Journal of Computational Science 68 (2023) 101956

Contents lists available at ScienceDirect

Journal of Computational Science

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

Federated Learning for improved prediction of failures in Autonomous Guided Vehicles

Bohdan Shubyn [a,b,*], Daniel Kostrzewa [a], Piotr Grzesik [a], Paweł Benecki [a], Taras Maksymyuk [b], Vaidy Sunderam [c], Jia-Hao Syu [d], Jerry Chun-Wei Lin [e], Dariusz Mrozek [a]

[a] Silesian University of Technology, Department of Applied Informatics, Akademicka 16, Gliwice, 44-100, Poland
[b] Lviv Polytechnic National University, Department of Telecommunications, Profesorska 11, Lviv, 79000, Ukraine
[c] Department of Computer Science, Emory University, Atlanta, 30322, GA, USA
[d] Department of Computer Science and Information Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei, 10617, Taiwan
[e] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Inndalsveien 28, Bergen, 5063, Norway

ARTICLE INFO

Keywords:
Federated Learning
Autonomous Guided Vehicles
Smart manufacturing
Prediction
Industry 4.0

ABSTRACT

Autonomous Guided Vehicles (AGVs) are nowadays an indispensable component of production lines in smart manufacturing. Managing the fleet of AGVs covers not only the delegation of operational tasks but also the monitoring of AGVs activity and health condition by applying tailored Machine Learning-based methods to detect anomalies in various signals gathered by edge IoT devices mounted on board. Detecting anomalies requires appropriate prediction of selected signals based on multiple types of sensor readings. Momentary energy consumption is one of the signals that can indicate abnormal states in AGVs. In this paper, we show that the prediction of this signal can be improved with the Federated Learning (FL) approach that involves exchanging experience gained by particular AGVs. This paper significantly extends the conference paper (Shubyn et al., 2022) with the new multi-round approach to building global prediction models and recent experiments on real data streams produced by AGVs designed by the AIUT company. The results of our experiments prove that in the AGV operational environments with distributed knowledge Federated Learning performs better than traditional centralized approaches and that frequent synchronization of experience may lead to better prediction quality.

1. Introduction

Autonomous Guided Vehicles (AGVs) are unmanned vehicles controlled by appropriate navigation systems capable of transporting production components on manufacturing lines without the need for direct operator support [1]. AGVs are used mainly in those manufacturing cases that require reducing the costs of business activities and in companies focused on continuous production optimization. Technical requirements for communication cause that AGVs primarily operate in transportation works inside factories, warehouses, office buildings, and in closed areas [2]. Moreover, AGVs perform well in cases of production involving cyclical processes, for the transport of heavy loads, as well as in the presence of hazardous conditions or conditions negatively affecting work efficiency [3].

Therefore, it is not surprising that AGVs are widely used in the field of process automation and have become an essential element of automated production in the era of Industry 4.0 [4]. Indeed, Industry 4.0 relies on automated production, where decisions are frequently made in real time. However, making decisions in the production environment requires the operation of the whole manufacturing equipment, which is constantly monitored, controlled, and coordinated. In fact, modern manufacturing relies on a complex ecosystem that consists of many elements, including various sensors, communication infrastructure, intelligent devices, various IT systems, and people. AGVs are a part of this ecosystem, which also means they are not a separate technological solution and completely independent entities but must operate in cooperation with the elements of this ecosystem to optimize the ongoing production tasks and to ensure automatic cooperation with assembly stations.

Ensuring the proper operation of the AGVs requires constant monitoring of the operational cycles they are involved in and the various types of signals they generate every second of the operational cycle. Such monitoring is a rich source of data that can be collected

* Corresponding author at: Silesian University of Technology, Department of Applied Informatics, Akademicka 16, Gliwice, 44-100, Poland.
E-mail address: boshubin@gmail.com (B. Shubyn).

https://doi.org/10.1016/j.jocs.2023.101956
Received 15 October 2022; Received in revised form 28 December 2022; Accepted 24 January 2023
Available online 1 February 2023

and analyzed to quickly recognize the potentially dangerous situations, abnormal usage, anomalous states of the vehicles or non-optimal operation of a single AGV or the whole fleet [5]. Artificial Intelligence (AI) and Machine Learning (ML) play an increasingly important role in such data analysis, especially in complex cases, where simple threshold-based alarm raising based on signal observation is difficult to implement due to the diversity of analyzed signals and features that may reflect the anomaly [6]. Moreover, since some decisions must be taken immediately after the critical anomaly is detected, AI-driven analytics should be performed at the edge IoT devices mounted on board. Edge computing or Edge analytics play a significant role in coordinating a fleet of AGVs and enabling robust production cycles. The analytics cover the development and embedding of appropriate Machine Learning (ML) algorithms in the edge IoT device to analyze the current behavior of each AGV and detect any possible problems, anomalies, or failures. Real-time analysis of the data produced by AGVs (like safety signals, navigation data, odometry information, motor activity, etc.) and advanced data exploration with ML techniques can enrich local and remote monitoring of the AGVs' health and improve the consistency of production. The experience gained by each AGV is precious, and appropriate predictive maintenance tools may detect the first signs of failure in industrial environments long before the appearance of the early alarms that precede failures of AGVs in a short period.

However, in the whole environment, AGVs operate as autonomous units with their own characteristics and sometimes implement different operational cycles in specific work environments. Thus, they experience various conditions during different or sometimes even the same operational cycles (e.g., multiple types of surfaces and surface contamination, different temperature and air humidity in the production hall or a warehouse, and different payloads on various routes for transporting components). Aggregating this experience in a natural production environment requires exchanging large amounts of valuable data, which is sometimes difficult, expensive, or could provide additional delays in detecting anomalies and reacting, which should be avoided.

Solving this problem on a broader scale and making AGVs in production more effective in anomaly and failure detection necessitates using more sophisticated ML techniques that fit such a distributed environment, where experience is dispersed among many members of the AGV society. We decided to investigate using Federated Learning (FL), which allows exchanging the experience between AGVs in the Edge computing-based system.

Federated learning emerged because much of the data containing useful information required to solve specific problems is difficult to obtain in sufficient amounts to train a powerful deep learning model. Moreover, besides useful information, training such a global model would cause transferring of additional operational data that is not relevant to the solved problem, increasing the volume of the transmitted data. The main idea of FL is that the same types of intelligent devices or AGVs working in production share their experience instead of data. Sharing experience reduces the data transfers, speeds up the building of the global view of the ongoing processes, and increases the amount of knowledge about various breakdowns of production, which allows better prediction and avoidance. Moreover, this approach increases the security of the whole solution since most of the data are stored and utilized locally on edge IoT devices. This prevents data from being stolen or intercepted as the experience is exchanged in the form of parameters of the artificial neural networks (their weights) suitable only for further processing at the highest level.

In the AGV reality, Federated Learning relies on the capability of IoT devices mounted on board to store all the data necessary for training the ML/FL local model. Therefore, there is no need to store vast amounts of training data in the monitoring data center, located, e.g., in the cloud (unless required for other purposes), which improves decentralized, edge-based data processing.

Predicting energy consumption in AGVs is essential for many reasons [7]. Abnormal energy consumption may indicate ongoing degradation of AGV components or improper usage of the AGV. For the whole AGV fleet prediction of energy consumption may decide on sending some of them to be charged, as all of them cannot be sent for charging at the same time. Thus, the prediction supports the management of the whole AGV fleet. In this paper, we show that FL improves the effectiveness of prediction of energy consumption performed on edge IoT devices by iterative creation of a global prediction model based on many local prediction models of particular AGVs and retraining the local models with new data. Before we start, we will review the related literature in terms of the use of Machine Learning models in smart industry and predictive maintenance (PdM) in Section 2. Section 3 provides characteristics of the Autonomous Guided Vehicles and Section 4 explains their industrial environment and data acquisition flow. In Section 5, we describe our FL-based approach for anomaly detection in AGVs, which relies on sharing and exchanging experience between edge devices mounted on AGVs. Section 6 presents results of experiments we conducted to appropriate strategy for sharing the experience of AGVs and increasing the performance of local energy consumption prediction models in a smart manufacturing environment. Finally, Section 7 summarizes our work with the discussion of obtained results in the context of traditional ML-based techniques and related works.

2. Literature review

2.1. Smart industry

The explosive growth of information and communication technology fuels the ever-growing development of smart industry and manufacturing [8,9]. Internet of Things (IoT) devices and advanced communication technologies facilitate the interconnection between machine–human (M2H) and machine–machine (M2M), enabling real-time monitoring and automatic control without human intervention. Additionally, artificial intelligence (AI) algorithms are adopted to analyze, diagnose, and predict industrial environments. Through data mining patterns and machine learning predictions, AI algorithms in innovative industries can achieve self-monitoring [10,11] and improve efficiency [12], and attract widespread attention in various industrial fields, including manufacturing [9,13,14], energy generation [15,16], robotics [17,18], and finance [19–21].

Machine learning (ML) is a subset of AI designed to build models, learn from data, and make predictions. ML is actively and widely used in various industries; Kotsiopoulos et al. [14], and Sharp et al. [22] stated that ML can improve the agility and energy efficiency of manufacturing systems and further optimize the production process. Traditional machine learning algorithms are lightweight models with low computational complexity and computational time, including regression, decision trees, random forests, support vector machines, etc [23].

With the exponential growth in computational capacity, numerous deep learning (DL) algorithms have been developed for robust feature extraction and accurate prediction. Recurrent Neural Networks (RNNs) are designed to memorize and analyze the temporal behavior of input data. Extended RNN architectures include Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), which are feasible and widely used in the smart industry. Wang et al. [24] proposed a hybrid prediction scheme for intelligent manufacturing, which consists of a novel deep heterogeneous GRU model and a local feature extraction mechanism. Essien and Giannetti [25] developed a novel autoencoder with deep convolutional LSTM neural networks for machine speed prediction, which employs a sliding window approach to reconstruct input sequences into a supervised learning framework. Convolutional Neural Network (CNN) is a shared weight architecture that can extract features from different scales and filters and exhibits outstanding performance in image processing and computer vision [26]. Melinte and

Vladareanu [27] utilized various CNN-based networks (VGG, Inception, ResNet, and Faster R-CNN) for facial expression recognition of human–computer interaction. Luo et al. [28], and Benjdira et al. [29] developed a CNN-based image detection model for unmanned aerial vehicles.

However, industrial circumstances may not have correct and deterministic answers. Instead of supervised learning, reinforcement learning (RL) lets intelligent agents take appropriate actions to maximize rewards and learn from trial-and-error in the environment. Zhou et al. [30] developed a multi-agent RL-based model for online scheduling in smart factories. Faryadi and Mohammadpour Velni [31] proposed an RL-based approach for modeling unknown fields for autonomous ground vehicles. Wu et al. adopted RL for financial portfolio establishment [21] and risk management [32]. The literature mentioned above demonstrates the considerable research attention to AI technology in smart industries.

