Future Generation Computer Systems 152 (2024) 225–238

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Prediction-based data reduction with dynamic target node selection in IoT sensor networks

Bartłomiej Płaczek

Institute of Computer Science, University of Silesia, Bedzinska 39, Sosnowiec, 41-200, Poland

ARTICLE INFO

Keywords:
Internet of Things
Wireless sensor networks
Transmission reduction
Prediction
Proximity monitoring
Mobile computing

ABSTRACT

In various applications of IoT sensor networks, special attention is paid to sensor nodes that register higher or lower values of a monitored parameter compared to other nodes. Thus, methods are necessary that enable the selection of target nodes while minimizing the number of data transmissions. The reduced amount of data transmitted during target node selection contributes to energy savings and prolongs the lifetime of battery-powered sensor nodes. This paper introduces a method for reducing data transmissions in sensor networks, where target nodes are selected to perform monitoring tasks. The proposed method allows the IoT gateway to determine when the sensor nodes should report their data readings. The gateway utilizes a prediction algorithm to determine whether a new target node should be selected. In this scheme, data are transmitted only when necessary for the correct selection of the target node. Experiments were conducted for a wireless sensor network consisting of mobile devices, where the target node is the device closest to a given location. The experimental evaluation confirmed that the proposed method significantly reduces the amount of transmitted data and ensures the proper selection of the target node. It was also demonstrated that the introduced approach reduces data transmissions more effectively than the state-of-the-art prediction-based algorithms. During the experiments, the amount of data transmitted from sensor nodes to the IoT gateway was analyzed. The proposed method has reduced the amount of transmissions by 94%, while the highest transmission reduction rate achieved by the existing techniques was 81.4%.

1. Introduction

Data transmission reduction is crucial in IoT (Internet of Things) sensor networks to ensure power efficiency, optimize bandwidth, mitigate network congestion, and reduce costs [1–4]. IoT devices, especially sensor nodes, are often battery-powered and designed to operate for extended periods without manual intervention. Transmitting data consumes a significant amount of energy compared to other tasks [5]. By reducing unnecessary transmissions, the power consumption of the devices can be minimized, resulting in longer battery life and reduced maintenance efforts.

IoT sensor networks often operate in environments with limited bandwidth or constrained network resources. By reducing the amount of data transmitted, the overall network capacity can be utilized more efficiently. This is particularly important in scenarios where multiple IoT devices are communicating over the same network infrastructure [6]. Moreover, a large number of devices and excessive data transmissions can result in network congestion. This congestion leads to increased latency, reduced network efficiency, and potential data loss. The network congestion can be alleviated by reducing unnecessary transmissions and ensuring smooth, reliable data communication.

It should also be noted that transmitting data in IoT networks often incurs costs, especially when using cellular or satellite communication services [7]. The associated costs can be significantly reduced by minimizing unnecessary transmissions, making IoT deployments more economically feasible.

In the IoT sensor network under consideration, the term "target node" refers to the sensor node that currently records a higher or lower value of a monitored parameter compared to other nodes. The identification of relatively high or low sensor readings is essential in various applications of IoT sensor networks [8,9]. For instance, if a sensor network is intended to monitor local events in a large area, finding target nodes with extreme readings is helpful for determining the current event location. In the case of fire monitoring, the target sensor nodes are those with high-temperature readings [10]. Another example is a mobile network, where target nodes correspond to mobile devices that are closest to a given geographical location or a moving object [11,12]. The selected target node can activate additional sensors, such as vision-based sensors, to collect detailed information about the local event or object of interest. In IoT systems with sensors and

E-mail address: bartlomiej.placzek@us.edu.pl.

https://doi.org/10.1016/j.future.2023.11.007
Received 2 June 2023; Received in revised form 31 October 2023; Accepted 3 November 2023
Available online 7 November 2023

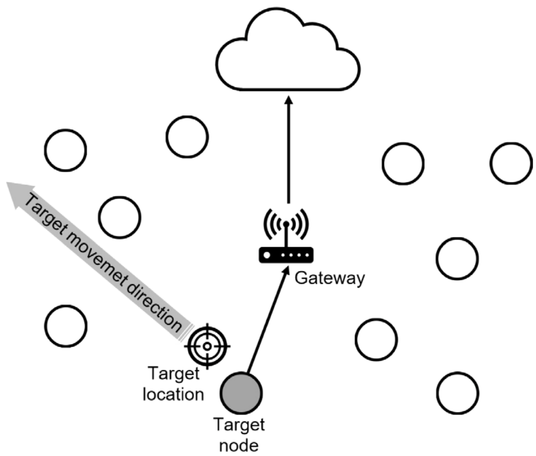

Fig. 1. Dynamic target node selection in IoT sensor network.

actuators, the selected target node can utilize an actuator to appropriately influence the monitored process or environment.

An example of target node selection in an IoT sensor network is illustrated in Fig. 1. In this scenario, the selection of the target node is based on the proximity of the sensor node to the target location (e.g., a sound source). It means that the sensor node closest to the target location is designated as the target node. Then, the target node sends the information regarding registered target parameters to a gateway. The gateway collects and analyzes the data before transmitting them to the cloud for further processing.

In many applications, the target node needs to be selected dynamically [13]. In that dynamic scenario, a given sensor node is designated as a target node for a limited period of time. Then, another target node is selected when the situation in the monitored area changes (e.g., when the monitored target moves to a different location, as shown in Fig. 1). The process of selecting the target node involves gathering data from multiple sensor nodes and transmitting it extensively to the gateway, which is responsible for making the selection. Therefore, in dynamic environments where the measured parameters or node positions change continuously, the target node is frequently selected, resulting in a large number of transmissions.

Prediction techniques can effectively minimize the amount of data transmitted and improve the overall efficiency of the sensor network [14]. Prediction algorithms enable the sensor nodes or devices at the network edge to determine the need for data transmission. It means that the transmission will only be initiated when the predictions indicate that the registered data can be useful for selecting a new target node. For instance, let us consider the example of dynamic target node selection shown in Fig. 1. When the target moves away from the current target node in an unknown direction, the new target node is selected based on data collected from all adjacent sensor nodes. Suppose the direction of target movement is predicted, as shown by the gray arrow in Fig. 1. In that case, the gateway can collect the necessary data for selecting a new target node from a subset of the sensor nodes that are close to the expected trajectory of the target. This simple example demonstrates that the amount of transmitted data can be effectively reduced by predicting a smaller set of expected future states.

In the literature, prediction-based transmission reduction schemes have been proposed to collect data that accurately reflect the actual sensor readings. The main objective of these methods is to predict the most probable future sensor readings and ensure that the difference between collected data and real measurements is below a predetermined threshold. According to the best author's knowledge, there is a lack of methods intended to minimize the data transmission during target node selection when the objective is to determine the order relation between data readings from different sensor nodes. The existing methods are ineffective in the considered scenario because they aim to continuously provide accurate estimates of the actual measurements. For target node selection, high accuracy of the collected data is required only when the readings of different sensor nodes are similar. Moreover, the prediction-based methods currently available assume that the sensor node must collect measurements periodically and then compare them with the prediction results. Thus, the amount of sampled data is not reduced, which results in significant energy consumption, particularly when active sensors are utilized. The state-of-the-art transmission reduction algorithms were designed for implementation in sensor networks that lack external storage and computing resources. Little research has focused on allocating the necessary data processing operations between edge devices and cloud servers.

The objective of the research reported in this paper is to improve the efficiency of prediction techniques in reducing the amount of sensed and transmitted data for selecting target nodes in IoT sensor networks. This study involves analyzing the advantages and disadvantages of various prediction algorithms and developing a new approach. The research questions are as follows:

- Can the transmission reduction rate of prediction-based methods be improved by forecasting possible changes in sensor readings instead of the most probable values?
- Is it possible to effectively reduce the amount of both transmitted and sampled data by implementing a prediction model on the network edge?
- How can computing resources in the cloud be utilized to store and process historical datasets for prediction purposes?

This paper introduces a new prediction method that allows us to reduce the data transmissions necessary for selecting the target node. The proposed method utilizes a novel algorithm to predict possible changes in sensor readings. On this basis, we can estimate the duration during which the sensor readings for each sensor node will not change significantly. By considering predictions for all sensor nodes, the gateway determines when a given sensor node will be eligible for potential selection as a new target node and when the node needs to transmit its data readings. This way, the gateway creates a transmission schedule for sensor nodes, ensuring that they only sample and send the necessary data to select the target node.

The originality of this work lies in the prediction-based scheduling of data sampling and transmission operations, which is based on the predicted increase and decrease rates of sensor readings. This approach allows the sensor nodes to avoid unnecessary data transmissions and sampling. Therefore, the energy consumption of sensor nodes can be reduced more effectively compared to existing transmission reduction methods that require periodic collection of data samples by sensor nodes. The presented method does not rely on any predetermined accuracy threshold or error bounds. A novel aspect of this approach is the prediction algorithm, which utilizes an instance-based learning technique to assess potential future change rates of sensor readings.

In the following section, we delve into the existing literature on the topic and justify the contribution of this paper. This is followed by a detailed explanation of the proposed prediction method and data transmission algorithms in Section 3. Subsequently, Section 4 presents the experimental results and highlights the advantages of the proposed approach compared to state-of-the-art methods. Finally, Section 5 concludes the paper by summarizing the main findings, discussing their significance, and suggesting directions for future research.

2. Related works

Various sensor node selection methods have been proposed in the literature to address specific application requirements and different performance objectives of IoT sensor networks [11,15–19]. These methods require collecting information about the current values of parameters registered by multiple sensor nodes in order to select the target node.

Thus, extensive data transmissions from a number of sensor nodes are performed when making a decision about selecting a new target node. When the process monitored by the sensor network undergoes continuous dynamic changes, frequent selection of the new target node is necessary, and high transmission rates are required. In this paper, a method is presented to reduce data transmission by using predictions instead of actual sensor readings to determine the necessity of updating the target node.

Usually, it is assumed that the sensor nodes make measurements and transmit data readings at constant, predetermined time intervals when collecting the necessary information to select a target node. The most straightforward approach to reducing the data burden in the communication network is to increase the interval between successive measurements and data transfers. This approach can be implemented using adaptive sampling algorithms [20]. However, the disadvantage of such a solution is that essential changes in the measured parameter can remain unregistered if they occur for a short duration and the time interval between measurements is relatively long [21]. Thus, this method of data collection can lead to an incorrect selection of the target node.

The above problem was addressed by introducing the send-on-delta data transmission strategy [22]. According to the send-on-delta concept, each sensor node performs measurements at short, constant time intervals. The data transmission is triggered if the measurement result indicates a significant change compared to the previously transmitted value. Namely, the sensor node transmits its current data reading if the difference between this reading and the previously reported value is above a given threshold [23].

