AGNOSTIC LEARNING FOR PACKING MACHINE STOPPAGE PREDICTION IN SMART FACTORIES

Gabriel Filios[1], Ioannis Katsidimas[1], Sotiris Nikoletseas[1], Stefanos H. Panagiotou[1], Theofanis P. Raptis[2]
[1]Department of Computer Engineering and Informatics, University of Patras, Greece and Computer Technology Institute and Press "Diophantus", Patras, Greece, [2]Institute of Informatics and Telematics, National Research Council (IIT-CNR), Pisa, Italy

NOTE: Corresponding author: Stefanos H. Panagiotou, spanagiotou@ceid.upatras.gr

Abstract – The cyber–physical convergence is opening up new business opportunities for industrial operators. The need for deep integration of the cyber and the physical worlds establishes a rich business agenda towards consolidating new system and network engineering approaches. This revolution would not be possible without the rich and heterogeneous sources of data, as well as the ability of their intelligent exploitation, mainly due to the fact that data will serve as a fundamental resource to promote Industry 4.0. One of the most fruitful research and practice areas emerging from this data-rich, cyber-physical, smart factory environment is the data-driven process monitoring field, which applies machine learning methodologies to enable predictive maintenance applications. In this paper, we examine popular time series forecasting techniques as well as supervised machine learning algorithms in the applied context of Industry 4.0, by transforming and preprocessing the historical industrial dataset of a packing machine's operational state recordings (real data coming from the production line of a manufacturing plant from the food and beverage domain). In our methodology, we use only a single signal concerning the machine's operational status to make our predictions, without considering other operational variables or fault and warning signals, hence its characterization as "agnostic". In this respect, the results demonstrate that the adopted methods achieve a quite promising performance on three targeted use cases.

Keywords – Industry 4.0, machine learning, production line stoppages, prognostics, smart factory

1. INTRODUCTION

Cyber-physical systems are an inevitable outcome of the fourth industrial revolution (also coined as Industry 4.0). Embedded computing, Internet communication, and ubiquitous control have now become fundamental components of modern engineered products and their manufacturing processes [1]. The cyber–physical convergence is opening up new business opportunities for industrial operators. The vision of a virtual world, that is overlaid on the physical world to continuously monitor it and take intelligent actions to adapt the cyber world to industrial needs, is part of an emerging trend of in pervasive and mo- bile computing [2]. The need for deep integration of those interrelated worlds opens up a rich business agenda to- wards establishing a new system and network engineering that is both physical and virtual [3]. Consequently, industrial cyber-physical systems employment is expected to revolutionize the way enterprises conduct their business from a holistic viewpoint, i.e., from shop-floor to business interactions, from suppliers to consumers, and from design to testing across the entire product and service lifecycle [4].

This revolution would not be possible without the rich and heterogeneous sources of data, as well as the ability of their intelligent exploitation; this is mainly due to the fact that data will serve as a fundamental resource to promote Industry 4.0 from machine automation to information extraction and then to knowledge discovery [5]. Smart factories already operate using sophisticated sensors, actuators and communication technologies. Internet of Things (IoT) devices are not seen any more as "dumb things" generating individually a few bytes, but as industrial devices, generating data of variable size and signifi- cant importance, still operated via batteries to make them more flexible and cheap to assemble, install and manage [6]. Smart factories perform adaptive responses by continuously monitoring and extracting information from physical objects (e.g., machines, work pieces, robotic elements, etc.) and production processes [7]. In this setting, large amounts of data are generated and collected, requiring advanced big data processing methodologies to build an integrated environment in which the production processes can be mirrored transparently and administrated in a more efficient way [8]. One of the most fruitful re- search and practice areas emerging from this data-rich, cyber-physical, smart factory environment is the data-driven process monitoring field, which applies multivariate statistical methods to enable prognostics, diagnostics and fault detection for industrial process operations and production results [9]. By applying big data analytics, it is possible to find interpretive results for strategic decision- making, providing novel insights which lead to significant production improvements, such as, maintenance cost decrease, early fault detection, machine stoppage prediction, spare parts inventory reduction, increased production, improvement in operator safety and repair verification [10].

©International Telecommunication Union, 2022
Some rights reserved. This work is available under the CC BY-NC-ND 3.0 IGO license: https://creativecommons.org/licenses/by-nc-nd/3.0/igo/.
More information regarding the license and suggested citation, additional permissions and disclaimers is available at:
https://www.itu.int/en/journal/j-fet/Pages/default.aspx

Novel developments in specialized fields of information and communication technologies and availability of easy-to-use, often freely available, software tools and off-the-shelf hardware components, offer great potential to transform the smart factory domain and their impact on the smart factory data pools effectively. One of the most trending developments is in the area of Machine Learning (ML). The utilization of ML is motivated by its enhanced capabilities to spare resources, machining time and energy, and its improved operational capacity where traditional methods have reached their limits [11]. However, the field of ML is highly diverse and many different algorithms, theories, and methods are available. For many industrial operators, this represents a barrier regarding the adoption of powerful ML tools and thus may block the us- age of the huge amounts of data which are more and more becoming available [12]. Additionally, industrial technology providers have chosen to deploy standalone systems which act as black boxes from which, in most cases, vital data is not possible to be acquired. Furthermore, it is also worth noting that, in many cases, the lack of qualified infrastructure, due to cost-related reasons such as the high commission fees of third party providers, is also a common problem.

Based on the aforementioned assumptions, it is evident that there are large and complex production lines which suffer from the absence of the necessary, sophisticated prediction infrastructure and, therefore, from the inability to receive the corresponding data and maintain a rich data pool. In this paper, we argue that even those can become able to establish a reliable and efficient prediction ability, in order to activate the required Predictive Maintenance (PdM) mechanisms. Specifically, we aim to predict a packing machine's stoppages on the following three targeted use cases:

1. Forecasting of the total daily duration of stoppages in the near future

2. Prediction of whether the packing machine will be stopped for more than 10 minutes in the next hour (later mentioned as "Minor Stoppage Duration Exceeding" case)

3. Prediction of whether a specific type of stoppage (breakdown event) will occur in the next hour (later mentioned as "Breakdown Occurrence" case)

In order to apply our proposed methodologies for these cases, we are taking advantage of univariate time series data coming from a production line machine of a large manufacturing plant from the food and beverage domain[1]. We aim at using the entire behavior of the machine as a single piece of data, or, in other words, we investigate whether a sequence of interruptions of a specific duration can potentially lead to the production line's stoppage in the near future.

Specifically, we first transform the collected raw dataset with several essential and required data preprocessing procedures and then we leverage this transformed time series data with two core approaches. The first one, consists of univariate time series forecasting algorithms (Prophet, ARIMA, HWAMS, TBAT, N-BEATS) that are employed for forecasting the first aforementioned use case. We also produce forecasting ensembles of these models to examine if we can achieve higher accuracy using multiple combinations of individual models. The second approach leverages machine learning algorithms to model our time series data, reframing the latter into a feature- based dataset. We utilize tree-based machine learning regression algorithms (decision tree, XGBoost regressor, extra-trees regressor, AdaBoost, gradient boosting regressor), both individually and as ensembles, in the same use case we apply the time series forecasting algorithms and compare their performance and results. Lastly, we investigate the next two aforementioned use cases as bi- nary classification problems, using the random forest algorithm to make our predictions.

To make this methodology clear, we offer a high-level visual representation of our overall workflow in Fig. 1.