2.2. Predictive maintenance

When it comes to anomaly detection, predictive maintenance is a critical issue in smart manufacturing by monitoring production status to discover potential anomalies (atypical states) and providing early warnings that seriously affect production quality, efficiency, and even safety. Predictive maintenance is designed to maximize the service life of components in manufacturing equipment, avoid unplanned downtime, and minimize planned downtime [33]. Traditional predictive maintenance is based on constant time or unquantified experience, which can be significantly improved by AI technologies of IoT sensors [33,34], web platform [35], and ML models [36,37]. Javed et al. [37] utilized ensemble, attention, RNN, and CNN techniques for anomaly detection in automated vehicles. Ahmad et al. [38] proposed an unsupervised learning algorithm with hierarchical temporal memory to detect anomalies in streaming data, which outperformed the traditional LSTM and GRU models on the real-world data streams, Numenta Anomaly Benchmark (NAB). Malawade et al. [39] developed neuroscience-inspired algorithms for predictive maintenance, which also employed hierarchical temporal memory and were evaluated on NAB.

Unsupervised anomaly detection can be seen as a one-class classification problem. While learning, models fit the non-anomalous data. In the next step, new data is evaluated by the model to generate a number called an anomaly score, which is used to decide whether the upcoming samples can be considered anomalous [40,41]. As time series is a specific input data for anomaly detection, the process is usually started from forecasting the expected following values of the series [42,43]. Those are compared with actual upcoming values, and the error is analyzed in further processing [42].

Last years brought many methods based on machine learning, especially on RNNs used for signal forecasting in many applications, among them: finance [44], signal prediction [45], finding anomalies in satellite telemetry [46,47] and medicine [48]. Multivariate approach for forecasting using LSTM, GRU, and derived bidirectional models was studied in [49]. In this paper, we focus on univariate signal prediction used together with federated learning-based training.

Distributed learning and federated learning [50] are potential techniques for predictive maintenance. They are decentralized learning methods with high parallelism, data privacy, and security, and state-of-the-art research for smart manufacturing [51]. Zhang et al. [51] introduced a real-time tuning architecture with two-level deep federated learning and a real-time automatic configuration tuning mechanism, where local servers obtain experience and share it with cloud servers and then aggregate knowledge to build a robust federated model. Our previous work related to FL-based anomaly detection published in [52] verified the usefulness of this technique on the mentioned NAB data set and analyzed different architectures for averaging shared experience. However, it did not explore the data streams from the real AGV environment. This work goes one step forward, showing that flattened architecture for FL is sufficient for obtaining high-quality predictions and that appropriate upgrading of prediction models increases the quality.

3. AGV characteristics

AGVs may differ in allocation and construction, but they also share the general purpose they were invented for. Before we go further into the details of our approach, we will provide more information about the AGV environment we performed our industrial tests.

3.1. The test vehicle

Our experiments use an AGV developed by AIUT Ltd. in cooperation with the Silesian University of Technology (SUT) in Gliwice, Poland [53]. While the company provides many versions of the vehicle, the paper focuses on Formica-1 (see Fig. 1). It is an AGV capable of moving specialized passive trolleys or carrying a general payload. In the first case, it is equipped with special lifting pins which interact with a trolley. The latter situation requires a lifting plate to be mounted on top of the vehicle.

With its own weight of ca. 250 kg, Formica-1 can handle up to 1000 kg of payload. Its battery (a single module, the same as used in electric buses) is sufficient for at least 8 h of continuous operation.

3.2. Data provided by AGV

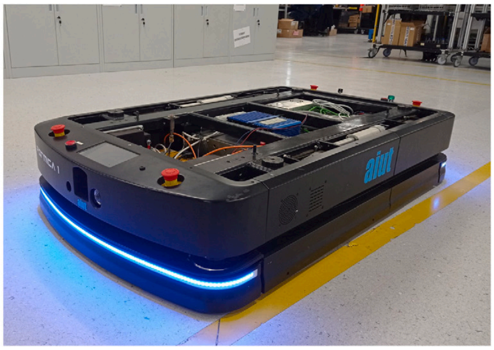
The tested AGV provides a variety of signals and statuses. The groups of signals are as follows:

- energy measurements - floating-point values representing momentary energy consumption (MEC) in [W] and battery cell voltage in [mV]; the energy consumption is measured only at the battery, with no separate values on motors and electronics;
- motors (separately left and right) - boolean statuses of engines - if a motor is active and how it has been activated (manual/automatic);
- odometry - numbers showing momentary frequencies of left/right encoder pulses and cumulative distance on each wheel;
- brakes - boolean values which indicate the current state of brakes - whether they are active and how they have been activated (manual/automatic);
- payload lifting - boolean states of pins (used for passive trolleys) or lifting plate for another type of payload;
- status of LED strips - booleans indicating activity of specific light strips; especially blue LED consume considerable amounts of energy;
- natural navigation and path control - numbers representing coordinates, heading, speed, position confidence, and path segment id;
- alarms and warnings - boolean values indicating various system failures and safe-related situations, e.g., problems with internal Profinet communication, no data from natural navigation, uneven distribution of payload, errors of safety devices (safety scanners, bumpers);
- safety signals - violation of safety zones and activation of bumpers;
- weight strains (depending on AGV version) - four weight gauges measuring the amount and location of the payload;
- driving modes - if the vehicle is manual/automatic/docking mode.

3.3. The dataset

To test the suitability of the Federated Learning approach in the prediction of the AGV energy consumption, we gathered the data from nine test runs with an average sequence length of ca. 1600 data points. In total, the whole dataset contained ca. 14,000 timestamps. The frequency of data was 1 Hz.

Test drives were executed using the following scenarios: repeated circular clockwise and counter-clockwise paths of diameter 2 m with a speed of 0.25 $\frac{m}{s}$, driving forward and backward with the speed of 0.2



(a) AGV without the lifting plate

(b) half-loaded complete AGV

Fig. 1. Formica-1, the AGV used as a source of data and the test device.

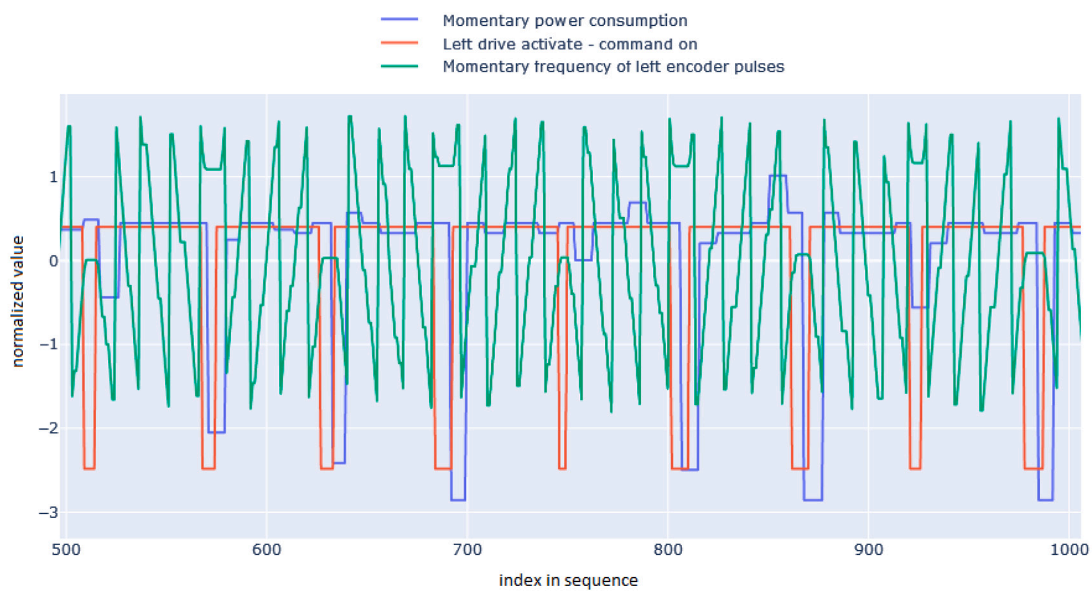


Fig. 2. Fragment of a time series from the dataset (run 3). Momentary energy/power consumption (MEC) and a few selected features are presented. Values were normalized to fit on a chart.

$\frac{m}{s}$, repeated emergency braking and fast acceleration in both directions, lifting plate up and down movements; each of those scenarios was executed on an empty vehicle and with half-loaded payload compartment (425 kg). Total weight of the vehicle correlates positively with energy consumption, so it needs to be considered. The experiments presented in Section 6 make use of all test drives without creating boundaries between runs, so the energy prediction model learns from data acquired on different load levels. The version of AGV used while testing was not equipped with weight strains, so the model learns this relation implicitly. Each sequence consists of more than 40 features mentioned above. However, in this paper, we deal only with univariate forecasting of momentary energy consumption. An example fragment of the time series containing MEC can be seen in Fig. 2.

4. Industrial environment and data acquisition flow

AGVs operate in industrial conditions and cooperate with several IT systems. This necessitates dedicated solutions for communication and data acquisition. This section provides an overview of the operational environment.

4.1. Industrial environment

The considered industrial environment is based on the model of two separate production factories equipped with two independent AGV systems. AGVs perform their regular operational activity while communicating with various IT systems in the manufacturing environment, including the Transport Management System (TMS, used for transportation operations), the Manufacturing Execution System (MES, which optimizes manufacturing operations and production efficiency), the Warehouse Management System (WMS, which supports inventory management), and Automation systems and field devices. Apart from these systems, the activity of the vehicles is monitored by Analytical systems that independently exist next to the production and warehouse management systems. The FL-based prediction methods we designed and developed are integrated with the analytical systems. The analytical system monitors the signals from Formica-1 AGV (described in detail in Section 3.1). It consists of three main components: Edge Federated Learning module, Edge Data Acquisition module, described in more detail in Section 4.2, and OPC UA Server. The OPC UA Server defines to its clients (i.e., OPC UA Clients) the set of services it offers and the process data format that it uses for communication. Each AGV communicates with external services running in the cloud environment, which hosts the data center. The cloud platform consists of a centralized industrial Data Lake for data storage, the IoT Hub service, which is responsible for maintaining communication between edge and cloud, and the Cloud Federated Learning module (for building a global prediction model). Fig. 3 presents the conceptual diagram of the considered industrial environment.

