ICRIE 2023 4th INTERNATIONAL CONFERENCE ON RECENT INNOVATIONS IN ENGINEERING

*Proceedings of the 4th International Conference on Recent Innovation in Engineering*
*ICRIE 2023, University of Duhok, College of Engineering, 13th – 14th September 2023*
*(Special issue for Passer journal of basic and applied sciences)*
*Paper No. 18*

# Real-Time Predictive Maintenance System of Industrial Equipment without Historical Failure Data

**Mohammed E. Bahar[1,*], Abed A. Schokry[2] and Mohammed A. Alhanjouri[3]**
[1]Industrial Engineer, Ministry of National Economy, Palestine
[2]College of Engineering, Islamic University of Gaza, Palestine
[3]College of Engineering, Islamic University of Gaza, Palestine
* Corresponding author: Industrial Engineer, Ministry of National Economy, Gaza Strip, Palestine. Tel: +970595918793.
mohammed.e.bahar@gmail.com; aschokry@iugaza.edu.ps; mhanjouri@iugaza.edu.ps

**ABSTRACT**

Predictive maintenance (PdM) appears to predict faults before they occur because unexpected industrial equipment failures directly affect workers' safety, cost, and work continuity. In this context, developing a PdM system needs historical data for failures, but those data are not available in our case, which is a sewage centrifugal pump, where failures had not been recorded before. Therefore, this research aims to develop a system for PdM that works efficiently in real-time and does not need historical data for failures; it can also predict failures at different periods and conditions. The "Data-Driven" method is a suitable methodology to apply to conditions data; also, time series forecasting and anomaly detection (AD) are the most applicable models for the studied case. The Main Bearing Temperature was chosen in the models because it is the most applicable parameter. The models' performance was evaluated in two ways: accuracy and resource consumption (execution time and RAM). After that, the most important accuracy metrics are a root mean square error (RMSE) for forecasting models and excess Mass and Mass Volume for AD models. The experimental results presented "TBATS" as the best forecast model and QuantileAD from the "ADTK" model as the best AD method.

**KEYWORDS:** Anomaly Detection, Data-Driven, No Historical Failure Data, Predictive Maintenance, Time Series Forecasting.

## 1 INTRODUCTION

It is expected that authors will submit carefully written and proofread material. Careful checking for spelling and grammatical errors should be performed. The number of pages of the paper should be from 4 to 8.

Papers should clearly describe the background of the subject of the author's work, including the methods used, results and concluding discussion on the importance of the work. Papers are to be prepared in English, and SI units must be used. Technical terms should be explained unless they may be considered to be known to the conference community. The references should be numbered [1], or [2, 3], or [1, 4-6].

Maintaining a suitable work environment for workers and production equipment and preventing breakdowns are the most critical factors that industrial facility owners must consider. Production and operation costs affect the continuous provision of services and products. All the previously mentioned factors, directly and indirectly, affect maintenance, as its costs lie between 15% and 40% of total production costs (Löfsten, 2000). The main maintenance strategies are as follows: **Corrective Maintenance** (CM) is described as repair work when a malfunction occurs, but it produces interruptions in work (S. C. Lee, 2010). **Preventive Maintenance** (PM) inspects the components at periodic intervals to prevent unexpected machine breakdowns, but it is costly (S. Lee & Kim, 2018). **Condition-Based Maintenance** (CBM) reduces uncertainty and depends on the operator sensing the failure in the machine through abnormal conditions. After developing industry 4.0 techniques like Artificial Intelligence (AI) and the Internet of Things (IoT), the concept of **Predictive Maintenance** (PdM) was developed to predict failures before they happen based on industry 4.0 and without relying on the human factor. So, PdM is defined as techniques used to help detect equipment's status to determine when maintenance action should be done (Mobley, 2002). PdM is one of the most efficient maintenance processes in saving time, money, and effort, unlike costly PM or CM, which may cause an interruption in work (Candanedo et al., 2018). Statistically, according to the U.S. Department of Energy, PdM is saving about 8% to 12% higher than PM and 30% to 40% higher than CM; moreover, ten times return on investment, 70% to 75% elimination of breakdowns, 20% to 25% increase in production (Sullivan et al., 2010).

This research introduces a PdM system based on Industry 4.0 about centrifugal pumps without any historical failure data. The system will be implemented at the (P7) sewage pumping plant built in the Al-Zaytoun neighbourhood of Gaza City. In addition, this plant is under the Gaza municipality's supervision. This research extends to a funded project by the University College of Applied Sciences Technology Incubator (UCASTI) with a budget reaching $5,000. The system's significance is to predict and correct problems before they occur. An interruption of services in this sewage plant will negatively affect its surrounding environment (*Sewage Nightmare Due to Electricity in Gaza*, 2012). The system impacts sewage services in the community via more availability without interruption. It also impacts reducing maintenance costs for the plant operator.

Briefly, the PdM system uses a microcontroller with a portable modem to collect and store data from the preinstalled controller system on the pump. Then, send data to a cloud service integrated with software to store, process, and analyze data as well as a dashboard to view results through an internet browser; see Figure 1. Different programming languages will be used in this system because it contains more than one framework. This work involves many scientific fields, such as control, electrical circuits, ML, programming languages, databases, IoT, cloud computing, and communications. Before implementing a PdM system in the current case, the main problem is that the pump does not have historical data for failures. Moreover, the pump has not failed since its installation in 2018, which makes it difficult to predict its failure in the near term. Traditional PdM system based on supervised learning needs previous failure data to detect, identify and classify failure patterns to work adequately, which is not available in the current case. Therefore, models of supervised machine learning can not be used. So, the challenge is to use a learning method that differs from "classification" to develop a PdM system with the smallest margin of error in the current case.

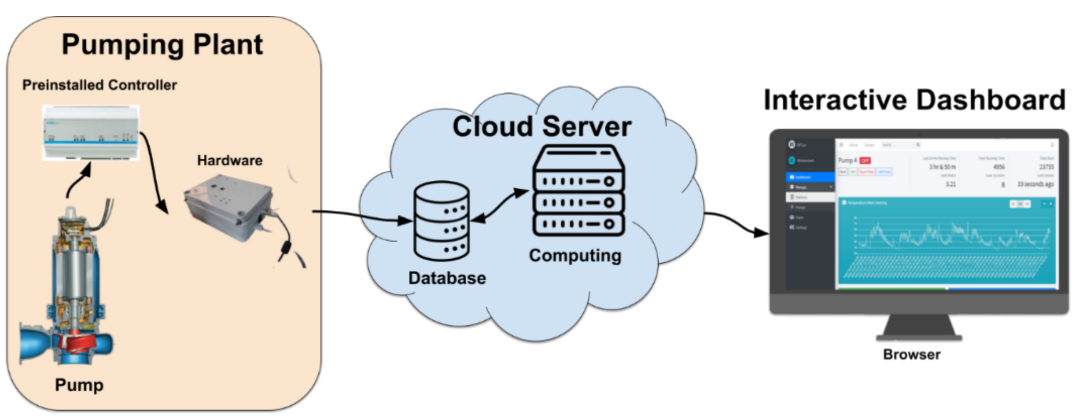


**Figure 1.** Data Flow of PdM System

This research aims to develop a real-time PdM system of industrial equipment lacking historical failure data and works efficiently at different prediction periods and operating conditions. Also, Its objectives are: 1) Collect real-time data required to predict conditions from the centrifugal pump and instantly verification to ensure the data are not distorted; 2) Process data by filtering, cleaning, and adjusting the data without affecting the results; and determine optimum methods; 3) Identify models, algorithms, and processes with minimum computing time and the highest accuracy to obtain the best performance in real-time; 4) Evaluate models and algorithms in terms of execution time, RAM consumption, and accuracy of outcomes; 5) Introduce results in a simple presentable way to make the appropriate decision as soon as possible before a problem occurs. Also, the main contributions of this research are: 1) Introducing a modified method for predicting the conditions of new or non-defective industrial equipment; in other words, applying PdM somehow so that it suits the current case. 2) Developing a detailed methodology for producing an integrated system to predict equipment conditions, accurately classify failure before it occurs, and display results to make a correct decision. 3) Designing unsupervised learning models to be applied in a PdM system are designed to be applicable to several environments and variables. 4) Define criteria and choose the optimum techniques from models and algorithms at the best performance to get the most accurate results.

## 2 RELATED WORKS

As equipment failures dangerously affect production, the need for PdM systems is increasingly essential; PdM's importance is in predicting when failure will occur. Many recent studies have focused on using various methods in the PdM concept through data classification to predict the next failure date or during a specific period (Çınar et al., 2020). Few researchers have taken methods different from classification to apply PdM's concept. The related works aim to provide concepts and ideas to achieve a PdM system in real-time methods that do not rely on data classification. In the beginning, the application of PdM has been widely associated with collecting and analyzing various data sources. In a study by (Paolanti et al., 2018), it was mentioned that three essential data sources must be used to implement PdM, which are as follows: 1) Fault history, 2) Maintenance/repair history, and 3) Machine conditions. Many recent studies have relied on using one source of data (Amruthnath & Gupta, 2018b, 2018a; Tinga & Loendersloot, 2019), and some have used more than one source of data (W. J. Lee et al., 2019; Paolanti et al., 2018). Through these studies, the authors can conclude that using one or more sources can sufficiently use PdM with high efficiency and accuracy. However, there is a lack of research to implement a PdM system that relies on machine conditions only, without historical failure data or a history of maintenance and repair.