2. RELATED WORK

Recently, ML services in the context of Industry 4.0 and PdM are trending in the ICT and manufacturing fields. More and more applied research is conducted to come up with reliable and insightful predictive analytics workflows and solutions which can be adopted by industrial manufacturing companies. Here, we present some indicative research work which can be considered as the most relevant to our research. In order to narrow down the presentation, we focus on approaches which use intelligent techniques and take advantage of real data coming from actual industrial plants.

The authors of [13], present a data-driven approach for estimating the probability of machine breakdown during specified time interval in the future, in the context of discrete parts manufacturing systems. Specifically, they utilize data such as historical log messages, event logs and operational information from milling machines in order to estimate the Remaining Useful Life (RUL) by using the random forest machine learning algorithm. Results demonstrate that machine failures can be reliably predicted up to 168 hours in advance and that feature selection algorithms highly impact log-based Predictive Maintenance (PdM).

Research work [14] studies a predictive maintenance application in the white goods/home appliances sector. The use case concerns a single production line that produces drums for dryers and the objective is to predict

[1] A non-disclosure agreement prevents us from providing more information regarding the company, the plant, the equipment and the related data.

 © International Telecommunication Union, 2022

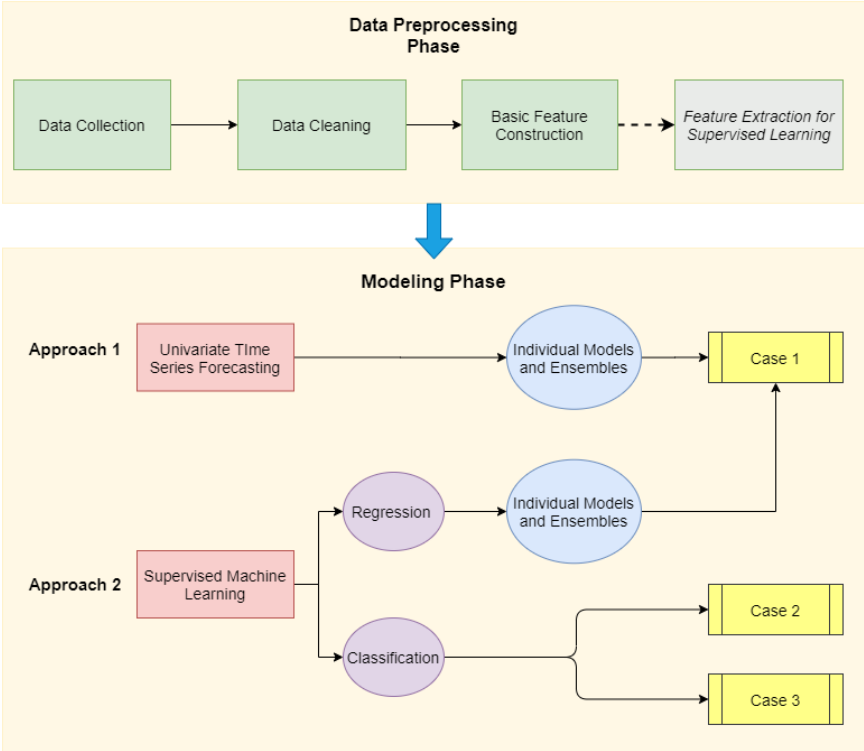


Fig. 1 – Methodology workflow

if the factory is going to have a breakdown on the next day, using legacy and operational data. Several machine learning algorithms are employed for modeling this binary classification problem such as Support Vector Machine (SVM), the multinomial naïve Bayes classifier, k-NN, decision tree, random forest, and the Multilayer Perceptron (MLP), from which decision tree reaches the highest accuracy score. Another interesting finding is the usage of an additional custom metadata feature in the modeling, calculated in the original dataset, that helped to increase the accuracy from 77.7% to the final and best score of 95%.

In [15], the authors develop a Bayesian network for fault assessment of an electrical motor. Their proposed model is able to calculate through inference the probability of rotor fault of an induction motor and define the weakest branch in the structure of the Bayesian network that leads to failure by determining the probabilities of intermediate events. Their methodology is validated in a motor of a real industrial site, using its history file and the results of conducted diagnostics to determine the a priori probabilities of several potential fault causes and thus enabling the Bayesian network to define the most likely faults.

The authors of [16] examine a predictive maintenance use case, proposing a method for early detection of faults in boiler feed pumps. In particular, they utilize existing event data and measurements acquired from SCADA systems of a coal fired power plant. The aim is to produce a model that can detect deviations from the normal operation state based on regression and to check which events or failures can be detected by it. Their proposed algorithm consists of a bag of regression models and the experimental results indicate that it outperforms decision trees and Multilayer Perceptron (MLP) classification algorithms.

In research work [17], a data-driven soft sensing methodology is presented for monitoring the quality of an industrial pasteurization process. The methodology is based on machine learning and benchmarks various algorithms such as decision tree, ridge regression, extra trees for estimating the temperature of products during the pasteurization process. The work studies a real beer pasteurization process in collaboration with Heineken's plant in Patras, Greece and the results demonstrate notable performance in temperature prediction accuracy, with an average Root Mean Square Error (RMSE) of 1.85°C in the test sets.

The authors of [18] investigate the machinery involved in the production of high-quality steel sheets by predicting the degradation of the drums within the involved heating coilers. They incorporate valuable sources of information like expert knowledge, configuration parameters and real-time information coming from sensors mounted on the machines, in order to develop a predictive model based on a Discrete Bayesian Filter (DBF) that estimates and predicts the gradual degradation of such machinery. The DBF model reaches high success, compared with knowledge-based and classic machine learning models and appears to be robust against noisy and fluctuating

© International Telecommunication Union, 2022

data and integrate well expert knowledge with production data.

In [19], authors propose a data-driven algorithm to predict throughput bottlenecks in a real-world production system from an automotive production line. They employ an Auto-Regressive Integrated Moving Average (ARIMA) method to predict the active run periods of the machine and integrate it with a data-driven active period technique to form a binary classification bottleneck prediction problem. Authors of [20] present a methodology that utilizes process sensor data from operation periods to forecast possible equipment stoppages (or faults) of a specified industrial equipment, which is used in the anode production process for the aluminum industry. The classifiers employed were the decision tree, random forest, the Gaussian/Bernoulli naive Bayes and multilayer perceptron including the logistic regression. The visualization of the features patterns and the simulation results show that a warning timeframe of around 5-10 minutes before the incident occurs is a feasible goal.

Contribution and motivation of our work Different to the aforementioned research work, and as an extended version of our previous work in [21], this paper includes the following novelties:

- Investigation of a packing machine's stoppages in a production line of a manufacturing plant from the food and beverage domain.
- Complex utilization of a single production scale machine signal in order to create a dataset that will allow the modeling of the machine's stoppage behavior. The machine is located at the final stages of the packing manufacturing process and there are multiple machinery, human and material reasons and issues that appear earlier in the process and cause it to stop. However, we are not able to monitor these and we only monitor the signal describing its operational state, hence the characterization of our methodology as "agnostic", which contradicts approaches that apply ML techniques and require large pools of data coming from different sources.
- We not only attempt to investigate interesting, to industrial operators and the general literature, use cases but we also aim at testing the potential self-forecastability of this univariate time series, by predicting the machine's behavior based on the past measurements of its operational state.
- Extraction of custom metadata features from the original signal data in order to enable the employment of machine learning classification and regression algorithms.
- Comparison of traditional forecasting and machine learning regression models, as well as their generated ensembles, in the applied context of Industry 4.0.
- The generated ensembles are proven to produce more accurate forecasts than simple, naive models.