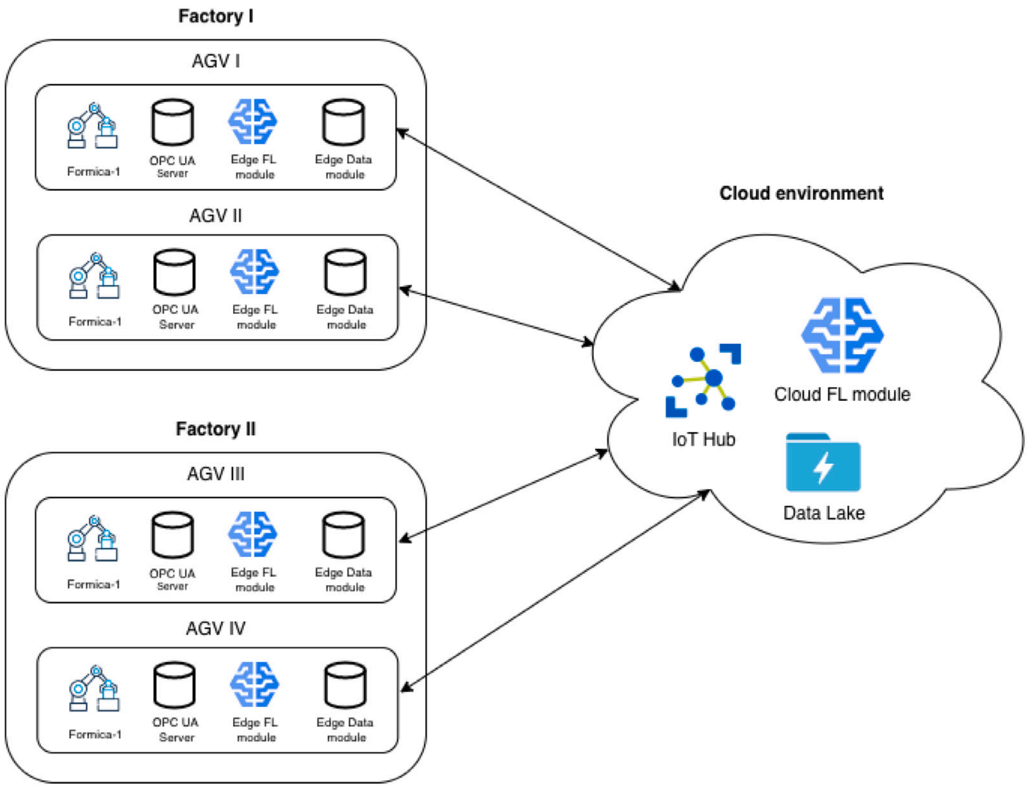


Fig. 3. Conceptual diagram of the industrial environment.

4.2. Data acquisition flow

The signals provided by AGVs, described in Section 3.2, are exposed by AGVs via OPC UA Server, which integrates signals from various internal sources, such as PLC controller, proximity sensors, and more. The OPC UA Server is running directly on the AGV. In order to integrate with it, a dedicated OPC UA Client module was implemented to retrieve the data for further analysis and use in an edge-based federated learning approach. The OPC UA Client is a part of the Edge Data module presented in Fig. 3, which runs on a separate edge device mounted on the AGV. This module is responsible for connecting to OPC UA Servers on AGVs, managing subscriptions to selected data nodes, retrieving data, pre-processing and aggregating it, and storing it in a local database before sending the required information (incl. parameters of local prediction models built on AGVs) to the cloud environment for long-term storage and further processing (e.g., building a global prediction model). The locally stored data are then used to train local models on the edge device mounted on an AGV, described in more detail in Section 5.

The proposed implementation consists of two different approaches to data retrieval: "subscription-based" and "periodical fetch." The first of them, called "subscription-based," relies on the built-in subscription mechanism of the OPC UA Server. Subscriptions allow us to separately subscribe for updates to each data node in the OPC UA Server, resulting in a precise and granular stream of updates for each considered data signal. It also reduces the number of unnecessary calls to the OPC UA Server. The downside of this approach is that since each data node is observed separately, the corresponding data streams are updated independently and have to be joined together to produce a view of the whole system at a given point in time, which is required to train local prediction models. To obtain such a complete view of all signals in a particular moment, we implemented a dedicated periodic aggregator module that produces such a view for predefined time windows. In the case of our setup, the view was produced every second. After periodic aggregation, the data are saved back in the local database. Fig. 4 presents the components of the "subscription-based" approach.

In order to simplify the "subscription-based" approach, especially to speed up and ease obtaining the data from initial experiments, we also implemented another approach called "periodic fetch." In this approach, instead of relying on a subscription mechanism, the data are retrieved and persisted in the local database for each signal at predefined time intervals (time windows). Thanks to that, there is a corresponding snapshot of the state of the whole AGV system for given points in time. As the data are already aggregated, they only need minimal processing and cleaning before they can be considered for further analysis and building local prediction models. This approach also has its downsides, such as inducing excessive load on OPC UA servers or producing redundant data. The components of the "periodic fetch" approach are presented in Fig. 5. In the future, we aim to still use a combination of both approaches, but with a more significant focus on a "subscription-based" approach, due to the benefits listed above.

However, this architecture was also used to compare the FL-based approach with the traditional ML-based approach, relying on the view of all the data that requires sending all the data to the cloud data center.

5. Federated learning for Anomaly detection in AVGs

Anomaly detection is identifying anomalous observations that do not fit the expected pattern of other observations in a data set. Anomaly detection has become a central research issue for intelligent devices (particularly AGVs) in the smart manufacturing environment.

Anomalous data can indicate significant and critical incidents that may require urgent attention. Anomaly detection is an essential concept in data analysis and is widely researched. In intelligent manufacturing, an anomaly is considered an unexpected change in the state or behavior of an Industrial IoT (IIoT) system that deviates from the norm.

Any sudden failure of the machine will lead to an undesirable loss of quality and productivity. Anomaly detection helps alleviate and reduce these problems. However, the limited availability of historical data and

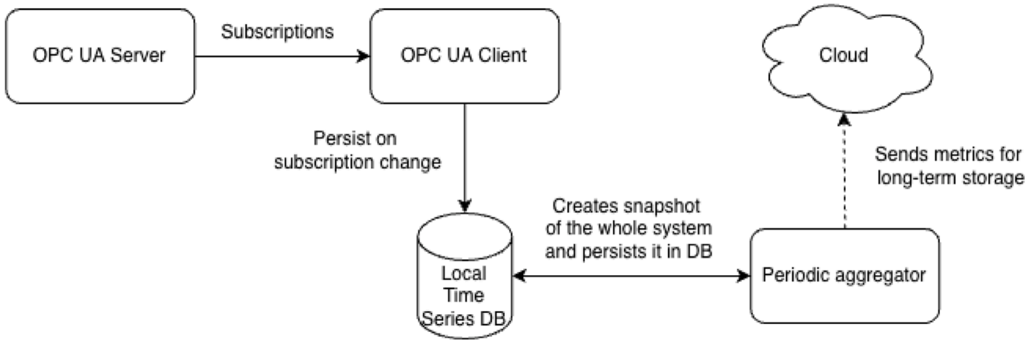


Fig. 4. Diagram of a "subscription-based" approach to data acquisition from OPC UA Server.

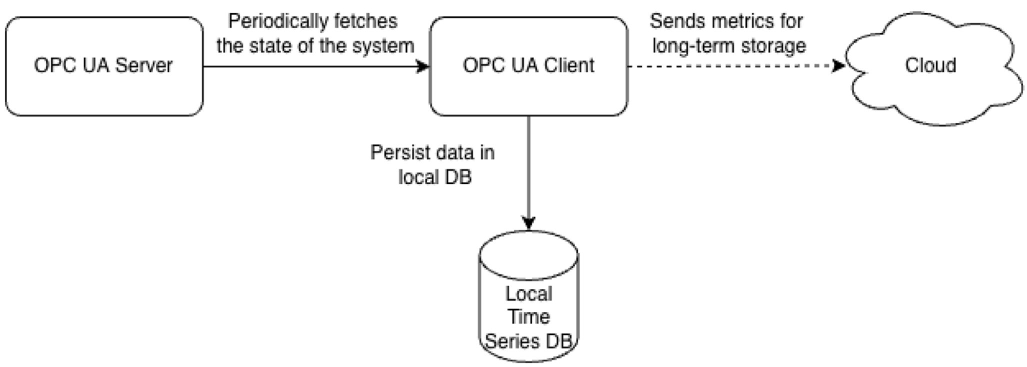


Fig. 5. Diagram of a "periodic-fetch" approach to data acquisition from OPC UA Server.

the security of industrial data make anomaly detection a challenging and complex process in an intelligent manufacturing environment.

It is often hard to get data with labeled anomalies, which is also the case for AGV-acquired data. Thus it is crucial to have an unsupervised method to classify data as anomalous. It can be seen as a one-class classification problem, where the only class represents non-anomalous data. One of the typical anomaly detection schemes involves the analysis of error between the actual and forecasted value of an investigated feature [42]. The forecasting model learns from data describing normal behavior and then is fed with upcoming time series to generate the expected sequence. Compared to the actual observations, this allows predicting anomaly score [46]. A suitable forecasting method is pivotal for successful anomaly detection [49], and we focus on that aspect in this paper.

To detect anomalies more accurately, we suggest sharing experiences between AGVs using approaches based on Federated Learning. Federated Learning is a machine learning technique that allows training machine learning models without moving data from the devices where this data is generated. Therefore, it has the inherent characteristics of preserving data privacy and reducing the amount of transferred data. These characteristics are required for industrial IoT environments that need data processing solutions working in real-time. So, the main idea of FL is that smart elements or AGVs of the same type being in production can share experiences to increase the amount of knowledge about various breakdowns, which will allow us to predict them more accurately.

We have been introducing this idea in the production environments operating based on the fleet of AGVs that the AIUT company in Poland manufactures. Fig. 1 shows the loaded Formica-1 AGV we have been supplementing with edge-based AI/FL methods.

5.1. The overview of the Federated Learning-based prediction algorithm

The complete process of exchanging experience between AGVs will be called a *round*. The round operates according to Algorithm 1 (also graphically visualized in Fig. 6). First, each AGV trains a local model on a specific data set locally (lines 1–4). In the second step, all AGVs send updated local models to the server in the cloud (line 5). Next, all local models are averaged on the server to create a global model that takes into account the experience of all AGVs (lines 6–10). Finally, the server sends the updated global model back to the AGVs to update their local model with the new global model (lines 11–13).