The send-on-delta method is not effective in reducing data transmissions when frequent signal changes occur. Let us consider a situation where a sensor reading is continuously increasing. An example could be a temperature measurement taken during a summer morning or an increase in road traffic intensity during rush hours. In such a case, according to the send-on-delta approach, data transmission has to be performed periodically, even if the increase of the monitored parameter is expected and can be easily predicted. Observations of this kind have motivated the development of the dual prediction scheme [24], in which both sensor nodes and the gateway predict the future changes of the monitored parameters using the same model of the monitored process. Transmissions of sensor readings are avoided as long as the predictions are consistent with the actual measurements [25].

When using the dual prediction scheme, the sensor node must sample data and make a prediction at each time step of the sensor network operation. Then, the difference between the sampled data and prediction results is evaluated. When the difference is lower than a predetermined threshold, the sensor node skips data transmission. The gateway substitutes the missing sensor reading with its own prediction result [26].

The dual prediction mechanism has found versatile applications in IoT systems. In [24], it was demonstrated that the choice of prediction algorithm heavily influences the extent of data transmission reduction in an IoT system. The authors conducted a comparative analysis of various prediction algorithms, evaluating their accuracy, delay, and percentage reduction in transmission. Moreover, they proposed the utilization of neural networks and long short-term memory networks for dual prediction schemes with real-time model training. The dual prediction technique was implemented to minimize data transmission between IoT sensor nodes and a medical server [27]. This implementation involves sending sensor readings to the server only if predictions deviate from the readings or if the data falls within a predefined interval. Experimental findings have confirmed the effectiveness of this method in reducing energy consumption, extending network lifespan, and ensuring the required levels of reliability, throughput, and end-to-end delay.

In [28], the researchers integrated a dual prediction scheme with a deep learning approach. This scheme utilizes a long short-term memory (LSTM) network model to decrease the frequency of data transmission among interconnected IoT devices. The predictive model is dynamically updated based on a defined set of tracking parameters that monitor its performance throughout the deployment, adapting to shifts in data characteristics over time. The algorithm introduced for updates guarantees the consistency of models deployed at both sensor nodes and the gateway.

The concept of dual prediction was also studied in [29]. The authors have proposed a window-based time series forecasting technique to apply this concept to resource-constrained IoT sensor nodes. According to this technique, the next data sample is predicted by adding the average of the differences between the adjacent previous samples and the current sample. The prediction is calculated based on samples from a time window of a specified size.

A normalized quantile regression was introduced for the dual prediction scheme in [30]. The fundamental idea behind this approach is to combine two autonomous prediction strategies based on quantile regression and cosine distance. The authors argue that quantile regression is effective in cases of data outliers, heteroskedasticity, and high skewness. On the other hand, the cosine distance method is suitable for dealing with data that have frequent fluctuations. Thus, these two methods were integrated to enhance the quality and reduce the uncertainty of predictions.

In [31], a predictive technique for content-based search of sensor nodes in IoT systems was presented. The study showed that the relevant sensor nodes can be swiftly found by forecasting the current output of sensors. The system used a dual prediction scheme to perform real-time updates of prediction models. This strategy has enabled the prediction model to maintain its effectiveness for long-range forecasting in the ever-changing IoT environment.

An alternative technique, based on the dual prediction scheme, was presented in [32] to tackle problems caused by unreliable sensor data. This approach encompasses two key phases: data reduction and data prediction. In the data reduction phase, the primary aim is to minimize the frequency of transmission and eliminate any erroneous sensor readings. The erroneous data points are discarded and replaced with estimated values to ensure data integrity. In the data prediction phase, the sensor readings that were not transmitted are forecasted using the Kalman filter.

The method outlined in [33] involves data prediction and compression. In this scheme, predictions are made using linear regression. Operations are organized into periods. Data readings acquired by a sensor node in the first period serve as the basis for predicting sensor readings in the subsequent period. The sensor readings from the first period are consistently compressed and transmitted to the gateway. After collecting new readings, the similarity measure is computed between the predicted data and the new sensor readings. If the calculated similarity score meets or exceeds a predefined threshold, the new data will not be sent to the gateway.

Another approach, which uses linear regression, was introduced in [34]. Here, two stages are executed at every step of data collection. During the first stage, the coefficients of the linear regression model are calculated based on recent sensor readings and then communicated to the gateway. In the second stage, the gateway predicts the data that was not transmitted in the first stage. If the prediction error exceeds a predefined threshold, the first stage is repeated with new sensor readings.

The transmission reduction schemes discussed above do not allow the sensor nodes to save energy by remaining in the sleep state for longer periods. The sensor nodes need to be awakened at regular intervals in order to activate the sensor and perform the measurement. Thus, these methods can only reduce the number of transmitted data, while the amount of data sensed and processed by sensor nodes remains unchanged. This drawback is significant, for example, in the case of nodes with energy-expensive active sensors or tiny nodes with limited computational resources.

Simultaneous reduction of sensed, processed, and transmitted data can be achieved by implementing a single prediction scheme. In this scheme, the gateway forecasts the monitored parameters and determines when the sensor nodes should collect their readings. This decision is based on the reliability of the current predictions [35]. The single prediction scheme enables the answering of user queries without directly retrieving data from sensor nodes. In this approach, each query specifies both the required data and the acceptable level of error. The gateway responds by indicating that the actual measurements fall within a given range if the confidence of prediction is high enough to meet the user's defined error threshold. To achieve this, the gateway employs statistical models [36]. The statistical models are used to assess the confidence level associated with forecasts of current sensor readings. On this basis, the gateway determines whether additional measurements need to be collected from the sensor nodes.

The work in [37] involved the application of the single prediction technique in a cluster-based sensor network. In this network, the cluster head is responsible for collecting data from its member sensor nodes. The cluster head evaluates the level of spatio-temporal correlations among data samples from the member nodes. Subsequently, the sampling rate of these sensor nodes is adjusted according to the calculated correlation level. A prediction algorithm is used to ensure data completeness, allowing a base station to reconstruct any missing data points.

An alternative single prediction strategy posits that the operation of a sensor network follows a schedule consisting of data transmission periods followed by data prediction intervals, as detailed in [38]. During these data prediction phases, the retrieval of real-time measurements from sensor nodes is omitted. The sensor readings are estimated using a predictive model created with deep learning techniques. That approach was designed for implementation in an industrial IoT system [39].

A transmission reduction protocol, which exploits redundancy at the temporal and spatial levels in clustered IoT sensor networks, was proposed in [40]. This protocol is implemented at the sensor node level and utilizes a modified kNN algorithm. According to this approach, the sensed data are classified into several classes using the kNN technique. Every class includes the most similar sensed data. The protocol selects a representative reading from each class and sends it to the sink instead of sending the entire data of each class. Moreover, similar classes are combined in order to further reduce the amount of data being transmitted.

In [41] a distributed prediction-compression mechanism was proposed. This mechanism periodically decides whether to send the data to the gateway. It utilizes the autoregressive integrated moving average (ARIMA) prediction method to forecast the data for the next period and determine whether the current data should be transmitted to the gateway. When the decision is made to send the data to the gateway, a compression technique is employed to eliminate redundant data. This approach combines various techniques for reducing data transmission, including adaptive piecewise constant approximation, differential encoding, and symbolic aggregate approximation.

The existing single prediction methods are suitable for applications in sensor networks where sensors are closely located and exhibit a well-defined spatiotemporal correlation. These correlations are used to generate a model that predicts the data and checks if the results fall within a confidence interval [42]. The application of this scheme for target node selection in sensor networks is challenging as it requires additional algorithms to determine when the predicted data are sufficient and when the collection of actual sensor readings can be skipped [43].

Based on the literature review above, the overall technical gaps have been identified, which has served as motivation for designing the proposed methodology. These gaps can be outlined as follows.

- The existing prediction-based methods for reducing transmission can determine whether new sensor readings need to be transmitted at the current time step. However, these methods are not suitable for deciding when the next data transmission will be necessary. Thus, they cannot be used to determine the time of the next transmission in advance.
- The available methods use a predetermined threshold of error or accuracy to decide if the prediction results can substitute the actual sensor readings. Yet, in the case of target node selection, the required accuracy changes dynamically, and a specific threshold level cannot be predetermined.
- Dual prediction methods are not suitable for effective reduction of data sampling.

In this paper, a new data collection method is introduced based on the single prediction scheme. The presented method takes into account the most significant changes in historical sensor readings to evaluate possible future ranges of the collected data and determine when the condition for updating the target node can be met. The sensor nodes suppress their transmissions and sampling until the prediction result indicates that the selection of a new target node is possible.

The state-of-the-art prediction algorithms for data reduction in sensor networks were designed to forecast the actual value of sensor reading [14]. In contrast, this paper introduces a modified prediction algorithm to determine a range (interval) of possible values that a sensor reading can take. The interval of possible sensor readings is evaluated by taking into account the largest expected increase and the largest expected decrease of the monitored parameter. This approach allows us to describe how the uncertainty of information about monitored parameters increases over time when the sensor node does not report its actual readings. On this basis, the gateway decides how long a given sensor node can remain in sleep mode before transmitting the next sensor reading.

The main contributions of this paper include:

- the development of a data collection method that enables accurate selection of target nodes and requires fewer data transmissions compared to state-of-the-art approaches;
- the introduction of a new prediction algorithm that determines intervals of future sensor reading values based on observed increase and decrease rates in historical data;
- strategies for implementing the proposed approach in the IoT sensor network.

Details of the proposed solutions are presented in the next section.

3. Proposed method

According to the proposed method, the time is estimated in which the values observed by sensor nodes can change significantly, and there can be a necessity to select a new target node. To explain the basic concept behind the proposed approach, we will consider a sensor network, where a sensor node has to act as the target node if its sensed value reaches or exceeds a given threshold. This scenario is illustrated in Fig. 2, where the value sensed by a non-target sensor node increases with time (solid line), and the threshold is denoted by the symbol w (dashed line). Symbols $t_1, \dots, t_4$ in Fig. 2 correspond to time instances when the sensor node transmits its readings to the gateway. In this example, the gateway has to decide when the subsequent transmission will be necessary after receiving sensor readings at time t_1. To this end, it predicts the maximum possible increase rate of the sensed value ($u(t_1)$) based on recently collected data. The upward-sloping dotted lines in Fig. 2 show the maximum expected increase of the sensed value. The time of the next transmission is determined as follows:

$$t_2 = \frac{t_1 + (w - v(t_1))}{u(t_1)}, \tag{1}$$

where $v(t_1)$ is the sensor reading registered at time t_1, and the meaning of the remaining symbols is explained above.