The use cases investigated in this work are arguably very interesting, as the packing machine is considered to be the bottleneck of the production line. In this way, a potential successful predictive modeling can give shop-floor operators the opportunity to take proactive actions and scheduling of the production line, resulting in great cost reduction and higher plant efficiency and productivity.

3. DATA ANALYSIS AND PREPROCESSING

In this section we describe how the raw data is collected, analysed and transformed in order to create a meaningful data representation for our modeling approaches. All of our proposed steps below are implemented using Python and Pandas, the powerful data analysis library [22].

3.1 Data collection and outline

As we have already mentioned, our factory application is related to the food and beverages sector. An IoT controller is installed to receive digital signals regarding the packing machine running state (*RUN* and *STOP* states) of the production line. In particular, the controller is a Raspberry Pi 3 Model B that is connected to the output of a Siemens SIMATIC S7-300 Programmable Logic Controller (PLC) and receives and stores the data locally for further analysis. Data acquisition is implemented in a clear and transparent way without interfering in any way to the process, which may cause profit loss. More signals and information require great effort, as data is gathered in different data storage modules, which in many cases consider "black box" systems, the topology does not allow an easy wired data transfer, and current infrastructure does not offer many options on data transmission. In addition, the presence of many metal surfaces, motors and inverters cause a lot of noise and prevents the use of wireless communications, which in practice appear to have low reliability. Thus, working with the current dataset was our only option.

The dataset consists of an irregular event-based time series, containing the stoppages and running states of an industrial packing machine which alternate at arbitrary times and as a result the spacing of the observation times is not constant. It is collected in a total period of one year, from October 2018 to October 2019 and contains 73576 entries from which 36775 are stoppages. Each time the machine's operation state changes either from *RUN* to *STOP* and vice versa, it is logged as an event with three columns:

- The timestamp in which the state changed.
- The binary value of the machine's state (1 for *RUN* and 0 for *STOP*).

 © International Telecommunication Union, 2022

- The duration of the previous state in seconds by calculating their timestamp difference.

Table 1 reflects a short snapshot of the raw data.

Table 1 – Example of the collected raw data

Timestamp	State	Duration
2019-05-30 14:44:09.298	1	2.141
2019-05-30 14:44:12.866	0	3.567
2019-05-30 14:44:24.994	1	12.128
2019-05-30 14:44:30.127	0	5.133
2019-05-30 14:45:15.596	1	45.469

It is also worth mentioning that industrial operators further discriminate the stoppages based on their duration, resulting in three stoppage categories:

- *Minor stoppages*, those that last between 10 seconds and 5 minutes.
- *Breakdown stoppages*, those that last between 5 and 40 minutes.
- *Major stoppages*, those that last more than 40 minutes.

In table 2 and 3, the packing machine's daily operational RUN time statistics in hour scale and count of stoppages per category are presented respectively for the time period of October 2018 to October 2019.

Table 2 – Daily RUN time statistics in hour scale

Metric	Value
min	2.1
max	20
Daily Mean	14
Daily Median	14.5
Daily St.Deviation	3.5

Table 3 – Machine's stoppages count statistics

Count	Value
Daily Mean (all categories)	180
Minor	32441
Breakdown	2545
Major	289

3.2 Dataset cleaning

The first step of the data preparation process involves the application of some core preprocessing steps in order to clean up the data from outliers and noisy, inconsistent information that can decrease the predictive performance of the models.
First of all, a small number of "duplicates" (consecutive rows with the same value, either 1 or 0) is removed from the dataset as it is important to have strictly alternating states. Afterwards, we need to assign the duration of each event in its own row. So, based on the aforementioned raw format, a shifting operation is applied in the duration column in order to set for each row the duration of it's next row. A third step is to tidy up the data based on the normal operation days and hours of the machine (and the whole production line in general). Any events that might have happened outside these (e.g after Saturday evenings, before the first shift of Mondays) are considered as special events (e.g for testing or maintenance purposes) and they are removed.

3.3 Feature engineering

This is the most important step prior to applying our proposed modeling approaches. Section 3.3.1 describes the core methodology to create a meaningful dataset. This dataset is used to model univariate time series forecast- ing problems, as presented in Section 4.1. In Section 3.3.2 we build upon this dataset and extract additional features to be used in our machine learning regression and classi- fication approaches.

3.3.1 Basic data preparation

It is evident that the described dataset in its raw format is not capable of providing useful information as input to a prediction model. Therefore, following the cleaning and preprocessing steps it is essential to add more representative information about the machine's functional patterns and we accomplish this by extracting features from the same data we have.
One common operation is to "resample" (group) the data in different time intervals each time (e.g by 5, 10, 15, 30 minutes, hourly or daily resampling) and calculate aggregation and statistical features such as the sum, mean, standard deviation of the durations or the count of the stoppage and running events in each interval.
It is worth mentioning that this resampling procedure requires additional preparation of the aforementioned dataset in order to calculate accurately the above features. As an example, assuming the resampling by 5 minutes, an interval 09:05-09:10 and a stoppage occurring at 09:09 until 09:15 (6 minutes duration), the event must be split into two events, one occurring at 09:09 with a duration of 1 minute and one occurring at 09:10 with a duration of 5 minutes. In this way, the initial information of the duration is not altered but it is just distributed correctly between the intervals 09:05-09:10 and 09:10-09:15 for the feature calculations. In Fig. 2, a visual representation of this procedure is presented, while Table 4 depicts a dataset example concerning the total duration (in seconds) and count of stoppages per hour.

3.3.2 Feature extraction

Supervised machine learning algorithms like the ones we use in our experiments depend on meaningful features, in order to correlate and map the input data with the output

© International Telecommunication Union, 2022

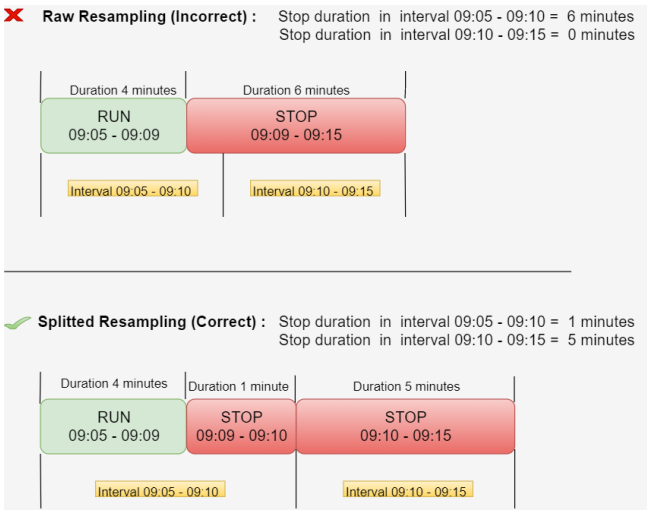


Fig. 2 – *Split resampling representation*

Table 4 – Example of an hourly resampled dataset of stoppages

Timestamp	Sum	Count
2019-05-30 14:00:00	1176.647	14
2019-05-30 15:00:00	1133.857	23
2019-05-30 16:00:00	44.975	2
2019-05-30 17:00:00	818.693	7
2019-05-30 18:00:00	298.273	5