Algorithm 1: Algorithm of the round

Data: m_i (Local model on AGV), M (Global model), $AGVs$ (the fleet of AGVs), N (the number of AGVs), SM (Server with a global model), LM_{inf_i} (the influence of the local model in the formation of the global model), X_{lm_i} (calculated values of MSE or Validation losses for specific AGV)

Result: $upAGVs$ (AGVs updated by global model)

1 **for** $i \leftarrow 1$ **to** N *(on AGV_i)* **do**
2   Train the RNN_i of AGV_i locally on unique, AGV-specific data;
3     $m_i \leftarrow$ weights of the local RNN;
4   Calculate X_{lm_i} on training data;
5   Send m_i and X_{lm_i} to the SM;
6 $X_{sum} \leftarrow \sum_1^N X_{lm_i}$;
7 **for** $i \leftarrow 1$ **to** N **do**
8     $ratio_i \leftarrow X_{lm_i}/X_{sum}$;
9     $LM_{inf_i} \leftarrow$Formula (3) $(ratio_i)$;
10 $M \leftarrow$ average of m_i-s with the influence of LM_{inf_i} by using Formula (4);
11 **for** $i \leftarrow 1$ **to** N **do**
12     $m_i \leftarrow M$;
13   Send m_i to AGV_i;

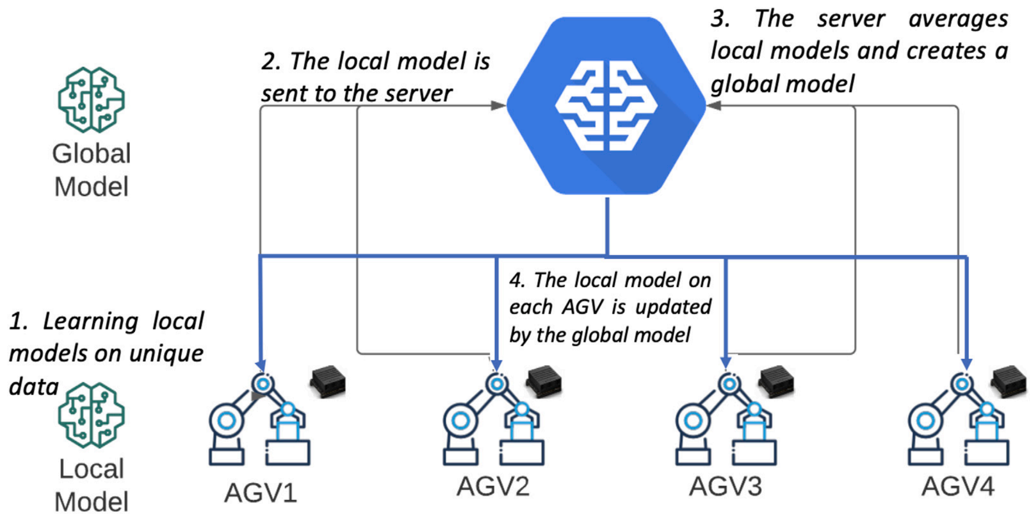


Fig. 6. The complete process of exchanging data between devices (Round).

5.2. The averaging of local models

A key element affecting the global model's performance is how we will average the local models. When forming a global model, it is necessary to correctly determine local models' influence. We want local models that perform better to impact the global model more significantly than local models that perform worse.

Considering that we will not have test data in real life, we decided that we should use available training data of the AGV (including historical training data that may not be used for training in a particular round but is still stored on the device) to determine the influence of local models. In this way, we will be able to determine which models are better trained and make a more significant impact on the global model.

In order to compare the effectiveness of local models, we decided to use the values of MSE (Formula (1)) and validation losses (Formula (2)) calculated as a result of running local models on training data.

$$MSE = \frac{\sum_i^n (y_i - \hat{y}_i)^2}{n}, \tag{1}$$

where y is the observed value, $\hat{y}$ is the corresponding predicted value and n the number of observations.

$$Val_{loss} = \frac{1}{n}\sum_i^n f(\hat{y}_i, y_i), \tag{2}$$

where f is the loss function. Validation loss is a metric used to assess the performance of a deep learning model on the validation set. The validation set is a portion of the dataset to validate the model's performance. The validation loss used a similar function to the training loss and is calculated from a sum of errors (differences $(y - \hat{y})$) for each observation in the validation set.

Based on one of these values, we will determine the influence of a specific local model when forming a global model using the Formula (3).

$$LM_{inf_i} = \frac{1 - (X_{lm_i}/X_{sum})}{N - 1}, \tag{3}$$

where X_{lm_i} is calculated values of MSE or Validation losses for specific AGV, X_{sum} is a sum of X_{lm_i} from all clients, and N is the number of AGVs.

Next, to obtain the global model, we use the averaging formula (4).

$$M = \sum_i^N m_i * LM_{inf_i} \tag{4}$$

where m_i local model on AGV_i, and $*$ is the point-wise multiplication.

In Section 6, we will use both MSE values and validation losses in Formula (3). In order to compare the obtained results and determine which of these values is more appropriate to use when averaging local models. We will also use the Mean Absolute Percentage Error (MAPE) metric that defines the accuracy of a forecasting method:

$$MAPE = \frac{1}{n}\sum_{i=1}^n \left|\frac{(y_i - \hat{y}_i)}{y_i}\right|. \tag{5}$$

The meaning of the variables is the same as for Formula (1).

6. Experimental results

We conducted several experiments with the real Formica-1 AGVs, obtaining various data from them, including momentary and cumulative power consumption, battery cell voltage, motor RPM, energy consumption and current consumption, cumulative distances, bearing temperatures, transportation pin actuator signals, and momentary frequencies, as described in Section 3.2.

For experiments reported in this article, we built a simulation environment in which we created several virtual clients that play the role of AGVs. For each of these virtual clients, we loaded the data from real AGVs. We focused on the data with "Momentary energy consumption" and tried to predict this value over time.

Energy consumption is one of the essential monitored parameters for the proper operation of many production machines. For example, changes in the energy consumption of some motors may suggest its failure and, consequently, the shutdown of the production machine or increased energy consumption and shorter operating time of the AGV.

For this study, we used FL architecture with the AI/FL implemented on the IoT device monitoring the AGV. This option does not require additional local servers for separate production lines [52]. It also provides better security for industrial data, as all the data will be processed locally on the devices and will not be sent anywhere, reducing the communication needs (and the amount of transferred data).

6.1. Choosing artificial neural network model

Given the fact that we work with time series, we decided to use recurrent neural networks (RNNs). Therefore, we focused on modified RNN architectures based on long short-term memory (LSTM) cells.

A key component of the LSTM cell-based architecture is the state of the cell. It goes directly through the whole cell, interacting with several operations. The information can easily flow on it without any changes. However, LSTM can remove information from the cell state using

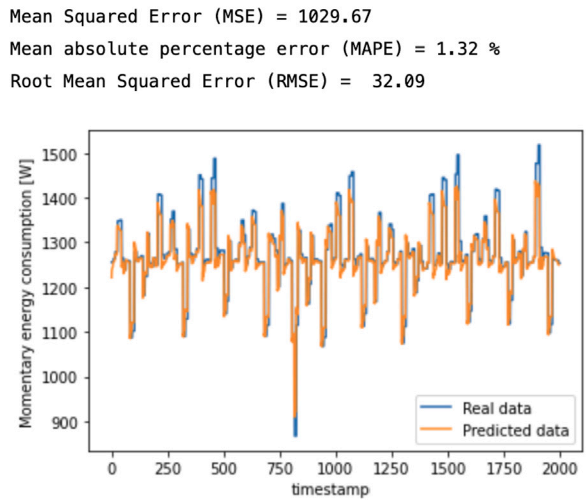


Fig. 7. Prediction performance of a single LSTM model trained on the whole training data set.

filters. Filters allow skipping information based on some conditions and consist of a sigmoid function layer and element-multiplication operation. LSTM is well-suited to predict time series given time lags of unknown duration. We used a back-propagation approach to train our model.

We decided to train our model on all available data, which we will use later for Federated Learning to prove that FL allows us to get a better result of predicting values of "Momentary energy consumption" over time. The results of the prediction of the LSTM model without FL on test data are presented in Fig. 7.

6.2. Experiment 1. Averaging based on MSE or validation loss

In this experiment, we wanted to check whether it is more appropriate to determine the influence of local models according to Formula (3). This formula can be based on the MSE values obtained from running the local model on the client's training data or on the validation losses obtained during the training of local models.

6.2.1. Splitting of data

We divided the data set into four main parts. The first three parts of the data (each one of them was 28.6 percent of the data set) were used as training data for three different virtual clients. Then, we used the last part of this data set (14 percent) to test the efficiency of local models from the virtual clients and the global model to compare their effectiveness with each other.

6.2.2. Experiment summary

The whole experiment was organized as follows:

1. We divided the whole data set into four parts. Three parts were used to conduct training on virtual clients (using the LSTM). The fourth part of this data set was used to test the effectiveness of the local models.
2. On each virtual client, we conducted the training and saved the trained model in the form of weights of neural networks.
3. The models of these three virtual devices were transferred to a separate server, which averaged the models and created two different global models based on the MSE and Validation loss values.
4. Based on the two global models, we predicted the Momentary energy consumption for a test set and compared their effectiveness.

6.2.3. Model averaging process

It is also essential to describe the process of averaging:

1. We calculate the MSE value after running the local model on training data or validation losses calculated from training the local model.
2. Using Formula (3), we find the influence of each local model separately on the formation of the global model. Formula (3) calculates the ratio of validation losses (or MSEs), inverts, and normalizes it by dividing them by N-1 (N - number of virtual clients).
3. When creating a global model, the neural weight of each client is multiplied accordingly by the influence value of this local model (Formula (4)).

6.2.4. Evaluation of models' effectiveness

To verify the effectiveness of the built models, we used the Mean Squared Error (MSE). The effectiveness of the local and global models is shown in Fig. 8.

6.2.5. Results

The obtained results demonstrate that both global models work better than local models from clients on the test data set. This is achieved by exchanging experiences between clients. Also, we decided to compare the global model results based on the values of MSE and the validation losses. We can see that we got better predictions (lower MSE) by averaging the global model using the MSE values. It is also worth noting that the validation loss value is better optimized when there is a large amount of data and a large number of training epochs of the LSTM network. Therefore, with a limited amount of data, we recommend using the MSE values by running models on training data.

6.3. Experiment 2. Multi-round averaging

In this experiment, we tested the performance of Federated Learning when we split the client's data for use in different rounds. We wanted to check how the global model would change with each round since we would train local models upgraded by the global model with the new data for each round.

We conducted several independent experiments and split our data to be able to produce 2/4/8 rounds. In general, this experiment was organized as follows:

1. We divided the data available for each client into two/four/eight parts (depending on the number of rounds). We trained local models on the first part of the data.