In a paper published by (Zonta et al., 2020), it is mentioned that the most common predictive methods are divided into three main sections as follows: 1) Classification like Physical model-based, knowledge-based, and data-driven; 2) Artificial Neural Networks (ANNs), ML, and algorithms; 3) CBM and Remaining Useful Life (RUL); In addition, it was mentioned that the classification is the most used. On the other hand, several recent studies have used time series forecasting and Anomaly Detection (AD) in dealing with time series in real time to achieve PdM (S. Lee & Kim, 2018; Lin et al., 2019; Rodrigues & Zarate, 2019). From the above, it can be concluded that it needs to apply these methods to overcome the current situation. Therefore, time series forecasting and AD are the most appropriate methods to achieve a PdM system without historical failure data.

### 2.1 Time Series Forecasting

With the development of AI, ML, and IoT techniques, many developments have occurred in industrial processes. Predicting industrial equipment conditions in the future is based on appropriate analysis of time series in the past. The following are four advanced studies on time series forecasting of PdM based on real-time: The first one, a study by (Amihai et al., 2018), aimed to show that industrial conditions can be predicted by ML applied to data from industrial sectors. Vibration monitoring was used in this study, as data were collected from industrial pumps. The Random Forest (RF) algorithm was used as a failure prediction algorithm. The study results showed that predicting the machine's state in the industrial sector is possible. It is difficult to repeat the study on the case being worked on because the vibrations differ in their classification from one industrial equipment to another. The second study by (Lin et al., 2019) found that the Remaining Useful Life (RUL) method needs improvement to implement PdM, so they propose a Time Series Prediction (TSP) algorithm based on an autoregressive integrated moving average (ARIMA) model to avoid random processes. Data were collected from Throttle Valve and Oilless Bushing. In addition, the Pre-Alarm Module (PreAM) was used to make an alert of real-time maintenance. Finally, this study contains techniques and models that can be used and modified to enhance the current research.

A third study prepared by (Fernandes et al., 2020) aimed to develop a platform for handling data produced by Industry 4.0 and provide a framework for acquiring, storing, and analyzing data and knowledge extraction. Data were collected from household water boilers from ten different device models. Long Short-term Memory (LSTM) has been studied, which is a recurrent neural network (RNN) architecture designed to handle sequential data. The LSTM model was prepared when it had been fed one day of data to predict failures for the next seven days. This study contains usable ideas such as data processing and arranging operations in the system. The fourth, a new comprehensive maintenance support system, was proposed by (Jain et al., 2021). In addition, Weibull developed a survival theory-based real-time prognostics module named (WTTE-RNN). Data were collected from cutting tools. This study's contribution supports a comprehensive maintenance system: 1) a Real-time prognostics approach; 2) a Decision model that immediately evaluates operational and maintenance costs; 3) To determine the right time for implementing maintenance activities. This study can enhance this research in more ways to measure when maintenance is required.

Finally, it has benefited from previous studies on the subject of time series forecasting, as many different techniques for predicting time series in real-time have been identified, and the most appropriate models to be applied in this research have been identified. (Lin et al., 2019) studied data are close to this research, so their models will be better examined. The Studies (Amihai et al., 2018; Fernandes et al., 2020; Jain et al., 2021) include data processing techniques and the design of framework processes that are beneficial for the research.

## 2.2 Anomaly Detection

One of the applications used in industrial fields is anomaly detection. This application benefits the industrial field as it helps remote detection of abnormal operation of industrial equipment, and it is one of the applications of predictive maintenance. Some of the studies in time series AD will be as follows: (Guo et al., 2018) proposed a Gated Recurrent Unit (GRU) based on Gaussian Mixture (GM) and Variational Autoencoder (VAE) system for anomaly detection called GGM-VAE. The system experimented on the Intel Berkeley research lab dataset and the Yahoo AD dataset. This study is consistent with the research being worked on, as it presents a method for dealing with unsupervised learning on multidimensional time-series data and provides ideas for processing and comparing data. In a study conducted by (S. Lee & Kim, 2018), a method was used for AD called an Anomaly Detection System using SARIMA and STL (ADSaS). This method uses two approaches to detect anomalies, which are the Seasonal Autoregressive Integrated Moving Average (SARIMA) and the Seasonal Trend decomposition using Loess (STL). The researchers used datasets in the experiment from Numenta Anomaly Benchmark (NAB) and “P” corporation datasets. Furthermore, ADSaS succeeded in detecting anomalies right in real-time and can be applied to the actual industry. This paper presents a practical system to detect anomalies in real-time, as the advantages of this study are fast algorithms and do not consume resources.

Authors (Gao et al., 2020) developed the Robust Time Series Anomaly Detection (RobustTAD) framework by integrating robust seasonal-trend decomposition and convolutional neural networks (CNN). This framework is used as an online service and used by Alibaba Group. Collected time series from the actual traffic to some Yahoo websites. Finally, researchers concluded that RobustTAD works better on public Yahoo benchmark datasets compared with forecasting-based algorithms, decomposition-based algorithms, and recent neural network-based algorithms. In this paper, the researchers presented a new method for analyzing data and time series anomaly detection, strengthening this study because it tested and worked on a real case. An unsupervised time series model was introduced by (Xu et al., 2021) for anomaly detection, and this model is based on Nouveau VAE (NVAE). The model to detect the anomaly is Time Series to Image VAE (T2IVAE). The data were collected from NAB, NASA, and network switches. AD accuracy is one of the factors that strengthen this study as better than the other mentioned models, but one of the weaknesses is working in specific cases because it works on univariate time series.

The authors benefited from previous studies on AD, where many techniques and models that were characterized by accuracy and speed in real-time AD were identified. (Gao et al., 2020; Guo et al., 2018) Provided new technologies and models for processing and analyzing data and detecting anomalies. (S. Lee & Kim, 2018; Xu et al., 2021) presented models distinguished by accuracy, speed, and low consumption of resources, but one of their research disadvantages is working on a single variable.

# 3 MATERIALS AND METHODS

## 3.1 Predictive Maintenance in Industry 4.0

PdM aims to reduce downtime and the cost of maintenance. The assumption to achieve PdM is zero defects in industrial equipment by continuously monitoring it and predicting failure before it occurs. It was difficult to reach zero failures in the past except through costly preventive maintenance. However, with the development of cloud computing, big data, and the IoT, it became possible to reach zero failures at the lowest cost through PdM (J. Lee et al., 2014). Industry 4.0 is based on intelligent systems and internet-based technologies and refers to the fourth industrial revolution.

PdM has many of the benefits mentioned in this research. However, due to the incredible complexity and high flexibility of modern industrial systems, stakeholders face many barriers when implementing the system (Vuksanović Herceg et al., 2020). From a study published by (Ran et al., 2019), the following are the three most important things that must be taken into consideration when implementing a PdM system: 1) System Architecture, as multiple systems participate in PdM system operation so the systems must be compatible with industry standards, easily integrate with current or future techniques, and meet the essential requirements of PdM system like forecast and anomaly detection; 2) System Purpose, Since minimizing cost and reliability are the two primary purposes of PdM, the primary purpose must be defined and balanced as much as possible because increasing one purpose affects the other, e.g. cost and reliability; 3) PdM Approaches, to obtain the best results, a PdM approach should be selected appropriately for the situation being worked on because the methods and models are varied and customized according to each case.

Prediction models are used to obtain a prediction of the future and to detect anomalies within the system. Models differ regarding the nature of data and the results to be reached. Also, they are divided into three categories as follows: 1) Classification Models: These models aim to identify if the machine will fail soon or not; 2) Regression Models: These models aim to predict RUL by analyzing time-series data, including a dependent variable and one (or more) independent variables; 3) Anomaly Detection Models: These models aim to identify rare events or observations via determining significant differences from the majority of the data.

## 3.2 Artificial Intelligence-Based Predictive Maintenance

The simplest way to describe AI is that intelligence is proven by a machine instead of natural intelligence, which contains emotions and consciousness. Some concepts of AI appeared and improved with the development of mathematics. However, with the emergence of semiconductors, computers, and the rapid development of mathematics in the 20th century, a qualitative leap occurred in the sciences of AI and ML (Ekmekci & Arda, 2020). ML is defined as the study of computer algorithms that improve automatically through data and experience, and it is part of artificial intelligence. A subset of ML is related to cloud statistics, so it is sometimes called statistical learning that focuses on making a prediction. ML algorithms are widely used and have many fields, such as medicine, industry, and agriculture. In this research, work has been done in the industry field, specifically PdM, and taking advantage of the many benefits of AI.

The most popular algorithms used in ML for PdM are Artificial Neural Network (ANN), Support Vector Machine (SVM), Decision Tree (DT), k-nearest Neighbors (k-NN), and finally, Random Forest (RF). And algorithms in DL are: Autoencoder (AE), Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), and Generative Adversarial Network (GAN). According to the purpose for which it is planned and the data type in PdM, these algorithms have different advantages and disadvantages, and they differ in their applications, speed, accuracy, and resource consumption and are often combined with other algorithms to improve their performance.

