited Boltzmann machines to process data obtained from various sensors.

The work [25] proposes the SmartFog architecture. This architecture allows for low latency decision making as well as adaptive network resource management. According to the authors' assurances, the proposed architecture makes it possible to emulate some of the functions of the human brain.

The authors of [26] focused on monitoring the patient's health and choosing the optimal solution by combining machine learning methods and Markov chains. Using the proposed Directional Mesh Network (DMN) framework, time sensitive data is analyzed near the signal source using various machine learning methods. As part of this work, the peripheral device can make a “smart” decision about whether to send the received data to a central server for processing or not. This solution is achieved using various machine learning techniques.

Machine learning methods are used in [27], which presents the concept of Smart Cargo, which allows making decisions in various situations.

Analysis of the proposed work in the field of machine learning in the fog computing model allows us to conclude that the vast majority of works use unsupervised learning techniques to improve decision-making ability in the fog computing model. Most of the work is based on the application of machine learning technologies in conditions of limited resources and unexpected situations.

## 2. DEEP LEARNING FRAMEWORK DESIGN

Let us assume that there is $K$ number of different types of input sensors $I$, where $I = \{I_k\}, n \in \{1, \dots, K\}$. Let us take a closer look at $I_k$, that generates a series of measurements over a certain period of time. These measurements can be represented though a matrix $M$ that would have a size of $d^{(k)} \times z^{(k)}$ for the collected data points, and by a $z^{(k)}$-sized vector $v$ for the time-series data. Where $d^{(k)}$ is the dimensionality of each measurement (for example, a 3-axis movement sensor), and $z^{(k)}$ is the number of measurements. By breaking up the input matrix $M$ and vector $v$ into time segments in order to form a series of non-overlapping time intervals with a width $\tau$ then we get $\Lambda = \left\{\left(M_t^{(k)}, v_t^{(k)}\right)\right\}$, where $|\Lambda| = T$. Let us assume that $\tau$ remains constant.

Then, we can apply a Fast Fourier Transform (FFT) for each of the elements in Λ, because the frequency domain contains the best local frequency patterns that are not dependent on the time domain behavior.

After that, the tensor $X^{(k)}$ of dimension $d^{(k)} \times 2f \times T$ is formed from these outputs. The set of final tensors $\chi = \{X^{(k)}\}$ for each sensor is fed to the input of the framework.

The proposed framework is shown in Figure 1 and has three components: the first component consists of many convolutional layers, the second component consists of recurrent layers, and the third component is represented by the output layer.

### 2.1. Convolutional nets

The convolutional layers can be divided into two categories: a set of particular convolutional nets for each tensor $X^{(K)}$of the input sensor, and one combination subnet, the input to which are the outputs $K$ of the convolutional nets.

As the structure of the particular convolutional net for various sensors is the same, let us look in

more detail at one of the convolutional network, which has a tensor $X^{(k)}$ feeding into it as an input. As established previously, $X^{(k)} \in \mathbb{R}^{d^{(k)} \times 2f \times T}$, where $d^{(k)}$is the dimensionality of the data received from the sensor, $f$ – the dimensionality in the frequency domain, and $T$ – is the number of time intervals.

For each interval of time $t$, the matrix $\mathrm{X}_{\cdot\cdot t}^{(k)}$ is passed as an input to the convolutional network with three layers. The goal is to extract two types of relations out of $\mathrm{X}_{\cdot\cdot t}^{(k)}$. The relation in the frequency domain and one between the measurements done by the sensors. The frequency domain contains a lot of local patterns with neighboring frequencies. The relations between the measurements done by the sensors are usually connected to all the dimensionalities.

Because of this property, we can apply two-dimensional filters in the form of $(d^{(k)}, cov1)$ to $\mathrm{X}_{\cdot\cdot t}^{(k)}$to obtain the relations between the measurements and patterns in the frequency domain, by obtaining $\mathrm{X}_{\cdot\cdot t}^{(k,1)}$as an output. After this step, filters in the form of $(1, cov2)$ and $(1, cov3)$ will be applied to $\mathrm{X}_{\cdot\cdot t}^{(k,1)}$ in sequence, to obtain $\mathrm{X}_{\cdot\cdot t}^{(k,2)}$ and $\mathrm{X}_{\cdot\cdot t}^{(k,3)}$.

After this the matrix $\mathrm{X}_{\cdot\cdot t}^{(k,3)}$ is flattened out into the vector $x_{\cdot\cdot t}^{(k,3)}$. Afterwards $K$ vectors from particular convolutional nets joined together to form the matrix $X_{\cdot\cdot t}^{(3)}$with $K$ lines, that is fed as an input to a single fusion net.