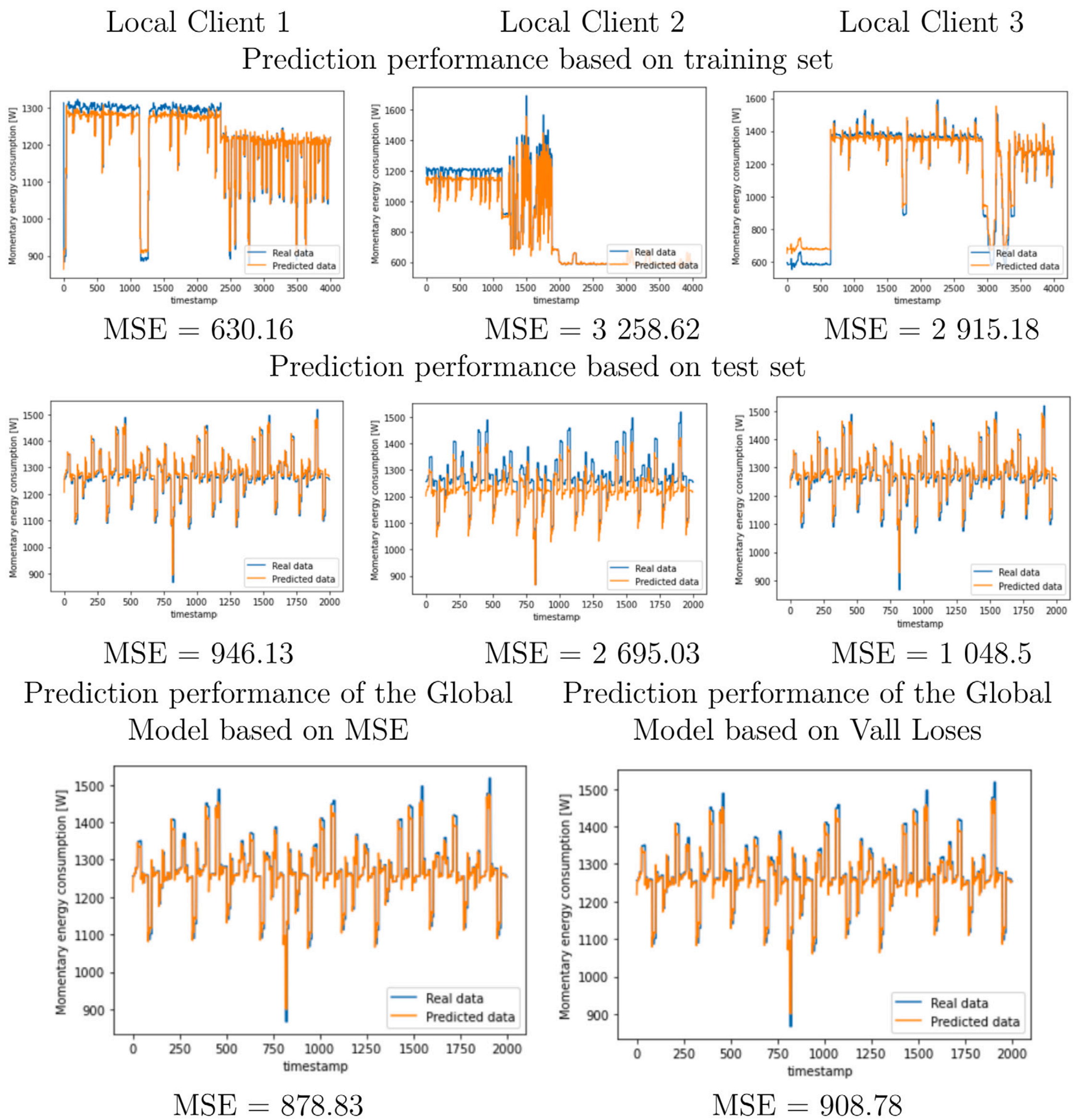


Fig. 8. Prediction performance for Local (LM) and Global Models (GM) for experiment 1.

2. We performed averaging to obtain the first global model.

3. Based on the first global model, we conducted client training by using the second part of the client's data.

4. We averaged the new local models to obtain the second global model.

5. We repeated steps 3–4 until we obtained the assumed number of rounds.

We also divided this experiment into two sub-experiments. We had two options for running the local models and calculating MSE values for averaging. We calculated the MSE values for averaging by running local models on the training data from specific rounds or on historical training data (this data includes not only the data used for training in the particular round but also the data used for training in previous rounds).

6.4. Multi-round averaging based on training data

In this experiment, we divided our client data to conduct 2, 4, and 8 rounds of Federated Learning separately. For further averaging, we calculated MSE values by running the local models only on the training data with which they were trained in a specific round.

6.4.1. Two-round averaging based on training data

As a first step, we split our data to run two rounds. So, we have data on our three clients. We divided this data into two parts. We used the first part of the data to train the local models of clients in the first round and conducted averaging. As a result, we get the global model of the first round. We used the second part of the data set from clients in the second round when our clients had the global model of the first round loaded on them. When training on the second data set, the local models uniquely changed according to training on the client's unique training data. Moreover, after the second round of averaging, we obtained a more powerful global model because we added a lot of new experiences from the clients. The prediction performance of the local models on the training data and the global model on the test part of data after the first and second rounds are presented in Fig. 9.

As a result of this experiment, we can see that the global model after the second round has become much more accurate on the test data than the global model after the first round. We obtained this result because more client data were considered when averaging the global model after the second round.

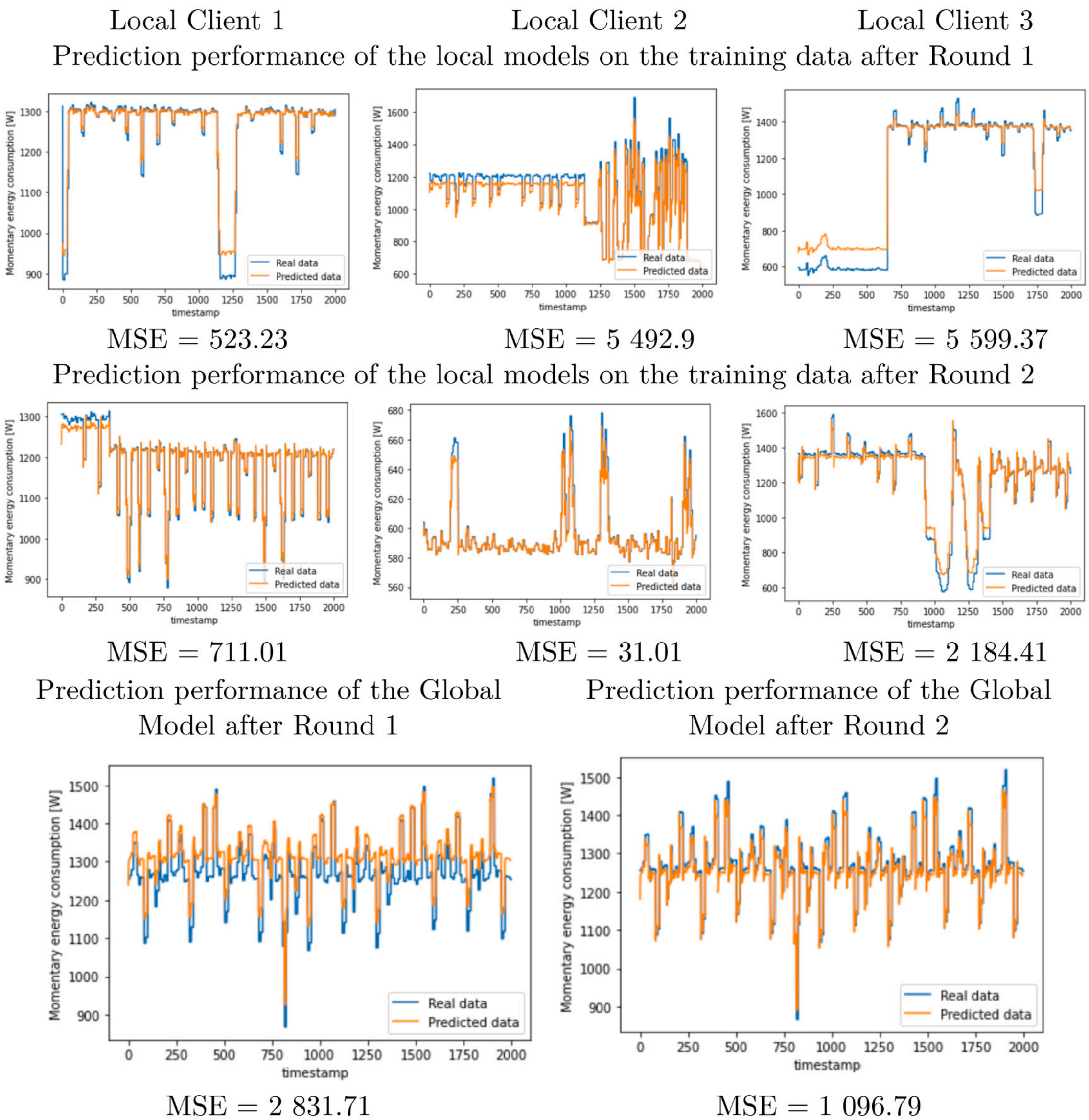


Fig. 9. Prediction performance for local and global models in 2-round averaging based on training data only.

Table 1
Prediction performance for each Round in 4-round averaging based on training data only.

Round №	MSE	MAPE
GM Round 1	2898.24	3.89%
GM Round 2	1422.04	2.28%
GM Round 3	1004.86	1.32%
GM Round 4	868.15	1.02%

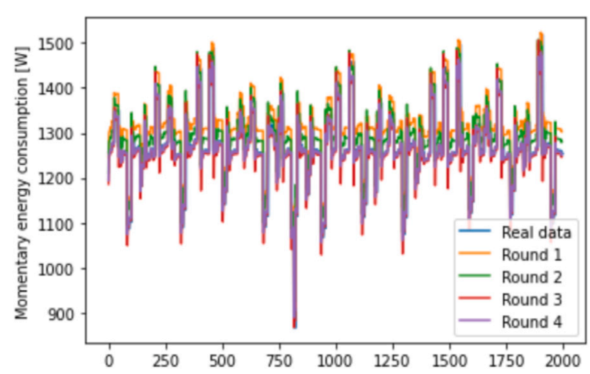


Fig. 10. Effectiveness of the global models for each Round in 4-round averaging based on training data only.

6.4.2. *Four-round averaging based on training data*

In the second part of the experiment, we divided data from our clients into four parts and conducted four rounds of averaging. At each round, the MSE was calculated only based on the training data that was used to train the local models at that particular round. The result of the global models after the 1st, 2nd, 3rd, and 4th rounds can be seen in Fig. 10 and in Table 1.

From this part of the experiment, we can see that the global model becomes more and more powerful with each round and each new batch of new data. Despite using the same amount of data overall for the 2- and 4-round experiments, we obtained a significantly better prediction result for the final global model for the 4-round variant. This can be explained by the fact that we performed more averaging of the experience. Due to this, we obtained a better global model since we modified the local models according to the new data each time through the averaging process.

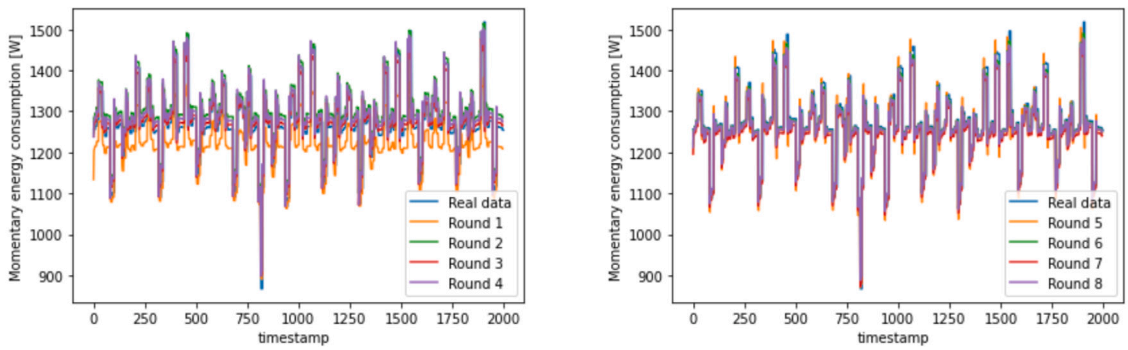


Fig. 11. Effectiveness of the global model after Round 1, 2, 3, 4, 5, 6, 7, and 8 in 8-round averaging based on training data only.

Table 2
Prediction performance for each Round in 8-round averaging based on training data only.