class of interest. The relevancy of features with the class to be predicted is a factor of high importance for the successful modeling and prediction performance of the algorithms. In our case, we generate a rich feature set by adding various features based on the process described in Section 3.3.1, and the knowledge of the industrial operators, reaching into a final number of 115 features.
The core feature list consists of the total duration (*sum*) and the number (*count*) of the occurred stoppages and run events in a given interval. Specifically, concerning the stoppages, we calculate sum and count features for each aforementioned stoppage category. Some performance indicators can be also synthesized based on *Minor stoppages* as they are the most frequent type of stoppages and arguably the most important for analysing the machine's behavior. For example, *Mean Time Between Stops (MTBS)* and *Mean Time Between Runs (MTBR)*, give an intuitive reflection on how frequently do stoppages and run events respectively occur, during the specified interval, and are calculated as follows :

$$MTBS = \frac{sum(run)}{count(stops)} \quad (1)$$

$$MTBR = \frac{sum(stop)}{count(runs)} \quad (2)$$

Furthermore, based on the feature list above *(sum and count of [minor_stop, breakdown_stop, major_stop], sum and count of runs, MTBS, MTBR)*, we extract two distinct feature categories which are calculated for each feature of the former. The first one, consists of the so-called lag features, which are variables containing values from prior time intervals and particularly, our lags range from 1 to 5 prior time steps. For example, assuming an hourly input window interval and a specific row in the dataset corresponding to the data for the input interval between 09:00-10:00, *count_stop* represents the count of stops of that hour and *count_stop_t_1*, *count_stop_t_2* the count of stops occurred in the intervals 08:00-09:00 and 07:00-08:00, respectively. The second category is about rolling statistics features, where we calculate the moving arithmetic mean and standard deviation of the core features with their lag values.

At last, we also create the *time_since_[major_stop, breakdown_stop]* which is the time elapsed (in minutes) from the last occurrence of a *[major_stop, breakdown_stop]* until each input time interval. Besides this, we decompose the timestamp column of Table 4 into single date time features like *day_of_week*, *day_of_month*, *week*, *hour* for each input time interval. The purpose is to provide the models with an understandable representation of the time nature of the data so they can potentially catch and model any underlying time related patterns such as trends and seasonalities.

4. PROPOSED MODELING APPROACHES

Our proposed modeling approaches are distinguished between two core categories. The first one, is univariate time series forecasting, which takes into account only one time dependent (numerical) variable and by modeling only its historical values, it produces forecasts for future time points. The second one, is the supervised machine learning approach, which differentiates from the former by using features in order to model and map the input data to an output, the prediction target of interest.

4.1 Univariate time series forecasting

Univariate time series data is a collection of observations with chronological order. In particular, it consists of the independent variable, time, and just one dependent variable, which varies over time. Forecasting this kind of data means that past measurements are analysed by the algorithms to model the underlying pattern and behavior of the series and make predictions for future points. Our use case concerns the forecasting of the daily total sum of minor stoppages, i.e, how much time will the investigated machine be in stop state per day in the future, due to minor stoppages. In particular, the dataset consists of the timestamp column, which ranges from January 2019 to August 2019 and the *sum_minor_stop* feature column. Daily resampling of the observations is chosen as it is an important indicator for industrial operators when monitoring the machine's efficiency. It is not by coincidence that we examine this use case, as minor stoppages are the most frequent type of stoppages and are considered to be the bottleneck of the machine's operation. Concerning the selected date period, in these dates the machine is more busy than any other period during the year, and

© International Telecommunication Union, 2022

thus, succeeding in generating insightful forecasts is vital to industrial operators. Model wise, we use the AtsPy time series library [23] for the implementation of the models described below. After producing forecasts for each individual model, we create their ensembles as described in Section 4.3.

- *ARIMA*, stands for Autoregressive Integrated Moving Average model, which is expressed as ARIMA(p,d,q). Parameters p, d, and q are integer values that decide the structure of the time series model; parameter p, q is the order of the Auto-Regressive (AR) model and the Moving Average (MA) model respectively, while parameter d is the level of difference applied to the data, if non stationarity is detected [24, 25].

- *Prophet*, is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It is robust to missing data and shifts in the trend and typically handles outliers well [26].

- *TBAT*, is a method that accounts for multiple seasonalities, using a combination of Fourier terms, exponential smoothing models, and Box-Cox transformations in a completely automated manner. TBATS also allows for the seasonal patterns to dynamically adjust over time [27]. In the ensembles generation, TBATS is a simple variation of TBAT, crafted in the AtsPy library for using seasonal transformation and TBAT without seasonal.

- *N-BEATS*, is a deep neural architecture based on backward and forward residual links and a very deep stack of fully-connected layers incorporated for solving univariate times series point forecasting problems [28].

- *HWAMS*, implements Holt Winter's exponential smoothing with additive trend and multiplicative seasonality [29].

4.2 Supervised machine learning

In this part we attempt to reframe the univariate time series forecasting approach as a supervised machine learning problem. The motivation for this feature-based time series modeling is to explore new valuable, for industrial operators, prediction use cases, such as the prediction of a stoppage occurrence in the near future. Moreover, we can potentially overcome some limitations of the data, that univariate time series forecasting methods usually cannot handle efficiently. For example, such limitations include the lack of massive historical data from a long time period to capture seasonality, and the fact that the past measurements of just the prediction target cannot be enough to model it's behavior, as it might be also dependent to other exogenous factors.

Considering the above, we transform the univariate time series data into a feature-based dataset, that machine learning algorithms can take advantage of. To achieve this, the dataset must undertake a considerable transformation as there is no concept of input and output features in time series. A supervised learning framing of a time series means that the data needs to be split into multiple samples that the models can learn from, and generalize across. Each sample (row) must have both an input set of features (columns) and an output component, i.e, the prediction target, also known as the class of the problem. We extract and generate the input set of features as presented in Section 3.3.2. Regarding the prediction target, we describe our use case decisions accordingly in Sections 4.2.1 and 4.2.2. The model implementation and evaluation are made using Python and the Scikit-Learn library [30].

4.2.1 Regression approach

A machine learning regression approach considers the modeling of input data to predict numerical (integer or continuous) values. In this part, we leverage machine learning regression algorithms, both individually and as ensembles (see Section 4.3), for the same use case and experiment details we applied the univariate time series forecasting models (time period ranging from January 2019 to August 2019 and taking into account only information from minor stoppages). The dataset is constructed in a way so that the input features (see Section 3.3.2) correspond to the previous day of the prediction target. In other words, input features of each sample (row) are mapped with the next day's value of *sum_minor_stop* target. Model-wise, we utilize the following tree-based regression algorithms.

- *Decision Tree*, a simple regression tree implementation.

- *Extra Trees*, implement a meta-estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various subsamples of the dataset and uses averaging to improve the predictive accuracy and control overfitting [31].

- *Gradient Boosting Regressors (GBR)* is a machine learning technique for regression problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. At each step, a new tree is trained against the negative gradient of the loss function, which is analogous to (or identical to, in the case of least-squares error) the residual error [32].

- *eXtreme Gradient Boosting (known as XGBoost and denoted with XGB in the ensembles)*, is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. In particular, it is about a scalable and accurate implementation of

gradient boosting machines and it has proven to push the limits of computing power for boosted trees algorithms as it was built and developed for the sole purpose of model performance and computational speed [33].

- AdaBoost, an AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases [34, 35].

4.2.2 Classification approach

In our machine learning classification approach, we investigate two distinct binary classification tasks.