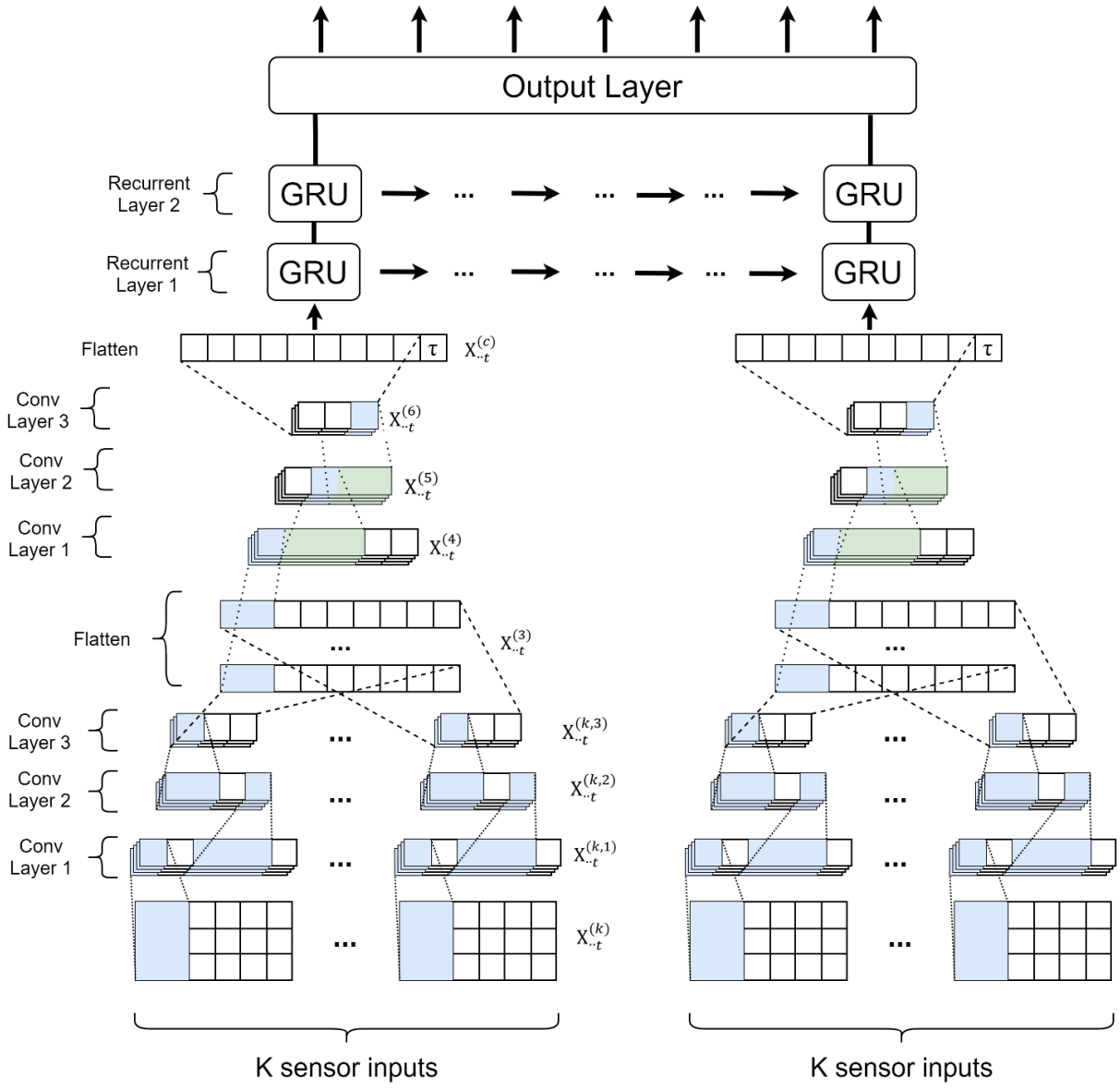



*Fig. 1.* **Distributed framework architecture**

***Source:*** **compiled by the authors**


ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

The structure of the single fusion net is similar to the structure of the particular convolutional net. First, a two-dimensional filter is applied in the form of $(K, cov4)$, to find the relation between all of the $K$ sensors with the output $X_{..t}^{(4)}$and then subsequent sequential application of filters $(1, cov5)$ and $(1, cov6)$ to obtain outputs $X_{..t}^{(5)}$ and $X_{..t}^{(6)}$ respectively.

Each of the convolutional layers uses 64 filters and rectified linear unit as the activation function. Also, each layer will also have batch normalization applied to it, in order to lower the internal covariance shift.

In the interest of simplifying the structure of proposed framework, it was decided to exclude the residual structures.

As a last step of this stage, the obtained matrix $X_{..t}^{(6)}$is flattened into a vector $x_{..t}^{(f)}$, and then conjugated with the intervals $[\tau]$ and fed as an input to the recurrent layer structure.

### 2.2. Recurrent net

Recurrent neural networks are powerful tool that can approximate functions and obtain useful data out of time-series sets. Vanilla recurrent networks have a hard time dealing with the task of processing long sets of time-series data however. For these applications two new models were introduced, the Long Short-term Memory (LSTM) and Gated Recurrent Unit (GRU).

In order to reduce the complexity level of the network, it was decided to use GRU, due to the fact that this model shows a similar level of effectiveness to LSTM on a large number of tasks but possesses a simpler structure.

Within the scope of this work a sequential GRU structure with two layers was used. In comparison to a single-layer model, sequential models allow to effectively increase the capacity of the model. In comparison to dual-directional GRU, that contains two time-flows from beginning to end, and back, a sequential GRU can operate sequentially, which allows it to process the dataflow faster.

In contrast to that a dual-directional GRU can only operate when all of the time-series sequence is already known, which is barely applicable to such applications as tracking for example.

In order to regulate the relations between the layers of the GRU, a dropout procedure was applied, as well as a recurrent batch normalization in order to lower the internal covariance shift between the time intervals. The inputs $\{x_t^{(c)}\}$ for $t = 1, \dots, T$ from the previous convolutional net are fed as an input to the sequential GRU, that generates $\{x_t^{(r)}\}$ for $t = 1, \dots, T$ as outputs, that are fed to the final output layer.

### 2.3. Output layer

The output value of the recurrent net is a set of vectors $\left\{x_t^{(r)}\right\}$ for $\mathrm{t} = 1, \dots, T$. For a regression problem, since the value of each element of the vector $x_t^{(r)}$ is within the range of $\pm 1$, $x_t^{(r)}$ encodes the physical output of the interval $t$. In the output layer, we want to examine the set $W_{out}$ with offset $b_{out}$ to convert $\mathrm{x}_t^{(r)}$ to $\hat{y}_t$, where $\hat{y}_t = W_{out} \cdot x_t^{(r)} + b_{out}$. Thus, the output layer is a fully connected layer on top of each bin with common parameters $W_{out}$ and $b_{out}$.

