IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 4 July 2024, accepted 11 July 2024, date of publication 15 July 2024, date of current version 26 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3428866

APPLIED RESEARCH

Long Short-Term Memory-Based Prediction Solution Inside a Decentralized Proactive Historian for Water Industry 4.0

ADRIAN KORODI [1], **ANDREI NICOLAE** [1], **DRAGOŞ BRISC** [1], **IONEL DRĂGHICI** [2], **AND ADRIAN CORUI** [2]

[1]Department of Automation and Applied Informatics, Faculty of Automation and Computers, Universitatea Politehnica Timişoara, 300223 Timişoara, Romania

[2]SC Aquatim SA, 300081 Timişoara, Romania

Corresponding author: Adrian Korodi (adrian.korodi@upt.ro)

This work was supported in part by the Executive Unit for the Financing of Higher Education, Research, Development and Innovation (UEFISCDI) through the "Transfer to the Economic Operator" Program (Project Code: PN-III-P2-2.1-PTE-2021-0039) under Contract 77PTE/2022.

ABSTRACT Improvement possibilities of industrial systems are driven by Industrial Internet of Things (IIoT) and Industry 4.0 concepts. The basic enablers are related to interoperation, and some efficiency and safety increasing solutions may be direct outcomes. Other benefits of interoperability are related to data integration, and the ability to analyze the collected information and to establish recipes for improvements. One of the most important targets required by the industry is the prediction capability, to forecast equipment faults or process values. The water industry with its specific characteristics is following the same path and needs efficiency increasing solutions that could be applied on its many legacy systems. Any data-driven solution is a case study driven solution. Therefore, the current research is started with deploying pilots that consists of proactive historians applied to functioning legacy water sector facilities. The work is presenting a Long-Short-Term-Memory (LSTM) neural networks-based prediction solution within the low-cost decentralized proactive historian. The exposed case study technological process is a wastewater treatment plant, where sludge pump failures are predicted to improve maintenance activities, as well as a chemical oxygen demand water quality indicator to improve process control strategy adjustments. The algorithm is generated as a result of a batch training and afterwards it is adapted also to incremental training. The solution is conceived for the hardware and software conditions of the proactive historian and the deployment within the real scenario proved excellent results, with the ability to provide 5 hours ahead correct predictions.

INDEX TERMS Proactive historian, IIoT, LSTM neural networks, industry 4.0, water industry, industrial automation.

I. INTRODUCTION

The Industrial Internet of Things (IIoT) and Industry 4.0 concepts are representing the pillars for new evolution in the industrial world [1], [2]. The key enabler is the interoperability and interoperation between entities and systems [3], [4]. Although many interoperability solutions are researched and solved [5], various challenges are still encountered related to protocol improvements [6] and legacy systems integration [7]. Interoperability and interconnected systems assure that data can be transferred. Interoperation is expected to improve the process functioning [8], but it usually has to follow an improvement strategy that could derive from information/recipes obtained from data analysis [9].

Neural networks (NN) are used in the industry to develop an improved decision making when data is accumulated (e.g. image processing-based defect detection using deep learning [10], convolutional NN (CNN) used for layout fixtures for sheet metal designs in automotive parts [11]. A synthesis of deep learning techniques in IIoT is presented in [12].

The associate editor coordinating the review of this manuscript and approving it for publication was Stefano Scanzio.

Nowadays, the industrial world needs also prediction, for future faults in the production lines that could cause delays and resource expenditures, or for other uncertainties. Many of the prediction approaches are including Long-Short-Term-Memory (LSTM) recurrent NN for the data-driven models. Work [13] presents a solution to predicts faults using a prediction of an expected image 1 second before, while [14] predicts the production progress in a manufacturing workshop. There are also other types of solutions, that may be suitable for specific scenarios, like paper [15] that presents a probabilistic temperature prediction in additive manufacturing. An important issue to be tackled is that a process and data driven research targeting an improving strategy, should rely on real data and should finally depict how to be applied on real industrial systems. Research [16] proposes an interesting, optimized LSTM based strategy to diagnose faults in chemical processes, but the work should continue to increase the technological readiness level and the industrial deployment. Paper [17] also relies on LSTM in order to assess short-term voltage stability for power systems, proving better results than the traditional methods.

Therefore, in Industry 4.0, systems have to be interfaced and connected, data must be collected and analyzed, improving strategies must be researched, developed, respectively applied on real systems. Generally, the industrial world focuses in this moment on all mentioned aspects trying to find the best answers to questions as "how to": interface, collect data, use data, react. The water industry makes no difference and presents particular characteristics like geographical and chronological development dispersion, various local technologies and solutions, many systems involved in a complete water cycle, sometimes considerable amount of time necessary to move between functional regimes in the treatment process, large lifecycles for local automation and consequently many legacy systems, etc. The water sector is focused on improving issues related to consumption, equipment failures and consequent improvements of the maintenance strategies, safety, etc. (e.g. [18], [19]).

An issue that was intensely analyzed and approached in the water sector is the interfacing. The Open Platform Communication Unified Architecture (OPC UA) protocol is considered a key enabler of Water Industry 4.0, as in the manufacturing industry [20], but other legacy protocols are still present in the water sector (e.g. Modbus TCP, S7, Ethernet/IP). Other aspects required in the water domain are related to improvements and the majority are data-driven. The industry and the literature are oriented towards NN (e.g. [21] where a CNN is used for pipe leakage detection and localization, [22] where CNN and LSTM NN are used for the underground drainage system for software sensing, [23] where CNN and LSTM NN is used along with a Raspberry Pi based image processing local system added to a mechanical water meter to detect leakages). But few works are encountered on prediction. Paper [24] presents a good solution to forecast water quality extremes, but it is unlikely to adapt the method to a local water/wastewater facility. Work [25] focuses on forecasting pipe failures in a water supply system, the provided data does not clarify completely the prediction timeframe, respectively the complete applicability. Work [26] was among the online analyzed initiatives, with a good focus on a water company needs, but the prediction timeframe required for a real maintenance or control adjustments has to be much larger and the authors could not obtain even the claimed prediction timeframe using the provided guidelines.

The current interdisciplinary research team approaches efficiency increase issues in the water sector focusing real case studies and several installed pilots (see the research strategy in chapter II). The current work is presenting the prediction solution within low-cost decentralized proactive historian applied on legacy systems. The research relies on LSTM NN that are used to predict equipment faults and analogue process values (see chapter III). The presented case study technological process is a wastewater treatment plant, where sludge pump failures are predicted as well as a chemical oxygen demand water quality indicator (see chapter III). The algorithm is conceived as result of a batch training and afterwards is adapted also to incremental training. The solution is tailored for the hardware and software conditions of the proactive historian, and it is applied on real scenario with excellent results, being able to make correct predictions 5 hours ahead. Discussion and conclusion is presented in chapter IV.