Round №	MSE	MAPE
GM Round 1	3923.97	4.24%
GM Round 2	1646.56	2.62%
GM Round 3	933.92	1.37%
GM Round 4	1202.75	1.94%
GM Round 5	956.82	1.23%
GM Round 6	840.35	0.92%
GM Round 7	1096.57	1.66%
GM Round 8	890.85	1.12%

6.4.3. Eight-round averaging based on training data

In the third part of the experiment, we divided data from our clients into eight parts and conducted eight rounds of averaging. The results of the global models after the 1st, 2nd, 3rd, 4th, 5th, 6th, 7th, and 8th rounds can be seen in Fig. 11, and Table 2.

In general, we can see that the prediction performance of the global model after round 8 is quite similar to what we got in the part of the experiment when we used only four rounds. This result indicates that the global model was optimized well in both cases. However, we can also see that the performance of the global model does not improve steadily, and we can see some jumps when it gets worse, such as in round 4. This is because, in this experiment, we used only training data from the specific round for running local models and calculating MSE for averaging. In this regard, the values of MSE between different local models could not be highly correlated with each other, for example, due to the small dimensionality of the data in a specific part of the data set, which led to a small value of MSE. Therefore, we decided to conduct another experiment. To calculate the MSE for local models, we used not only new available training data on the device but also historical training data that were used on the client in the past rounds.

6.5. Multi-round averaging based on extended training data

In this experiment, we divide our client data to conduct 2, 4, and 8 rounds of Federated Learning separately. However, at this time, for calculating MSE values for further averaging, we ran local models of clients, not only on the training data with which they were trained in a specific round but also all available training data on the client. In our case, it will be data from the current round and data from the previous rounds.

6.5.1. Two-round averaging based on extended training data

As a first step, we split our data to run two rounds. We used the first part of the data to train the local models of clients in the first round and conducted averaging. As a result, we got the global model of the first round. The prediction performance of the local models and the global model after the first round is the same as in experiment 6.4.1 for the case with two rounds. The main changes happened in the second round because we ran local models on all available training data and used updated values of MSE to make more optimized averaging of local models. Prediction performance of the local models and global model after the first and second rounds are presented in Fig. 12.

Table 3
Prediction performance for each Round in 4-round averaging based on extended training data.

Round №	MSE	MAPE
GM Round 1	2898.24	3.89%
GM Round 2	1087.32	1.74%
GM Round 3	1209.24	1.79%
GM Round 4	875.07	1.16%

As a result, for this part of the experiment, we can observe that the prediction performance of the global model after round 2 is much better when we take the MSE value from running local models on the entire training data set than when we used the MSE value by running local models only on the training data set data of a specific round. Therefore, by calculating MSE values for further averaging on larger data sets, we can obtain more qualitative estimates of the performance of their predictions.

6.5.2. Four-round averaging based on extended training data

In the second part of this experiment, we divided data from our clients into four parts and conducted four rounds of averaging. The MSE was calculated based on the training data used to train the local models at current and previous rounds. The result of the global models after the 1st, 2nd, 3rd, and 4th rounds can be seen in Fig. 13 and Table 3.

Results obtained in this part of the experiment are close to those we got in the part of the experiment when we calculated MSE values for further averaging by running local models only on the training data of a specific round. However, these results are still better than the result of the model without averaging or when using two rounds of averaging. It also proves that we can get a global model to make better signal predictions using Federated Learning.

6.5.3. Eight-round averaging based on extended training data

In the third part of the experiment, we divided data from our clients into eight parts and conducted eight rounds of averaging. The results of the global models after the 1st, 2nd, 3rd, 4th, 5th, 6th, 7th, and 8th rounds can be seen in Fig. 14 and in Table 4.

We obtained the best prediction performance from the result obtained in this part of the experiment. This proves that exchanging experience between AGVs more frequently allows for better prediction of energy consumption. It also proves that it is better to calculate MSE values for further averaging by training local models on all available training data on the device, as it allows more efficient averaging of local models.

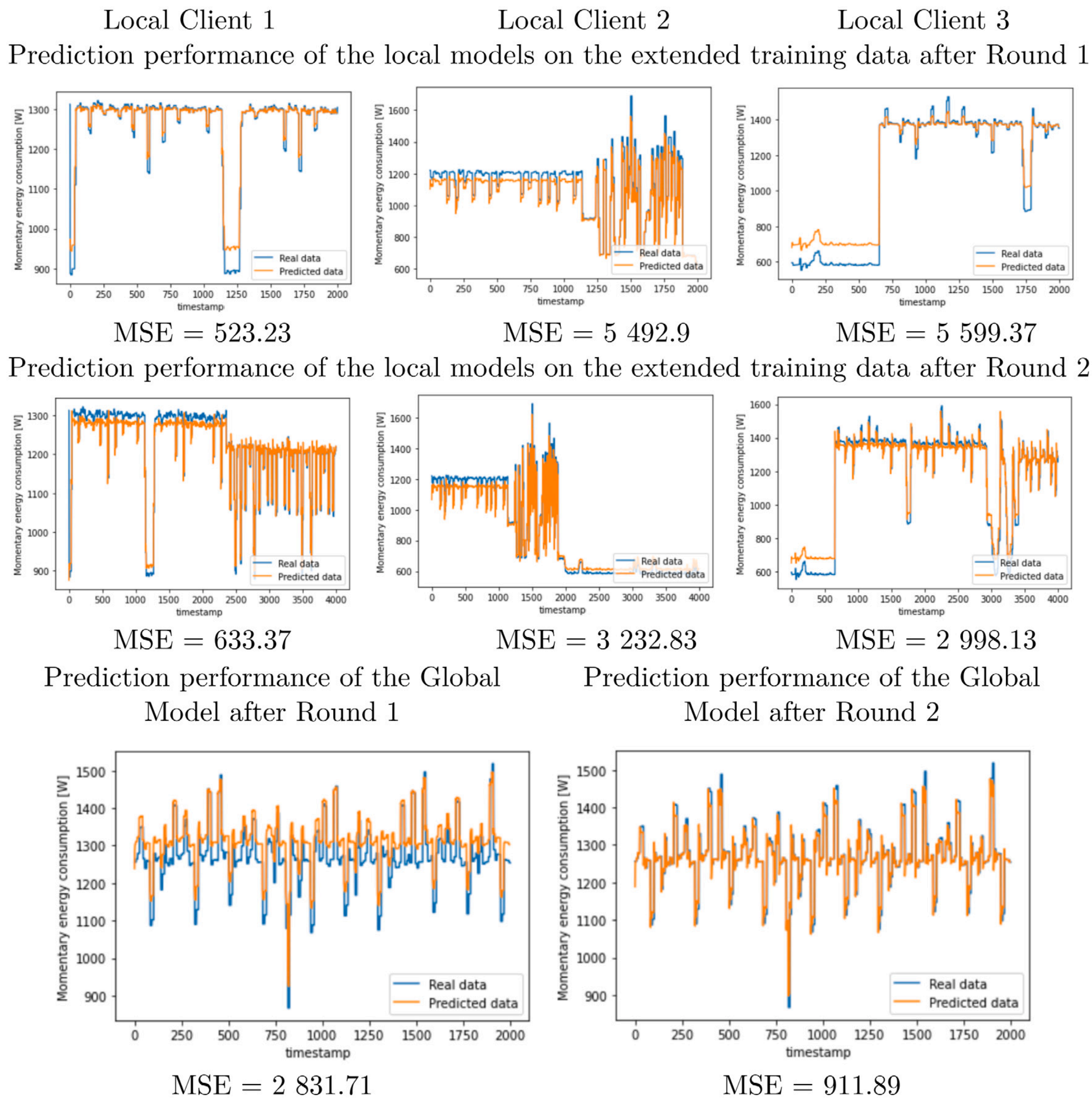


Fig. 12. Prediction performance for local and global models in 2-round averaging based on extended training data.

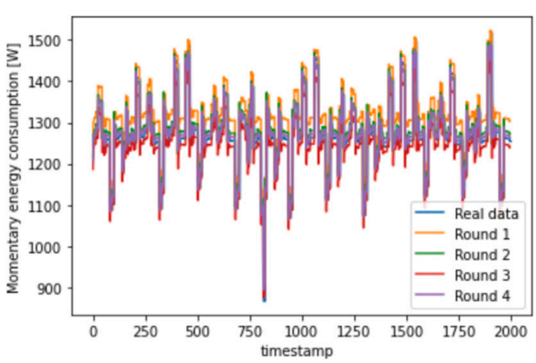


Fig. 13. Effectiveness of the global model after each Round in 4-round averaging based on extended training data.

Table 4
Prediction performance for each Round in 8-round averaging based on extended training data.

Round №	MSE	MAPE
GM Round 1	3923.97	4.24%
GM Round 2	1626.64	2.59%
GM Round 3	1116.07	1.81%
GM Round 4	999.67	1.54%
GM Round 5	954.54	1.28%
GM Round 6	841.26	0.92%
GM Round 7	872.01	1.04%
GM Round 8	833.72	0.92%

The fluctuation of values of the MSE and MAPE visible in Tables 3 and 4 results from the division of the data sets and different characteristics of the data sets provided to agents in different rounds (which can also be observed in results presented in Figs. 8, 9, and 12). However, a common approach is to use the Federated Learning technique with Continuous Learning, which makes the division of the data set irrelevant (after a certain number of rounds) because local models and the global model are continuously updated over time.

6.6. Execution time and resource consumption

We also verified the execution time spent on continuous training local LSTM models with parts of the training data sets and how it corresponds to the training time performed with the whole training data set. For comparison, we took the time spent on the experiment

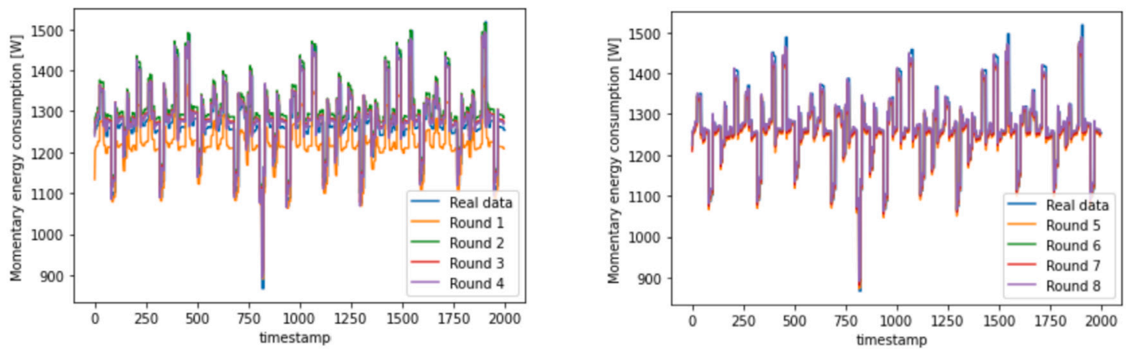


Fig. 14. Effectiveness of the global model after Round 1, 2, 3, 4, 5, 6, 7, and 8 in 8-round averaging based on extended training data.

by splitting data into eight rounds of training on the extended data set and the time spent on training on the entire data set (baseline) without using Federated Learning.