The gateway will receive further sensor readings at time t_2. It should be noted here that the sensor node sends a sequence of recently sensed

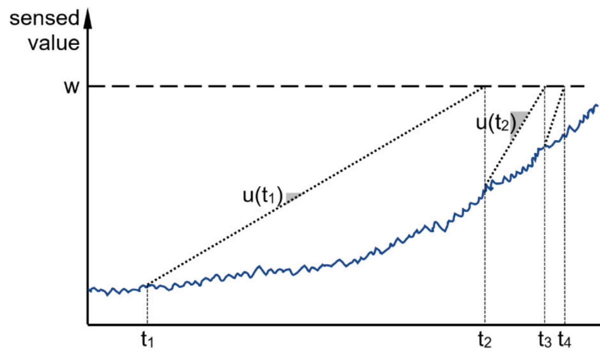

Fig. 2. Transmission scheduling for threshold-based target node selection.

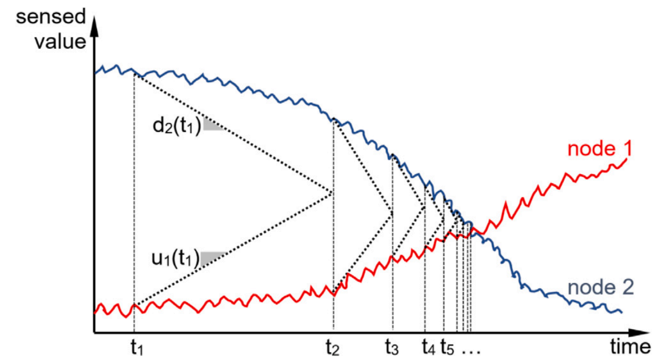

Fig. 3. Transmission scheduling for order-based target node selection.

values. During preliminary experiments, it was observed that several subsequent sensor readings have to be taken into account as input data for the prediction algorithm in order to capture the local trend of the monitored parameter. The prediction accuracy is significantly lower when only the current sensor reading (e.g., $v(t_2)$) is used for prediction purposes. Thus, in our example, the sensor node can be put into sleep mode at time step t_1 and has to be awakened at time step $t_2 - c + 1$ to collect the sequence of c data readings.

In Fig. 2, it was assumed that the predicted maximum increase rate $u(t_2)$ is greater than $u(t_1)$ to illustrate the fact that the prediction results depend on the local trend of the monitored parameter. Namely, $u(t_2)$ in this example is greater than $u(t_1)$ because the increase in the sensed value at time t_2 is visibly faster than that observed at time t_1.

As shown in Fig. 2, the transmissions are performed more frequently when the sensed value is closer to the threshold, above which the sensor node has to be selected as the target node. Therefore, when using the presented approach, the gateway can be immediately informed when the sensor reading reaches the threshold level. This effect is achieved since the time between two successive transmissions has the limit 0 as sensed value $v(t)$ tends to threshold w:

$$\lim_{v(t)\to w} \Delta t = 0, \tag{2}$$

where Δt is the time between transmissions, which corresponds to the difference $t_2 - t_1$ in Eq. (1).

Hereinafter, we will consider another, more complex scenario where a constant threshold is not determined, and the selection of the target node depends on the order relationship between values currently sensed by specific nodes. It means that the new target node has to be selected when there is a change in the order of sensor nodes. This change is determined by sorting the nodes based on the sensed value. For instance, one might be interested in selecting the target node that currently records the maximum or minimum value of a monitored parameter. In this case, we predict the maximum possible increase rate $u(t)$ and the maximum possible decrease rate $d(t)$ of sensor readings to estimate when the new target node can be selected, as illustrated in Fig. 3.

The example in Fig. 3 shows the values of a monitored parameter for two sensor nodes. It is assumed in this example that node 1 has to be selected as the target node when its sensed value is above the sensor reading of node 2. The sensor readings are transmitted to the gateway at time t_1. Then the gateway decides when the subsequent transmission will be necessary based on a prediction of the maximum possible increase rate for node 1 ($u_1(t_1)$) and the maximum possible decrease rate for node 2 ($d_2(t_1)$). These predicted rates are shown in Fig. 3 by the sloping dotted lines. The formula below determines the time of the second data transmission from the sensor node:

$$t_2 = \frac{t_1 + (v_2(t_1) - v_1(t_1))}{(d_2(t_1) + u_1(t_1))}, \tag{3}$$

where $v_1(t_1)$ and $v_2(t_1)$ denote the value sensed at time t_1 for nodes 1 and 2, respectively.

The operations mentioned above are repeated after subsequent data transfers at time steps $t_2, t_3, t_4, t_5, \ldots$, as illustrated in Fig. 3. It should be noted in this example that the transmissions are performed more frequently when the difference between sensed values is decreasing and when it becomes more probable that the order relation between sensor readings will change soon. Thus, the frequency of data transfers is appropriately adjusted to ensure the correct selection of the target node.

The concepts presented in the above examples enable us to formulate algorithms that allow the gateway to collect the necessary data from sensor nodes and select the target node appropriately. As was already mentioned, we assume that the gateway needs to determine the current order of sensor nodes concerning their sensed values. It means that the gateway must determine which sensor node currently records the highest value of the monitored parameter, which sensor node records the second highest value, and so on. This assumption allows us to apply the proposed algorithms to various target node selection rules. For instance, the algorithms can be used to pick the target node with the maximum sensed value or choose 10% of nodes with the highest sensor readings.

Operations performed by the gateway are presented by Algorithm 1. Algorithm 2 shows the operations of a sensor node. Table 1 lists the symbols used in the algorithms described next.

Table 1
List of symbols.

Symbol	Description
i, j	Identifiers of sensor nodes
t	Time step of sensor network operation
$u(t)$	Maximum possible increase rate of sensor readings
$d(t)$	Maximum possible decrease rate of sensor readings
$v(t)$	Sensor reading registered at time step t
V	Sequence of recent sensor readings
c	Number of sensor readings in sequence V
A	Set of sensor nodes transmitting data at current time step
h	Time horizon for calculating change rates of sensor readings
s, s_i	Number of time steps to next transmission
k	Number of the most similar sequences used for predictions
M_U	Prediction model used to determine $u(t)$
M_D	Prediction model used to determine $d(t)$
T	Length of time series containing historical data

The data exchange between the sensor nodes and the gateway occurs in time steps. The gateway waits for data transfers from sensor nodes at each time step. It receives data from the target node or nodes, as well as optionally from other non-target sensor nodes that were scheduled to wake up and report their current sensor readings at the specified time step. Symbol A in Algorithm 1 denotes the set of all sensor nodes that have transmitted their readings at time step t. For each of these nodes, the gateway predicts the maximum possible increase rate ($u_i(t)$) and the maximum possible decrease rate ($d_i(t)$). Details of the prediction method are presented later in this chapter. Then, the time to the subsequent transmission for sensor nodes i and j is determined, as shown in lines 7-11 of Algorithm 1. Consequently, the variable $s_{i,j}$ stores the number of time steps after which the current order relation between readings of nodes i and j can change.

Additionally, we require that $s_{i,j}$ is not lower than one and not greater than $s_{\max}$ (line 12 in Algorithm 1). The parameter $s_{\max}$ allows us to set the maximum time after which each sensor node has to report its readings. In line 14 of Algorithm 1, the gateway selects the target node (or nodes) based on recently reported sensor readings (v_i). Then, the time to the next transmission (s_i) is determined for particular sensor nodes (lines 16–20 in Algorithm 1). It should be noted that the target nodes send their data at each time step. Thus, the variable s_i is set to 1 for the selected nodes. Finally, the gateway sends an acknowledgment containing s_i to the sensor nodes in A.

Algorithm 1 Gateway operations

1: **for** each time step t **do**
2: Receive V from sensor nodes $i \in A$
3: **for** each sensor node $i \in A$ **do**
4: Predict $u_i(t)$ and $d_i(t)$
5: **end for**
6: **for** each pair of sensor nodes (i, j) such that $i \in A$, $j \in A, i \neq j$ **do**
7: **if** $v_i(t) \geq v_j(t)$ **then**
8: $s_{i,j}$:= floor$\left(\left[v_i(t) - v_j(t)\right] / \left[d_i(t) + u_j(t)\right]\right)$
9: **else**
10: $s_{i,j}$:= floor$\left(\left[v_j(t) - v_i(t)\right] / \left[d_j(t) + u_i(t)\right]\right)$
11: **end if**
12: $s_{i,j}$:= min(max($s_{i,j}$,1),$s_{\max}$)
13: **end for**
14: Select target node(s)
15: **for** each sensor node $i \in A$ **do**
16: **if** sensor node i is selected as target node **then**
17: s_i := 1
18: **else**
19: s_i := $\min_{j \in A, j \neq i} s_{i,j}$
20: **end if**
21: Send acknowledgment to sensor node i with s_i
22: **end for**
23: **end for**

As mentioned earlier in this section, the sensor node has to deliver a sequence of recently sensed values to the gateway at a predetermined time step. The length of the data sequence is denoted by parameter c. Thus, following Algorithm 2, the sensor node creates a list V of data readings and sends the collected data from this list to the gateway. When the requested time to the subsequent transmission is shorter than c time steps ($s_i < c$), the sensor node will only collect and send the s_i most recent data readings. This is because the remaining $c - s_i$ sensor readings have already been reported to the gateway. Variable r in Algorithm 2 denotes the number of data readings that need to be collected and appended to V before transmission. The number of time steps that the sensor node can spend in the sleep state is determined by s.

Algorithm 2 Sensor node operations

Input: V := empty list, s := 0, r := c
1: **do**
2: Sleep for s time steps
3: Read $v(t)$ from sensor
4: Append $v(t)$ to V
5: $r := r - 1$
6: **if** $r > 0$ **then**
7: $s := 1$
8: **else**
9: Send V to gateway
10: Receive acknowledgment with s_i
11: Clear V
12: r := min(s_i, c)
13: $s := s_i - r + 1$
14: **end if**
15: **while** true

According to the proposed approach, the gateway has to predict the largest possible increase rate ($u_i(t)$) and the largest possible decrease rate ($d_i(t)$) of the sensor readings. The objective of prediction is to estimate $u_i(t)$ and $d_i(t)$ for a given sequence V of c recent sensor readings:

$$V = \{v_i(t-c+1), v_i(t-c+2), \dots, v_i(t)\} \tag{4}$$

In order to make predictions, we take into account the increase and decrease rates observed in a historical dataset, which includes previously collected sensor data.