II. RESEARCH STRATEGY FOR THE PREDICTIVE HISTORIAN

The largely dispersed local systems in the water sector are centralized in local, regional, central SCADA control centers. But, going up the pyramid shaped hierarchical levels, the information is filtered and reduced, the general knowledge is increasing, the local processes related knowledge is decreasing, respectively the capability to automatically react towards the local automation is reduced. Therefore, most suitably, the improving solution that exchanges data with the local process and implements the LSTM NN based strategy should be placed in the fog computing area, near the local SCADA system or the automation panels.

The proactive historian is conceived as a low-cost decentralized solution that is able to interface the local legacy system and to collect and analyze data, to identify dependencies, to reach objective functions considering constraints in a process aware manner, respectively to act on the functioning local automation in order to implement the obtained recipes. The mentioned characteristics were proven in some specific scenarios, a complete solution to increase energy efficiency for drinking water facilities being addressed in [27]. But, many opportunities for research arise for such proactive historian in the water sector when data from industrial functioning systems are considered in specific scenarios and local process structuring. This implies tailoring, longer term testing and mandatory partnership with water companies. The research team started in July 2022 an intensive project to develop and test the capabilities of the proactive historian in a context of real scenarios. The steps of the research strategy along with marking the limits of the current status are shown in Figure 1. Pilot structures were started in connection with real functioning drinking water and wastewater legacy systems. Proactive historians were tailored and extended for specific scenarios.

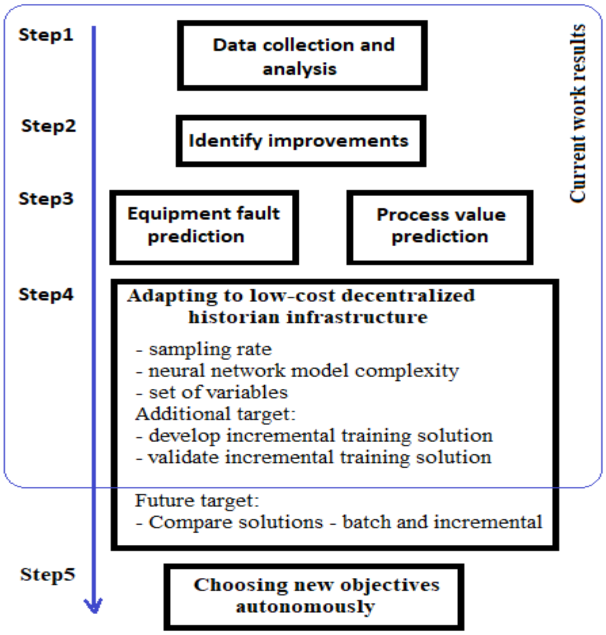

FIGURE 1. The steps of the research.

After proper interfacing (mainly through OPC UA protocol), hardware and software deployment of the historian with possibilities to monitor and easily upgrade the solution, local technological process basic knowledge being integrated, the following first two steps were covered:

Step 1: Data collection and analyze. Historians gathered data for more than 8 months in continuous functioning (see Figure 2). Data was continuously analyzed, dependencies were obtained.

Step 2: Identify improvements. Available solution allowed to validate, adapt and extend energy reduction strategies in the drinking water sector and to identify recipes and scenarios in this direction. But, together with process and mechanical engineering specialists from the water company, the following improvement possibilities were targeted: reducing the energy consumption; reducing the consumption of substances; non-invasive process control adjustments on existing functioning automation; equipment fault prediction.

The current research is focusing on prediction. Prediction is essential for improvements and in order to have a meaning it is mandatory to be engendered using real scenarios and systems. Two prediction targets were established:

- the main target consisted in equipment fault prediction, as being an important research requirement in the industry.

- additional target was related to predict analog process values to be used in non-invasive process control adjustments on existing automation. The necessity to predict values is reasoned by the higher time constants that are present in

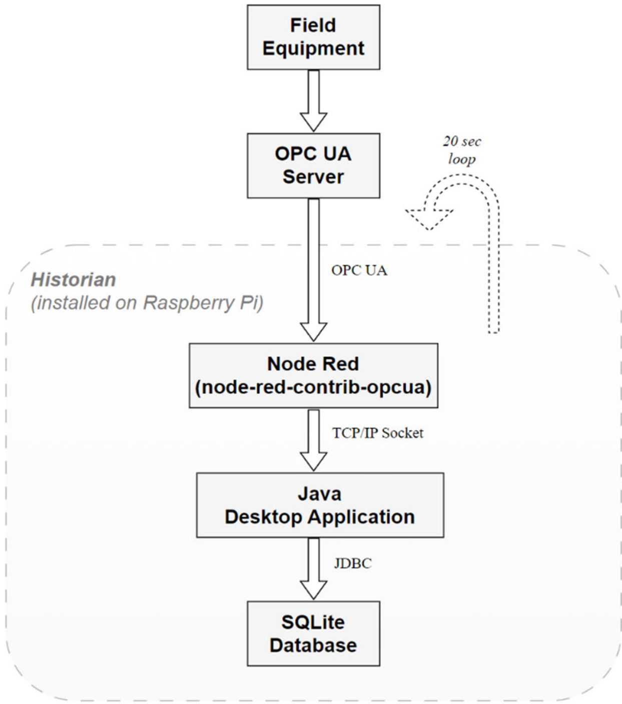

FIGURE 2. Data collecting using the installed historian.

parts of the treatment process in the water industry. It means that sometimes is not possible to generate the desired effect by adjusting suddenly the control strategy. This types of situations are usual in the biological phase of the wastewater treatment, but also in the drinking water treatment (e.g. chlorine correction).

The research continued with the third step divided into two branches for the prediction algorithm, and with the fourth step that adapts the strategy to the low-cost historian infrastructure:

Step 3 – Branch 1: Fault prediction. The fault prediction strategy required the presence of equipment faults within the data collected by the historians. All process equipment was monitored, and the process and mechanical experts from the research teams pointed out elements that are important for improvements and time objectives for prediction. The main condition for training and validation of the algorithm is to observe previous equipment faults (the quantity influences correctness/completeness) and to correlate the faults with other tag values in the system.