1. Prediction of whether the packing machine will be stopped for more than 10 minutes in the next hour ("Minor Stoppage Duration Exceeding" case)

2. Prediction of whether a specific type of stoppage (breakdown event) will occur in the next hour ("Breakdown Occurrence" case)

Both cases constitute a very important and interesting challenge. A potential accurate predictive modeling could warn industrial operators and in this way give them enough time to take preventive and corrective maintenance actions in the machine's production line or the investigated machine itself. As a result, this could lead to avoiding stoppages from occurring, and thus, to significant cost reductions for the unit. Regarding the dataset, in both cases, we calculate the aforementioned features of Section 3.3.2, for an hourly input window and resampling interval. This means that we gather and aggregate the data per hour and we map each hour interval (row) to the prediction target (class) for the next hour.
We use the random forest classifier as the modeling algorithm for both cases. A random forest is a meta-estimator that fits a number of decision tree classifiers on various subsamples of the dataset and uses averaging to improve the predictive accuracy [36]. We chose this classifier as it is considered a highly accurate and robust method because of the number of decision trees participating in the process. It does not suffer from the overfitting problem, canceling out the biases by generating an internal unbiased estimate of the generalization error as the forest building progresses.

4.3 Average ensembles generation

Forecasting accuracy can potentially be improved by combining forecasts, produced by different algorithms [37, 38]. Thus, we attempt such an ensemble approach as well, generating two distinct ensemble categories from the aforementioned univariate forecasting and regression models, respectively. In particular, multiple combinations of the individual models compose an ensemble by *averaging* their predictions for each data point in the test data.
We produce, in a brute force manner, all possible combinations of ensembles of different lengths (subsets) between the models, generating iteratively ensembles consisting of

$$i = 2...k \;\; individual \;\; models \tag{3}$$

where i, is the number of individual models in each ensemble and k, is the total number of individual models in each model category (forecasting and regression).
Taking forecasting category as an example, which consists of five (5) individual models, we generate all possible ensembles that consist of 2, 3, 4 and 5 individual models respectively. So, for $i = 2$ we get ensembles produced by two (2) individual models like *ARIMA_Prophet*, *ARIMA_TBAT*, *ARIMA_N-BEATS* ... *Prophet_TBAT* etc. Accordingly, for $i = 3$, we get ensembles produced by three (3) individual models like *ARIMA_Prophet_TBAT*, *ARIMA_Prophet_N-BEATS* ... etc. Of course, only unique ensemble combinations are created, i.e, duplicate ensembles like *Prophet_ARIMA* are not generated.

5. EVALUATION AND RESULTS

In this section we first present the metrics used to evaluate our proposed approaches and afterwards we evaluate and compare the time series forecasting methods with the machine learning regression algorithms for the same prediction problem. In the last section, the results of the two classification use cases are evaluated and discussed.

5.1 Metrics

The following metrics were used to evaluate the results of the univariate time series forecasting and machine learning regression models :

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|e_t| \tag{4}$$

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{e_t}{y_t}\right| \tag{5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}e_t^2} \tag{6}$$

$$MASE = \frac{MAE}{MAE_{in-sample,naive}} \tag{7}$$

where e_t is the absolute error, n is the sample size and y_t is the actual value

The first three are quite common in the literature, while the last one is an interesting metric that gives us an alternative baseline for determining the quality of our forecasts.

 © International Telecommunication Union, 2022

- *Mean Absolute Percentage Error (MAPE)* expresses the average of the absolute percentage errors. While it is easily interpretable and one of the most popular metrics, it has also certain disadvantages [39, 40] that lead us to utilizing other metrics as well for a more reliable and complete evaluation of the models.
- *Mean Absolute Error (MAE)* is a data scale-dependent metric which indicates how big of an error we can expect from the forecast on average.
- *Root Mean Squared Error (RMSE)* is the standard deviation of the residuals (prediction errors). It is a data scale-dependent metric sensitive to outliers, putting a heavier weight on larger errors.
- *Mean Absolute Scaled Error (MASE)* is a measure of forecast accuracy which compares the model's forecast against a naive benchmark method calculated in-sample. This measure is data scale-independent, useful in cases where there are different scales in the data or values which are negative or close to zero. In addition, it is easily interpretable: when $MASE < 1$, it implies that the forecasts of the proposed method perform, on average, better out-of-sample than the in-sample one-step forecasts of the naive method [41].

  As a benchmark reference, a simple naive model sets all forecasts to be the value of the last observation. That is, at the time t, the k-step-ahead naive forecast F_{t+k} is "predicted" with the observed value at time $t(y_t)$:

$$F_{t+k} = y_t \tag{8}$$

  The naive static average model sets the forecasts to the average value using the expanding window method and the moving median and average models set the forecasts with the median and average values calculated with the rolling window method.

Concerning our classification approach, the evaluation metrics we use are the following:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$\text{F-Measure} = 2 * \frac{\text{Precision * Recall}}{\text{Precision+Recall}} \tag{12}$$

where TP, TN are the true positives and negatives and FP , FN the false positives and negatives predicted outcomes of the model

- *Accuracy* is intuitively the overall fraction of predictions our model got right. However, it is not enough to evaluate our models using only this metric, especially for class-imbalanced datasets like ours. In this way, we examine the following metrics to get a more clear view of the model performance.
- *Precision* represents the proportion of positive identifications that were actually correct.
- *Recall* represents the proportion of the actual positives that were identified correctly.
- *F-measure* is a measure of a test's accuracy. It provides a way to combine both precision and recall into a single measure that captures both properties.

5.2 Comparison between forecasting and regression models

In this part, the comparative performance of the regression algorithms and the univariate forecasting models is discussed. Our purpose is to examine which model category and which specific models perform better, what is the performance of the ensembles and what is the over- all predictive ability for our use case of interest. We em- ploy the aforementioned model categories with their respective individual models and ensembles, to make the prediction for our use case. We perform the training of the models with 75% of the data, while the evaluation is conducted with the remaining 25%, of both sets of the original dataset. Due to time series indexing, this means that the most recent data is used for the actual forecasts. To evaluate the overall performance in this use case we choose to present the following top five models produced per modeling category.

Forecasting ensembles :

1. *ARIMA, TBAT, NBEATS, TBATS*
2. *ARIMA, TBAT, NBEATS*
3. *TBAT, NBEATS, TBATS*
4. *Prophet, TBAT, NBEATS, TBATS*
5. *ARIMA, TBAT, TBATS*

Regression ensembles :

1. *AdaBoost, XGB, Decision Tree*
2. *GBR, AdaBoost, XGB, Decision Tree*
3. *GBR, AdaBoost, Decision Tree*
4. *XGB, Extra Tree, Decision Tree*
5. *XGB, Decision Tree*

© International Telecommunication Union, 2022

Table 5 – Results of the best five forecasting models

				MASE			
Model	MAPE	MAE	RMSE	moving_mean	static_mean	moving_median	naive
1	19,765 %	32,536	42,399	0,926	0,882	0,920	0,664
2	19,833 %	33,026	43,255	0,940	0,895	0,934	0,674
3	20,007 %	33,808	44,282	0,962	0,916	0,956	0,690
4	20,135 %	33,165	43,435	0,944	0,899	0,938	0,677
5	20,156 %	33,489	42,568	0,953	0,908	0,947	0,683

Table 6 – Results of the best five regression models

				MASE			
Model	MAPE	MAE	RMSE	moving_mean	static_mean	moving_median	naive
1	20,753 %	31,085	37,988	0,890	0,842	0,866	0,632
2	20,984 %	31,727	38,368	0,909	0,860	0,884	0,645
3	21,017 %	31,466	37,879	0,901	0,853	0,877	0,639
4	21,080 %	31,458	38,553	0,901	0,852	0,877	0,639
5	21,252 %	32,118	39,735	0,920	0,870	0,895	0,653

In Tables 5 and 6 the detailed results of these best five models are presented per modeling category. We note that results for the MAE and RSME metrics are expressed in minute scale. Considering both tables, we first point out the fact that ensembles make the top of the leader board as their resulting combined averaged forecasts are more accurate than those of the individual models and we also notice a comparable performance from both modeling categories. In particular, forecasting methods did slightly better in terms of relative errors (MAPE) but regression models were slightly better regarding the mean absolute and root mean squared errors. So, taking into account all these metrics, with each one having it's specific importance, we can conclude that in this use case the two modeling categories were equally competitive. In addition, we should also highlight the results of the MASE metric, i.e, the benchmark comparison against the four naive methods described in Section 5.1. We can tell that our proposed models produced on average more accurate forecasts than naive models. This is interesting, as it means that we can employ a meaningful model, in terms of implementation worthiness, when compared to a naive forecasting solution.