### 3.3 System Structure

The project started in March 2020, when the general structure of this project was agreed to be created, from planning until closing. The fieldwork began with a search for the case to be worked on; determinants of choosing this case are as follows: 1) Unplanned downtime of industrial equipment during operation is not acceptable; 2) the unplanned stoppage of equipment does cause damage to the surrounding environment; 3) The equipment is difficult to access or work around it is not easy. The public sector has been chosen, as some services in the public sector meet all the determinants. The sewage pumping plant (P7) in the Al-Zeitoun neighborhood in the Asqula area is the most needed PdM; because if it stops for 12 hours, most of the surrounding area will be affected through over flood. Moreover, the lack of a natural slope in that area will cause the water not to drain to any other area, (Gaza Municipality Confirms Its Readiness for the Storm, 2015; Sewage Collapse in Southern Gaza, 2016). Also, it is not easy to stay around these pumps during normal operating conditions. These reasons increase the need to keep pumps operating at optimal efficiency without interruption, so this pump was chosen to work on it. This plant was built in 1998 and was exposed to several problems that caused damage to the surrounding environment and residents. It is worth noticing that the plant serves 250 thousand people and is operated by the Gaza Municipality; it has a capacity of 1200 cubic meters per hour and contains four pumps (Wastewater Treatment Plants Are Overflowing at Every Moment, 2008).

It was previously mentioned that the plant has four pumps, and all the pumps are the same but differ in the type of built-in controllers. Pump No. 4 was selected in the plant, and the reason for choosing this pump is that the control room contains an MAS 711 controller, where this controller can send and receive data from the pump. This pump (FLYGT 3312/775) specifications are as follows: 2000 kg, 985 rpm, and 550 mm; and motor (43-56-6BC): 1400kg, 180 kW, 360 A, 985 rpm, and IP68. All of the above (pump,

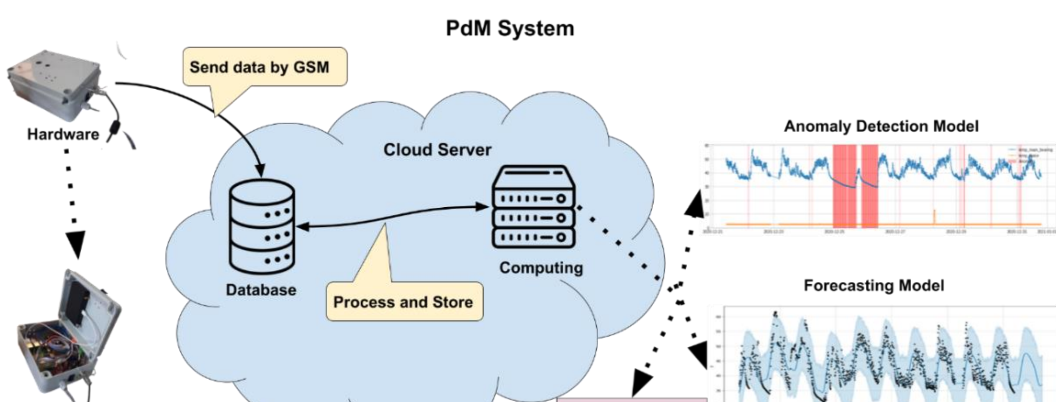


**Figure 2.** PdM system in detail

motor, and controller) are from Xylem Inc., which is a large water technology provider from the United States (Xylem Water Solutions & Water Technology | Xylem US, 2021).

It is necessary to explain the system inputs and where they came from, system outputs, and how the interaction between them happened, including all internal processes and stages of the system, to display the structure of the PdM system. Four sensors (two for temperature and two for leakages) are built-in pump that sends data to the MAS 711 controller. Data are stored every 10 minutes for 24 hours; this period is not enough to implement PdM, so an external unit has been developed and implemented to support running the PdM system with the best results. From this point, the PdM system application begins; the designed device receives the data, then verifies the data (whether the values are valid or not), and stores it every 0.5 minutes for extended periods (years), which helps to implement AI models and obtain relevant results. Additionally, the stored data are sent to a cloud server via a 4G modem. It is worth noting that the device was designed with external assistance from an electronics engineer.

The data are received on a cloud server (VPS) and stored in a database (SQL), then the data are processed and prepared to apply two AI models (Forecasting and AD). Then, the outputs of the two models are stored in a database for the Interactive Dashboard to communicate with the database separately Figure 2; at this point, the PdM system ends. Finally, data, predictions (model outputs), and alerts (if they occur) are accessed through an Interactive Dashboard; these results can be displayed on any device with a browser.

### 3.4 Tools and Techniques Used

The PdM system implemented is based on several platforms, software, hardware, programming languages, and libraries; all of the above can be described and detailed in two parts.

The first part is hardware: The designed device consists of a Raspberry Pi 3, 4G Modem, Electricity Regulator, Fan, Electrical and Data Connections, and 20 A Battery. Also, an outer case is needed to collect these parts to protect them from shocks and dust; additionally, the Raspberry Pi version is basic. The specification of the cloud server (VPS) that was used to process and display data in the Dashboard is 2 CPU Core, 2 GB RAM, 32 GB HDD, and 20 TB traffic. In addition, Google servers from Drive (Google Colab) were used to analyze data and test AI models and algorithms to get the most reliable results. Colab server specifications are as follows: CPU is Intel(R) Xeon(R) @ 2.30GHz, one socket, one core per socket, two threads per core, L3 cache is 46080K, RAM is 13GB, and hard disk available is 68GB; specifications may vary for each experiment (sessions) but will be maintained the stability of specifications for all experiments (Exactly similar).

The second one is software: The programming language used mainly in the system is Python. This language was used in the program installed on the designed device. This language was used in data analysis, application, and evaluation of AI models and algorithms. The main libraries (frequently used) that were relied upon in the PdM system data analysis are Pandas, NumPy, Matplotlib, Sklearn, etc.

### 3.5 Data Collection

Data goes through several stages before being stored in the database, so it must be ensured that all required data are stored correctly and without interruption. In the first stage, the data are collected by four sensors built-in in the pump and sent data to MAS 711 controller; the sensors and thresholds are as follows: 1) Main Bearing Temperature (T < 140 ºC); 2) Stator Winding Temperature (20 ohm < R < 3000 ohm); 3) Leakage Stator Housing (3 mA < I < 22 mA); 4) Leakage Junction Box (3 mA < I < 22 mA), see Figure 3. Figure and Thresholds in this paragraph from MAS 711 Installation and Operation manual.

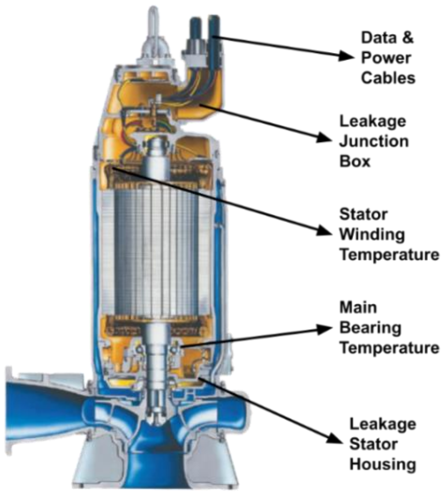


**Figure 3.** Pump Sensors and Cables

In the next stage, data are stored (limitedly) in a MAS 711 controller, which determines whether the pump needs to be stopped or not and shows pump status and other data; after which the designed device (Raspberry Pi 3) receives the data extensively as it adds several new readings; the data that is added from MAS 711 and stored in the device is as follows: 1) State of Pump (Running or not); 2) Total Starting Times (number of running times); 3) Total Running Time (Running hours from installation time); 4) Four alert readings that include the four sensors; 5) time and date of creating and storing; 6) ID (request number); 7) Pump ID (pump number). At this stage, the correctness of readings coming from the controller is verified by Raspberry Pi to reduce errors as possible. Data are received by sending and receiving requests every half a minute through the RS-485 port. In the final stage, these data are sent to the server using an API, where it is stored in a SQL database.

### 3.6 Data Preprocessing

This subsection explains the method of preprocessing raw data to become ready for applying AI models; look at Algorithm 1. Data preprocessing is divided into several levels, as follows. In the beginning, to clean the data, the sensors sometimes give incorrect readings such as (Celsius: 5000 or 0), (ohm: 0 or negative), (date: 1970), Not a Number (NaN), etc. All incorrect readings are immediately deleted from the database so as not to affect the system prediction results, A and B in Figure 4.