For historical data, the change rates of sensor readings are determined as illustrated in Fig. 4. The example in Fig. 4 shows the change rates estimated for time instance t with the use of three subsequent data points, registered at time steps $t+1, t+2, t+3$. It should be noted that the rates of change r_1, r_2, and r_3 correspond to the slope of the thin black lines, i.e., $r_1 = v(t+1) - v(t), r_2 = (v(t+2) - v(t))/2$, etc. Thus, based on the data presented in Fig. 4, one can expect that the sensed value may grow as fast as described by r_3 ($u(t) = 3$) when recent sensor readings are similar to the presented sequence $v(t-2), v(t-1), v(t)$. This approach has two main parameters: the length of the input sequence (c) and the length of the time horizon for which the change rates r are calculated (h). For the example presented in Fig. 4, it was assumed that $c = 3$ and $h = 3$.

Let us assume that the historical sensor readings are available for time steps $t = 1, 2, 3, \dots, T$, then the largest rates of increase $u(t)$ and decrease $d(t)$ are determined for the time instances $t = c, c+1, \dots, T-h$. For each time instance, we calculate h values of change rate r_d using the following formula:

$$r_d(t) = \frac{v(t+d) - v(t)}{d}, d = 1, 2, \dots, h \tag{5}$$

Subsequently, the largest decrease rate $d(t)$, and the largest possible increase rate $u(t)$, are evaluated as the maximum and minimum of the $r_d(t)$ values:

$$\begin{aligned} d(t) &= \min\{r_1(t), r_2(t), \dots, r_h(t)\}, \\ u(t) &= \max\{r_1(t), r_2(t), \dots, r_h(t)\}. \end{aligned} \tag{6}$$

In the historical dataset, we can usually find many sequences of sensor readings similar to a given sequence (V) that are followed by significantly different change rates. An example is presented in Fig. 5, where we observe opposite trends for sequences V_1 and V_2 at

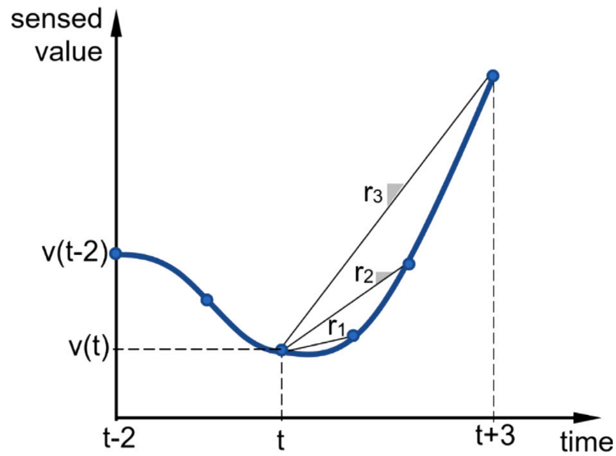

Fig. 4. Change rates estimation for sensor readings.

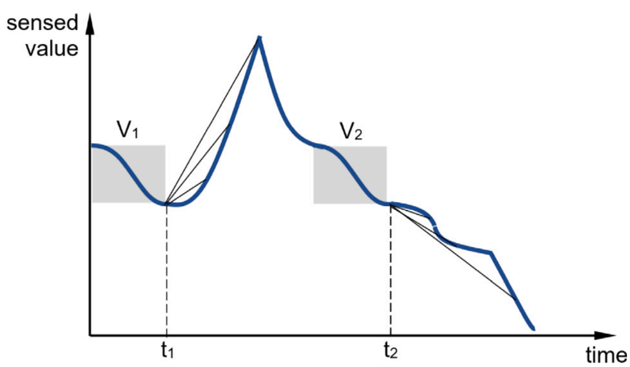

Fig. 5. Evaluation of maximum increase and decrease rates for a sequence of sensor readings.

times t_1 and t_2. In this example, the possible maximum increase and decrease rates cannot be correctly evaluated for V when considering the change rates observed at time t_1 or t_2 separately. Thus, it is justified to aggregate the change rates evaluated for several sequences from the historical dataset when determining the possible increase and decrease rates. It means that we need to find similar sequences (nearest neighbors) in the historical data and then take into account the extreme change rates for the collection of similar sequences.

In this study, we introduce and examine two strategies (online and offline) to predict the maximum possible increase rate $u(V)$ and the maximum possible decrease rate $d(V)$ for a given sequence of sensor readings V. Both strategies are inspired by the k-nearest neighbor algorithm [44].

According to the online strategy, the search for similar sequences in historical datasets is performed on-demand when the gateway receives recent sensor readings V and initiates the prediction procedure (see line 4 in Algorithm 1). Details of these operations are presented in Algorithm 3. We use the Euclidean distance as the similarity measure to select k historical sequences that best match the given sequence V of recent sensor readings. The output of Algorithm 3 establishes the prediction results for Algorithm 1, i.e., $u_i(t) = u(V)$ and $d_i(t) = d(V)$.

In practical implementations, the historical dataset is often large and cannot be stored on a gateway with limited memory and computing resources. Moreover, searching similar data sequences is a computationally-expensive task that must be completed in real-time. A feasible solution is to store and process the historical data in the cloud, as illustrated in Fig. 6. In such a case, Algorithm 3 is executed by a cloud server. The gateway sends a request to the cloud server each time it receives the recent data readings from the sensor node

Algorithm 3 Online prediction strategy

Input: Sequence V, Historical dataset $v(1), ..., v(T)$, decrease rates $d(c)$, ... ,$d(T-h)$, increase rates $u(c), ..., u(T-h)$)

Output: $u(V), d(V)$

```
for t = c to T − h do
    distance(t) = Euclidean distance between sequences V
    and {v(t − c + 1), v(t − c + 2), ..., v(t)}
    Select k time instances t_j (j=1, 2, ...,k) for which
    distance(t) takes the lowest values
    u(V) = max{u(t_j), j = 1, 2, ..., k}
    d(V) = min{d(t_j), j = 1, 2, ..., k}
end for
```

Table 2
Input data of Algorithm 4 (example).

Sequence of historical data			Decrease rate	Increase rate
$v(1)$	$v(2)$	$v(3)$	$d(3) = -4$	$u(3) = 1$
$v(2)$	$v(3)$	$v(4)$	$d(4) = -2$	$u(4) = 3$
$v(3)$	$v(4)$	$v(5)$	$d(5) = -8$	$u(5) = 0$
$v(4)$	$v(5)$	$v(6)$	$d(6) = -1$	$u(6) = 7$
$v(5)$	$v(6)$	$v(7)$	$d(7) = -1$	$u(7) = 5$

Table 3
Euclidean distance between data sequences (example).

t	3	4	5	6	7
3	0	29	32	37	26
4	29	0	24	42	39
5	32	24	0	34	27
6	37	42	34	0	15
7	26	39	27	15	0

(V) and needs to predict the values of $u(V)$ and $d(V)$. This approach is associated with additional communication costs and is vulnerable to communication delays.

A diagram of the offline strategy is presented in Fig. 7. In this strategy, similar sequences in the historical dataset are searched for and grouped. Algorithm 4 provides a detailed explanation of these operations. In lines 1–3 the distance between all sequences in the historical dataset is calculated. For each sequence, a group of k nearest neighbors is found (lines 6 and 7). Then, the maximum possible increase and decrease rates (u, d) are determined for each group of similar sequences. A machine learning algorithm utilizes these results to train prediction models. Finally, the trained models are sent to the gateway. The above operations are not repeated every time the gateway receives new sensor readings. They are executed in advance before the gateway starts to collect data in accordance with Algorithm 1.

To illustrate how Algorithm 4 works, we will use a simple example in which a historical time series of 10 sensor readings ($v(1) \dots v(10)$) is analyzed ($T = 10$). It was assumed in this example that the number of lags and the time horizon for calculating the change rate are the same ($c = h = 3$). Table 2 shows the sequences of sensor readings extracted from the input time series, along with the corresponding increase and decrease rates. It should be noted that, for the assumed values of c and h, the change rates $d(t)$ and $u(t)$ can be determined for time steps $t = 3 \dots 7$. Algorithm 4 calculates the Euclidean distances between the available data sequences (see line 3 in the pseudocode). Hypothetical results of these calculations are presented in Table 3, where t denotes the last element in the data sequence (e.g. $t = 4$ corresponds to the sequence $\{v(2), v(3), v(4)\}$).

After evaluating the distances between sequences, Algorithm 4 searches for the nearest (most similar) sequences (see lines 6–7 in

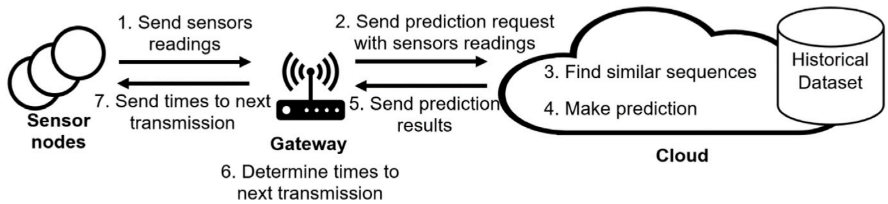

Fig. 6. Online strategy.

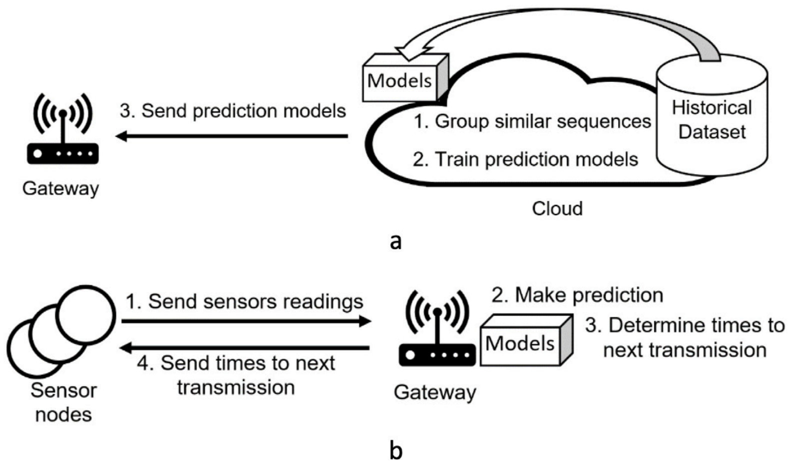

Fig. 7. Offline strategy: (a) offline operations, (b) online operations.