Examining the winning ensembles more in depth, we observe on the one side that the TBAT algorithm with its variation, TBATS, dominates in the forecasting ensembles. We also notice the model type variation in those winning ensembles, as we do not get ensembles of closely related models, but we witness the combinations of different algorithms, especially those which include the N- BEATS deep neural architecture. On the other side, we observe that decision tree and secondly XGBoost, are core contributor models in each ensemble.

In Fig. 3 we demonstrate a visualized comparison of the actual test data and the forecasts produced by the best model from the forecasting and regression category. The data set comprises 34 daily data points ranging from 12 July to 30 August, as depicted in the x-axis and the total sum of minor_stoppages expressed in seconds. We notice the steady, seasonal alike prediction pattern of the forecasting ensemble which is close to the mean of the data. This is kind of expected, due to the underlying modeling process of the individual models. In contrast, the regression model differentiates by producing higher and more custom forecasts, in some cases it even results interestingly in really high accuracy, like in the dates 16, 27 July and 14 August. Closing up, we acknowledge the decent performance of the models in regards to our agnostic modeling. In particular, the reframing of the time series data as a supervised regression problem can be regarded successful, considering that the regression models achieved equal performance with the forecasting models, which are tailored for specific univariate time series problems. Research-wise, we could extend the experiments in the future with the collection of additional data, in order to have a more holistic view of the trends, seasonality and other patterns and thus, achieve better forecasting accuracy. We could also try working on different subsets of the data to examine the performance of the models in relation to the dataset size. Lastly, it would be interesting to further experiment with ensembles between models that have been trained in different subsets of the data as well as ensembles between the forecasting and regression models.

5.3 Classification use cases

In this part, we present the results of the random forest classifier for the two aforementioned prediction use cases. For simplifying our findings evaluation, we can regard for each use case accordingly, the positive class as an alarm generation towards the industrial operators and the negative class as an ignore state, assuming that the predictions were incorporated in a real-time predictive maintenance system. For both of them we perform the

 © International Telecommunication Union, 2022

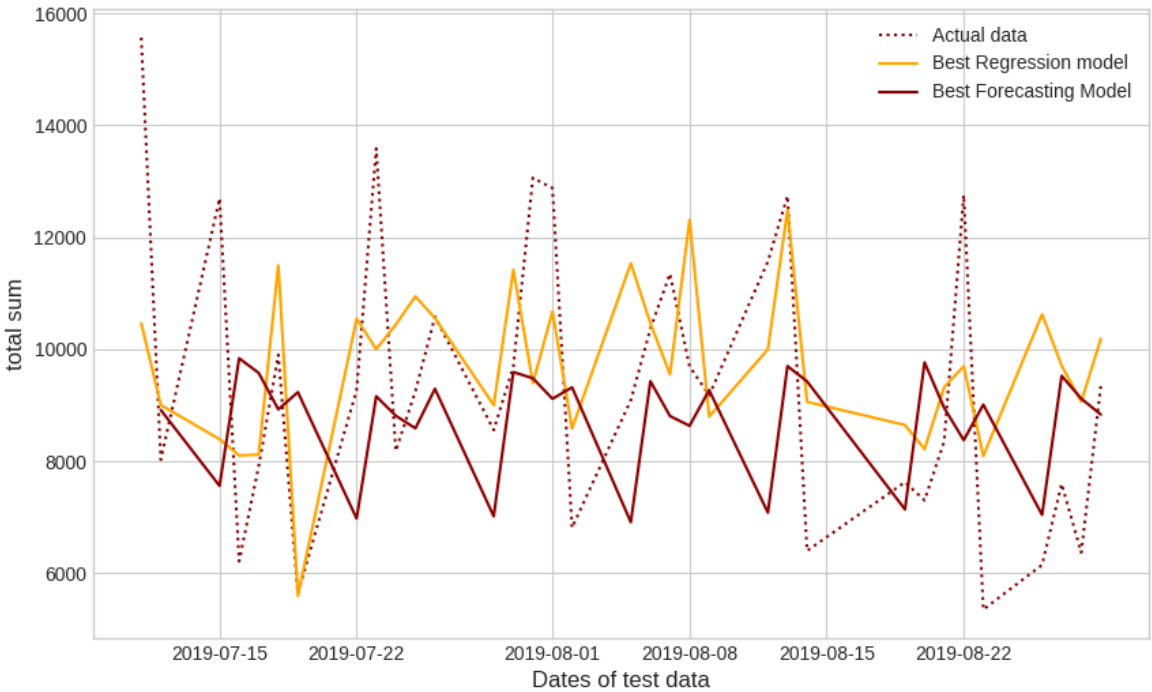


Fig. 3 – Actual data and model performance visualization

training and evaluation of the models in the same way as in our regression approach, applying a 75% / 25% percentage split between the training and test sets. This is also approximately the class distribution for both cases, 75% and 25 % for the negative and positive class respectively. For each use case we present the generated confusion matrix[2] (tables 7 and 9) which consists of the predicted true and false positives and true and false negatives, based on which the classifications metrics are derived in tables 8 and 10.

5.3.1 *Minor stoppage duration exceeding*

In the dataset we treat the classes as follows: positive (1) is the class describing that the total sum of minor stoppages will exceed the duration of 10 minutes in the next hour and negative (0) is the class about the opposite. The dataset focuses on the period of April 2019 till July 2019, which is the operational peak of the production line and the packing machine. Considering tables 7 and 8, we observe that the classifier detects only 11 % of the actual alarms it should detect, which is in fact low. However, we notice an optimistic result, the almost absolute identification of the ignore class, with 99% recall for ignore class and 78% precision for the alarm class, meaning that the classifier at least would not signal many false alarm warnings. This is indeed a valuable predictive ability; otherwise, producing frequent false alarms may lead to the industrial operator's distraction. Or even worse, inaccurate predictions can make them take wrong preventive and corrective decisions for the production line and the investigated machine.

Table 7 – Confusion matrix of *Minor Stoppage Duration Exceeding* case

		Predicted	
		ignore	alarm
Actual	ignore	369	4
	alarm	113	14

Table 8 – Performance of *Minor Stoppage Duration Exceeding* case

Accuracy	Class	Precision	Recall	F-measure
77 %	ignore	77 %	99 %	86 %
	alarm	78 %	11 %	19 %

5.3.2 *Breakdown occurrence*

This use case is modeled between January 2019 and September 2019, the period containing the most observations in our dataset. In this way, our aim is to gather in the input data as many breakdown events and relevant

[2] Also known as an error matrix, it is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class.