The chosen prediction strategy was the LSTM recurrent NN. The approach in step 3 of the research was to:

- train the LSTM NN based on the first set of stored data;
- validate the obtained NN based on a second dataset;
- validate the LSTM network using newly acquired data.

Step 3 – Branch2: Process value prediction. The actions within branch 2 were somehow similar with those in branch 1 but for stateless variables. The target was to obtain similar algorithm with its instance tailored for analog value prediction.

Step 4: Adapting to low-cost decentralized historian infrastructure. Regarding that the proactive historian is designed for a decentralized perspective, applied in connection with local legacy systems using a low-cost infrastructure (e.g. Raspberry Pi), the prediction strategy had to be adapted for: reduced sampling rate (the higher the sampling rate, the bigger the volume of stored data and the processing time); reduced NN model complexity; reduced set of variables.

An additional target of the research was to obtain independent and automatic improvement of the algorithm by: developing an incremental training solution for the algorithm within the historian infrastructure; validating the incremental training solution.

After this point, future research that will be undertaken will consist of comparing the on-condition triggered batch training with the incremental training solutions.

Also, as future step of the research would be:

Step 5: Choosing new objectives autonomously. Other AI techniques will have to be added and evaluated to be able to:

- establish prerequisites for predicting faults (e.g. number of faults limit for training) or process value (e.g. process value that could cause better system performance influencing higher time-constrained process parts).
- identify proper variable sets to predict chosen tag value.
- autonomously validate incremental training.
- establish conditions and constrains for the proactive historian to properly evaluate, validate and deploy the newly obtained algorithm in junction with the legacy system (process knowledge and functioning related criteria, including safety and ethical analysis, especially in the case of adjusting legacy control structures; algorithm upgrade procedure).

III. PREDICTION SOLUTION IN THE PROACTIVE HISTORIAN

A. GENERAL PREDICTIVE CAPABILITIES

In order to develop the ability to predict future evolutions, complex patterns must be identified from historical data, aspect which perfectly suits the use of artificial intelligence technologies for this purpose. For the practical implementation of the current research, Microsoft Visual Studio Code was used as source code editor and Python (version 3) as programming language. Also, for structuring and displaying data, the following software libraries were used: Pandas, NumPy, Matplotlib, Scikit-learn (train_test_split), Beautiful Soup. For creating the artificial intelligence model, the following software libraries were used: Tensorflow (Sequential, Layers, MeanSquared, Adam) and Scikit-learn (MinMaxScaler, Normalize, Accuracy_score).

The subsequent step consisted in identifying the type of NN model which would be most appropriate for the purpose of predicting values in the future. Because the available data represents a sequence recorded at successive equally spaced points in time, the desired outcome of the research is a solution to a time series forecasting problem. The authors examined various types of NN that are applicable for this kind of problem and decided to use LSTM because of its ability to identify and maintain long-term dependencies to make predictions in future time steps. Also, LSTM's input, output and forget gates allows to selectively retain or forget information, which is very useful, especially under varying operational conditions. The data is sequentially processed, which preserves the order of events, making them ideal for modeling time series data. Non-linearity and non-stationary data are accommodated as well by LSTM, which are inherently flexible, all of those attributes making them suitable for the current case.

Moving forward with the research, 2 generic LSTM models were generated: a complex model with approximately 100000 parameters and a simpler model with approximately 19000 parameters. The complex model is presented in Figure 3 and the simpler model is presented in Figure 4.

```
#Building the LSTM Model
nr_inputs2 = X2_train.shape[2]
nr_outputs2 = y2_train.shape[1]

model2 = Sequential()
model2.add(InputLayer((timesteps2, nr_inputs2)))
model2.add(LSTM(128, return_sequences=True))
model2.add(Dropout(0.2))
model2.add(LSTM(64))
model2.add(Dense(64, activation='relu'))
model2.add(Dropout(0.2))
model2.add(Dense(8, activation='relu'))
model2.add(Dense(nr_outputs2, activation='linear'))
```

FIGURE 3. The complex model detailed.

```
model2 = Sequential()
model2.add(InputLayer((timesteps2, nr_inputs2)))
model2.add(LSTM(64))
model2.add(Dense(8,'relu'))
model2.add(Dense(nr_outputs2,'linear'))
```

FIGURE 4. The simpler model detailed.

Following later tests, the simpler model continued to offer identical or superior results compared to the more complex model (R^2 0.546; MSE 1.568; RMSE 1.252 for the complex model, compared to the values from Table 4 below), which supported the decision to advance the current research based only on the simpler model. However, it is possible that the more complex model might be better suited for other applications than the ones considered in the current paper.

The used simpler model, presented in Figure 4, is a sequential model and consists of the following layers:

- InputLayer – receives the input data and specifies its format. The size of the input is defined by *timesteps* (the number of time periods considered in each sequence) and nr_inputs (the number of characteristics/OPC UA tags used in each datapoint).
- LSTM – this layer consists of 64 LSTM units, which allow the model to learn complex sequential relations and to capture temporal dependencies in input data.
- Dense – this layer consists of 8 units and has a Rectified Liner Unit (ReLU) activation function, which introduces a non-linearity in the model. The ReLU function returns 0 for all negative values and keeps positive values unmodified,

which helps at learning non-linear characteristics in input data.

- Dense – the last layer contains a number of *nr_outputs* units, which correspond to the number of output characteristics. This layer uses a linear activation function, which produces continuous outputs and allows the model to generate predicted values without restrictions.

An interesting perspective is brought by the possibility to build an automated learning ability for the AI model through the implementation of Incremental Training concept, which could help improve the accuracy of the model in time, after the real world deployment. Also, the Incremental Training concept allows the model to adapt to changes in data over time and to adjust its predictions based on the new information. However, it is important to establish and enforce mechanisms through which to ensure that the incremental changes to the model will not have an unwanted negative impact over its accuracy. Also, adding the Incremental Training ability requires an evaluation of the processing requirements and the capability of the hardware platform that supports the smart software tool to meet those requirements as compared to the similar model without Incremental Training. The trade-off between more demanding processing requirements and possibility to obtaining accuracy improvements must be carefully considered for each particular application.

B. INTEGRATION INTO AUTOMATED SOFTWARE SOLUTION

To integrate the predictive capabilities obtained as a result of the current research into an optimization strategy applied in an autonomous manner by a software tool, the authors elaborated the software architecture from Figure 5 for assimilating the aforementioned capabilities into the proactive historian solution.