Table 4
Calculation of maximum possible increase and decrease rates (example).

t_1	t_2	$d(t_1)$	$d(t_2)$	$u(t_1)$	$u(t_2)$	$d_{min}(t)$	$u_{max}(t)$
3	7	−4	−1	1	5	−4	5
4	5	−2	−8	3	0	−8	3
5	4	−8	−2	0	3	−8	3
6	7	−1	−1	7	5	−1	7
7	6	−1	−1	5	7	−1	7

the pseudocode). Let us assume that $k = 2$. In this case, one similar sequence is selected for each sequence. The resulting pairs of sequences are denoted by t_1 and t_2 in Table 4. For instance, the values $t_1 = 3$ and $t_2 = 7$ in the first row indicate that the most similar sequence to $\{v(1), v(2), v(3)\}$ is $\{v(5), v(6), v(7)\}$. Table 4 also includes the largest possible increase and decrease rates $(u_{max}(t), d_{min}(t))$ determined for each sequence. These results are then used to train the prediction models (M_U and M_D). To be specific, the training dataset for the model M_U includes the sequences of historical data from Table 2 and the $u_{max}(t)$ values from Table 4. The sequences are considered as inputs during model training, while the $u_{max}(t)$ values are the expected outputs. Similarly, when training the M_D model, the input vectors consist of sequences of historical data, and the expected outputs are determined by $d_{min}(t)$.

Algorithm 4 is versatile in its ability to apply any machine learning model for predicting the rates of increase and decrease. The machine learning models considered in this study include support vector machine, random forest, and recurrent neural network.

An advantage of the online strategy is that the prediction models can be updated in parallel while Algorithms 1 and 2 are being executed during the data collection process in a sensor network. This approach enables the gateway to obtain predictions without having to perform the time-consuming tasks of searching for similar sequences or training models. Each time the gateway needs to evaluate $u(V)$ and $d(V)$, the prediction is performed locally using the available models.

Algorithm 4 Offline prediction strategy

Input: Historical dataset $v(1)$, ..., $v(T)$, decrease rates $d(c), ..., d(T-h)$, increase rates $u(c), ..., u(T-h)$)
Output: Prediction models M_U, M_D

```
for t_a := c to T − h − 1 do
    for t_b := t_a to T − h do
        distance(t_a, t_b) := Euclidean distance between
            sequences {v(t_a − c + 1), v(t_a − c + 2), …, v(t_a)}
            and {v(t_b − c + 1), v(t_b − c + 2), …, v(t_b)}
    end for
end for
for t := c to T − h do
    select k time instances t_j (j = 1, 2, ..., k) for which
        distance(t, t_j) takes the lowest values
    u_max(t) := max{u(t_j), j = 1, 2, …, k}
    d_min(t) := min{d(t_j), j = 1, 2, …, k}
end for
Train model M_U to predict u_max(t) based on {v(t − c + 1), v(t − c + 2), …, v(t)}
Train model M_D to predict d_min(t) based on {v(t − c + 1), v(t − c + 2), …, v(t)}
```

Before implementing the proposed method for reducing transmission in a sensor network, the parameters c, h, and k need to be tuned. The process of parameter tuning can be performed using optimization procedures, such as grid search or random search [45]. The parameter

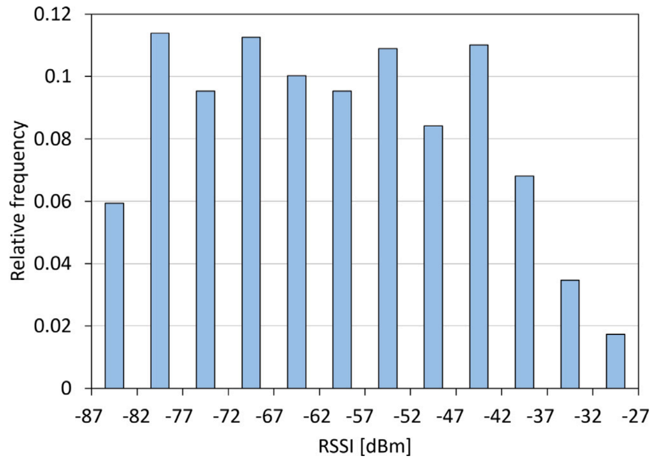

Fig. 8. Histogram of RSSI data.

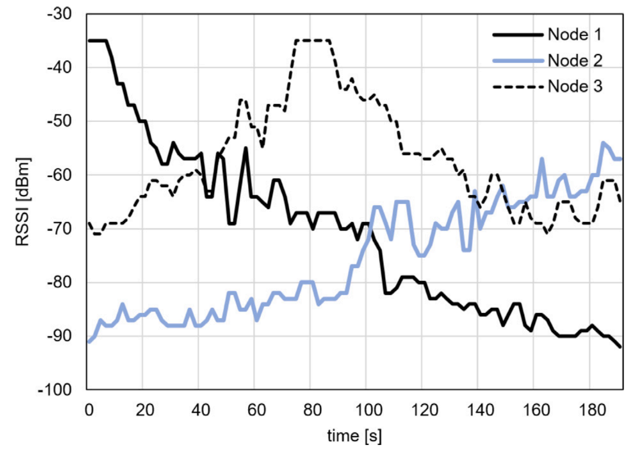

Fig. 9. Example of raw RSSI data from three sensor nodes.

optimization is based on previously collected time series of sensor readings.

4. Experimental evaluation

4.1. Experimental setup

Experiments were conducted assuming a model of a sensor network that consists of three mobile sensor nodes transmitting data directly to the gateway via one hop. During a single time step of the network operation, all sensor nodes can collect data samples and report them to the gateway.

The proposed methods were evaluated experimentally using real-life data collected from mobile devices (smartphones). The mobile devices have acted as sensor nodes to measure the strength of the radio signal received from a Wi-Fi access point (RSSI). When collecting the experimental data, the smartphones were kept by persons moving in an indoor area of 135 square meters. The target node selection was performed to determine which smartphone recorded the highest RSSI value at a given time. This information can be useful in various applications, e.g., when it is necessary to find the target node (or the user of a mobile device) with the shortest distance to the access point location.

The RSSI data were collected using three mobile devices for 60 h. Measurements were performed at a frequency of 1 Hz. The experimental dataset is a multivariate time series consisting of 215 450 records. Each record includes three RSSI values in dBm, acquired by three devices. The RSSI values are negative integers ranging from −86 dBm to −28 dBm. The distribution of RSSI data is visualized by the histogram in Fig. 8. We can categorize the records into three classes, where a different device registers the highest RSSI value. The differences in the number of records for such classes are not greater than 20%.

An example of the raw RSSI data collected during a 190-second period is presented in Fig. 9. At the preprocessing stage, the median filter [46] was used to reduce the noise in raw RSSI measurements. Computational experiments were conducted using the R programming language and related packages, including Keras, Caret, and e1071.

Three metrics were taken into account to evaluate the performance of the proposed approach and the compared methods described in the following subsection. The metrics considered include transmission reduction, error in target node selection, and sampling reduction:

$$Transmission\ reduction = \frac{NTR}{NT} \cdot 100\%, \tag{7}$$

where NT is the total amount of data transmissions from sensor nodes when all nodes send data to the gateway at each time step; NTR denotes the number of transmissions from sensor nodes performed in the considered time period if using the given transmission reduction method. The error of target node selection is defined as follows:

$$Error = \frac{TSE}{TS} \cdot 100\%, \tag{8}$$

where TS is the total number of time steps in the considered period of sensor network operation; TSE denotes the number of time steps when the target node cannot be correctly selected based on the data delivered to the gateway. Let us assume that the gateway selects node α as the target node at time step t. We accept this selection as made correctly if the following condition is satisfied:

$$max_i\{RSSI_i(t)\} - RSSI_\alpha(t) < 2 \text{ dBm}, \tag{9}$$

where i denotes the id of the sensor node ($i \in \{1, 2, 3\}$), and $RSSI_i(t)$ is the RSSI value registered by sensor node i at time step t.

Finally, the sampling reduction is defined as follows:

$$Sampling\ reduction = \frac{NSR}{NS} \cdot 100\%, \tag{10}$$

where NS represents the number of data samples recorded when all nodes collect data readings at each time step, and NSR denotes the number of data samples captured in the specified time period after implementing the given method for reducing transmission.

4.2. Compared methods

During the experiments, the proposed method was compared with seven representative transmission reduction approaches from the literature. The compared methods include five dual prediction algorithms based on different models [28–30,47,48], the combinational data prediction method [34], and an algorithm that performs global predictions of the increase and decrease rates for the considered sensor readings [49].

In the dual prediction scheme, the gateway and the sensor node utilize a prediction model to estimate the current RSSI reading. Additionally, the sensor node evaluates a prediction error by calculating the absolute difference between the predicted value and the actual measurement. Then, the current reading is transmitted to the gateway only if the prediction error exceeds a predetermined threshold. In the opposite scenario, the transmission from the sensor node is omitted, and the gateway chooses the target node based on the prediction result. This scheme was implemented using different prediction models (naïve [47], ARIMA [48], LSTM [28], Window-based [29], and Normalized quantile regression [30]).

According to the dual prediction method using the naïve model, the prediction is set to be the last RSSI value transmitted to the gateway.

This simple model does not require any training or tuning procedures. For the dual prediction using Auto-Regressive Integrated Moving Average (ARIMA) models, the best prediction models were identified using the Hyndman–Khandakar algorithm [50]. This algorithm involves unit root tests, the minimization of the Corrected Akaike Information Criterion (AICc), and maximum likelihood estimation.

The implementation of dual prediction with LSTM (*Long Short-Term Memory*) was preceded by preliminary experiments that enabled us to select the most appropriate recurrent neural network architecture. The architectures proposed in [28] were tested with various hyperparameter configurations during these experiments. The neural network with 16 LSTM cells and 1 hidden layer was chosen for further testing. The number of cells for the hidden layer was set to 8, and the ReLU (*rectified linear unit*) activation function was used. The dropout rate was equal to 0. The training procedure for the LSTM model utilized the Adam optimizer, which is a replacement optimization algorithm for stochastic gradient descent. The batch size was set to 16 samples, and the loss function used was the mean squared error.

In the case of the window-based forecasting technique [29], the next sensor reading is calculated as the sum of the current reading and the average difference of adjacent previous readings in a time window. The parameter of this method, which determines the width of the time window (w), was set to 3 based on preliminary experiments. The normalized quantile regression method includes three stages: initialization, model building, and prediction. These stages were implemented based on the pseudocode presented in [30].

The combinational data prediction method is based on linear regression, as discussed in Section 2. The implementation of this method involves training the model using the least squares algorithm, predicting data, and calculating errors. Details of these operations are described in [34].