For the classification problem, $x_t^{(r)}$ is a vector of features on the interval $t$. On the output layer, you first need to convert $\left\{x_t^{(r)}\right\}$ to a feature vector of fixed length for later processing.

One way is to average the features over time. Also, more complex methods can be applied to generate final features.

One way or another, the features are averaged over time to obtain the feature $x^{(r)} = \frac{\sum_{t=1}^{T} x_t^{(r)}}{T}$. Next, we feed $x^{(r)}$ to the input of the activation layer with the softmax function to get the probability $\hat{y}$ of the predicted class.

### 2.4. Framework customization for specific tasks

In general, the following steps are required to apply the framework to a specific task:

1) determine the number of inputs $K$. Perform preliminary transformation of inputs into a set of tensors $X = \left\{X^{(k)}\right\}$ as input data of the framework;

2) determine the type of problem, whether it is a classification or regression problem;

3) develop a specific loss function for this task or use a standard function (the mean square error for regression problems or cross-entropy for classification problems).

If you use the default framework, you need to determine the inputs $K$, Perform preliminary trans-

formation of the measurements of the sensors, and determine the problem.

Pre-processing stage consists in the fact that the data from the sensors are formed into blocks, after which the Fourier transform is applied to each block. For particular sensor, the results of transformation are combined into tensors $X^{(k)}$ of dimension $d^{(k)} \times 2f \times T$, where $d^{(k)}$ is the dimension of measurements, $f$ is the dimension of intervals, and $T$ is number of intervals.

The $K$ parameter determines not the physical number of sensors, but the number of sensory modalities. If there are more than one sensor of the same modality (for example, a couple motion sensors or altitude sensors), we consider them as one multidimensional sensor.

## 3. EXPERIMENTAL RESULTS

For an experimental evaluation of the proposed framework, let us consider its work on two tasks: recognition of user actions (HHAR, Heterogeneous Human Activity Recognition) and user identification by analyzing his movement.

As a dataset for these two tasks, the dataset presented in [28] was used. The dataset contains the measurements of two motion sensors (accelerometer and gyroscope). The readings were fixed while the subjects performed certain actions without a strictly agreed order in advance. A smart watch and a smartphone were used to obtain sensor measurements. The dataset contains data from 9 users who performed 6 actions (cycling, sitting, standing, walking, climbing stairs, descending stairs), as well as data taken from 8 mobile devices.

For each task, the inputs to the proposed framework are data from the accelerometer and gyroscope. For the task of recognizing user actions, the actions performed by the subjects were taken as categories, and for the identification task, the subjects themselves were selected as categories.

The measurements were split into 5 second intervals. Further, each interval was additionally divided into intervals of τ 0.25 seconds long. Next, the frequency response of the sensors was calculated for each time interval and the data from the time intervals were combined into tensors for further use as input to the framework.

A mobile device with a Qualcomm Snapdragon 820 CPU and Edison Compute Module was chosen as the hardware platforms for the experiments. The framework was trained on a stationary computer with a GPU. The trained networks were run on hardware platforms only using the power of the central processor, without the use of GPUs and DSPs.

Three modifications of the proposed framework were used as analogs for two tasks. Also, for each individual problem, additional algorithms based on other classifiers were used. Let us give a brief description of the framework options that were used as a comparison:

1) a framework with a single GRU layer (framework-1) with a higher dimension while observing the number of parameters. This comparison will allow you to check the increase in the efficiency of the framework when adding another GRU layer;

2) a framework without a particular convolutional net for each input (framework-2). Instead, combination of input is used, and for each interval its own single matrix is created, which is fed directly to the input of the fusion net;

3) framework without fusion net (framework-3). In this version of the proposed framework, the output of each particular convolutional network is transformed into a vector, after which the vectors are combined and the resulting vector is fed to the input of the recurrent network.

For the task of recognizing user actions, the following analogs were used:

1) the features proposed in [29] and [30] were taken and a random forest was used as a classifier (ActivityRec-RF);

2) the same features as in the previous analogue were used, but the vector reference machine (ActivityRec-SVM) was used as a classifier;

3) this approach was proposed in [31] and consists in using a stack of limited Boltzmann machines (ActivityRec-BM1);

4) the approach was proposed in [32] and consists in using a limited Boltzmann machine for each sensor input, after which another limited Bolman machine is used to merge the results (ActivityRec-BM2).

For the task of identifying users, the following analogs were taken:

1) identification of a person by gait was proposed in [33] and consists in extracting gait patterns from the data and using a reference machine of vectors for comparison with a template (USER-ID-SVM1);


ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

2) another approach based on gait identification was proposed in [34]. The approach is based on the extraction of gait with the help of a convolutional network and matching with a model using a SVM and validation of the results using the Wald test (USER-ID-SVM2).

### 3.1. HHAR task

For the task of recognizing human actions, an assessment for individual objects was used (that is, the full amount of data of one person was used as a test sample). Three metrics were used for the assessment: accuracy, macro-averaged F1 score, and micro-averaged F1 score with a 95 % confidence interval. The results are shown in Fig. 2.

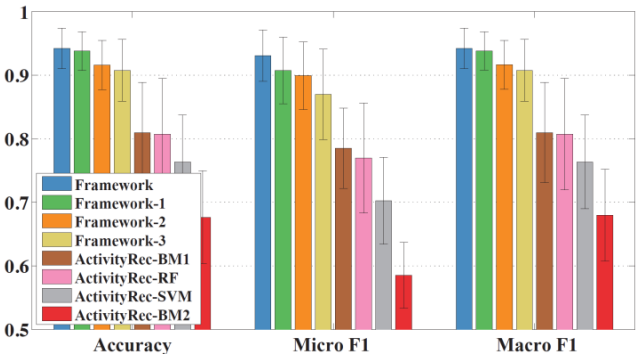



*Fig. 2.* **HHAR task evaluation**
***Source:*** **compiled by the authors**

As can be seen from the obtained metric values, the proposed framework and its variants have surpassed other methods. It should be noted that the ActivityRec-RF and ActivityRec-SVM approaches use hand-selected features, while the proposed framework allows you to automatically extract more flexible features that are better generalized to users that are not in the training sample.