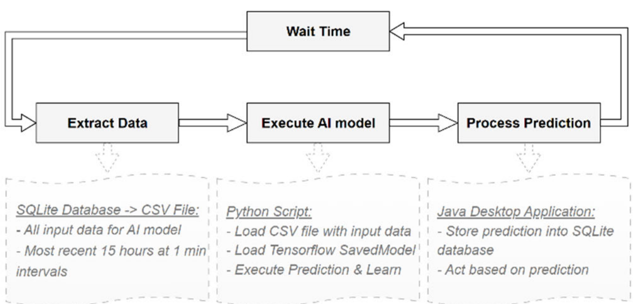

FIGURE 5. Software architecture for integrating predictive capabilities into the proactive historian solution.

Figure 5 highlights both the 4 general actions that are required to be implemented for integrating an AI model which generates predictions into a smart software tool and the particularities suited for the practical applications addressed by the current research.

The architecture consists of a loop managed by the Java application from historian, the first step being represented by the extraction of the required data from the SQLite database into a Comma Separated Values (CSV) file. For this, all characteristics that are expected at input for the trained LSTM model must be considered from a time interval appropriate for the specific application (the most recent 15 past hours from the moment of execution, with a 1 min. interval between successive data records in the case of the current research). In the second step, the LSTM AI model must be executed through the use of a Python script, which must initially load the extracted input data from the CSV file and also load the Tensorflow SavedModel from a file. Afterwards, the script executes the computing of the prediction, which, in case of integrating a self-learning model, also represents the incremental learning step for the model. The obtained prediction is finally processed by the Java application, which stores the prediction into the SQLite database for future reference. Also, the prediction represents the foundation on which the Java application will act upon, the type of action being dependent on the particular implemented optimizing strategy or objective. After waiting a specific amount of time, the described loop repeats, thus providing periodic and continuous applying of the proactive, intelligent capabilities over the technical system's operation.

If applicable, as in the case of the current research, the Incremental Training means that the LSTM model supports a new training round after executing each prediction, but limited to the size of the newly received information only. In order to persist, the last state of the model must be locally saved in a file with .h5 extension, each iteration of the loop requiring the file loading. After predicting, the new training data must be defined by removing the unwanted layers of previous data and a training process with low number of epochs can take place. Basically, the new set of data is added to the model without the need to train it from scratch, a snippet of the developed Python script for this feature being presented in Figure 6.

```
contor = 0

model.load_weights('model.h5')

predictions = model.predict(X_newData)

model.layers[contor].trainable = False

model.fit(X_newData, y_newData, epochs=5)

contor = contor + 1

model.save_weights('model.h5')
```

FIGURE 6. Python script snippet for incremental training.

The adaptive mechanisms to incorporate real-time data updates and unforeseen operational conditions include data preprocessing, extraction of rolling averages and moving medians, as well as algorithms for anomaly detection in real-time, model retraining and updating (Incremental Training and periodic Full Training when substantial changes in operational patterns are detected) and continuous predictive model performance monitoring using key performance indicators (e.g. prediction accuracy, precision). Several strategies could be considered for ensuring long-term accuracy and relevance: (*i*) regular model validation using latest real-time data, (*ii*) ensemble learning, (*iii*) adaptive hyperparameter tuning using Bayesian optimization and (*iv*) domain expert review for fine-tuning.

The Incremental Training is minimizing the prediction latency, is computationally less demanding per update, is more scalable than Batch Training, especially when the size of data grows, but requires constant computational resources and it cannot leverage the more advanced optimization techniques that Batch Training can. The less frequent updates in Batch Training can lead to slower responsiveness to new trends or anomalies in the data. A hybrid approach, with Batch Training during off-peak times and Incremental Training for continuous updates ensures that the system remains scalable, efficient, and responsive to real-time data changes.

Frequent retraining can be avoided by adjusting the learning rate dynamically, which allows quick adapting to recent changes without drastically altering established knowledge. Also, a sliding window approach is appropriate, shifting as new data arrives, the size of the sliding window being adjustable based on the rate of change in the data. Other solutions are: (*i*) prioritizing the memory update, ensuring that significant changes are quickly incorporated while minor fluctuations are smoothed out or (*ii*) regular mini-batch updates at regular intervals.

The concept drift can be detected through usage of statistical tests, such as the Page-Hinkley test or the Adaptive Windowing (ADWIN) method.

C. APPLICATION – CASE STUDY AND RESULTS

The WWTP targeted by the case-study serves an area containing around 6000 inhabitants and is based on a classical sequential technological process. Briefly, two wastewater treatment lines are implemented, and after mechanical treatment of the inlet, the biological treatment contains two sequential batch reactor tanks where the aerobic and anoxic treatment is time based. The sludge line and bypass lines are also included. The control system consists of 9 S7-1200 PLCs that are connected with the redundant SCADA WinCC V13 SCADA system using a redundant fiber optic ring. The pilot structure integrated the proactive historian by interfacing the two OPC UA servers of the WinCC SCADA servers.

By deploying the proactive historians, two issues were identified. First, two sludge pumps within the biological reactors were frequently failing causing problems for process continuity and maintenance. Secondly, the process engineer pointed out that for large sudden variations of the chemical oxygen demand (CODcr) the biological treatment is not able to respond properly and it would need some more time to adapt. If the pump fault would be predicted at least 3 hours ahead, respectively if the CODcr indicator would be forecasted at least 1 hour in advance, the company and the local control system would be able to react properly.

To apply the general purpose algorithm to the case study, a data preprocessing was required before implementing and testing. The available data was stored starting with July 2022 by the historian application in a SQLite database, which has been retrieved and used in the current research. 460 different OPC UA tags were monitored by the historian, using a value sampling of approximately 20 seconds. The data is structured in a database table containing 461 columns (for each OPC UA tag + timestamp), a new row being added at 20 sec. intervals. An insight on the data format is presented in Figure 7.

<table>
<tr><th></th><th>timestamp</th><th>ns=1;s=t|01.ST.01_ComandaDistanta</th><th>ns=1;s=t|P7.Analogice_QIC-01.103</th><th>ns=1;</th></tr>
<tr><td>1</td><td>2022-09-07 14:17:09</td><td>false</td><td>13.59682464599609</td><td>5.313</td></tr>
<tr><td>2</td><td>2022-09-07 14:17:30</td><td>false</td><td>13.58778381347656</td><td>5.310</td></tr>
<tr><td>3</td><td>2022-09-07 14:17:50</td><td>false</td><td>13.64655685424805</td><td>5.314</td></tr>
<tr><td>4</td><td>2022-09-07 14:18:11</td><td>false</td><td>13.63751602172852</td><td>5.311</td></tr>
<tr><td>5</td><td>2022-09-07 14:18:31</td><td>false</td><td>13.66012191772461</td><td>5.313</td></tr>
</table>

FIGURE 7. An insight on the available data's format.