The global prediction method [49] utilizes information about the largest expected increase and decrease rates of sensed values, which is similar to the proposed approach. The main difference is that, in the case of the global method, the expected increase and decrease rates are calculated based on the entire historical dataset. In contrast, the proposed approach utilizes a subset of historical data that is chosen based on its similarity to the current sensor readings. Specifically, for the global method, the largest increase rate u is determined as the $(100-p)$th percentile of the positive values of the difference between two successive measurements in the historical dataset. The largest decrease rate d is determined as the pth percentile of the negative values of the difference between two successive data readings.

In order to evaluate the proposed method, it was implemented using both online and offline strategies. For the offline strategy, three variants were considered that utilize different machine learning algorithms for training the prediction models. These variants include Support Vector Machines (SVM) [51], Random Forest (RF) [52], and Recurrent Neural Networks (LSTM). The training of SVM and RF models was conducted using the machine learning algorithms implemented in R packages (caret and e1071). The parameters of these models were tuned using the grid search method with 10-fold cross-validation. The LSTM neural network for predicting the maximum increase and decrease rates was designed and tuned in the same manner as described above for the dual prediction scheme.

4.3. Results and discussion

During the experiments, the amount of data sampled and transmitted from the sensor nodes to the gateway was analyzed for the proposed approach and compared with the state-of-the-art methods described in Section 4.2. The author has also verified the impact of data transmission on the accuracy of selecting target nodes. The main objective of the proposed approach is to minimize data transmission and sampling while ensuring the accurate selection of the target node. It means that the sensor node with the highest RSSI reading must be selected at each time step. Thus, the analysis was performed to determine the level of transmission reduction for a given method, provided that the target node is correctly selected for all time instances considered.

The results of this analysis are presented in Fig. 10. It can be observed from these results that the proposed method achieves the highest reduction in transmission. The proposed method, specifically with the online strategy, has successfully eliminated 94% of the transmissions. The offline strategy has also achieved high levels of transmission reduction, close to 90%. A significantly lower transmission reduction was achieved with the state-of-the-art methods. The global prediction method has, on average, eliminated 81.4% of transmissions. Nearly 70% of the transmissions were suppressed by the combinational method. In the case of the dual prediction approach, the reduction level was near 50% for all examined prediction algorithms. Similar reduction levels were achieved by the window-based and normalized quantile regression methods.

When comparing the results obtained from two implementation strategies of the proposed method, it becomes apparent that the online strategy allows us to achieve a higher level of transmission reduction compared to the offline strategy. The reason is that the online strategy uses historical data directly for prediction. In contrast, during the training of the prediction model, the offline strategy resulted in the loss of some information that was originally included in the historical dataset. The type of machine learning algorithm does not significantly affect the performance of the offline strategy. A slightly higher level of transmission reduction was achieved for the RF algorithm (average value of 90.9%) than for SVM (90.6%) and LSTM (89.4%).

In order to evaluate the accuracy of the RF, SVM, and LSTM models developed for offline strategy, the normalized root mean squared error (NRMSE) was assessed. For this experiment, the dataset was randomly split into 70% for model training and 30% for testing. Results obtained from 20 test runs are presented in Fig. 11. The error bars in Fig. 11 correspond to the values of the 5th and 95th percentiles. These results show that the RF algorithm achieved a lower prediction error than SVM and LSTM. The improved prediction accuracy of the RF model contributes to a greater reduction in transmission. It should be noted that, for the experimental dataset, the actual values of the increase and decrease rates are equal to 0 for a number of time instances. Thus, the NRMSE metric was used in this analysis. The root mean squared error was normalized using the difference between the maximum and minimum observed values.

The experimental results in Fig. 10 reveal that the dual prediction method achieves similar levels of transmission reduction for all the prediction models considered. The average reduction in transmissions was equal to 50.7% for the naïve prediction, 50.0% for LSTM, and 50.5% for the ARIMA model. The low differences between transmission reduction levels for the analyzed dual prediction methods are due to the fact that the implemented prediction models achieve similar accuracy in predicting the RSSI data. Fig. 12 shows the dependency between the prediction horizon and the mean average percentage error (MAPE) for the compared models. It should be noted that the difference in MAPE between prediction models is not greater than 0.5%. When the prediction is made for the subsequent RSSI measurement (prediction horizon of one timestep), the lowest error is observed with the naïve algorithm (0.54%). LSTM and ARIMA errors are equal to 0.61% in this case. However, when considering longer prediction horizons, the LSTM neural network yielded the lowest MAPE values. The shorter prediction horizons are used more frequently for the dual prediction approach being considered. Thus, the accuracy of the model for the shortest prediction horizons has the most significant impact on reducing transmission. As a result, the naïve predictor can achieve a slightly higher level of transmission reduction compared to ARIMA and LSTM. Moreover, we should note that the naïve approach is usually effective in predicting chaotic time series with unexpected changes, such as those in the RSSI measurements.

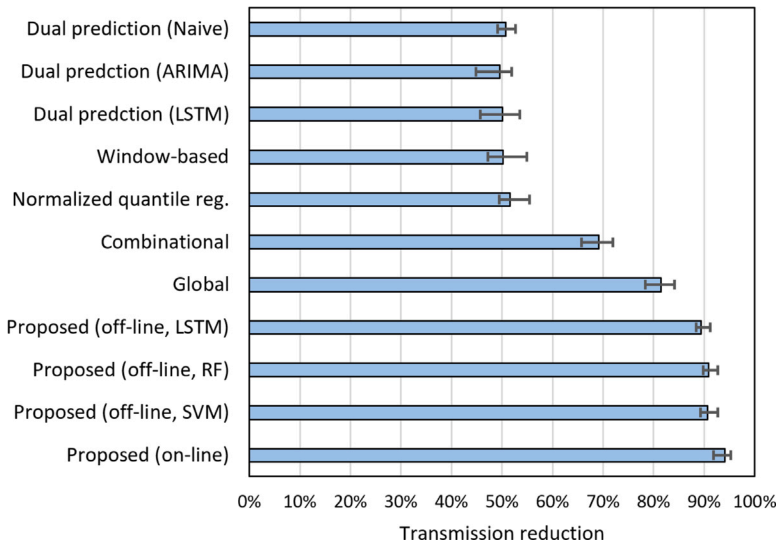

Fig. 10. Data transmission reduction for the compared methods.

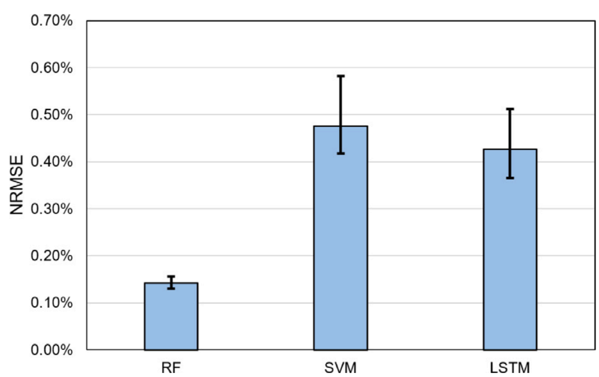

Fig. 11. Error of maximum increase and decrease rates prediction for machine-learning models implemented in offline strategy.

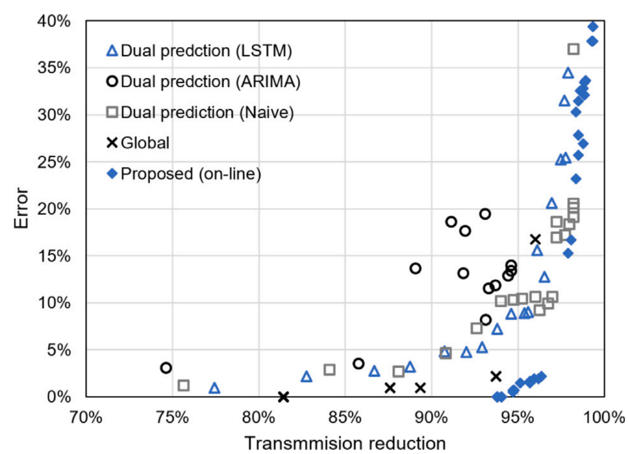

Fig. 13. Error of target node selection for the compared methods.

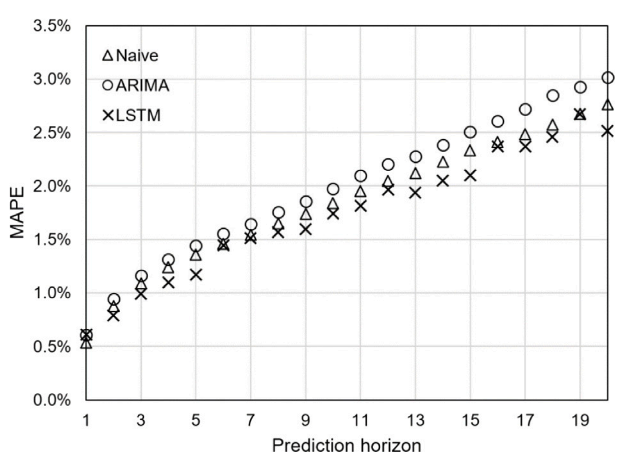

Fig. 12. Prediction error.

As mentioned earlier in this section, the transmission reduction levels presented in Fig. 10 were achieved by tuning the parameters of the compared methods to ensure correct target node selection for all time instances. It was required to have a target node selection error of 0%. For other settings, those methods can achieve higher levels of transmission reduction; however, this comes at the cost of errors in selecting target nodes. The dependencies between transmission reduction and target node selection error for the compared methods are presented in Fig. 13. The results for the dual prediction algorithms were obtained by changing the prediction error threshold from 1 dBm to 20 dBm. It was observed that increasing the threshold for the dual prediction scheme leads to a decreased number of transmissions and more frequent errors in target node selection. In the case of the global prediction method, the experiments were performed for various values of the parameter p ranging from 0 to 30.

Additional experiments were conducted to verify the possibility of controlling the trade-off between reducing transmission and minimizing errors in target node selection for the proposed method. To achieve this goal, an auxiliary parameter (β) was introduced in Algorithm 1. This parameter enables us to decrease the values of the change rates $u_i(t)$ and $d_i(t)$ predicted by the gateway. Specifically, we modify the values of $u_i(t)$ and $d_i(t)$ determined in line 4 of Algorithm 1 as follows: $u_i(t) := u_i(t) \cdot \beta, d_i(t) := d_i(t) \cdot \beta$. The experiments were performed for $\beta \in [0.5, 1]$. It should be noted that for $\beta < 1$ the predicted increase and decrease rates are reduced. This results in less frequent data transmissions from sensor nodes, but it also carries the risk of selecting the incorrect target node. As shown in Fig. 13, the proposed approach achieves a transmission reduction level of up to 96% with an error of target node selection below 2.5%. This result is significantly better than that of the state-of-the-art methods.