© International Telecommunication Union, 2022

information around their occurrence as possible, in order to increase the chances of a successful prediction. We treat as positive (1) class, the event of a breakdown occurence in the next hour and as negative (0) class, the exact opposite event. Reviewing the results in tables 9 and 10, we notice that our classifier performed better in this case than the previous one, in the prediction of the actual occurrences of the main class of interest (in this case, "alarm" for a possible breakdown). The improvement was by a factor of 10 %, reaching a recall of 21% (compared to 11% of the previous case). However, this was caused, by the reduced precision, i.e, the reduced ability of the model to distinguish clearly between the actual and false alarms and so the alarm class has only, on average, a 51% chance to be predicted correctly (compared to 78% of the previous case). This is a usual phenomenon in machine learning, also known as precision and recall trade-off, in which improving the former metric might worsen the latter and vice versa. The predictive power for the ignore (negative) class is slightly worse than the first use case, as precision dropped from 77% to 75%, but it is still a relatively good sign that the model can at least perform well for one of the metrics.

Table 9 – Confusion matrix of *Breakdown Occurrence* case

		Predicted	
		ignore	alarm
Actual	ignore	804	69
	alarm	274	72

Table 10 – Performance of *Breakdown Occurrence* case

Accuracy	Class	Precision	Recall	F-measure
72 %	ignore	75 %	92 %	82 %
	alarm	51 %	21 %	30 %

In Fig. 4, the performance of random forest is provided to visualize the differences in each metric, per use case and prediction class. The figure at the top depicts the performance for the ignore (negative) class, while the bottom one is for the alarm (positive) class.

Given these results for both use cases and looking further into the F-measure values, we can sum up that our feature-based time series classification approach achieved an overall moderate, weak performance in the alarm (positive) class. On the contrary, it reveals interesting results for the ignore (negative) class, which seems to be predicted in a quite accurate way. The tests indicate that we miss essential information in order to achieve high accuracy in the prediction of alarm cases and this is not to our surprise, as we do not monitor the different factors that may cause a stoppage and in general disturbances in the production line. This means that future research activities will include the collection of those exogenous factors and the further modeling experimentation of the aforementioned use cases. Nonetheless, our

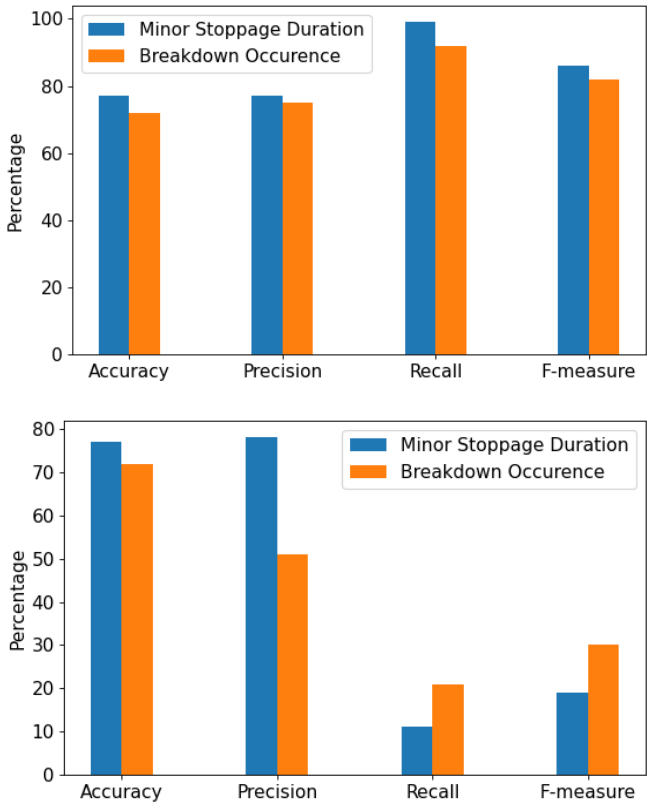


Fig. 4 – Performance of random forest in both classification use cases, per class. The figure at the top depicts the performance for the ignore (negative) class, while the bottom one is for the alarm (positive) class.

agnostic methodology so far can be still regarded as decent work, considering the results produced while having only one signal at our disposal. In this way, the regression and classification approaches we developed can be utilized as baseline work for these future activities.

6. CONCLUSION

In this work, we investigated the prediction of industrial packing machine stoppages by applying the approaches of traditional univariate time series forecasting and supervised machine learning. Our highest forecasting and classification accuracy of 80% and 77% respectively does not imply a perfect, state-of-the-art performance but we can still regard these results as promising considering our agnostic methodology, indicating that it is feasible, at least to a certain extent, to model the machine's behavior based only on its raw past operational state measurements. We acknowledge that the presence of noise and random fluctuations in the data as well as the lack of strong seasonalities, clear structural patterns, and most importantly the absence of the stoppage factors weaken the forecasting and predictive ability of the models. Therefore, we note that better results can be potentially achieved with the use of more production line data. Such data could be derived from additional sensors and operational variables, fault or warning signals from the same machine as well as from other machines which operate in the same line.

© International Telecommunication Union, 2022

This is due to the fact that such data affects the general flow of the entire packaging line and consequently the operation of the packing machine. Hence, it is clear that prospects for future research include the collection of such data and use it in our modeling. In addition to the aforementioned update and enrichment of our data pool, our proposed methodologies can act as a reference for future enhancements with more suitable predictive models and techniques (such as deep learning, feature selection methods, classification ensembles, and class imbalance handling) that can eventually lead to a more robust and accurate prediction framework.

ACKNOWLEDGEMENT

This research was supported by (i) the "Andreas Mentzelopoulos Foundation", (ii) the CNR Short Term Mobility (STM) program 2019, and (iii) the European Union's Horizon Europe research and innovation programme RE4DY (European Data as a PRoduct Value Ecosystems for Resilient Factory 4.0 Product and ProDuction ContinuitY and Sustainability) under grant agreement No 101058384.

REFERENCES

[1] A. Barnard Feeney, S. Frechette, and V. Srinivasan, *Cyber-Physical Systems Engineering for Manufacturing*. Cham: Springer International Publishing, 2017, pp. 81–110. [Online]. Available: https://doi.org/10.1007/978-3-319-42559-7_4

[2] M. Conti, S. K. Das, C. Bisdikian, M. Kumar, L. M. Ni, A. Passarella, G. Roussos, G. Troester, G. Tsudik, and F. Zambonelli, "Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber-physical convergence," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 2 – 21, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119211001271

[3] J. Sztipanovits, "Cyber physical systems - convergence of physical and information sciences," *it - Information Technology*, vol. 54, no. 6, pp. 257 – 265, 2012. [Online]. Available: https://www.degruyter.com/view/journals/itit/54/6/article-p257.xml

[4] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin, "Industrial cyberphysical systems: A backbone of the fourth industrial revolution," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 6–16, 2017.

[5] T. P. Raptis, A. Passarella, and M. Conti, "Data management in industry 4.0: State of the art and open challenges," *IEEE Access*, vol. 7, pp. 97 052–97 093, 2019.

[6] ——, "Distributed data access in industrial edge networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 915–927, 2020.

[7] E. Molina, O. Lazaro, M. Sepulcre, J. Gozalvez, A. Passarella, T. P. Raptis, A. Ude, B. Nemec, M. Rooker, F. Kirstein, and E. Mooij, "The autoware framework and requirements for the cognitive digital automation," in *Collaboration in a Data-Rich World*, L. M. Camarinha-Matos, H. Afsarmanesh, and R. Fornasiero, Eds. Cham: Springer International Publishing, 2017, pp. 107–117.