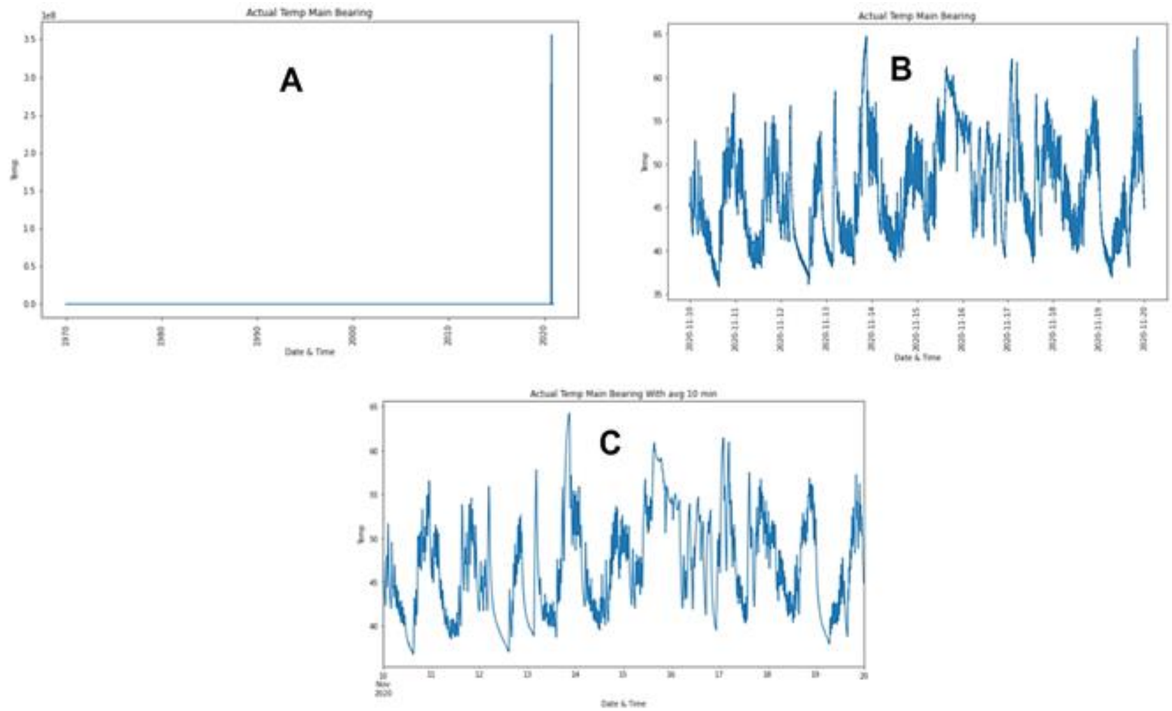


**Figure 4.** A) Temperature before deleting incorrect readings; B) After deleting it; C) After

Next, the selected attributes to be worked on are the time of the MAS 711 controller requesting data ('data_time'), main bearing temperature ('temp_main_bearing'), and stator winding temperature ('temp_stator'). The other two sensors are neglected because they do not change over time during the data collection period (not valid for prediction). The other attributes are helpful for use in the dashboard but not helpful for training AI models. The next step is to choose the time as an index in a dataset and sort the data from oldest to newest (to fit with used models), then delete the seconds to reduce noise in the data, as this level of accuracy is not required. Finally, every ten minutes is collected (resample) in one point to train the system where the average values are obtained during this period, C in Figure 4. A period of ten minutes was chosen after data analysis, as it is faster in processing and more stable (less sensitive to alerts) and gives the appropriate prediction accuracy for this type of case. The person responsible for maintaining the pump wants to know the faults through three hours at least (every ten minutes is suitable in this case). This accuracy (10 min) allows predictions for the next 24 hours.

**Algorithm 1: Data Preprocessing**

**Input:** Raw data imported from a database
**Output:** Processed data to use in models
**Step 1**
Importing Raw data to DataFrame named **df_raw**
Delete rows from **df_raw** if ('temp_main_bearing' = 0) or ('temp_main_bearing' > 200) or ('temp_stator' = 0) or ('temp_stator' > 200) or ('data_time' contain "1970") or ('total_start' = 0) or ('total_running_time' = 0) or ('leak_junction' = 0)
Drop (NaN) from **df_raw** if it exists.
**Step 2**
Choose three attributes 'data_time' and 'temp_main_bearing' and 'temp_stator' from **df_raw** to a new DataFrame named **df**
**Step 3**
**df** index as 'data_time' and define it as date time
Sort **df** index (data_time) from oldest to newest
Remove seconds from **df** index (data_time)
**Step 4**
Resample **df** values ('temp_main_bearing' and 'temp_stator') to 10 Minutes with mean and drop NaN, importing results to DataFrame named **df_10min**

```
Reset df_10min index to default values
Round values in df_10min to 2 decimal
End
```

## 3.7 PdM Models

Models and libraries were searched on the internet to apply the algorithms found and choose the most suitable for the data type being worked on. This subsection will cover in detail these models and libraries and how to use them.

### 3.7.1 Forecasting Models

1) Seasonal Autoregressive Integrated Moving Average (SARIMA)

Autoregressive Integrated Moving Average (ARIMA) is one of the most popular univariate time series forecasting methods. This method deals properly with trends but fails to deal with seasonality in the data. Therefore, seasonal parameters were added to the equation for this method, and it was called SARIMA or seasonal ARIMA. The SARIMA model used in this research was imported from stats models, a Python module containing several statistical models. There are seven parameters in SARIMA model {SARIMA(p,d,q)×(P, D,Q)s}, which are as follows: 1) "p" is trend autoregression; 2) "d" is trend difference; 3) "q" is a trend moving average; 4) "P" is seasonal autoregressive; 5) "D" is seasonal difference; 6) "Q" is seasonal moving average; 7) "m" is a number of time steps for a seasonal period. In addition to the above, this parameter is used in the SARIMA model, in which "t" is the Trend parameter. A loop was used, trying all possibilities and giving the lowest ten values from the RMSE of SARIMA parameters. Now, all the inputs, which are the optimum parameters and the processed data, are ready. The modelling starts from this step, which is to put the inputs into the model, train the model, and then enter the period to predict it.

2) SARIMAX with Fourier

The Seasonal Autoregressive Integrated Moving Average with Exogenous Variables (SARIMAX) model is not very different from SARIMA except for one additional variable, a vector of exogenous variables named X. SARIMAX model has the same parameters, X used in internal calculations of a model. Fourier transform has been added to the model because this addition can deal with more than one season in the data; in this way, it is possible to improve the forecast accuracy. The method used to implement the SARIMAX model does not require many steps or even determine parameters; it only needs small and large seasonal testing periods and forecast periods. The main libraries are imported to handle and display data. The next step is to determine the large and small seasonality and test period from the data (determined by the nature of the data), then apply the Fourier terms to add it to the model that will be trained. Next, the model is trained according to the selected data, parameters, and Fourier terms. Then the forecast process is carried out according to the period specified for the forecast; thus, a prediction was obtained by the applied method.

3) Facebook Prophet

The Prophet Model is open-source software released by Facebook; it is a time series forecasting procedure that works with non-linear trends and fits with seasonality (year, week, day). The model works with strong seasonal effects and with extreme values and missing data. From the published paper of (Taylor and Letham, 2017), on which the Prophet model is based, it is clear that it works by calculating more than one feature within the data, and each feature has an equation to obtain the best results in less time. Application of this model on data does not require many steps, as it is characterized by ease and speed. After importing data, the confidence interval and mode of seasonality (additive or multiplicative) are determined. Then, start training (fit) the model. The next stage is to choose the period to be forecasted then forecast the data.

4) TBATS

TBATS is a model for forecasting time series that contains complex seasonality; the method used for forecasting is exponential smoothing. It is characterized by dealing with more than one seasonal pattern in a univariate time series and extracting forecasting from them. Examples of complex predictions that it can handle are non-integer and non-nested seasonality, in addition to large-period seasonality. TBATS is an acronym for "T" (Trigonometric seasonality), "B" (Box-Cox transformation), "A" (ARIMA errors), "T" (Trend), and "S" (Seasonal components). The model is built in two steps: 1) selection of seasonal periods and training of the system; 2) choosing forecast period and forecast. Model reference is (De Livera et al., 2011).

5) LightGBM

Light Gradient Boosting Machine (LightGBM) is open-source software developed by Microsoft. It is based on a gradient boosting framework and decision tree algorithms; plus, it is used in regression and classification. According to (Ke et al., 2017) from Microsoft, LightGBM is a highly efficient gradient-boosting decision tree, and it is faster than all algorithms that use the same method. When implementing the algorithm, the LightGBM library must be installed. Firstly, the parameter must be chosen by testing specific numbers for window length and choosing an optimum number to get the most reliable prediction. The next step is creating a forecaster with LightGBM with optimum parameters; finally, training and forecasting the model.

6) Keras

Keras is an open-source software library programmed in Python, running on the TensorFlow platform. According to Keras Homepage, this platform is fast and easy to use, and its efficiency has been proven, so it is used in research centers such as NASA, CERN, and others. Keras must be imported from Tensorflow before applying the model. The steps for building a Keras model are as follows: 1) Data processing by normalizing it (all data between zero and one); 2) Define all the parameters that will be useful in building the model (split fraction, past, future, learning rate, batch size, and epochs); 3) Choose training and validation periods; 4) Build the model by selecting the appropriate layers with the specific data type or method for forecasting such as LSTM or basic Keras; 5) Start training after placing all the parameters in the model; 6) Predicting and displaying results. More than one model has been built using Keras.

### 3.7.2 Anomaly Detection Models

1) Anomaly Detection Toolkit (ADTK)

Anomaly Detection Toolkit (ADTK) is a package that runs in Python, and it is for unlabelled time series because the type of anomaly detection for this package is unsupervised AD. ADTK has many methods (detectors) to detect anomalies; it is easy to use because of a few implementation steps, plus displaying data clearly. The detectors used in building the model are as follows: 1) ThresholdAD: to detect thresholds; 2) QuantileAD: to detect anomalies based on quantiles of data; 3) PersistAD: to detect a change of value from its preceding average or median; 4) VolatilityShiftAD: to detect a shift of volatility; 5) SeasonalAD: to detects values away from seasonal patterns; 6) AutoregressionAD: to detects anomalous from autoregression property; 7) GeneralizedESDTestAD: to detects anomaly based on generalized extreme studentized deviate (ESD) test; these are not all detectors, but the most used. Before implementation, the data must be verified by ADTK to work correctly with the detectors. The first step is to import the detector from the package and then define the required variables for each detector (differ from one to another). Finally, choose the data start anomaly detector and display the results.