Compared to the models ActivityRec-BM1 and ActivityRec-BM2, which are also based on deep learning, the proposed model generates better and more flexible features by using the time factor, as well as taking into account the relationship between multiple sensors.

Compared to the framework options, the standard version achieves better performance: accuracy 0.938 ± 0.034, macro F1 0.936 ± 0.038 and micro F1 0.938 ± 0.034.

Consider the error matrix for the proposed deep learning framework (Fig. 3).

It can be seen that the prediction of the activities “Sitting” and “Standing” gives the largest error. This is due to the fact that the sensors used give similar measurements when the subject is in one of these states. Also, there is a small error when trying to distinguish between the states “Climbing a ladder” and “Descending a ladder”.

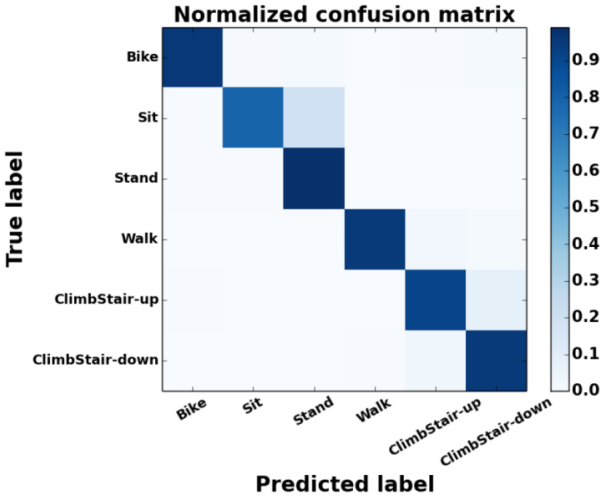



*Fig. 3.* **Confusion matrix for HHAR task**
***Source:*** **compiled by the authors**

### 3.2. User identification task

This task is designed to identify the user by analyzing his movement. A 10-block cross-validation was used to evaluate the models.

Figure 4 shows the evaluation of the models' operation at 5 time intervals at an interval of 1.25 seconds.

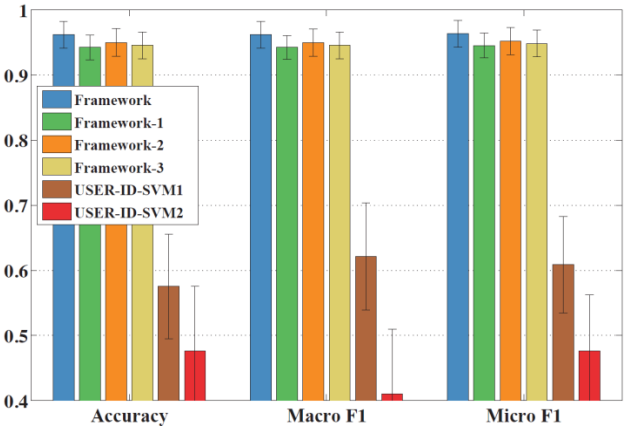



*Fig. 4.* **Task evaluation for 5 time intervals**
***Source:*** **compiled by the authors**

Figure 5 shows the evaluation of the models' performance at 20 time intervals at an interval of 5 seconds.

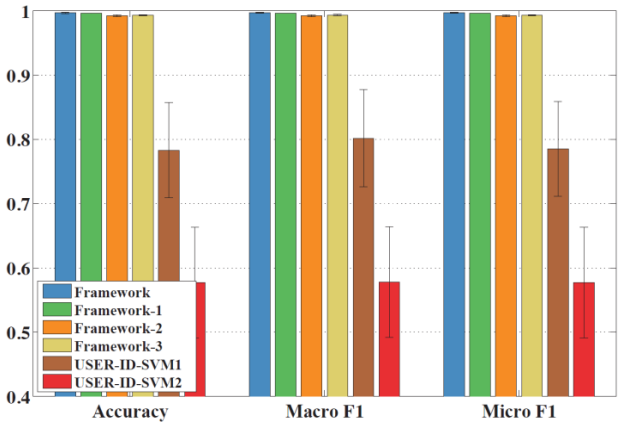



*Fig. 5.* **Task evaluation for 20 time intervals**

***Source:*** **compiled by the authors**

Three metrics were used for the assessment: accuracy, macro-averaged F1 score, and micro-averaged F1 score with a 95 % confidence interval.

The proposed framework and its three variants are superior to analogous models. Compared to the USER-ID-SVM1 approach, which is to extract patterns and then match against a pattern, the proposed framework can automatically extract features from the data, which work well not only when the identified person walks, but also when performing other actions.

If we compare the proposed framework with the USER-ID-SVM2 approach, which consists in extracting templates and then using a neural network to extract features, then the proposed framework solves the problem in a fully automatic mode.

The manual data processing process is completely removed, and the use of local, global and temporal relations gives the best result.

Despite the fact that in the case of using 5 time intervals, all variants of the proposed framework show similar efficiency, at 20 time intervals the standard version achieves better efficiency: accuracy $0.994 \pm 0.006$, macro F1 $0.994 \pm 0.006$ and micro F1 $0.996 \pm 0.006$.

Consider the error matrix for the user identification problem (Fig. 6).

Based on the results obtained, we can conclude that the proposed framework shows excellent results in this task.

On average, two misclassifications occurred in each iteration of testing.

Let us compare the work of the proposed framework with its three variants based on the number of used time intervals (Fig. 7). Let us evaluate the accuracy of the framework and its variants when changing the number of time intervals from 5 to 20, which corresponds to a duration from 1 to 5 seconds.