For the batch training, the available data was exported into a CSV file, which was used in the Python preprocessing script.

In order to manage the large volume of information (database size was approximately 3.5 GB) and reduce the processing effort, a 10 min. timeframe window was chosen further (every 30th row of initial data was considered).

After filtering initial data, the preprocessing script structured the data in a format that is appropriate for usage in a LSTM model, by performing the following operations:

- Select only the relevant columns for LSTM's input and output (only a subset from the total of 460 OPC UA Tags were used for applications).

- Convert the data in NumPy format.

- Build the input and output datasets using a specified number of time steps for each sample.

- Store the input values into a matrix and the corresponding output values into a vector.

The last step in the preprocessing of data consisted in dividing the data into separate datasets for training, testing and validating, which was done by making use of *train_test_split* function from Scikit-learn library. The outcome of this approach was the splitting of data in the following shares: 63% of data for training, 30% of data for testing and 7% of data for validating.

All boolean values were replaced with the numerical values 0 and 1 respectively. All the records that contained at least one missing value (*null* or *nan*) were removed. The input data was normalized in the [0-1] numerical interval to ensure consistent scaling. For anomaly detection, the statistical method of Z-Score analysis was used, with a threshold of 3 being chosen for anomaly filtering. In order to safeguard against erroneous predictions, the method of Cross-Validation was used for model validation and testing, using historical data to ensure that the model generalizes well to unseen data.

The generic LSTM NN model presented in chapter III A represented the basis for the 2 different practical applications in the current research, both of which being further detailed in the current section.

1) STATUS PREDICTION OF SLUDGE PUMP

The first practical application is oriented towards the prediction of the status information of a sludge pump from the WWTP, in line with the predictive maintenance objective fixed for the current research.

The studied pump is one of the 2 pumps responsible for disposal of sludge from the first biological reactor from the 2 reactors available in the WWTP.

The status of the pump is available in the data recorded by the historian, as a numerical value of a specific OPC UA tag, the association from Table 1 being applicable.

TABLE 1. Mapping of sludge pump status values to their significance.

Value	Significance
0	Unknown status
1	Fault
2	Not running
3 and 4	Normal running
9	Warning

The characteristics (corresponding OPC UA tags) from Table 2 were used from the available data as input data in order to train the aforementioned generic LSTM NN. The pump status was used as output data.

The list of characteristics from Table 2 has been chosen by the authors after a series of experimental tests in which the performance of the trained model was analyzed. The research team observed that adding any other characteristic to the list did not bring a performance improvement, while removing any of the mentioned characteristics from the list resulted in negative impact over performance.

TABLE 2. List of characteristics used as input data for neural network training – sludge pump status prediction.

Characteristic significance	Measurement unit	Type of value
Pump status	See Table 1	Numerical integer
Pump automatic mode	-	Boolean
Pump remote command	-	Boolean
Pump frequency	Hz	Numerical float
Pump power	kW	Numerical float
Pump current	A	Numerical float
Pump energy meter	kWh	Numerical integer
Biological reactor sludge stabilization flow	m^3/h	Numerical float
Biological reactor sludge stabilization volume	m^3	Numerical float

The sample (time steps) was set at 30 units, which ensures a prediction spanning over the course of 5 future hours and the number of passes over the database (epochs) was set at 10.

The LSTM network parameters configuration included a learning rate of 0.001, batch size of 32, 100 epochs and the loss function was Mean Squared Error (MSE).

The authors applied both the training and testing of the LSTM network for the status prediction of the sludge pump on 2 scenarios. The difference between the 2 considered scenarios is represented by the time intervals from which the data was used for training, respectively testing, the same characteristics being chosen and same LSTM generic model serving as starting point. The time intervals used for training and validating the LSTM model were extracted are summarized for both scenarios in Table 3.

TABLE 3. Time intervals from which data was extracted to train and test the AI model – sludge pump status prediction.

Description	Scenario 1	Scenario 2
Training		
Start	7 Sept. 2022 14:17:09	7 Sep. 2022 14:17:09
End	14 Dec. 2022 00:56:17	13 Mar. 2023 04:02:41
Testing		
Start	22 Dec. 2022 23:07:22	28 Dec. 2022 01:30:38
End	29 Jan. 2023 07:54:36	18 Jun. 2023 09:08:13

Graphical representations of the test results are presented for both scenarios to compare the prediction with the actual status recorded in the database (Figure 8 for scenario 1, Figure 10 for scenario 2), and from the perspective of displaying exclusively the prediction (Figure 9 for scenario 1, Figure 11 for scenario 2).

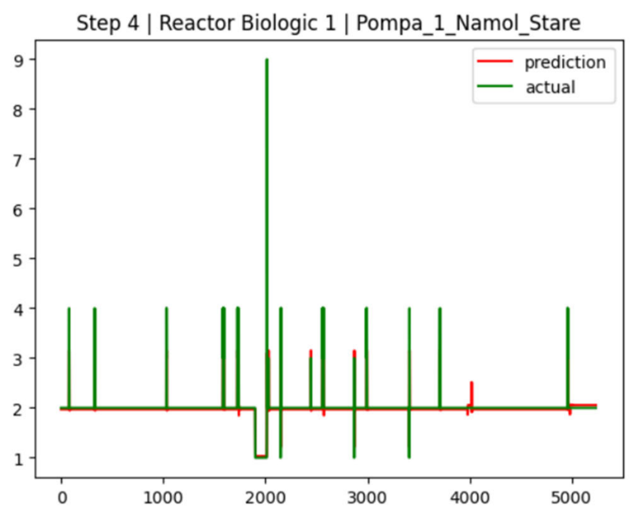

FIGURE 8. An insight on the available data's format.

In order to determine the accuracy of the trained model, 4 approaches were followed. Firstly, the *r2_score* function from Scikit-learn library was used, which returns the coefficient of determination (R^2) between predicted values and actual values, performing a computation with a 0 error margin. Also, the standard MSE and RMSE techniques were considered. However, because the pump status values are

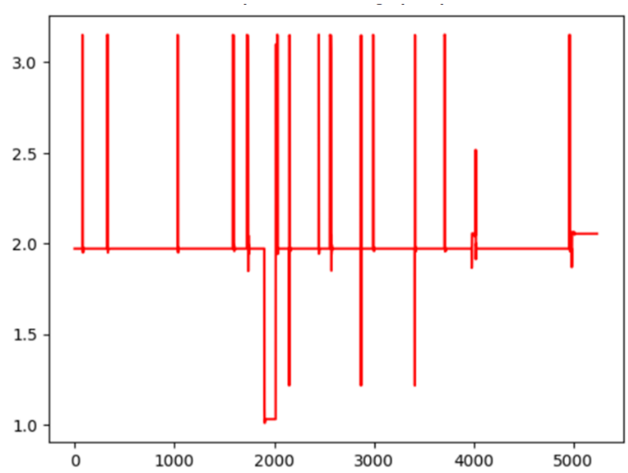

FIGURE 9. Scenario 1: sludge pump status prediction.