2) PyCaret

PyCaret is an open-source low-code ML library in Python; it works in minutes, from preparing data to launching the model, as shown on the PyCaret Homepage. The PyCaret library is distinguished by many characteristics that distinguish it from other libraries. There are fifty options to deal with the data when installing the AD module and linking it to the data. The first step is to make the model by selecting the detector and parameters identified for it. At next step is to start the detector and store the points identified as anomalies (model results) in a data frame. Finally, the data is stored using the same library, characterized by interacting with the data in real time.

3) Keras

The application of an AD model by Keras library requires more steps and more knowledge and experience to design the model. The steps for applying the model are as follows: 1) Split data for training and test data and normalize the training data to fit the model; 2) Create sequences in data and reshape training data based on it; 3) Build a model, this step requires a good knowledge of the nature of the data and contains several layers to obtain the most appropriate training result; 4) The training is started based on selected parameters, then evaluate the model through learning curve; 5) Based on the trained model, the threshold for detecting anomalies is determined from MAE loss; 6) The last step is to prepare the test data and then detect anomalies by threshold.

## 3.8 Evaluation

Selecting the appropriate model must be scientific and measurable, and measuring performance must be standardized for all models, where each type of prediction model (forecasting or AD) is evaluated separately. When choosing the model to be applied in the PdM system, two main factors must be considered: the model's performance in terms of output quality (Model Performance Metrics) and the model's performance in terms of computer resource consumption (Code Performance Metrics).

### 3.8.1 Model Performance Metrics

A. Evaluating Forecasting Models

The following five metrics can be used in this research to evaluate the forecasting models mentioned in Table 1. The evaluation was done by calculating the same predicted time and date with the actual values; keep in mind that they were described by smaller values (more minor is better) when comparing these metrics. To read this table clearly, understand that the actual data are collected during period X and predicted data are the predicted values during period X.

**Table 1.** Performance Metric of Forecasting Models

| Performance Metric | Abbreviations | Formula |
|---|---|---|
| Mean Squared Error | MSE | $MSE = \frac{1}{n}\sum_{i=1}^{n}(actual - predicted)^2$ |
| Root Mean Squared Error | RMSE | $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(actual - predicted)^2}$ |
| Mean Absolute Error | MAE | $MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert actual - predicted\rvert$ |
| Mean Absolute Percentage Error | MAPE | $MAPE = \frac{1}{n}\sum\left\lvert\frac{actual - predicted}{actual}\right\rvert$ |
| Symmetric Mean Absolute Percentage Error | sMAPE | $sMAPE = \frac{1}{n}\sum\frac{\lvert predicted - actual\rvert}{(actual + predicted)/2}$ |

The previous metrics will be discussed here to determine the most appropriate metrics for this research. According to (Hyndman & Athanasopoulos, 2018), MSE, RMSE, and MAE are "Scale-dependent errors" metrics, and MAPE and sMAPE are "Percentage errors" metrics. What distinguishes Scale-dependent is that it is easier to understand and is widely used for these types of metrics. However, its disadvantage is that it is not suitable for comparison at different units. Unlike the Percentage metrics, which can compare different units or large variation scales, it is preferred to use for these types of cases only because its accuracy is lower than the Scale-dependent. For Scale-dependent metrics, RMSE is preferred over MSE because RMSE is more useful when using large error values. In addition, MAE is not sensitive to outliers, but RMSE is better at dealing with data highly sensitive to outliers. Finally, all metrics will be used to compare the data. However, the primary reliance will be on the RMSE metric because it corresponds to research data as it has one unit (temperature), and outliers cannot be neglected.

B. Evaluating Anomaly Detection Models

When choosing a suitable method for AD evaluation, several challenges will be faced and have to be solved because the data are unlabeled. Plus, no one with experience can label the data to test the models, as one of the proposed methods for evaluating the models is labelling data and testing it. Because there were no failures during the data collection period, it was preferred to look for the operating roughness and evaluate it, but it is not possible to use it for the selected pump after an in-depth search for a method to evaluate unsupervised AD models. A research paper titled "How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms?" was published (Goix, 2016). Goix's paper relies on mathematical concepts of Excess Mass and Mass Volume in equations (1) and (2), where the results of this paper showed that two methods are suitable for unlabeled data evaluation.

$$EM_s(t) = \sup_{u \geq 0} \ \mathbb{P}(s(\mathbf{X}) \geq u) \ - \ t\mathrm{Leb}(s \geq u) \qquad (1)$$

$$MV_s(\alpha) = \inf_{u \geq 0} \ \mathrm{Leb}(s \geq u) \ \ s.t. \ \ \mathbb{P}(s(\mathbf{X}) \geq u) \geq \alpha \qquad (2)$$

The previously mentioned methods were applied through a library programmed based on Goix's paper when facing difficulties applying them to used models (Leary, 2021). After a request from the author, Leary updated the library to match some models used. In addition, library codes have been edited by the author, so the codes are compatible with all models used in this research. Finally, to simplify understanding of this method's results, the highest "Excess Mass" values are better, and the lowest "Mass Volume" values are better. The two evaluation methods will be used, and outliers will be ignored to interpret the results easily.

### 3.8.2 Code Performance Metrics

Improving code performance is essential for any program. Two reasons are increasing the importance of measuring program performance used in this research: 1) it depends on AI, which needs more resources; 2) it works on the cloud, as this solution is more expensive than local devices in high computing. Performance measurement is done by measuring execution time (running time required in milliseconds) and maximum RAM (peak size in MB) for specific lines of code. In comparison, this research is essentially concerned with models for forecasting or AD, not the whole code lines. For the comparison between the models to be valid, it relied on measuring the performance of specific lines of code for each model. In forecasting models, the performance of the training code and the fitting (predicting) code are measured. In AD models, only fitting (predicting) code performance is measured, except for models based on DL, which rely on training and fitting.

# 4 RESULTS AND DISCUSSION

## 4.1 Experimental Data

Data are a major focus in this research, as the Data-Driven approach was used to apply AI models. Data preprocessing is explained in the previous section to simplify the understanding of models' application methodology. Requests from the pump are stored every half a minute; the number of requests that have been stored is 235,678 from 09/28/2020 until 25/12/2020 (89 days); this period is enough to train the appropriate model because the prediction period does not exceed a day. After deleting the rows that contain errors in the data, a deleting method described previously, their number of stored requests has become 234013, where the percentage of loss is 0.706% of the data. It is worth noting that the pump running time during the data collection period is 530 hours 27 minutes, which is equivalent to 27.2% of the total time; pump running has a significant impact on the data, as will be shown later in this subsection.

**Table 2.** Descriptive statistics for practical variables

| | Main Bearing Temperature | Stator Winding Temperature |
|---|---|---|
| **Count** | 234013 | |
| **Mean** | 46.592578 | 2.530694 |
| **Std** | 6.896036 | 1.156287 |
| **Min** | 29.6 | 2.3 |
| **25%** | 41.7 | 2.4 |
| **50%** | 46 | 2.4 |
| **75%** | 51.2 | 2.4 |
| **Max** | 78.1 | 18.8 |

After cleaning the data, it has been observed that variables are stable (no changes) due to no faults occurring to the pump during the data collection period. All parameters except for two variables (changes over time), which are "main bearing temperature" and "stator winding temperature". Descriptive statistics measures were applied to the previous two variables. Therefore, it can be concluded from Table 2 that the main bearing temperature is the most appropriate variable for applying PdM principles because its readings indicate a continuous, logical change over time. What reinforces this decision is that the unit of measurement for the main bearing temperature is Celsius so that it can be easily estimated. Unlike stator winding temperature that uses the ampere unit, its movement is unpredictable as it increases and decreases rapidly. Also, the correlation coefficient between the two variables is completely non-existent (-0.000027), so the stator winding variable is not relied upon to give any predictions.

Now, presenting the data analysis where data to be used is average for every ten minutes (df_10min); the reason for this was explained in the previous subsection. Time series have Systematic and Non-Systematic components, systematic like the trend (increasing or decreasing in series) and seasonality (repeating cycle in series); non-systematic like noise or residual (random variation). In Figure 5, additive decomposition analysis is shown. In the first chart, observed data were selected for the analysis, where ten days were chosen with a minor data loss, the data selected from Nov 10 until Nov 20, 2020. In the second chart, the trend data were shown. It is noted that these data show the effect of pump operation on its temperature (without seasonality). From the above, the water consumption of the population served by the pump can be expected (weekly or monthly). Climate or precipitation affects this reading significantly. When reviewing the weather for the same period, it was found that rain fell on the 16th, which caused a significant increase. In the third chart, seasonal data are shown repeating cycles over a specific period. The readings have determined the effect of different population consumption during the day on the pump running and its temperature. It is also noted that weather does not affect this reading, as the temperature begins to rise at afternoon (13:00) and begins to decrease at midnight (00:00). The fourth and final chart shows residual data; as it can be seen, the data follow an unpredictable pattern, which causes uncertainty in the prediction.