[8] J. Yan, Y. Meng, L. Lu, and L. Li, "Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance," *IEEE Access*, vol. 5, pp. 23 484–23 491, 2017.

[9] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual Reviews in Control*, vol. 36, no. 2, pp. 220 – 234, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578812000399

[10] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcala, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360835219304838

[11] D. Weichert, P. Link, A. Stoll, S. Rüping, S. Ihlenfeldt, and S. Wrobel, "A review of machine learning for the optimization of production processes," *The International Journal of Advanced Manufacturing Technology*, vol. 104, no. 5, pp. 1889–1902, 2019.

[12] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016. [Online]. Available: https://doi.org/10.1080/21693277.2016.1192517

[13] C. Gutschi, N. Furian, J. Suschnigg, D. Neubacher, and S. Voessner, "Log-based predictive maintenance in discrete parts manufacturing," *Procedia CIRP*, vol. 79, pp. 528 – 533, 2019, 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S221282711930215X

[14] M. Pertselakis, F. Lampathaki, and P. Petrali, "Predictive maintenance in a digital factory shop-floor: Data mining on historical and operational data coming from manufacturers' information systems," in *Advanced Information Systems Engineering Workshops*, H. A. Proper and J. Stirna, Eds. Cham:

Springer International Publishing, 2019, pp. 120–131.

[15] A. Lakehal, A. Ramdane, and F. Tachi, "Probabilistic reasoning for improving the predictive maintenance of vital electrical machine: Case study," *Journal of Advanced Engineering and Computation*, vol. 2, no. 1, 2018.

[16] M. Moleda, A. Momot, and D. Mrozek, "Predictive maintenance of boiler feed water pumps using scada data," *Sensors*, vol. 20, no. 2, p. 571, Jan 2020. [Online]. Available: http://dx.doi.org/10.3390/s20020571

[17] G. Filios, A. Kyriakopoulos, S. Livanios, F. Manolopoulos, S. Nikoletseas, S. H. Panagiotou, and P. Spirakis, "Data-driven soft sensing towards quality monitoring of industrial pasteurization processes," in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2022, pp. 167–174.

[18] J.-R. Ruiz-Sarmiento, J. Monroy, F.-A. Moreno, C. Galindo, J.-M. Bonelo, and J. Gonzalez-Jimenez, "A predictive model for the maintenance of industrial machinery in the context of industry 4.0," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103289, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197619302489

[19] M. Subramaniyan, A. Skoogh, H. Salomonsson, P. Bangalore, and J. Bokrantz, "A data-driven algorithm to predict throughput bottlenecks in a production system based on active periods of the machines," *Computers & Industrial Engineering*, vol. 125, pp. 533 – 544, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360835218301608

[20] N. Kolokas, T. Vafeiadis, D. Ioannidis, and D. Tzovaras, "Forecasting faults of industrial equipment using machine learning classifiers," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*, 2018, pp. 1–6.

[21] G. Filios, I. Katsidimas, S. Nikoletseas, S. Panagiotou, and T. P. Raptis, "An agnostic data-driven approach to predict stoppages of industrial packing machine in near future," in *16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2020, pp. 236–243.

[22] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56 – 61.

[23] D. Snow, "AtsPy: Automated time series models in python (1.15)." 2020. [Online]. Available: https://github.com/firmai/atspy/

[24] S. Ho and M. Xie, "The use of arima models for reliability forecasting and analysis," *Computers & Industrial Engineering*, vol. 35, no. 1, pp. 213 – 216, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360835298000667

[25] P. Newbold, "Arima model building and the time series analysis approach to forecasting," *Journal of Forecasting*, vol. 2, no. 1, pp. 23–35, 1983. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3980020104

[26] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[27] A. M. D. Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, 2011.

[28] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," 2019.

[29] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International Journal of Forecasting*, vol. 20, no. 1, pp. 5 – 10, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207003001134

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, p. 3–42, Apr. 2006. [Online]. Available: https://doi.org/10.1007/s10994-006-6226-1

[32] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002, nonlinear Methods and Data Mining. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167947301000652

[33] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[34] H. Drucker, "Improving regressors using boosting techniques," 1997.

 © International Telecommunication Union, 2022

[35] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002200009791504X

[36] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, p. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[37] A. Graefe, J. S. Armstrong, R. J. Jones, and A. G. Cuzán, "Combining forecasts: An application to elections," *International Journal of Forecasting*, vol. 30, no. 1, pp. 43 – 54, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207013000423

[38] J. Armstrong, "Combining forecasts: The end of the beginning or the beginning of the end?" *International Journal of Forecasting*, vol. 5, no. 4, pp. 585 – 588, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0169207089900137

[39] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, vol. 9, no. 4, pp. 527 – 529, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0169207093900793

[40] J. S. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," *International Journal of Forecasting*, vol. 8, no. 1, pp. 69–80, June 1992. [Online]. Available: https://ideas.repec.org/a/eee/intfor/v8y1992i1p69-80.html

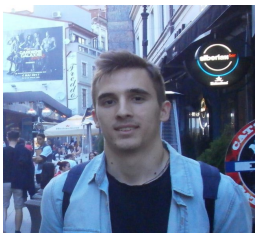
[41] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679 – 688, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207006000239

AUTHORS

Gabriel Filios is a postdoctoral researcher at the Computer Engineering and Informatics department in the University of Patras, Greece. His research interests include wireless sensor networks, energy efficiency in smart buildings and industries, feature extraction for human activity, crowd-sensing systems and privacy in IoT. He has coauthored several publications in international refereed conferences (IEEE ICC, ACM MobiWac, IEEE WF-IoT) and has participated in several EU projects (HOBNET, IoT Lab, PrivacyFlag, SAFESTRIP).

Ioannis Katsidimas is a postdoctoral researcher at the Computer Engineering and Informatics Department of Patras University, Greece. His research interests include wireless power transfer algorithms in ad hoc communication networks, wireless sensor networks, and Internet of Things. He has participated in several EU projects (IoT Lab, PrivacyFlag, SAFESTRIP).

Sotiris Nikoletseas is a professor at the Computer Engineering and Informatics Department of Patras University, Greece. His research interests include algorithmic aspects of wireless sensor networks and the Internet of Things (IoT), wireless energy transfer protocols, probabilistic techniques and random graphs, average case analysis and probabilistic algorithms and algorithmic engineering. He has coauthored over 300 publications in international journals and refereed conferences, three books (on the probabilistic method, on theoretical aspects of sensor networks, and on wireless power) and 30 invited chapters in books by major publishers.



Stefanos H. Panagiotou is a PhD candidate at the Computer Engineering and Informatics Department of Patras University, Greece. His research is being held under scholarship funding from University of Patras and his interests include machine learning applications in production lines, TinyML and AI-driven digital twins for structural health monitoring.

Theofanis P. Raptis received a Ph.D. degree from the University of Patras, Greece. He is currently a research scientist with the National Research Council, Italy. He has published in journals, conference proceedings, and books, more than 70 articles on industrial networks, wirelessly powered networks, Internet of Things testbeds, and platforms. He is also regularly involved in international IEEE and ACM sponsored conference and workshop organization committees, in the areas of networks, computing, and communications. He has been serving as an associate editor for the IEEE Access and IET Networks journals.

© International Telecommunication Union, 2022