In general, the average training time when using Federated Learning on each client is 7.2 s per round, and if we sum up the training on all clients for eight rounds, the total time is 2 min and 52 s. However, we should emphasize that we used three independent agents, and their training process was parallelized. As a result, the total training time, considering the parallelization, is about 58 s. For the baseline (i.e., when training the prediction model on the whole data set without using Federated Learning), the training time took 2 min and 18 s. From these results, we can see that for the baseline model, the training time was shorter (there is no need for knowledge sharing and weight averaging). Still, when using Federated Learning, each device will train the prediction model in an average of 58 s.

We also verified the possibility of running the FL on the edge devices. Limited capabilities of edge devices pose constraints on the Machine Learning models that can be used in the anomaly detection. The most resource consuming is the phase of training the local prediction model. The FL approach that we developed is targeted for use with the NVidia Jetson Nano (ARM Cortex A57 1.43 GHz Quad-Core CPU, Nvidia Maxwell GPU with 128 CUDA cores and 4 GB RAM) edge device. We experimentally checked the memory consumption during the training phase on the training data set divided into eight parts (for eight training rounds) as in the experiment presented in Section 6.4.3 (each part consisted of 500 observations). The training consumed on average 110 MB of the memory, which is not much. For comparison, training of the baseline model on the whole data set reached the consumption of 139 MB of memory. These results show that developed solution is light and feasible for the edge devices.

7. Discussions

As a result of all the conducted experiments, we verified the effectiveness of FL for exchanging experiences between AGVs. FL improves the performance of signal prediction, making it possible to detect and avoid anomalies much better by further analysis of the signal and finding significant variations.

The results of all conducted experiments are summarized in Table 5. Almost all experiments based on FL led to better signal prediction results than the traditional approach (single LSTM model). We obtained the best prediction performance as a result of an experiment in which we performed eight rounds of model averaging and experience synchronization based on the MSE values obtained from local models trained on the extended training data set. This result is 19% more effective than LSTM without FL.

This also confirms the usefulness of the method much stronger than related works. For example, Zhang et al. [51] conducted an experiment where they tried to reduce the percentage of defects in production, and they obtained a result that demonstrates that their approach based on

Table 5
Prediction performance for all experiments.

Experiment	MSE	Reference
LSTM without FL	1029.67	Fig. 7
1-Round FL based on Validation Losses	908.78	Fig. 8
1-Round FL based on MSE	878.83	Fig. 8
2-Round FL based on training set	1096.79	Fig. 9
2-Round FL based on extended training set	911.89	Fig. 12
4-Round FL based on training set	868.15	Fig. 10 and Table 1
4-Round FL based on extended training set	875.07	Fig. 13 and Table 3
8-Round FL based on training set	890.85	Fig. 11 and Table 2
8-Round FL based on extended training set	**833.72**	Fig. 14 and Table 4

federated learning allows them to gain an advantage over recurrent neural networks by close to 5%. Our approach provides even more substantial evidence of the suitability of the FL approach in terms of gained prediction performance, despite the inherent advantages of FL, like reduced data transfers and data security.

It has to be clearly stated that the data set used in this study is not publicly available yet. As part of our future work, we want to enlarge it significantly, which will then be published. Furthermore, the outcome of the present research is very encouraging, and no doubt, the results obtained will form the basis for further experiments. We plan to focus on anomaly detection based on the signal prediction made as well as on further improving the quality of the prediction itself.

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

The authors do not have permission to share data.

Acknowledgments

The research was supported by the Norway Grants 2014–2021 operated by the National Centre for Research and Development, Poland under the project *Automated Guided Vehicles integrated with Collaborative Robots for Smart Industry Perspective* (Project Contract no.: NOR/POL-NOR/CoBotAGV/0027/2019-00), pro-quality grant (02/100/RGJ23/0026) of the Rector of the Silesian University of Technology, Gliwice, Poland, and by research funds for young scientists and Statutory Research funds of Department of Applied Informatics, Silesian University of Technology, Gliwice, Poland (grants No 02/100/BK_23/0027 and 02/100/BKM22/0020) and the Polish Ministry of Science and Higher Education as a part of the CyPhiS program at the Silesian University of Technology, Gliwice, Poland (Contract No. POWR.03.02.00-00-I007/17-00) and by Reactive Too project that has received funding from the European Union's Horizon 2020 Research, Innovation and Staff Exchange Programme under the Marie Skłodowska-Curie Action (Grant Agreement No. 871163).

References

[1] Q. Li, A. Adriaansen, J. Udding, A. Pogromsky, Design and control of automated guided vehicle systems: A case study, IFAC Proc. Vol. 44 (1) (2011) 13852–13857, http://dx.doi.org/10.3182/20110828-6-IT-1002.01232, 18th IFAC World Congress. URL https://www.sciencedirect.com/science/article/pii/S1474667016458507.

[2] L. Tebaldi, G.D. Maria, A. Volpi, R. Montanari, E. Bottani, Economic evaluation of automated guided vehicles usage in a food company, Procedia Comput. Sci. 180 (2021) 1034–1041, http://dx.doi.org/10.1016/j.procs.2021.01.352, Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020). URL https://www.sciencedirect.com/science/article/pii/S1877050921004063.

[3] E.A. Oyekanlu, A.C. Smith, W.P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, D.J. Kuhn, J.D. Mcghinnis, S.C. Buonavita, N.A. Looper, M. Ng, A. Ng'oma, W. Liu, P.G. Mcbride, M.G. Shultz, C. Cerasi, D. Sun, A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications, IEEE Access 8 (2020) 202312–202353, http://dx.doi.org/10.1109/ACCESS.2020.3035729.

[4] A. Ziebinski, D. Mrozek, R. Cupek, D. Grzechca, M. Fojcik, M. Drewniak, E. Kyrkjebø, J.C.-W. Lin, K. Øvsthus, P. Biernacki, Challenges associated with sensors and data fusion for AGV-driven smart manufacturing, in: M. Paszynski, D. Kranzlmüller, V.V. Krzhizhanovskaya, J.J. Dongarra, P.M. Sloot (Eds.), Computational Science – ICCS 2021, Springer International Publishing, Cham, 2021, pp. 595–608, http://dx.doi.org/10.1007/978-3-030-77970-2_45.

[5] A. Ziebinski, M. Bregulla, M. Fojcik, S. Kłak, Monitoring and controlling speed for an autonomous mobile platform based on the hall sensor, in: N.T. Nguyen, G.A. Papadopoulos, P. Jędrzejowicz, B. Trawiński, G. Vossen (Eds.), Computational Collective Intelligence, Springer International Publishing, Cham, 2017, pp. 249–259, http://dx.doi.org/10.1007/978-3-319-67077-5_24.

[6] D. Grzechca, A. Ziębiński, P. Rybka, Enhanced reliability of ADAS sensors based on the observation of the power supply current and neural network application, in: N.T. Nguyen, G.A. Papadopoulos, P. Jędrzejowicz, B. Trawiński, G. Vossen (Eds.), Computational Collective Intelligence, Springer International Publishing, Cham, 2017, pp. 215–226, http://dx.doi.org/10.1007/978-3-319-67077-5_21.

[7] R. Cupek, A. Ziębiński, M. Drewniak, M. Fojcik, Estimation of the number of energy consumption profiles in the case of discreet multi-variant production, in: N.T. Nguyen, D.H. Hoang, T.-P. Hong, H. Pham, B. Trawiński (Eds.), Intelligent Information and Database Systems, Springer International Publishing, Cham, 2018, pp. 674–684, http://dx.doi.org/10.1007/978-3-319-75420-8_63.

[8] B.R. Haverkort, A. Zimmermann, Smart industry: How ICT will change the game!, IEEE Internet Comput. 21 (1) (2017) 8–10.

[9] B. Wang, F. Tao, X. Fang, C. Liu, Y. Liu, T. Freiheit, Smart manufacturing and intelligent manufacturing: A comparative review, Engineering 7 (6) (2021) 738–757.

[10] S. Bergweiler, Intelligent manufacturing based on self-monitoring cyber-physical systems, in: UBICOMM 2015, Vol. 121, 2015.

[11] J.-H. Syu, Y.-R. Yeh, M.-E. Wu, J.-M. Ho, Self-management portfolio system with adaptive association mining: A practical application on Taiwan stock market, Mathematics 9 (10) (2021) 1093.

[12] Y. Meng, Y. Yang, H. Chung, P.-H. Lee, C. Shao, Enhancing sustainability and energy efficiency in smart factories: A review, Sustainability 10 (12) (2018) 4779.

[13] M. Abubakr, A.T. Abbas, I. Tomaz, M.S. Soliman, M. Luqman, H. Hegab, Sustainable and smart manufacturing: An integrated approach, Sustainability 12 (6) (2020) 2280.

[14] T. Kotsiopoulos, P. Sarigiannidis, D. Ioannidis, D. Tzovaras, Machine learning and deep learning in smart manufacturing: The smart grid paradigm, Comp. Sci. Rev. 40 (2021) 100341.

[15] J.H. Syu, M.E. Wu, G. Srivastava, C.F. Chao, J.C.W. Lin, An IoT-based hedge system for solar power generation, IEEE Internet Things J. 8 (13) (2021) 10347–10355.

[16] J.H. Syu, J.C.W. Lin, G. Srivastava, Call auction-based energy management system with adaptive subsidy and dynamic operating reserve, Sustain. Comput.: Inform. Syst. (2022) 100786.

[17] J. Arents, M. Greitans, Smart industrial robot control trends, challenges and opportunities within manufacturing, Appl. Sci. 12 (2) (2022) 937.

[18] G. Wilson, C. Pereyda, N. Raghunath, G. de la Cruz, S. Goel, S. Nesaei, B. Minor, M. Schmitter-Edgecombe, M.E. Taylor, D.J. Cook, Robot-enabled support of daily activities in smart home environments, Cogn. Syst. Res. 54 (2019) 258–272.

[19] M.E. Wu, J.H. Syu, J.C.W. Lin, J.M. Ho, Effective fuzzy system for qualifying the characteristics of stocks by random trading, IEEE Trans. Fuzzy Syst. (2021).

[20] J.H. Syu, J.C.W. Lin, C.J. Wu, J.M. Ho, Stock selection system through suitability index and fuzzy-based quantitative characteristics, IEEE Trans. Fuzzy Syst. (2022).

[21] M.E. Wu, J.H. Syu, J.C.W. Lin, J.M. Ho, Portfolio management system in equity market neutral using reinforcement learning, Appl. Intell. 51 (11) (2021) 8119–8131.

[22] M. Sharp, R. Ak, T. Hedberg Jr., A survey of the advancing use and development of machine learning in smart manufacturing, J. Manuf. Syst. 48 (2018) 170–179.

[23] T. Pranckevičius, V. Marcinkevičius, Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification, Baltic J. Mod. Comput. 5 (2) (2017) 221.

[24] J. Wang, J. Yan, C. Li, R.X. Gao, R. Zhao, Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction, Comput. Ind. 111 (2019) 1–14.

[25] A. Essien, C. Giannetti, A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders, IEEE Trans. Ind. Inform. 16 (9) (2020) 6069–6078.