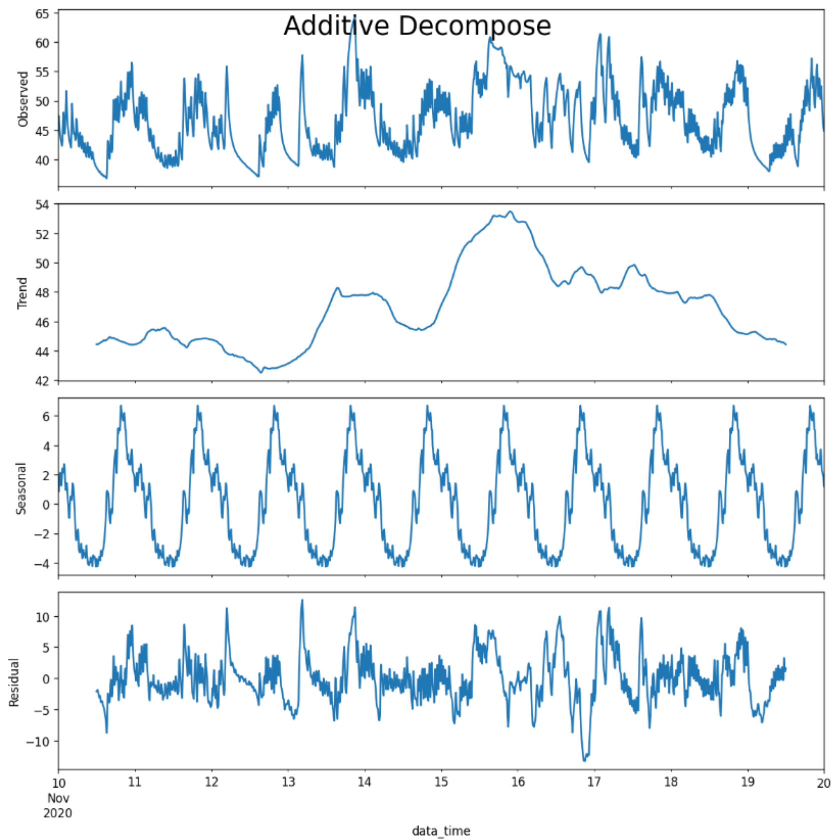


**Figure 5.** Additive decomposition

### 4.2 Forecasting Results

1) SARIMA & SARIMAX with Fourier

From SARIMA Model, the results of using Hyperparameters code show that it could not deal with seasonality in the data. The optimum results did not deal with seasonality. Bad results may be due to the massive noise of data and the appearance of more than one season.

From SARIMAX, this method is based on the Fourier series. It is an improvement over the SARIMA method, as it can handle two overlapping seasonality in the time series. Training the model with Fourier

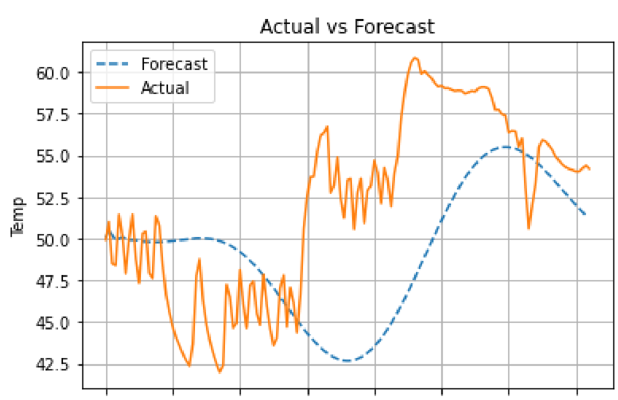


**Figure 6.** Forecasting by SARIMAX

results in the optimal parameters, and a forecast is obtained based on these parameters. The optimal parameters were as follows: non-seasonal order is (1, 1, 2), seasonal order is (1, 0, 0, 4), no trend, and optimizer is L-BFGS algorithm. The resulting parameters show that the model was able to identify seasonality in the data, which is reflected in the accuracy of the forecast, as shown in Figure 6.

2) Facebook Prophet

The Prophet's model is fast and easy to apply. It is also characterized by accurate forecasts, as the model can capture many features from time series. When looking at Figure 7, which shows a comparison of actual data with a forecast, significant improvement appears over the previous models. The model also provides another evaluation method (Cross-validation), but it will not be used because it's not existed in the rest of the models.

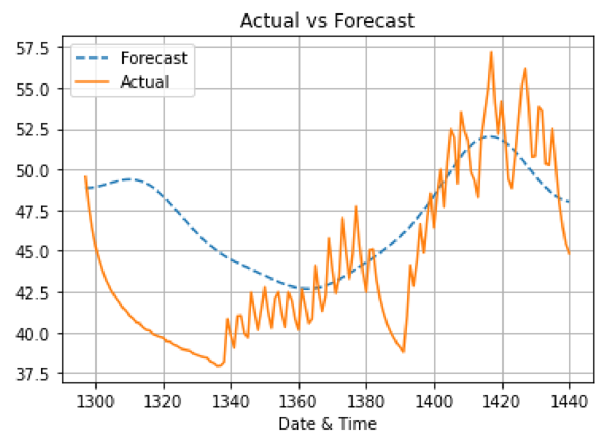


**Figure 7.** Forecasting by Prophet

3) TBATS

The results of training the model show that it has captured all the features available to the model. In Figure 8, the model deals with two seasons: the small season every 40 minutes and the large season every 24 hours. The model predicts the data trend with no increase or decrease from actual data (shifting). Performance results are better than the previous models.

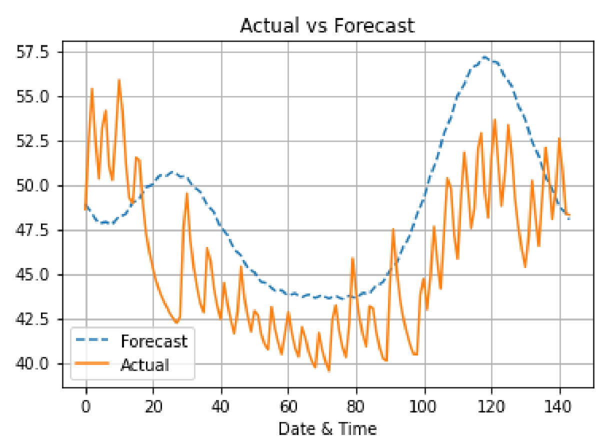


**Figure 8.** Forecasting by TBATS

4) LightGBM

This model shows learning results with few parameters, as the trained model does not show features that deal with complex time series. The results show the model dealt with one season. The performance evaluation of the model was weaker than the previous model.

5) Keras

Keras library based on TensorFlow contains many details and variables to build the model because these models (based on DL) deal with big data, extract features automatically, and make predictions. Three different models have been tested in terms of parameters and model layers, all models are based on Keras, and the results are as follows:

1) This model consists of four layers (one input, two LSTM, and one Dense); in addition, its total parameters are 30,651. The model showed effortless learning results, as the model did not capture the trend and seasonality.
2) This model consists of three layers (one input, LSTM, and Dense); in addition, its total parameters are 66,689. The model showed very high learning results (overfitting), knowing that the training data differed from the test data. In addition, it shows high capabilities to predict the details and features of the time series. However, it cannot be relied upon because it may give poorly accurate results when data periods differ (time series constantly changing), which is one of the overfitting problems.
3) This model consists of four layers (one input, two LSTM, and one Dense); in addition, its total parameters are 8,712. The model showed an ability to predict the trend and one season (24 Hrs.). It is also noted that the model is weak in predicting the short periods, which are essential for the case being worked on.

## 4.3 Anomaly Detection Results

### ADTK

This model uses many methods to achieve AD, including univariate methods. Seven methods were used for univariate. In Figure 9., the evaluated univariate methods are shown, which are: A) ThresholdAD: There is no anomaly because no values are exceeding the thresholds (high=130, low=30), this method does not help show operating anomalies because it shows pump failure after it has occurred; B) GeneralizedESDTestAD: This method did not detect any anomalies in the data, so it is not suitable for use with pump data; C) VolatilityShiftAD: This method detects volatility in the data, and this is very useful for the situation being worked on, as the model is not highly sensitive (required for fewer alerts); D) PersistAD: This method is not effective as the anomaly period is very short so that one point appears in each column, and this method is very sensitive; E) QuantileAD: It appears that the results are few and detect anomalies on the edges