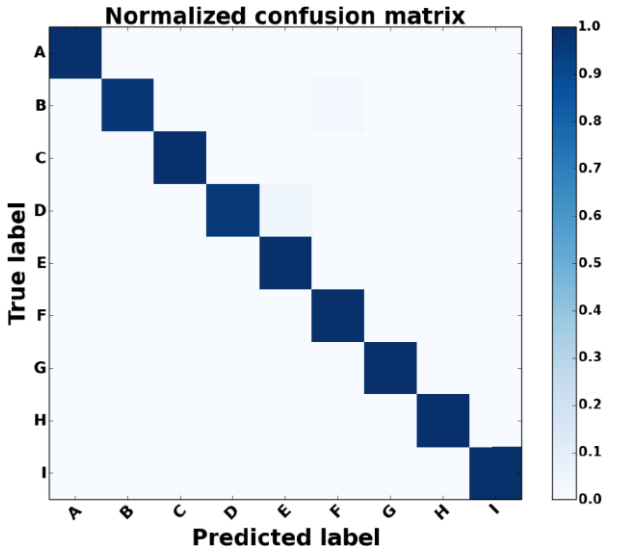



*Fig. 6.* **Confusion matrix for User ID task**

***Source:*** **compiled by the authors**

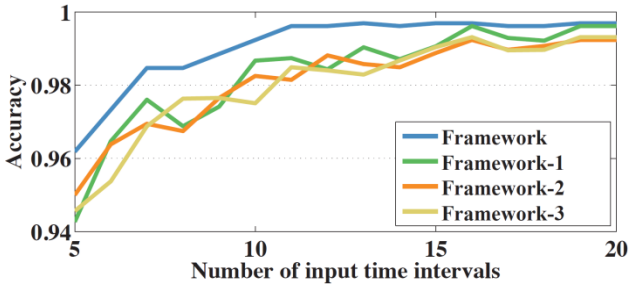



*Fig. 7.* **Accuracy over time intervals for User ID task**

***Source:*** **compiled by the authors**

### 3.3. Power consumption and inference time

Consider the energy consumption and inference time of the framework and analog models. Assuming that for a mobile device, computing task time and power consumption can be influenced by many other factors, measurements were performed on the Edison Compute Module using an external energy meter.

Since the task of recognizing human activity and the task of identifying a person is not periodic, the power consumption was measured at the moment of inference of the models.

Fig. 8 and Fig. 9 show energy consumption and inference time for the task of recognizing human activity.


ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

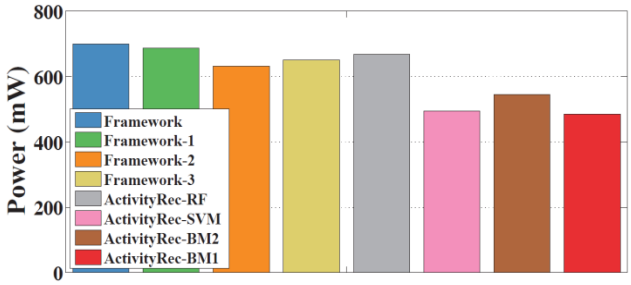



*Fig. 8.* **Energy consumption for HHAR task**

***Source:*** **compiled by the authors**

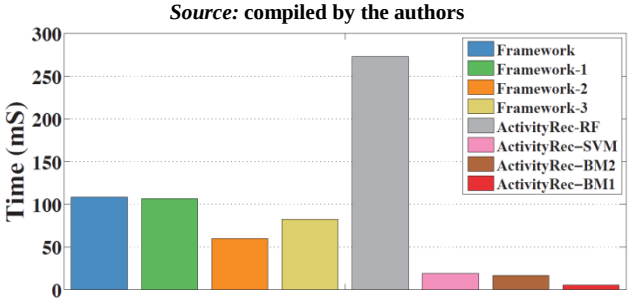



*Fig. 9.* **Inference time for HHAR task**

***Source:*** **compiled by the authors**

Fig. 10 and Fig. 11 show energy consumption and inference time for the task of identifying a person by his movement.

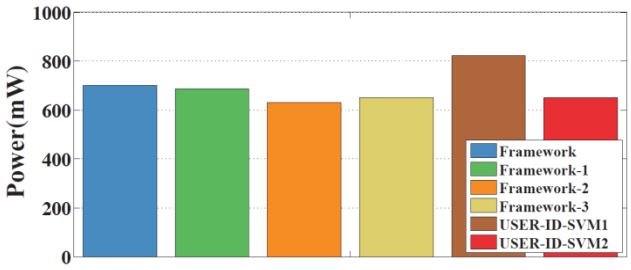



*Fig. 10.* **Energy consumption for User ID task**

***Source:*** **compiled by the authors**

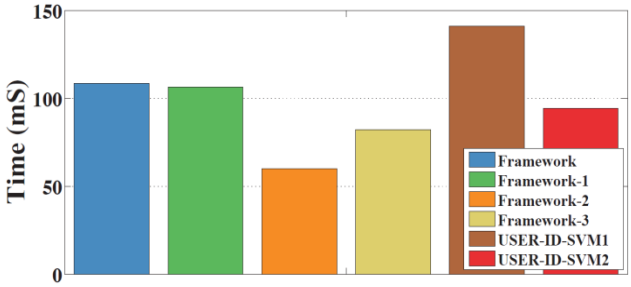



*Fig. 11.* **Inference time for User ID task**

***Source:*** **compiled by the authors**

For the task of recognizing human actions, the proposed framework and its variants consume a moderate amount of energy and cope with the task quickly enough.

It is noteworthy that the ActivityRec-RF approach, which is based on the random forest algorithm, shows a fairly long runtime.

This can be argued by the fact that this method is an ensemble bagging method, which consists in combining deep decision trees.

For the problem of human identification, all methods show approximately the same energy consumption and inference time, with the exception of the USER-ID-SVM1 method, which includes a preprocessing stage and a relatively large convolutional network, which leads to more energy consumption and slower operation.