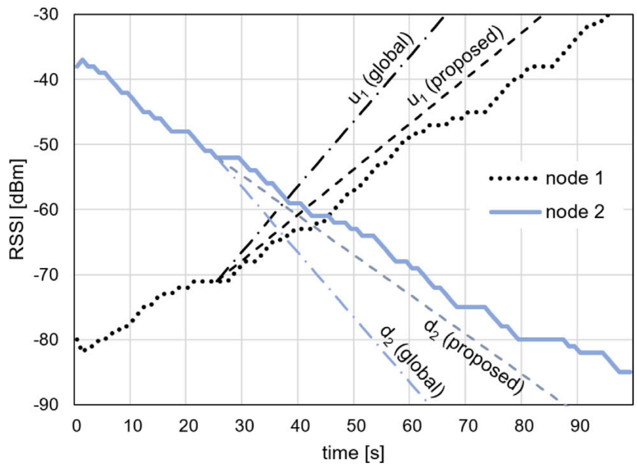

Fig. 14. Increase and decrease rates determined using the proposed method and the global approach (example).

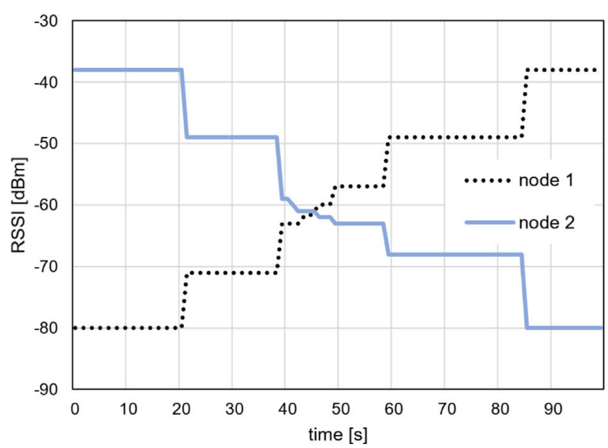

Fig. 15. Values of RSSI reported to gateway using the proposed method (example).

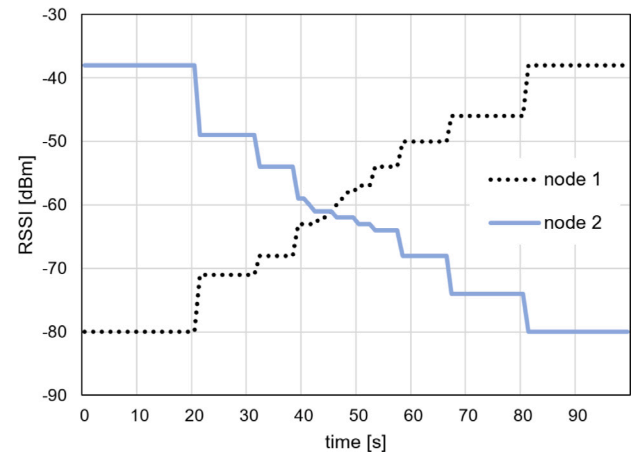

Fig. 16. Values of RSSI reported to gateway using the global method (example).

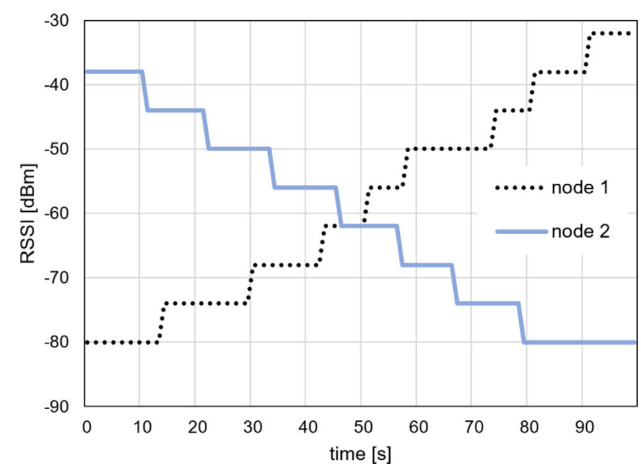

Fig. 17. Values of RSSI reported to gateway using the dual prediction method (example).

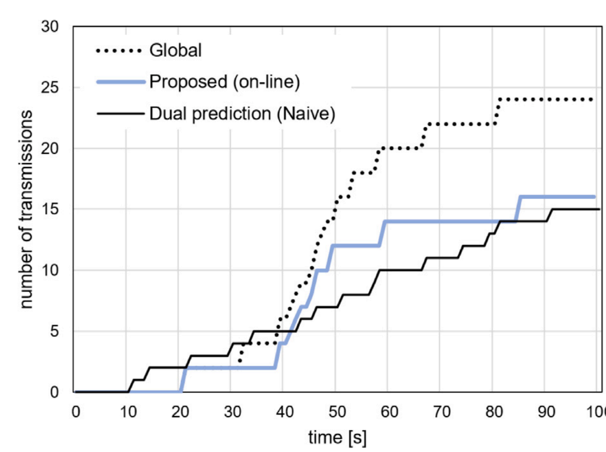

Fig. 18. Number of transmissions from sensor nodes for the compared methods (example).

An example illustrating the operation of the analyzed transmission reduction methods is presented in Figs. 14–18. Fig. 14 shows a sample time series of RSSI readings collected by two moving sensor nodes for a time period of 100 s. The time series was preprocessed using the median filter. Additionally, the dashed lines in Fig. 14 depict the maximum increase rates determined for sensor node 1 and the maximum decrease rates for node 2. These change rates were predicted using both the proposed and global methods for a single time instance (time = 25 s). It can be observed that the change rates computed by the proposed algorithm are lower and correspond better to the actual trend of the time series. Let us assume that the sensor nodes in the example from Fig. 14 have transmitted their data readings to the gateway at time = 25 s. Then, the global method schedules the next transmission to occur at time = 35 s. In contrast, the proposed method schedules the next transmission at time = 40 s. As a result, transmissions occur less frequently with the proposed approach, leading to a lower number of transmissions during the analyzed time period.

Figs. 15–17 show the RSSI values transmitted to a gateway by sensor nodes 1 and 2. The transmissions are scheduled using the proposed approach, global method, and dual prediction scheme. It should be remembered that the original measurements collected by the sensor nodes are presented in Fig. 14. The changes of RSSI in Figs. 15–17 correspond to the moments in time when the actual readings were transmitted from the sensor nodes. Cumulative numbers of transmissions for the compared methods are shown in Fig. 18.

The proposed approach (Fig. 15) and the global method (Fig. 16) perform frequent transmissions when the two sensor nodes being considered register RSSI values that are close to each other. Thus, for both methods, the gateway can correctly determine that the change

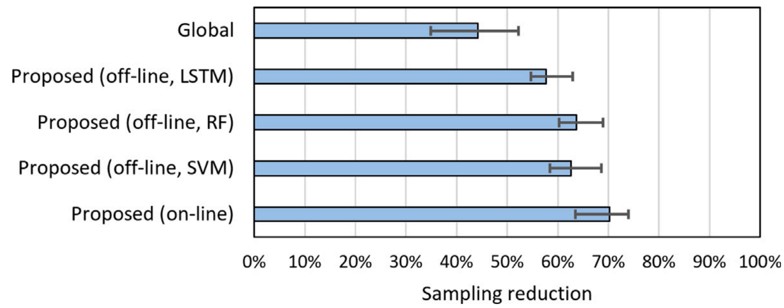

Fig. 19. Comparison of data sampling reduction.

of the target node should occur at time = 46 s. This indicates that both methods do not have any target selection errors (sensor node 1 is selected as the target node for the time period between 0 and 45, and node 2 is selected as the target node for the remaining time). However, the proposed approach requires fewer transmissions.

In the case of the dual prediction method (Fig. 17), the target node is not changed to node 2 at time = 46 s, as the data readings reported to the gateway do not contain the information that the actual RSSI reading of node 2 is higher than that of node 1. This information is delivered to the gateway with a delay, at time = 52 s. Therefore, errors in target node selection are encountered in this example for the dual prediction method at time steps 46–51. When analyzing the results in Fig. 18, we can observe that the number of transmissions for the dual prediction method is close to that of the proposed approach.

Further analysis was conducted to determine the level of reduction in sampled data for both the online and offline versions of the proposed approach, as well as for the global prediction method. The results of this analysis are presented in Fig. 19. The proposed method allows us to reduce up to 70% of the data sampled by sensor nodes. The lower percentage of reduced data samples (close to 60%) for the offline strategy is a consequence of the lower reduction in transmission. The correlation between sampling reduction and transmission reduction is observed in the proposed approach because each transmission requires the prior collection of c data samples.

In general, the above results confirm that the introduced method effectively reduces both the transmitted and sampled data. It should be noted that the remaining compared methods require the sensor node to acquire a data sample at each time step. Thus, the sampling reduction for these methods is 0%.

5. Conclusions

In various applications of IoT sensor networks, data transmissions are necessary to dynamically select a target node based on its sensor readings or location. The amount of data transmissions can be minimized by using well-designed prediction algorithms. This study demonstrates that transmission can be more effectively reduced by predicting a range of possible changes, rather than relying solely on predicted sensor readings. The ability to predict potential changes in sensor readings allows us to determine how long the current target node can continue its mission and when it becomes necessary to select a new target node. Thus, data transmissions are eliminated if they are unlikely to contribute to the decision regarding the selection of the target node. The experiments reported in this paper revealed that, in comparison with state-of-the-art methods, the presented approach achieves better trade-offs between reducing transmission and accurately selecting target nodes. The proposed prediction algorithm has been implemented for target node selection. However, this prediction technique can also be helpful in various applications for evaluating the future range of sensor readings. Future research directions include extending this approach to enable multivariate predictions while considering multiple sensed parameters.