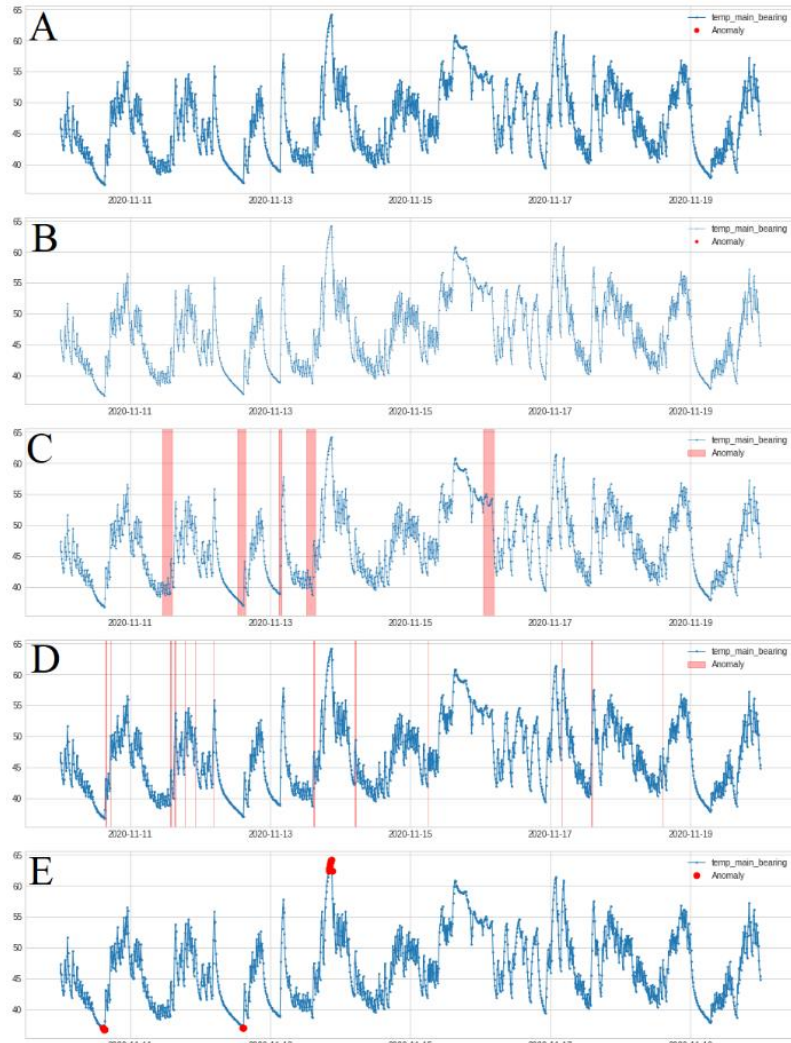


**Figure 9.** Evaluated Univariate Methods

**PyCaret**

This model uses 12 detectors (algorithms) to detect anomalies. After testing all of them, it was found that three detectors did not detect any anomalies; they are Angle-base Outlier Detection (ABOD), Subspace Outlier Detection (SOD), and Stochastic Outlier Selection (SOS). Therefore, these three detectors are not suitable for this type of data. In addition, it is noted that Connectivity-Based Local Outlier (COF) and Local Outlier Factor (LOF) detectors presented highly random results. Also, Histogram-Based Outlier (HBOS) and Principal Component Analysis (PCA) detectors presented results for only one side (up or down). The rest of the five detectors presented difficult results to review by observing and giving feedback because no problems appeared in them. This difficulty is because no problems occurred during the observation period, and the anomaly in operation is complex for the human to determine from the temperature only. Therefore, the statistical evaluation will be completely relied upon for all detectors in this model. The five detectors are as follows: 1) Minimum Covariance Determinant (MCD); 2) One-class SVM (OCSVM); 3) Isolation Forest (IForest); 4) Clustering-Based Local Outlier (CBLOF); 5) K-Nearest Neighbors (KNN).

1) Keras

Keras library can handle many applications, including anomaly detection. Also, this library is highly customized, where it changed data shape, layers characteristics, and threshold calculation method, so the calculation of AD is based on these properties. Two methods were used in this library, which are:

A. This model consists of eight layers (one input, two Conv1D, two Dropout, and three Conv1DTranspose); its total parameters are 9,505. The learning curve results (Train and Validation) showed that the model is a good fit (not underfitting or overfitting). It shows different results from previous models where unusual pump stops were detected. The pump stopped completely in the specified anomaly period, and this stop was not repeated during the test period.
B. The model consists of seven layers (one input, two LSTM, two Dropout, one RepeatVector, and one Dense); its total parameters are 198,273. The learning curve results (Train and Validation) showed that the model is fit, but model A is more suitable than this model. The method of calculating the threshold is 98% of MAE loss. Applying the threshold presents results whose quality is difficult to evaluate by the figure, so rely completely on the statistical evaluation.

### 4.4 Evaluation Results

Human evaluation of models based on observation is individual. Therefore, in this research, the primary use is statistical evaluation. In statistical evaluation, what matters is the results accuracy, then the time and space (RAM) for forecasting (fitting) because it occurs every ten minutes. Moreover, the time and space for training occur every 24 hours, so it is less important than other indicators in evaluation. In all tables below, the best result is (**bold**), and the second-best result is (*italic*).

1) Forecasting Models

From the forecasting evaluation results in Table 3 and

Table **4**, the TBATS model is the best, as it is the lowest value in the primary metric (RMSE) and the rest of the indicators except for MAE. The TBATS model also showed the best forecast results regarding speed and space (RAM) consumption. A model gives a long training time (4.15 minutes) compared to the rest of the models, but this time is less important because the model is trained once every 24 hours. The results also show that the Prophet model presents good forecast results but is slower and consumes more space from TBATS. It is also noted that the Keras No. 3 model gave lower forecast results than the rest of the models, but it was not chosen because it is on a different scale. A percentage metric (sMAPE) was used when evaluating this model; it presents the worst result (largest value) is 1.222. After reviewing the statistical evaluation results and comparing the forecast figures, it was found that the TBATS model is the best to use in the PdM system for this data.

**Table 3.** Forecasting Model Performance Metrics Results

| Model | MSE | RMSE | MAE | MAPE | sMAPE | Notes |
|---|---|---|---|---|---|---|
| **SARIMA** | 74.295 | 8.619 | 7.596 | 0.183 | 0.163 | |
| **SARIMAX with Fourier** | 41.044 | 6.407 | 5.083 | 0.096 | 0.101 | |
| **Prophet** | *21.764* | *4.665* | **3.713** | *0.088* | *0.083* | |
| **TBATS** | **20.627** | **4.542** | *3.924* | **0.086** | **0.082** | **The Best** |
| **LightGBM** | 61.513 | 7.843 | 6.915 | 0.165 | 0.15 | |
| **Keras No. 1** | 125.677 | 11.211 | 10.451 | 0.246 | 0.214 | |
| **Keras No. 2** | 51.745 | 7.193 | 5.834 | 0.126 | 0.126 | |
| **Keras No. 3** | 0.642 | 0.801 | 0.635 | -2.792 | 1.222 | Different scale |

**Table 4.** Forecasting Code Performance Metrics Results

| Model | Training Time (Milli-seconds) | Training Memory (MB) | Forecasting Time (Milliseconds) | Forecasting Memory (MB) | Notes |
|---|---|---|---|---|---|
| **SARIMA** | **102.29** | 2.501 | 38.334 | 1.353 | |
| **SARIMAX with Fourier** | 57732.101 | 242.63 | *22.051* | 0.449 | |
| **Prophet** | *2986.053* | 83.654 | 5063.3 | 46.493 | |

| | | | | | |
|---|---|---|---|---|---|
| **TBATS** | 249554.844 | **0.589** | **1.56** | **0.012** | **The Best** |
| **LightGBM** | 30418.8 | *2.333* | 34.523 | *0.104* | |
| **Keras No. 1** | 21176.803 | 5.643 | 8418.2 | 4.85 | |
| **Keras No. 2** | 16538.879 | 7.339 | 584.38 | 1.394 | |
| **Keras No. 3** | 108509.79 | 9.281 | 626.62 | 1.053 | |

2) Anomaly Detection Univariate Models

When evaluating AD models, two metrics were used, and the primary metric was not determined. Table 5 shows that the PersistAD method is best on Excess Mass metric and the HBOS method is best on Mass Volume metric. The QuantileAD method is the best on both metrics when determining the second-best result. It also shows that the QuantileAD method is the best on the two metrics of time and space (RAM) consumption, see

Table **6**. The most reliable method that can be used in the PdM system is QuantileAD, as it is on performance metrics very close to the best result, and it is also the least resource-intensive, as shown in

Table **6**.

**Table 5.** AD Univariate Model Performance Metrics Results

| Model | Methods | Excess Mass | Mass Volume | Notes |
|---|---|---|---|---|
| **ADTK** | VolatilityShiftAD | 0.05352977 | 18.0351 | |
| | PersistAD | **0.00010075** | 27.4189 | |
| | QuantileAD | *0.00010738* | *27.5049* | **The Best** |
| **PyCaret** | COF | 0.00518841 | 25.893 | |
| | Iforest | 0.00503137 | 26.9622 | |
| | HBOS | 0.00370902 | **27.5099** | |
| | KNN | 0.00503308 | 27.1503 | |
| | LOF | 0.00504811 | 26.8385 | |
| | OCSVM | 0.00504367 | 26.5093 | |
| | PCA | 0.00506277 | 26.287 | |
| **Keras** | A | 0.00754171 | 2.4743 | |
| | B | 0.005174 | 2.9709 | |

**Table 6.** AD Univariate Code Performance Metrics Results

| Model | Methods | Fitting Time (Milli-seconds) | Fitting Memory (MB) | Notes |
|---|---|---|---|---|
| **ADTK** | VolatilityShiftAD | *29.454* | 0.174 | |
| | PersistAD | 38.536 | 0.22 | |
| | QuantileAD | **18.751** | **0.063** | **The Best** |
| **PyCaret** | COF | 1378.6 | 32.178 | |
| | Iforest | 2039.013 | 1.385 | |
| | HBOS | 426.392 | *0.169* | |
| | KNN | 528.2 | 0.539 | |
| | LOF | 548.665 | 1.578 | |
| | OCSVM | 568.364 | 0.172 | |

| | | | | |
|---|---|---|---|---|
| | PCA | 411.43 | 0.177 | |
| **Keras** | A | 841.578 | 3.783 | |
| | B | 24898.186 | 9.934 | |

## 5 CONCLUSION AND FUTURE WORK

In this paper, a real-time PdM system was developed and applied to a centrifugal pump on a sewage pumping plant in Gaza City. The system is a device designed to receive, validate, store, and send data from the pump to an interactive dashboard; where the data is processed, displayed, and predict the next failure time by AI models run. Eight forecast models and thirteen AD methods were experimented with, and the result was that TBATS is the best forecast model and QuantileAD is the best AD method.