## CONCLUSIONS

In this paper, we propose a distributed framework that uses deep learning methods to process data generated by different types of sensors in transducer networks. The proposed framework allows solving classification and regression problems. The architecture of the framework consists of particular convolutional net for processing data from sensors of the same type, a fusion convolutional net, and a recurrent net that allows you to extract features from time series that are formed when data comes from sensors.

For experimental verification of the developed framework, the task of recognizing human actions and the task of identifying a person by movement were taken. Variations of the proposed framework with a modified architecture, as well as third-party approaches based on various machine learning methods, were used as analog methods.

For the task of recognizing human actions, the proposed framework surpassed the analogous methods by about 8 %. According to the metric of accuracy, the proposed framework demonstrates the value of 0.938 ± 0.034, according to the metric "macro F1" the value is 0.936 ± 0.048 and according to the metric “micro F1” the value is 0.938 ± 0.034.

For the task of identifying a person by his movement, the proposed framework surpassed the results of analogues by about 13 %. In terms of accuracy metrics, “macro F1” and “micro F1”, the proposed framework shows a value of 0.994 ± 0.006.

Also, a study was carried out on the energy consumption and inference time of the proposed framework. The proposed framework and its variants consume a moderate amount of energy and cope with the task quickly enough.

Possible directions for further work are further optimization of the structure of the fusion convolutional net, as well as setting up a recurrent net.

## REFERENCES

1. BSRIA. “The Smart Building Market in Asia will exceed $1036bn by 2020”. – Available from: https://www.bsria.co.uk/news/article/the-smart-building-market-in-asia-will-exceed-$1,036bn-by-2020. – [Accessed Feb 2021].

2. So, A. T. P. & Wong, K. C. “On the Quantitative Assessment of Intelligent Buildings”. *Facilities*. 2002; Vol. 20 No. 5/6: 208–216. DOI: https://doi.org/10.1108/02632770210435206.

3. Azevedo Guedes, A. L., Carvalho Alvarenga, J., Dos Santos Sgarbi Goulart, M., Rodriguez y Rodriguez, M. V. & Pereira Soares, C. A. “Smart Cities: The Main Drivers for Increasing the Intelligence of Cities”. *Sustainability*. 2018; Vol. 10. 3121. DOI: https://doi.org/10.3390/su10093121.

4. Antoshchuk, S. G., Shamin, I. I., Sharma, B. S. & Shcherbakova, G. Yu. “A Multi-Objective Optimization Problems of Clustering Protocols for Wireless Sensor Networks Using Meta-Heuristic Techniques”. *Herald of Advanced Information Technology*. *Publ. Nauka i Tekhnika*. Odesa: Ukraine. 2018; Vol. 1 No. 1: 21–27. DOI: https://doi.org/ 10.15276/hait.01.2018.2.

5. Lima, E. G., Chinelli, C. K., Guedes, A. L. A., Vazquez, E. G., Hammad, A. W. A., Haddad, A. N. & Soares, C. A. P. “Smart and Sustainable Cities: The Main Guidelines of City Statute for Increasing the Intelligence of Brazilian Cities”. *Sustainability*. 2020; Vol. 12 No. 3. 1025. DOI: https://doi.org/ 10.3390/su12031025.

6. McGlinn, K., O’Neill, E., Gibney, A., O’Sullivan, D. & Lewis, D. “Simcon: A Tool to Support Rapid Evaluation of Smart Building Application Design Using Context Simulation and Virtual Reality”. *Journal of Universal Computer Science*. 2010; Vol. 16 No. 15: 1992–2018. DOI: https://doi.org/10.3217/jucs-016-15-1992.

7. Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A. & Buyya, R. “Big Data Computing and Clouds: Trends and Future Directions”. *Journal of Parallel and Distributed Computing*. 2015; Vol. 79-80: 3–15. DOI: https://doi.org/10.1016/j.jpdc.2014.08.003.

8. Stojmenovic, I. & Wen, S. “The Fog Computing Paradigm: Scenarios and Security Issues”, *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*. 2014; Vol. 2: 1– 8. DOI: https://doi.org/10.15439/2014F503.

9. Dastjerdi, A. V., Gupta, H., Calheiros, R. N., Ghosh, S. K. & Buyya, R. “Fog Computing: Principles, Architectures and Applications”. *Internet of Things, Principles and Paradigms.* 2016: p. 61–75. DOI: https://doi.org/10.1016/B978-0-12-805395-9.00004-6.

10. Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M., A. Choudhury, N. & Kumar, V. “Security and Privacy in Fog Computing: Challenges”. *IEEE Access*. 2017; Vol. 5: 19293–19304. DOI: https://doi.org/10.1109/ACCESS.2017.2749422.

11. Khan, S., Parkinson, S. & Qin, Y. “Fog Computing Security: A Review of Current Applications and Security Solutions”. *Journal of Cloud Computing*. 2017; Vol. 6 No. 19. DOI: https://doi.org/ 10.1186/s13677-017-0090-3.

12. Bonomi, F., Milito, R., Zhu, J. & Addepalli, S. “Fog Computing and its Role in the Internet of Things”. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. 2012. p. 13–16. DOI: https://doi.org/10.1145/2342509.2342513.

13. Sareen, P. & Kumar, P. “The Fog Computing Paradigm”. *International Journal of Engineering & Technology*. 2016; Vol. 4 No. 8: 55–60.

14. Vaquero, L. M. & Rodero-Merino, L. “Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing”. *ACM SIGCOMM Computer Communication Review*. 2014; Vol. 44 No. 5: 27–32. DOI: https://doi.org/10.1145/2677046.2677052.

15. Saharan, K. P. & Kumar, A. “Fog in Comparison to Cloud: A Survey”. *International Journal of Computer Applications*. 2015; Vol. 122 No. 3: 10–12. DOI: https://doi.org/10.5120/21679-4773.