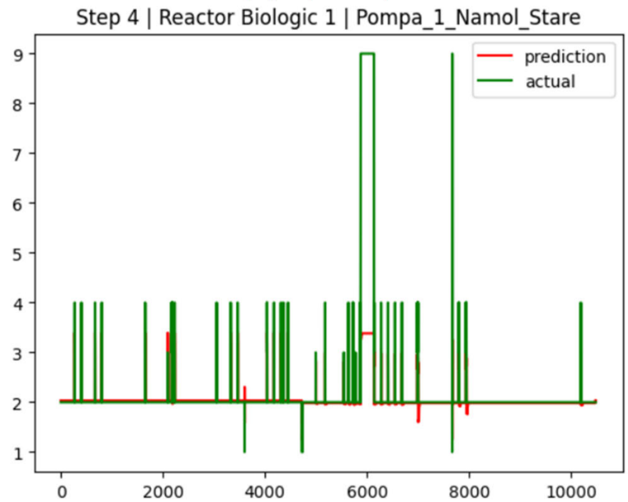

FIGURE 10. Scenario 2: comparison of sludge pump status prediction with actual values.

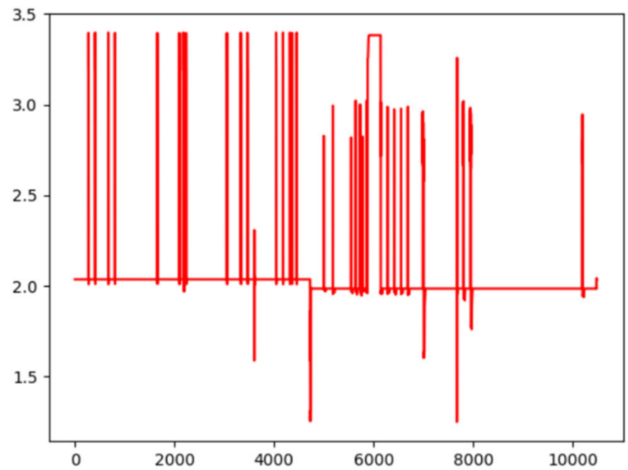

FIGURE 11. Scenario 2: sludge pump status prediction.

integers, a 0.5 error margin has been fixed for a value to be considered wrong in the last performance evaluation approach.

Values within the specified error margin can be easily extrapolated in order to compensate for the prediction error. For example, predicting the value 3.3 instead of 3 is considered correct in this second approach, which evaluates the ratio of correct predictions to total prediction points under those conditions. The described approaches for evaluating the performance were applied in both scenarios, the outcome being part of Table 4, which highlights very good results in terms of model accuracy.

TABLE 4. Accuracy of trained models – sludge pump status prediction.

Evaluation Approach	Scenario 1	Scenario 2
Coefficient of Determination (R^2)	0.546	0.546
Mean Squared Error (MSE)	1.056	1.056
Root Mean Squared Error (RMSE)	1.027	1.027
Accuracy (correct/total with 0.5 error margin)	98.625 %	96 %

2) CHEMICAL OXYGEN DEMAND PREDICTION AT WWTP INLET

The second practical application targeted the prediction of a quality indicator, CODcr. The same generic LSTM NN model serves as starting point, and the application pursued a second improvement objective, to optimize the legacy process control.

The list of characteristics that were chosen from the available data to be used as input data for the training of the LSTM model are enumerated in Table 5. All values were measured at the inlet of the WWTP. The output data was CODcr. Similar with the first application, a sample of 30 units, ensuring a 5 hours ahead prediction and 10 number of passes over the database were fixed.

TABLE 5. List of characteristics used as input data for neural network training – CODcr prediction at WWTP inlet.

Characteristic significance	Measurement unit	Type of value
CODcr	mg/l	Numerical float
Phosphate (PO4)	mg/l	Numerical float
Ammonium (NH4)	mg/l	Numerical float
Acidity (pH)	-	Numerical float [0-14] interval
Temperature	Celsius degrees	Numerical float
Flow from septic trucks	m3/h	Numerical float
Volume from septic trucks	m3	Numerical float

The LSTM model for the CODcr prediction was trained and then tested on data spanning in the time intervals presented in Table 6.

The LSTM network parameters configuration was: learning rate 0.01, batch size 32, epochs 50 and loss function MSE.

The result is shown in Figure 12 as a comparison between prediction and actual result and in Figure 13 as prediction only.

TABLE 6. Time intervals from which data was extracted to train and test the AI model – CODcr prediction at WWTP inlet.

Description	Time Interval
Training	
Start	7 September 2022 14:17:09
End	13 March 2023 04:02:41
Testing	
Start	28 December 2022 01:30:38
End	18 June 2023 09:08:13

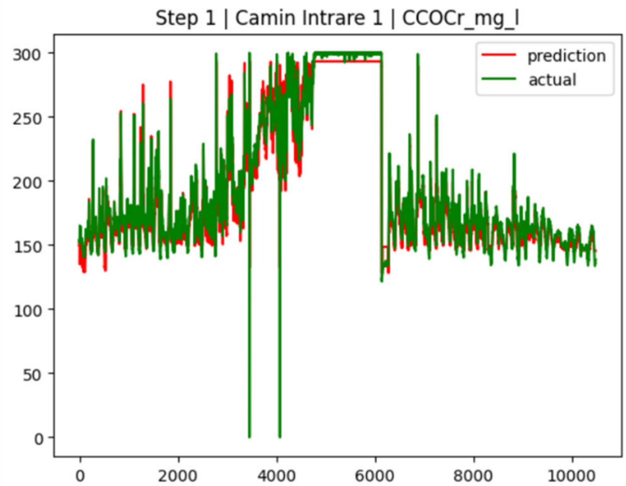

FIGURE 12. Comparison of CODcr prediction with actual values.