The authors suggest future works to develop a system by considering the following recommendations:

- Using multivariate increases the model's ability to predict equipment conditions.
- Classify the detected anomaly types (Vibration, Bearing Wear, Imbalance, etc.) and the degree of their impact (Extreme, High, Middle, Low, No).
- Determines parameters that can be applied during long prediction periods, such as efficiency or wear.
- Cyber security is considered when implementing the system at each data transmission and storage stage.
- Developing methods (metrics) for evaluating unsupervised AD models significantly improves choosing the best model.

## 6 ACKNOWLEDGEMENTS

The authors would like to express their appreciation and thanks to UCASTI, which received a grant from UNDP under the Swiss Agency for Development and Cooperation (SDC) for a START project, and Gaza Municipality, which allowed to work at a facility it operates.

## REFERENCES

1. Amihai, I., Gitzel, R., Kotriwala, A. M., Pareschi, D., Subbiah, S., & Sosale, G. (2018). An Industrial Case Study Using Vibration Data and Machine Learning to Predict Asset Health. *2018 IEEE 20th Conference on Business Informatics (CBI)*, *01*, 178–185. https://doi.org/10.1109/CBI.2018.00028
2. Amruthnath, N., & Gupta, T. (2018a). A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, *April*, 355–361. https://doi.org/10.1109/IEA.2018.8387124
3. Amruthnath, N., & Gupta, T. (2018b). Fault class prediction in unsupervised learning using model-based clustering approach. *2018 International Conference on Information and Computer Technologies (ICICT)*, 5–12. https://doi.org/10.1109/INFOCT.2018.8356831
4. Candanedo, I. S., Nieves, E. H., González, S. R., Martín, M. T. S., & Briones, A. G. (2018). Machine Learning Predictive Model for Industry 4.0. In *2018 First International Conference on Artificial Intelligence for Industries (AI4I)* (Vol. 1, Issue July, pp. 501–510). https://doi.org/10.1007/978-3-319-95204-8_42

5. Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. (2020). Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability*, *12*(19), 8211. https://doi.org/10.3390/su12198211
6. De Livera, A. M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing. *Journal of the American Statistical Association*, *106*(496), 1513–1527. https://doi.org/10.1198/jasa.2011.tm09771
7. Ekmekci, P. E., & Arda, B. (2020). History of Artificial Intelligence. In *SpringerBriefs in Ethics* (Issue December, pp. 1–15). https://doi.org/10.1007/978-3-030-52448-7_1
8. Fernandes, S., Antunes, M., Santiago, A. R., Barraca, J. P., Gomes, D., & Aguiar, R. L. (2020). Forecasting Appliances Failures: A Machine-Learning Approach to Predictive Maintenance. *Information*, *11*(4), 208. https://doi.org/10.3390/info11040208
9. Gao, J., Song, X., Wen, Q., Wang, P., Sun, L., & Xu, H. (2020). *RobustTAD: Robust Time Series Anomaly Detection via Decomposition and Convolutional Neural Networks*. 9. https://doi.org/https://arxiv.org/abs/2002.09545
10. *Gaza Municipality confirms its readiness for the storm*. (2015). SAFA Press Agency. https://safa.ps/p/142569
11. Goix, N. (2016). *How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms?* http://arxiv.org/abs/1607.01152
12. Guo, Y., Liao, W., Wang, Q., Yu, L., Ji, T., & Li, P. (2018). Multidimensional Time Series Anomaly Detection: A GRU-based Gaussian Mixture Variational Autoencoder Approach. *Proceedings of Machine Learning Research*, *95*(2001), 97–112. http://proceedings.mlr.press/v95/guo18a.html
13. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. *Principles of Optimal Design*, 504. https://otexts.com/fpp2/
14. Jain, A. K., Dhada, M., Hernandez, M. P., Herrera, M., & Parlikad, A. K. (2021). A comprehensive framework from real-time prognostics to maintenance decisions. *IET Collaborative Intelligent Manufacturing*, 1–9. https://doi.org/10.1049/cim2.12021
15. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 3147–3155.
16. Leary, C. (2021). *EMMV library*. https://pypi.org/project/emmv/
17. Lee, J., Kao, H., & Yang, S. (2014). Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment. *Procedia CIRP*, *16*, 3–8. https://doi.org/10.1016/j.procir.2014.02.001
18. Lee, S. C. (2010). Maintenance strategies for manufacturing systems using Markov models. *ProQuest Dissertations and Theses*, 118. http://hdl.handle.net/2027.42/77723
19. Lee, S., & Kim, H. K. (2018). ADSaS: Comprehensive Real-time Anomaly Detection System. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11402 LNCS*, 29–41. https://doi.org/10.1007/978-3-030-17982-3_3
20. Lee, W. J., Wu, H., Yun, H., Kim, H., Jun, M. B. G., & Sutherland, J. W. (2019). Predictive Maintenance of Machine Tool Systems Using Artificial Intelligence Techniques Applied to Machine Condition Data. *Procedia CIRP*, *80*, 506–511. https://doi.org/10.1016/j.procir.2018.12.019
21. Lin, C.-Y., Hsieh, Y.-M., Cheng, F.-T., Huang, H.-C., & Adnan, M. (2019). Time Series Prediction Algorithm for Intelligent Predictive Maintenance. *IEEE Robotics and Automation Letters*, *4*(3), 2807–2814. https://doi.org/10.1109/LRA.2019.2918684
22. Löfsten, H. (2000). Measuring maintenance performance – in search for a maintenance productivity index. *International Journal of Production Economics*, *63*(1), 47–58. https://doi.org/10.1016/S0925-5273(98)00245-X
23. Mobley, R. K. (2002). An Introduction to Predictive Maintenance (Second Edition). In *An Introduction to Predictive Maintenance (Second Edition)*.

24. Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., & Loncarski, J. (2018). Machine Learning approach for Predictive Maintenance in Industry 4.0. *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 1–6. https://doi.org/10.1109/MESA.2018.8449150
25. Ran, Y., Zhou, X., Lin, P., Wen, Y., & Deng, R. (2019). A Survey of Predictive Maintenance: Systems, Purposes and Approaches. *ArXiv*, *XX*(Xx), 1–36. http://arxiv.org/abs/1912.07383
26. Rodrigues, M. W., & Zarate, L. E. (2019). Time Series Analysis Using Synthetic Data for Monitoring the Temporal Behavior of Sensor Signals. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, *2019-Octob*, 453–458. https://doi.org/10.1109/SMC.2019.8913907
27. *Sewage collapse in southern Gaza*. (2016). https://www.24fm.ps/ar/news/1462297514
28. *Sewage nightmare due to electricity in Gaza*. (2012). Palestine Allan. https://paltimeps.ps/p/12754
29. Sullivan, G. P., Pugh, R., Melendez, A. P., & Hunt, W. D. (2010). Operations & Maintenance Best Practices. In *U.S. Department of Energy*.
30. Taylor, S. J., & Letham, B. (2017). Forecasting at Scale. *PeerJ Preprints*, *35*(8), 48–90. https://doi.org/https://doi.org/10.7287/peerj.preprints.3190v2
31. Tinga, T., & Loendersloot, R. (2019). Physical Model-Based Prognostics and Health Monitoring to Enable Predictive Maintenance. In *Predictive Maintenance in Dynamic Systems* (pp. 313–353). Springer International Publishing. https://doi.org/10.1007/978-3-030-05645-2_11
32. Vuksanović Herceg, I., Kuč, V., Mijušković, V. M., & Herceg, T. (2020). Challenges and Driving Forces for Industry 4.0 Implementation. *Sustainability*, *12*(10), 4208. https://doi.org/10.3390/su12104208
33. *Wastewater treatment plants are overflowing at every moment*. (2008). MA'AN Development Center. https://www.maan-ctr.org/magazine/Archive/Issue4/main_topic3/topic3.htm
34. Xu, L., Zheng, L., Li, W., Chen, Z., Song, W., Deng, Y., Chang, Y., Xiao, J., & Yuan, B. (2021). NVAE-GAN Based Approach for Unsupervised Time Series Anomaly Detection. *Proceedings of ACM Conference (Conference'17)*, *1*(1), 8. http://arxiv.org/abs/2101.02908
35. *Xylem Water Solutions & Water Technology | Xylem US*. (2021). https://www.xylem.com/en-us
36. Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*(August), 106889. https://doi.org/10.1016/j.cie.2020.106889