[26] K. O'Shea, R. Nash, An introduction to convolutional neural networks, 2015, arXiv preprint arXiv:1511.08458.

[27] D.O. Melinte, L. Vladareanu, Facial expressions recognition for human–robot interaction using deep convolutional neural networks with rectified adam optimizer, Sensors 20 (8) (2020) 2393.

[28] X. Luo, X. Tian, H. Zhang, W. Hou, G. Leng, W. Xu, H. Jia, X. He, M. Wang, J. Zhang, Fast automatic vehicle detection in uav images using convolutional neural networks, Remote Sens. 12 (12) (2020) 1994.

[29] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, K. Ouni, Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3, in: 2019 1st International Conference on Unmanned Vehicle Systems-Oman, UVS, IEEE, 2019, pp. 1–6.

[30] T. Zhou, D. Tang, H. Zhu, Z. Zhang, Multi-agent reinforcement learning for online scheduling in smart factories, Robot. Comput.-Integr. Manuf. 72 (2021) 102202.

[31] S. Faryadi, J. Mohammadpour Velni, A reinforcement learning-based approach for modeling and coverage of an unknown field using a team of autonomous ground vehicles, Int. J. Intell. Syst. 36 (2) (2021) 1069–1084.

[32] J.M.T. Wu, S.H. Lin, J.H. Syu, M.E. Wu, Embedded draw-down constraint reward function for deep reinforcement learning, Appl. Soft Comput. 125 (2022) 109150.

[33] M. Pech, J. Vrchota, J. Bednář, Predictive maintenance and intelligent sensors in smart factory, Sensors 21 (4) (2021) 1470.

[34] A. Chehri, G. Jeon, The industrial internet of things: examining how the IIoT will improve the predictive maintenance, in: Innovation in Medicine and Healthcare Systems, and Multimedia, Springer, 2019, pp. 517–527.

[35] K. Aksa, S. Aitouche, H. Bentoumi, I. Sersa, Developing a web platform for the management of the predictive maintenance in smart factories, Wirel. Pers. Commun. 119 (2) (2021) 1469–1497.

[36] S. Cho, G. May, I. Tourkogiorgis, R. Perez, O. Lazaro, B.d.l. Maza, D. Kiritsis, A hybrid machine learning approach for predictive maintenance in smart factories of the future, in: IFIP International Conference on Advances in Production Management Systems, Springer, 2018, pp. 311–317.

[37] A.R. Javed, M. Usman, S.U. Rehman, M.U. Khan, M.S. Haghighi, Anomaly detection in automated vehicles using multistage attention-based convolutional neural network, IEEE Trans. Intell. Transp. Syst. 22 (7) (2020) 4291–4300.

[38] S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134–147.

[39] A.V. Malawade, N.D. Costa, D. Muthirayan, P.P. Khargonekar, M.A. Al Faruque, Neuroscience-inspired algorithms for the predictive maintenance of manufacturing systems, IEEE Trans. Ind. Inform. 17 (12) (2021) 7980–7990.

[40] J. Yoon, K. Sohn, C.-L. Li, S.O. Arik, C.-Y. Lee, T. Pfister, Self-trained one-class classification for unsupervised anomaly detection, 2021, arXiv preprint arXiv:2106.06115.

[41] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (2009) http://dx.doi.org/10.1145/1541880.1541882.

[42] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &Amp; Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 387–395, http://dx.doi.org/10.1145/3219819.3219845.

[43] S. Chauhan, L. Vig, Anomaly detection in ECG time signals via deep long short-term memory networks, in: 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA, 2015, pp. 1–7, http://dx.doi.org/10.1109/DSAA.2015.7344872.

[44] O.B. Sezer, M.U. Gudelek, A.M. Ozbayoglu, Financial time series forecasting with deep learning: A systematic literature review: 2005–2019, Appl. Soft Comput. 90 (2020) 106181.

[45] Z. Han, J. Zhao, H. Leung, K.F. Ma, W. Wang, A review of deep learning models for time series prediction, IEEE Sens. J. 21 (6) (2019) 7833–7848.

[46] P. Benecki, S. Piechaczek, D. Kostrzewa, J. Nalepa, Detecting anomalies in spacecraft telemetry using evolutionary thresholding and LSTMs, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 143–144, http://dx.doi.org/10.1145/3449726.3459411.

[47] J. Nalepa, M. Myller, J. Andrzejewski, P. Benecki, S. Piechaczek, D. Kostrzewa, Evaluating algorithms for anomaly detection in satellite telemetry data, Acta Astronaut. (2022) http://dx.doi.org/10.1016/j.actaastro.2022.06.026.

[48] K.M. Tsiouris, V.C. Pezoulas, M. Zervakis, S. Konitsiotis, D.D. Koutsouris, D.I. Fotiadis, A long short-term memory deep learning network for the prediction of epileptic seizures using EEG signals, Comput. Biol. Med. 99 (2018) 24–37.
[49] P. Benecki, D. Kostrzewa, P. Grzesik, B. Shubyn, D. Mrozek, Forecasting of energy consumption for anomaly detection in automated guided vehicles: Models and feature selection, in: 2022 IEEE International Conference on Systems, Man, and Cybernetics, SMC, 2022.
[50] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) 1–19.
[51] Y. Zhang, X. Li, P. Zhang, Real-time automatic configuration tuning for smart manufacturing with federated deep learning, in: International Conference on Service-Oriented Computing, Springer, 2020, pp. 304–318.
[52] B. Shubyn, D. Mrozek, T. Maksymyuk, V. Sunderam, D. Kostrzewa, P. Grzesik, P. Benecki, Federated learning for anomaly detection in industrial IoT-enabled production environment supported by autonomous guided vehicles, in: D. Groen, C. de Mulatier, M. Paszynski, V.V. Krzhizhanovskaya, J.J. Dongarra, P.M.A. Sloot (Eds.), Computational Science – ICCS 2022, Springer International Publishing, Cham, 2022, pp. 409–421.
[53] T. Steclik, R. Cupek, M. Drewniak, Automatic grouping of production data in Industry 4.0: The use case of internal logistics systems based on automated guided vehicles, J. Comput. Sci. 62 (2022) 101693, http://dx.doi.org/10.1016/j.jocs.2022.101693.

Bohdan Shubyn is currently a Ph.D. student at the Department of Applied Informatics at the Silesian University of Technology, Gliwice, Poland, and at the Department of Automated Control Systems at Lviv Polytechnic National University, Lviv, Ukraine. He obtained his master's degree at Lviv Polytechnic National University in 2019. His research interests mainly focus on machine learning, data analytics, databases, the Internet of Things, and Federated Learning.

Daniel Kostrzewais working as an Assistant Professor in the Department of Applied Informatics at the Silesian University of Technology (Gliwice, Poland), where he received Ph.D. degree in computer science in 2015. His research interests are mainly focused on deep learning, machine learning, evolutionary algorithms applied to anomaly detection, image processing, including super-resolution reconstruction, continuous and discrete optimization problems, and dimensionality reduction.

He published over 50 papers published in international peer-reviewed journals and conference proceedings. Dr. Kostrzewa was also a research engineer in several European Space Agency funded and academic projects.

Piotr Grzesik, M.Sc., Eng., is a Ph.D. student at the Department of Applied Informatics at the Silesian University of Technology in Gliwice. His main research areas include Cloud computing, Serverless computing, the Internet of Things, Edge Computing, databases, and bioinformatics.

Pawel Benecki is currently a Teaching Assistant in the Department of Applied Informatics at the Silesian University of Technology (SUT) in Gliwice, Poland. He received his MSc. Eng. degree from SUT in 2007. Currently he is pursuing a Ph.D. at SUT. In his research he is interested in super-resolution of images and processing of multimedia data. He coauthored ca. 20 papers published in international journals and conference proceedings.

Taras Maksymyuk is an associate professor in the Department of Telecommunications at Lviv Polytechnic National University, Lviv, Ukraine. He is also acting as a senior systems engineer at Infineon Technologies, Lviv, Ukraine. He received his MS (2011), Ph.D. (2015) and D.Sc. (2021) degrees in telecommunication systems and networks from the Lviv Polytechnic National University and completed his postdoctoral research at the Internet of Things and Artificial Intelligence Lab., Korea University (2017). He serves as an Editor for the Internet of Things Series in IEEE Communications Magazine and an Editor for the Wireless Communications and Mobile Computing. His research experience includes various aspects of 4G/5G mobile networks technologies, Internet of Things applications and artificial intelligence. Currently his research interests are focused on 6G technologies, blockchain, metaverse applications, digital twins and deep learning.

Vaidy Sunderam is currently Samuel Candler Dobbs Professor, and Chair of the Department of Computer Science at Emory University. He is also Director of the Emory Computational and Life Sciences Strategic initiative. His research expertise is in high performance computing, distributed systems, collaboration frameworks, and data management. His recent work has explored issues in data privacy, information security, and analytics. Sunderam's research has made several conceptual and pragmatic contributions in these areas, including frameworks for metacomputing, scientific simulations, data storage and access, and collaboration. In the realm of data management, his projects include moving object databases, privacy-preserving crowdsensing, and trust-based information fusion. Sunderam's research has been recognized and supported by federal grants from a number of agencies, best paper awards, keynote talk invitations, and software systems that are widely used. Vaidy Sunderam helped launch several educational programs in Computer and Computational Science at Emory, including Ph.D. and MS programs in cooperation with Emory Health Sciences, and most recently, a BS in CS and Economics. Sunderam teaches Computer Science at all levels, serves as advisor to undergraduate as well as graduate and postdoctoral scholars.

Jia-Hao Syu is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. He obtained his bachelor's degree from the Department of Computer Science and Information Engineering, National Taiwan University, in 2019. His research interests are mainly in the areas of data science, artificial intelligence, quantitative trading, and financial computation.

Jerry Chun-Wei Lin received his Ph.D. from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan in 2010. He is currently a full Professor with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 500 research articles in refereed top-tier journals and conferences. His research interests include data analytics, soft computing, AI/ML, privacy preserving and security technologies, optimization, and fuzzy-set theory. He is the Editor-in-Chief of the International Journal of Data Science and Pattern Recognition. He has recognized as the most cited Chinese Researcher respectively in 2018, 2019, and 2020 by Scopus/Elsevier. He is a Fellow of IET (FIET), ACM Distinguished Member (Scientist), and IEEE Senior Member.

Dariusz Mrozek is currently an Associate Professor and Head of the Department of Applied Informatics at the Silesian University of Technology (SUT) in Gliwice, Poland. He received his Ph.D. degree from SUT in 2006. His research interests cover IoT, parallel and Cloud computing, databases, big data, and bioinformatics. He is now focused on developing IoT solutions for healthcare, elderly care, manufacturing, and nature monitoring.

He is the author of 150+ papers published in international journals and conference proceedings, author of two books on the use of Big Data analytics and high-performance computations in bioinformatics published by Springer, coeditor of fifteen other books devoted to databases and data processing, and editor of many special issues in reputable scientific journals. He has collaborated with qualified institutions by working in different research projects, like Imperial College of London in UK, Microsoft Research and Amazon in the USA.