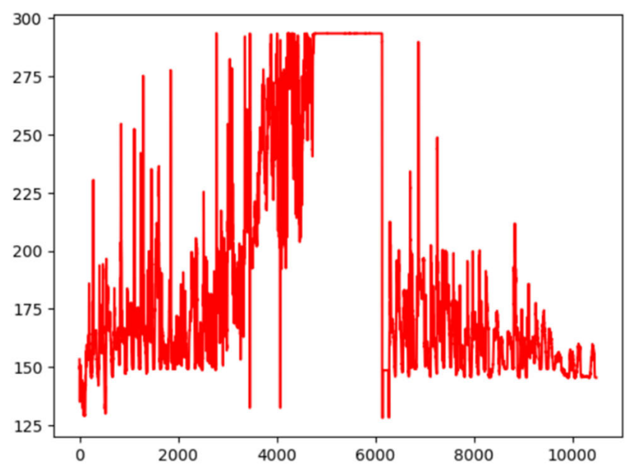

FIGURE 13. CODcr prediction.

To compute the accuracy of the model, the coefficient of determination (R^2) was initially evaluated through the *r2_score* function from Scikit-learn library. Also, the accuracy of the CODcr prediction was measured by using the *accuracy_score* function from Scikit-learn library and the standard MSE and RMSE techniques, the approaches concluding with the results presented in Table 7.

TABLE 7. Accuracy of trained model – CODcr prediction at WWTP inlet.

Evaluation Approach	Result
Coefficient of Determination (R^2)	0.422
Mean Squared Error (MSE)	173.535
Root Mean Squared Error (RMSE)	13.173
Accuracy (with Scikit-learn library)	97.237 %

IV. DISCUSSION AND CONCLUSION

Focusing on the equipment fault detection before occurrence, the accurate prediction of status obtained as a result of the current research proved to be promising. Also, the ability to forecast the water quality indicator to assure the necessary time for process control reconfiguration, together with the capability of the historian to react on the legacy system by adjusting the anoxic and aerobic time intervals, was considered successful. The 5 hours ahead prediction exceeded the requirements of the water company experts, offering the validation for tailoring the basic solution in other scenarios. Therefore, transition towards a different objective (e.g. air blower faults) will be approached. Mainly, the minimum number of defects within the stored data that is necessary for training the generic LSTM proposed network should be determined, and the dimension of the OPC UA tags set used as input for the network should be considered while following different objectives. The classic issues of under fitting or over fitting the model must be acknowledged.

Adapting and deploying the solution to work on lower hardware and software resources represented also an important target that was accomplished. Up to the authors knowledge, there is no such low-cost decentralized proactive historian as the one presented in the current paper.

Another reached objective was to include the possibility of incremental training within the proactive historian. Regarding the obtained results, there was no major difference from the batch training which proved excellent results, but no detailed and complete comparison was realized, this being a future goal as mentioned in the research strategy.

Regarding the research strategy, the duration of step 3 was consistent and had to be done for each pilot structure because of the process and technological specificities. Some issues were also impacting the results and needed corrections. For example, local maintenance actions that were only partially observable through local supervisory application tags resulted in removing the functioning hours counting variables from the algorithm training, although there would have been important in the decision making. The non-invasive character of the historian is important and the goal for the strategy was to function for any type of legacy local solution.

In order to quantify uncertainty in LSTM's predictions, several approaches are suitable, among which Monte Carlo Dropout, Bayesian inference or constructing prediction intervals through the Quantile Regression method. The resulting uncertainty measures can help facility operators in managing maintenance strategies by supporting informed decision making processes. Also, high certainty predictions could support the transition from reactive to preventive maintenance, with all the benefits of the latter included.

The methodology of assessing the impact of the proposed solution over the operational efficiency of the water treatment facility could include comparative analysis in which key performance indicators (KPIs) are compared from before and after the implementation or even control group comparison between different wastewater treatment plants, simulation studies and maintenance logs analysis. The most significantly affected performance indicators are: mean time between failures, mean time to repair, system uptime, equipment downtime, maintenance costs, energy consumption.

In conclusion, the current research presented the general overview of integrating predictive capabilities into a software tool able not just to collect data, but to analyze it and proactively improve the monitored technical system, in accordance with Industry 4.0 concepts. The paper presented two generic LSTM models, which were applied using real world industrial data. The solutions generated valuable results in terms of prediction accuracy, both for predicting equipment status and process parameter values.

REFERENCES

[1] M. Tabaa, F. Monteiro, H. Bensag, and A. Dandache, "Green Industrial Internet of Things from a smart industry perspectives," *Energy Rep.*, vol. 6, pp. 430–446, Nov. 2020.

[2] N. T. Ching, M. Ghobakhloo, M. Iranmanesh, P. Maroufkhani, and S. Asadi, "Industry 4.0 applications for sustainable manufacturing: A systematic literature review and a roadmap to sustainable development," *J. Cleaner Prod.*, vol. 334, Feb. 2022, Art. no. 130133.

[3] G. S. S. Leal, W. Guédria, and H. Panetto, "Enterprise interoperability assessment: A requirements engineering approach," *Int. J. Comput. Integr. Manuf.*, vol. 33, no. 3, pp. 265–286, 2020.

[4] I. González, A. J. Calderón, and J. M. Portalo, "Innovative multi-layered architecture for heterogeneous automation and monitoring systems: Application case of a photovoltaic smart microgrid," *Sustainability*, vol. 13, no. 4, p. 2234, Feb. 2021.

[5] I. Seilonen, V. Vyatkin, and U. D. Atmojo, "OPC UA information model and a wrapper for IEC 61499 runtimes," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2019, pp. 1008–1013.

[6] I. Alexandru and A. Korodi, "Improving OPC UA publish-subscribe mechanism over UDP with synchronization algorithm and multithreading broker application," *Sensors*, vol. 20, no. 19, p. 5591, 2020.

[7] K. Habib, M. H. M. Saad, A. Hussain, M. R. Sarker, and K. A. Alaghbari, "An aggregated data integration approach to the Web and cloud platforms through a modular REST-based OPC UA middleware," *Sensors*, vol. 22, no. 5, p. 1952, Mar. 2022.

[8] F.-C. Baiceanu, O. Ivanov, R.-C. Beniuga, B.-C. Neagu, and C.-M. Nemes, "A continuous multistage load shedding algorithm for industrial processes based on metaheuristic optimization," *Mathematics*, vol. 11, no. 12, p. 2684, Jun. 2023.

[9] S. Pachner and J. Miethlinger, "Smart data analysis for optimized manufacturing of powder coatings on corotating twin screw extruders," *AIP Conf. Proc.*, vol. 2055, no. 1, 2019, Art. no. 070010.