16. Antoshchuk, S. G., Lobachev, I. M. & Maleryk, R. P. “Method of the Sensor Network Resources Allocation in the Conditions of Edge Computing”. *Herald of Advanced Information Technology*. *Publ. Nauka i Tekhnika*. Odesa: Ukraine. 2018; Vol. 1 No. 1: 28–35. DOI: https://doi.org/10.15276/hait.01.2018.3.


ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

17. Shi, W., Cao, J., Zhang, Q., Li, Y. & Xu, L. “Edge Computing: Vision and Challenges”. *IEEE Internet Things Journal*. 2016; Vol. 3 No. 5: 637–646. DOI: https://doi.org/10.1109/JIOT.2016.2579198.

18. Mahmud, R., Kotagiri, R. & Buyya, R. “Fog Computing: A Taxonomy, Survey and Future Directions”. *Internet of Everything*. 2018: 103–130. DOI: https://doi.org/10.1007/978-981-10-5861-5_5.

19. Kamath, G., Agnihotri P., Valero, M., Sarker, K. & Song, W.-Z. “Pushing Analytics to the Edge”. *IEEE Global Communications Conference (GLOBECOM)*. 2016. p. 1–6. DOI: https://doi.org/10.1109/GLOCOM.2016.7842181.

20. Azimi, I., Anzanpour, A., Rahmani, A. M., Liljeberg, P. & Salakoski T. “Medical Warning System Based on Internet of Things Using Fog Computing”. *International Workshop on Big Data and Information Security (IWBIS)*. 2016. p. 19–24. DOI: https://doi.org/10.1109/IWBIS.2016.7872884.

21. Lu, L., Xu, L., Xu, B., Li, G. & Cai, H. “Fog Computing Approach for Music Cognition System Based on Machine Learning Algorithm”. *IEEE Transactions on Computational Social Systems*. 2018; Vol. 5 No. 4: 1142–1151. DOI: https://doi.org/10.1109/TCSS.2018.2871694.

22. Li, L., Ota, K. & Dong M. “Deep Learning for Smart Industry: Efficient Manufacture Inspection System with Fog Computing”. *IEEE Transactions on Industrial Informatics*. 2018; Vol. 14 No. 10: 4665–4673. DOI: https://doi.org/10.1109/TII.2018.2842821.

23. Sheremet, O. I., Korobov, O. Ye., Sadovoi, O. V. & Sokhina, Y. V. “Intelligent System Based on a Convolutional Neural Network for Identifying People Without Breathing Masks”. *Applied Aspects of Information Technology*. *Publ. Nauka i Tekhnika*. Odesa: Ukraine. 2020; Vol.3 No.3: 133–144. DOI: https://doi.org/10.15276/aait.03.2020.2.

24. Pérez, J. L., Gutierrez-Torre, A., Berral, J. L. & Carrera D. “A Resilient and Distributed Near Real-time Traffic Forecasting Application for Fog Computing Environments”. *Future Generation Computer Systems*. 2018; Vol. 87: 198–212. DOI: https://doi.org/10.1016/j.future.2018.05.013.

25. Kimovski, D., Ijaz, H., Saurabh, N. & Prodan R. “Adaptive Nature-inspired Fog Architecture”, *IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*. 2018. p. 1–8. DOI: https://doi.org/ 10.1109/CFEC.2018.8358723.

26. Lu, J., Xiang, X., Shen, D., Chen, G., Chen, N., Blasch, E., Pham K. & Chen, Y. “Artificial Intelligence Based Directional Mesh Network Design for Spectrum Efficiency”. *IEEE Aerospace Conference*. 2018. p. 1–9. DOI: https://doi.org/ 10.1109/AERO.2018.8396558.

27. Costa, R., Jardim-Goncalves, R., Figueiras, P., Forcolin, M., Jermol, M. & Stevens R. ‘‘Smart Cargo for Multimodal Freight Transport: When ‘Cloud’ Becomes ‘Fog’’. *IFAC-PapersOnLine*. 2016; Vol. 49 No. 12: 121–126. DOI: https://doi.org/10.1016/j.ifacol.2016.07.561.

28. Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjaergaard M. B., Dey, A., Sonne, T. & Jensen M. M. “Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition”. *SenSys '15: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. 2015. p. 127–140. DOI: https://doi.org/10.1145/2809695.2809718.

29. Figo, D., Diniz, P. C., Ferreira, D. R. & Cardoso, J. M. “Preprocessing Techniques for Context Recognition from Accelerometer Data”. *Personal and Ubiquitous Computing*. 2010; Vol. 14: 645–662. DOI: https://doi.org/10.1007/s00779-010-0293-9.

30. Hammerla, N. Y., Kirkham, R., Andras, P. & Ploetz, T. “On Preserving Statistical Characteristics of Accelerometry Data Using Their Empirical Cumulative Distribution”. *ISWC '13: Proceedings of the 2013 International Symposium on Wearable Computers*. 2013. p. 65–68. DOI: https://doi.org/ 10.1145/2493988.2494353.

31. Bhattacharya, S. & Lane, N. D. “From Smart to Deep: Robust Activity Recognition on Smartwatches Using Deep Learning”. *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 2016. p. 1–6. DOI: https://doi.org/ 10.1109/PERCOMW.2016.7457169.

32. Radu, V., Lane, N. D., Bhattacharya, S., Mascolo, C., Marina, M. K. & Kawsar, F. “Towards Multimodal Deep Learning for Activity Recognition on Mobile Devices”. *UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. 2016. p. 185–188. DOI: https://doi.org/10.1145/2968219.2971461.

33. Thang, H. M., Viet, V. Q., Thuc, N. D. & Choi, D. “Gait Identification Using Accelerometer on Mobile Ohone”. *International Conference on Control, Automation and Information Sciences (ICCAIS)*. 2012. p. 344–348. DOI: https://doi.org/10.1109/ICCAIS.2012.6466615.