CRediT authorship contribution statement

Bartłomiej Płaczek: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] O. Gulec, Extending lifetime of wireless nano-sensor networks: An energy efficient distributed routing algorithm for internet of nano-things, Future Gener. Comput. Syst. 135 (2022) 382–393.
[2] A. Jarwan, A. Sabbah, M. Ibnkahla, Data transmission reduction schemes in WSNs for efficient IoT systems, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1307–1324.
[3] P.H. Li, H.Y. Youn, Gradient-based adaptive modeling for IoT data transmission reduction, Wirel. Netw. 26 (8) (2020) 6175–6188.
[4] G. Min, L. Liu, W. Zhai, Z. Wang, W. Lu, An efficient data collection algorithm for partitioned wireless sensor networks, Future Gener. Comput. Syst. 140 (2023) 53–66.
[5] Y. Fathy, P. Barnaghi, R. Tafazolli, An adaptive method for data reduction in the internet of things, in: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), IEEE, 2018, pp. 729–735.
[6] A.N. Njoya, C. Thron, M.N. Awa, A.A.A. Ari, A.M. Gueroui, Lifetime optimization of dense wireless sensor networks using continuous ring-sector model, Future Gener. Comput. Syst. 129 (2022) 212–224.
[7] F. Vannieuwenborg, S. Verbrugge, D. Colle, Choosing IoT-connectivity? A guiding methodology based on functional characteristics and economic considerations, Trans. Emerg. Telecommun. Technol. 29 (5) (2018) e3308.
[8] P. Sarode, T. Reshmi, Study on query-based information extraction in IoT-integrated wireless sensor networks, in: Countering Cyber Attacks and Preserving the Integrity and Availability of Critical Systems, IGI Global, 2019, pp. 142–156.
[9] C. Zhu, L.T. Yang, L. Shu, V.C. Leung, T. Hara, S. Nishio, Insights of top-k query in duty-cycled wireless sensor networks, IEEE Trans. Ind. Electron. 62 (2) (2014) 1317–1328.
[10] F.T. Al-Dhief, R.C. Muniyandi, N. Sabri, M. Hamdan, N.M.A. Latiff, M.A.A. Albadr, M.H.H. Khairi, M. Hamzah, S. Khan, Forest fire detection using new routing protocol, Sensors 22 (20) (2022) 7745.
[11] L. Saadatdoost, H. Motameni, S.M. Hosseini Andargoli, R. Ghazalian, Sensing node selection and mobile sink displacement in the environments with multiple targets, IET Wirel. Sens. Syst. 11 (4) (2021) 147–160.
[12] B. Płaczek, Decision-aware data suppression in wireless sensor networks for target tracking applications, Front. Comput. Sci. 11 (2017) 1050–1060.
[13] B. Płaczek, M. Bernaś, Uncertainty-based information extraction in wireless sensor networks for control applications, Ad Hoc Netw. 14 (2014) 106–117.
[14] G. Sahar, K.B.A. Bakar, F.T. Zuhra, S. Rahim, T. Bibi, S.H.H. Madni, Data redundancy reduction for energy-efficiency in wireless sensor networks: A comprehensive review, IEEE Access 9 (2021) 157859–157888.

[15] S. Bhandari, N. Bergmann, R. Jurdak, B. Kusy, Time series analysis for spatial node selection in environment monitoring sensor networks, Sensors 18 (1) (2017) 11.
[16] C. Wang, X. Shen, H. Wang, H. Mei, Energy-efficient collection scheme based on compressive sensing in underwater wireless sensor networks for environment monitoring over fading channels, Digit. Signal Process. 127 (2022) 103530.
[17] A. Ghosh, S. Misra, V. Udutalapally, Multiobjective optimization and sensor correlation framework for IoT data validation, IEEE Sens. J. 22 (23) (2022) 23581–23589.
[18] Y.E. Hamouda, M.M. Msallam, Smart heterogeneous precision agriculture using wireless sensor network based on extended Kalman filter, Neural Comput. Appl. 31 (2019) 5653–5669.
[19] H. Yu, C. Liu, Y. Ren, N. Ji, C. Yang, Service node selection optimization for mobile crowd sensing in a road network environment, Veh. Commun. 22 (2020) 100203.
[20] D. Trihinas, G. Pallis, M.D. Dikaiakos, Low-cost adaptive monitoring techniques for the internet of things, IEEE Trans. Serv. Comput. 14 (2) (2018) 487–501.
[21] M.Z.A. Bhuiyan, J. Wu, G. Wang, T. Wang, M.M. Hassan, E-sampling: Event-sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems, ACM Trans. Auton. Adapt. Syst. (TAAS) 12 (1) (2017) 1–29.
[22] C. Santos, J.A. Jimenez, F. Espinosa, Effect of event-based sensing on IoT node power efficiency. Case study: Air quality monitoring in smart cities, IEEE Access 7 (2019) 132577–132586.
[23] M. Diaz-Cacho, E. Delgado, A. Barreiro, P. Falcón, Basic send-on-delta sampling for signal tracking-error reduction, Sensors 17 (2) (2017) 312.
[24] H. Liazid, M. Lehsaini, A. Liazid, Data transmission reduction using prediction and aggregation techniques in IoT-based wireless sensor networks, J. Netw. Comput. Appl. 211 (2023) 103556.
[25] G.M. Dias, B. Bellalta, S. Oechsner, The impact of dual prediction schemes on the reduction of the number of transmissions in sensor networks, Comput. Commun. 112 (2017) 58–72.
[26] V.W. Håkansson, N.K. Venkategowda, F.A. Kraemer, S. Werner, Cost-aware dual prediction scheme for reducing transmissions at IoT sensor nodes, in: 2019 27th European Signal Processing Conference (EUSIPCO), IEEE, 2019, pp. 1–5.
[27] F.A. Almalki, S. Ben Othman, F. A. Almalki, H. Sakli, EERP-DPM: energy efficient routing protocol using dual prediction model for healthcare using IoT, J. Healthc. Eng. 2021 (2021) 1–15.
[28] A. Fathalla, K. Li, A. Salah, M.F. Mohamed, An LSTM-based distributed scheme for data transmission reduction of IoT systems, Neurocomputing 485 (2022) 166–180.
[29] S. Suryavansh, A. Benna, C. Guest, S. Chaterji, A data-driven approach to increasing the lifetime of IoT sensor nodes, Sci. Rep. 11 (1) (2021) 22459.
[30] K. Jain, A. Singh, A two-vector data-prediction model for energy-efficient data-aggregation in wireless sensor network, Concurr. Comput.: Pract. Exper. 34 (11) (2022) e6898.
[31] F. Wu, Y. Chen, X. Chen, W. Fan, Y. Liu, An adaptive dual prediction scheme based on edge intelligence, IEEE Internet Things J. 7 (10) (2020) 9481–9493.
[32] H. Wang, Z. Yemeni, W.M. Ismael, A. Hawbani, S.H. Alsamhi, A reliable and energy efficient dual prediction data reduction approach for WSNs based on Kalman filter, IET Commun. 15 (18) (2021) 2285–2299.
[33] A.M. Hussein, A.K. Idrees, R. Couturier, Distributed energy-efficient data reduction approach based on prediction and compression to reduce data transmission in IoT networks, Int. J. Commun. Syst. 35 (15) (2022) e5282.
[34] K. Jain, A. Agarwal, A. Abraham, A combinational data prediction model for data transmission reduction in wireless sensor networks, IEEE Access 10 (2022) 53468–53480.
[35] G.M. Dias, B. Bellalta, S. Oechsner, A survey about prediction-based data reduction in wireless sensor networks, ACM Comput. Surv. 49 (3) (2016) 1–35.
[36] H. Malik, A.S. Malik, C.K. Roy, A methodology to optimize query in wireless sensor networks using historical data, J. Ambient Intell. Humaniz. Comput. 2 (2011) 227–238.
[37] G.B. Tayeh, A. Makhoul, C. Perera, J. Demerjian, A spatial-temporal correlation approach for data reduction in cluster-based sensor networks, IEEE Access 7 (2019) 50669–50680.
[38] M.A.P. Putra, A.P. Hermawan, D.-S. Kim, J.-M. Lee, Energy efficient-based sensor data prediction using deep concatenate mlp, in: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2021, pp. 1–4.
[39] M.A.P. Putra, A.P. Hermawan, D.-S. Kim, J.-M. Lee, Data prediction-based energy-efficient architecture for industrial IoT, IEEE Sens. J. (2023).
[40] A.K. Idrees, R. Alhussaini, M.A. Salman, Energy-efficient two-layer data transmission reduction protocol in periodic sensor networks of IoTs, Pers. Ubiquitous Comput. (2020) 1–20.
[41] A.M. Hussein, A.K. Idrees, R. Couturier, A distributed prediction–compression-based mechanism for energy saving in IoT networks, J. Supercomput. (2023) 1–37.
[42] C.R. Morales, F. Rangel de Sousa, V. Brusamarello, N.C. Fernandes, Evaluation of deep learning methods in a dual prediction scheme to reduce transmission data in a WSN, Sensors 21 (21) (2021) 7375.
[43] B. Płaczek, Selective data collection in vehicular networks for traffic control applications, Transp. Res. C 23 (2012) 14–28.
[44] F. Martínez, M.P. Frías, M.D. Pérez, A.J. Rivera, A methodology for applying k-nearest neighbor to time series forecasting, Artif. Intell. Rev. 52 (3) (2019) 2019–2037.
[45] T. Yu, H. Zhu, Hyper-parameter optimization: A review of algorithms and applications, 2020, arXiv preprint arXiv:2003.05689.
[46] A. Booranawong, N. Jindapetch, H. Saito, Adaptive filtering methods for RSSI signals in a device-free human detection and tracking system, IEEE Syst. J. 13 (3) (2019) 2998–3009.
[47] F.A. Aderohunmu, G. Paci, D. Brunelli, J.D. Deng, L. Benini, Prolonging the lifetime of wireless sensor networks using light-weight forecasting algorithms, in: 2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, IEEE, 2013, pp. 461–466.
[48] G. Li, Y. Wang, Automatic ARIMA modeling-based data aggregation scheme in wireless sensor networks, EURASIP J. Wireless Commun. Networking 2013 (2013) 1–13.
[49] M. Lewandowski, B. Płaczek, Data transmission reduction in wireless sensor network for spatial event detection, Sensors 21 (21) (2021) 7256.
[50] R.J. Hyndman, Y. Khandakar, Automatic time series forecasting: the forecast package for r, J. Stat. Softw. 27 (2008) 1–22.
[51] R.-E. Fan, P.-H. Chen, C.-J. Lin, T. Joachims, Working set selection using second order information for training support vector machines., J. Mach. Learn. Res. 6 (12) (2005).
[52] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.

Bartłomiej Płaczek, M.Sc. (2002), Ph.D. (2005), Habilitation (2016), is a professor at the University of Silesia, Institute of Computer Science, Poland, and an expert from the National Centre for Research and Development (NCBR). His research interests include sensor networks, smart sensors, intelligent systems, and image processing. He is the author or co-author of more than seventy papers in peer-reviewed journals and conferences and several books. He serves as a reviewer for many scientific journals, international conferences, and research foundations. Since 2010, Dr. Placzek has been a reviewer of grants and projects (including EU projects) in the field of information technologies.