[10] J. Song, Y. C. Lee, and J. Lee, "Deep generative model with time series-image encoding for manufacturing fault detection in die casting process," *J. Intell. Manuf.*, vol. 34, no. 7, pp. 3001–3014, 2023.

[11] J. Villena Toro, A. Wiberg, and M. Tarkian, "Application of optimized convolutional neural network to fixture layout in automotive parts," *Int. J. Adv. Manuf. Technol.*, vol. 126, no. 1, pp. 339–353, 2023.

[12] S. Latif, M. Driss, W. Boulila, Z. E. Huma, S. S. Jamal, Z. Idrees, and J. Ahmad, "Deep learning for the Industrial Internet of Things (IIoT): A comprehensive survey of techniques, implementation frameworks, potential applications, and future directions," *Sensors*, vol. 21, no. 22, p. 7518, Nov. 2021, doi: 10.3390/s21227518.

[13] N. H. Yu and S. Baek, "Fault detection in automatic manufacturing processes via 2D image analysis using a combined CNN-LSTM model," in *Advances in Production Management Systems. Smart Manufacturing and Logistics Systems: Turning Ideas Into Action* (IFIP Advances in Information and Communication Technology), vol. 663. Cham, Switzerland: Springer, 2022.

[14] W. Qian, Y. Guo, H. Zhang, S. Huang, and L. Zhang, "Digital twin driven production progress prediction for discrete manufacturing workshop," *Robotics Comp.-Integr. Manuf.*, vol. 80, Apr. 2023, Art. no. 102456.

[15] I. Sideris, F. Crivelli, and M. Bambach, "GPyro: Uncertainty-aware temperature predictions for additive manufacturing," *J. Intell. Manuf.*, vol. 34, no. 1, pp. 243–259, Jan. 2023.

[16] Y. Han, N. Ding, Z. Geng, Z. Wang, and C. Chu, "An optimized long short-term memory network based fault diagnosis model for chemical processes," *J. Process Control*, vol. 92, pp. 161–168, Aug. 2020.

[17] M. Zhang, J. Li, Y. Li, and R. Xu, "Deep learning for short-term voltage stability assessment of power systems," *IEEE Access*, vol. 9, pp. 29711–29718, 2021.

[18] K. H. Goh and K. F. See, "Twenty years of water utility benchmarking: A bibliometric analysis of emerging interest in water research and collaboration," *J. Cleaner Prod.*, vol. 284, Feb. 2021, Art. no. 124711.

[19] M. Grzegorzek, K. Wartalska, and B. Kaźmierczak, "Review of water treatment methods with a focus on energy consumption," *Int. Commun. Heat Mass Transf.*, vol. 143, Apr. 2023, Art. no. 106674.

[20] M. Ladegourdie and J. Kua, "Performance analysis of OPC UA for industrial interoperability towards industry 4.0," *Performance Analysis of OPC UA for Industrial Interoperability towards Industry 4.0*, vol. 3, no. 4, pp. 507–525, 2022.

[21] Y.-L. Tsai, H.-C. Chang, S.-N. Lin, A.-H. Chiou, and T.-L. Lee, "Using convolutional neural networks in the development of a water pipe leakage and location identification system," *Appl. Sci.*, vol. 12, no. 16, p. 8034, Aug. 2022.

[22] R. Palmitessa, P. S. Mikkelsen, M. Borup, and A. W. K. Law, "Soft sensing of water depth in combined sewers using LSTM neural networks with missing observations," *J. Hydro-Environ. Res.*, vol. 38, pp. 106–116, Sep. 2021.

[23] M. Carratù, S. D. Iacono, G. Di Leo, V. Gallo, C. Liguori, and A. Pietrosanto, "Smart water meter based on deep neural network and undersampling for PWNC detection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[24] K. T. N. Nguyen, B. François, H. Balasubramanian, A. Dufour, and C. Brown, "Prediction of water quality extremes with composite quantile regression neural network," *Environ. Monitor. Assessment*, vol. 195, no. 2, p. 284, Feb. 2023.

[25] A. Robles-Velasco, C. Ramos-Salgado, J. Muñuzuri, and P. Cortés, "Artificial neural networks to forecast failures in water supply pipes," *Sustainability*, vol. 13, no. 15, p. 8226, Jul. 2021.

[26] *LSTM for Predictive Maintenance on Pump Sensor Data*. Accessed: Jun. 4, 2024. [Online]. Available: https://towardsdatascience.com/lstm-for-predictive-maintenance-on-pump-sensor-data-b43486eb3210

[27] A. Korodi, A. Nicolae, and I. A. Drăghici, "Proactive decentralized historian-improving legacy system in the water Industry 4.0 context," *Sustainability*, vol. 15, no. 15, p. 11487, Jul. 2023, doi: 10.3390/su151511487.

ADRIAN KORODI received the B.S., M.S., and Ph.D. degrees from Universitatea Politehnica Timişoara, in 2003, 2005, and 2008, respectively.

He is currently an Associate Professor with the Department of Automation and Applied Informatics. He was the Director of various research grants and contracts, many of them involving the industry (e.g., water and wastewater, automotive, and BMS). Throughout the years, he was an Automation/SCADA Expert working mostly in the water industry and more than 100 contracts of works. His research interests include the IIoT and industry 4.0, with an emphasis on protocols, interoperation and efficiency increasing solutions in industry, security issues in automation, and SCADA systems.

ANDREI NICOLAE received the B.S., M.S., and Ph.D. degrees from Universitatea Politehnica Timişoara, in 2016, 2018, and 2021, respectively.

He has eight years of experience as a Software Developer in the industry. He is currently an Associate Researcher with the Department of Automation and Applied Informatics. His research interests include the IIoT and industry 4.0.

DRAGOŞ BRISC received the B.S. degree from Universitatea Politehnica Timişoara in 2024. He was already involved into several research projects. He is a Software Developer specialized in Python, with expertise in machine learning solutions for time series predictions. Experienced in implementing industry-standard techniques for data analysis and forecasting. Passionate about applying algorithms and data-driven approaches to solve real-world challenges in various industries.

IONEL DRĂGHICI received the Ph.D. degree from Universitatea Politehnica Timişoara in 2015. He has significant experience working in the industry at the largest water company in the area for many years and was also involved in different research grants or partnerships with the academic area.

ADRIAN CORUI received the Ph.D. degree from Universitatea Politehnica Timişoara in 2009. He has significant experience working in the industry at the largest water company in the area for many years and was also involved in different research grants or partnerships with the academic area.