34. Gadaleta, M. & Rossi, M. “Idnet: Smartphone-based Gait Recognition with Convolutional Neural Networks”. *Pattern Recognition*. 2018; Vol. 74: 25–37. DOI: https://doi.org/10.1016/j.patcog.2017.09.005.

**Conflicts of Interest:** the authors declare no conflict of interest

Received 18.01.2021
Received after revision 24.02.2021
Accepted 15.03.2021

**DOI: https://doi.org/10.15276/aait.02.2021.1**
**УДК 004.93.1**

# РОЗПОДІЛЕНИЙ ФРЕЙМВОРК, ЗАСНОВАНИЙ НА ГЛИБИННОМУ НАВЧАННІ ДЛЯ ТРАНСДЮСЕРНИХ МЕРЕЖ РОЗУМНОГО БУДИНКУ

**Іван Михайлович Лобачев**[1)]
ORCID: https://orcid.org/0000-0002-4859-304X; lobachev.i.m@gmail.com. Scopus ID: 57192559239
**Світлана Григорівна Антощук** [1)]
ORCID: https://orcid.org/0000-0002-9346-145X; asg@opu.ua. Scopus ID: 8393582500
**Микола Анатолійович Годовиченко**[1)]
ORCID: https://orcid.org/0000-0001-5422-3048; nick.godov@gmail.com. Scopus ID: 57188700773
[1)] Одеський національний політехнічний університет, пр. Шевченка, 1. Одеса, 65044, Україна

## АНОТАЦІЯ

Дана робота присвячена розробці розподіленого фреймворка, заснованого на глибинному навчанні для обробки даних з різних сенсорів, які генеруються трансдюсерними мережами, які застосовуються в області розумних будинків. Запропонований фреймворк дозволяє обробляти дані, які надходять з сенсорів різного типу для вирішення задач класифікації і регресії. Архітектура фреймворка складається з декількох складових: індивідуальних згорткових мереж, які обробляють вхід з сенсорів одного типу, єдиної згорткової мережі злиття, яка обробляє безліч виходів індивідуальних згорткових мереж. Далі, результат роботи єдиної згорткової мережі злиття подається на вхід рекурентної мережі, яка дозволяє витягувати значущі ознаки з тимчасових послідовностей. Результат роботи рекурентної мережі подається на вихідний шар, який генерує вихід фреймворка, виходячи з типу завдання, що вирішується. Для експериментальної оцінки розробленого фреймворка були взяті два завдання: завдання розпізнавання дій людини і завдання ідентифікації людини по руху. Датасет містив дані двох сенсорів (акселерометра і гіроскопа), які збиралися у 9 користувачів, які виконували 6 дій. У якості апаратних платформ було використано мобільний пристрій, а також апаратний пристрій Edison Compute Module. Для порівняння результатів роботи, були використані варіації запропонованого фреймворка з різною архітектурою, а також сторонні підходи, засновані на різних методах машинного навчання, включаючи опорні машини векторів, випадковий ліс, обмежені машини Больцмана і так далі. В результаті, запропонований фреймворк, в середньому, перевершив інші алгоритми приблизно на 8 % за трьома метриками в задачі розпізнавання дій людини і виявився ефективнішим приблизно на 13% в завданні ідентифікації людини по руху. Також було виміряно споживання енергії і час роботи запропонованого фреймворка і його аналогів. Було виявлено, що запропонований фреймворк споживає помірну кількість енергії, а час роботи можна оцінити як прийнятний.

**Ключові слова**: Розумна будівля; інтернет речей; глибоке навчання; згорткова нейронна мережа; вентильний рекурентний вузол; рекурентна нейронна мережа; довга короткочасна пам’ять

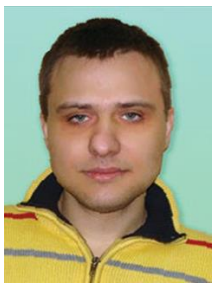
## ABOUT THE AUTHORS

**Ivan M. Lobachev,** PhD Student of Information Systems Department. Odessa National Polytechnic University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-4859-304X; lobachev.i.m@gmail.com. Scopus ID: 57192379296
***Research field*:** Deep Learning; Wireless Sensor Networks; Smart Cities; Embedded Systems; Quantum Computing; Data Mining

**Іван Михайлович Лобачев,** аспірант кафедри Інформаційних систем. Одеський національний політехнічний університет, пр. Шевченка, 1. Одеса, 65044, Україна

**Svitlana G. Antoshchuk,** Dr. Sci. (Eng), Professor, Head of Computer Systems Institute. Odessa National Polytechnic University, 1, Shevchenko Ave. Odesa, 65044, Ukraine.
ORCID: https://orcid.org/0000-0002-9346-145X; asg@opu.ua. Scopus ID: 8393582500
***Research field*:** Pattern Recognition; Deep Learning; Object Tracking; Face Recognition, Graphic Images Formation and Processing

**Світлана Григорівна Антощук,** доктор технічних наук, професор, директор Інституту Комп'ютерних Систем. Одеський національний політехнічний університет, пр. Шевченка, 1. Одеса, 65044, Україна



**Mykola Hodovychenko,** PhD (Eng), Associate Prof. of the Department of Project-based Learning in IT. Odessa National Polytechnic University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0000-0001-5422-3048; nick.godov@gmail.com. Scopus ID: 57188700773
***Research field*:** Deep Learning; Data Mining; Smart Cities; Video Processing; Motion Tracking; Project-based Learning; Patter Recognition

**Микола Анатолійович Годовиченко,** кандидат технічних наук, доцент кафедри Проектного навчання в ІТ. Одеський національний політехнічний університет, пр. Шевченка, 1. Одеса, 65044, Україна