A Deep Learning Approach for Short Term Prediction of Industrial Plant Working Status

Pierfrancesco Bellini, Daniele Cenni, Luciano Alessandro Ipsaro Palesi, Paolo Nesi, Gianni Pantaleo
Distributed Systems and Internet Tech lab, Department of Information Engineering, University of Florence
DISIT Lab, https://www.disit.org , https://www.snap4industry.org : email <name.surname>@unifi.it
Corresponding Author: Paolo Nesi, paolo.nesi@unifi.it

***Abstract* —** Predictive Maintenance has gained more and more research and commercial interests, being a pivotal topic for improving the efficiency of many production industrial plants to minimize downtimes, as well as to reduce operational costs for interventions. Solutions reviewed in literature are increasingly based on machine learning and deep learning methods for prediction of fault proneness with respect to normal working conditions. Many state-of-the art solutions are not actually applied in real scenarios, and have restrictions to be executed in real-time in the production environment. *In this paper*, a framework for predictive maintenance is presented. It has been built upon a deep learning model based on Long-Short Term Memory Neural Networks, LSTM and Convolutional LSTM. The proposed model provides a one-hour prediction of the plant status and indications on the areas in which the intervention should be performed by using explainable LSTM technique. The solution has been validated against real data of ALTAIR chemical plant, demonstrating an high accuracy with the capability of being executed in real-time in a production operative scenario. The paper also introduced business intelligence tools on maintenance data and the architectural infrastructure for the integration of predictive maintenance approach.

Keywords—Predictive Maintenance, Industry 4.0, Deep Learning, Convolutional Neural Networks, CNN, Long-Short Term Memory Networks, LSTM.

I. INTRODUCTION

In real world Industry 4.0 scenarios, it is necessary to maximize the efficiency and productivity of plants, in order to improve competitiveness in the market. To this end, a crucial role is played by the production plant maintenance. In addition to efficiency and productivity, good maintenance reduces operative costs, improves the product quality, and rationalizes resources. Typical kinds of maintenance policies are Corrective Maintenance (CM) and Preventive Maintenance (PM). The CM [Blanchard et al., 1995] or run-to-failure is quite expensive, it consists of the intervention after a failure in the production cycle that in most cases leads to the production plant stop. The PM is defined as maintenance carried out according to predetermined technical criteria [Gentles, 2020]. PM can reduce the number of failures/stops and can also be cyclical (time-based maintenance, TBM) and predictive (condition-based maintenance, CBM). In TBM the decisional process is determined on the basis of failure time analyses [Yam et al., 2001], [Jardine et al., 2006]. In complex production plants, different kinds of maintenance strategies may be adopted at the same time for different parts and production lines. For PM, solutions and techniques proposed in research literature can be classified in three approaches, based on: physical, data-driven and hybrid [Liao and Kottig, 2016].

In this paper, an integrated solution for predictive maintenance in chemical plant is presented. Most of the chemical plants are critical infrastructures which present a production process never stopping and running 24H/7D per week. The case taken into account presents a production process including chemical products which have to be carefully treated for their potential impact on the environment in case of accident. This implies that early warning and an efficient corrective maintenance are mandatory policies to be established to become operative. The aspects addressed in this paper are: (i) the usage of deep learning techniques for predictive maintenance, specifically Long-Short Term Memory Neural Networks, LSTM and Convolutional LSTM, with some technique for explaining the prediction which can be used to help the maintenance teams; (ii) the integration of workflow management system for maintenance with general control systems and data flow (also developing Node-Red library for integrating data flow and workflow ticketing system); and (iii) a business intelligence tool for maintenance. The solution has been developed exploiting the IoT Industry 4.0 development environment and framework called Snap4Industry, which in turn is based on Snap4City which is 100% open source (and licence free) and it is available at [Https://www.snap4city.org], [Badii et al., 2020a], [Badii et al., 2020b]. The new capabilities have been exploited to implement the higher-level control in the large chemical plant of ALTAIR.

This paper is structured as follows: in Section II, a review of related work in the context of Predictive Maintenance is reported. In Section III, the general architecture of the solution is presented, where the action to put in place a predictive maintenance aspects to work in real time are evident. Section III.B describes the Business Intelligence for the analysis of the maintenance data. In Section IV, an early version of the Predictive Maintenance Model based on LSTM is described with its assessment. Section V presents an advanced Predictive Maintenance Model based on CNN-LSTM and its validation results. All the validations have been performed by taking into account data of ALTAIR chemical plant. In section V.C, an approach for explain the results in real time and thus for exploiting the maintenance predictions for the identification of the area in which to operate has been reported. Finally, Section VI reports conclusions.

Reference	Techniques	Data	Type of predictions	Real time	Results
[Mathew et al., 2017]	SVM	The method is tested on a simplified simulated time-series data set	Estim. the remaining useful life (RUL) of systems and/or equipment by time series	Simulated	Traditional SVR obtained a RMSE=0.732
[Kanawaday and Sane, 2017]	ARIMA	Data from Slitting machine (Tension, Pressure, Width e Diameter.)	Predicts parameter values for the production cycle.	Yes	accuracy 94.46% for CART to 98,69% for DNN
[Shao et al., 2019]	Multi-Channel LSTM-CNN	Tennessee Eastman (TE) chemical process simulation	Five-sample time series to predict the next sample	Yes	Average F1-score of 92.03%
[Zhang et al., 2018]	LSTM	NASA's C-MAPSS dataset	Track the system degradation and to predict the remaining useful life	No	RMSE =18.07
[Sánchez et al., 2016]	Deep Boltzman machine	Two typical rotating machinery systems	Diagnosing the health of rotating mechanical systems by classification	Yes	Accuracy = 95.17%.
[Zhang and Zhao, 2017]	Deep belief Network (DBN)	Tennessee Eastman (TE) chemical process simulation dataset	Extracts the feature from a process data period and classifies the fault status	No	Average fault diagnosis rate of 82.1%.
[Tao et al, 2015]	Autoencoder - softmax regression	Bearing dataset of Case Western Reserve University	Bearing failures are diagnosed by receiving as input their characteristics	No	Accuracy > 90%

Table 1. Comparative overview of Related Work implementations

II. Related Work

The data-driven techniques used for the prediction of industry plant failures can be used as early warning, and firing activities of maintenance, as in CM. This approach aims to avoid/reduce the occurrence of major disasters that may happen on chemical plants. A typical approach consists of the following major steps: data acquisition, feature extraction, feature engineering, model training, finalizing a predictive model, and then perform predictive model validation/testing [Zhao et al., 2017]. In critical infrastructures, one has to set up any kind of planned maintenance, and resilience guidelines also impose to be ready for the unexpected unknowns [Bellini et al., 2019]. Data-driven based methods can be divided into two groups, based on two main approaches: traditional machine learning techniques (e.g., logistic regression, eXtreme Gradient Boosting (XGBoost) and Random Forest (RF) for supervised classification) and deep learning techniques.

In [Binding et al., 2019], **logistic regression, XGBoost** and **RF** techniques have been used on environment variables and machine temperature variables to predict 30-minute machine failure in real time. From the evaluation of the Receiver Operator Characteristic (ROC) curves, all three algorithms perform significantly better than a random classifier. In [Uhlmann et al., 2018], **k-means clustering** technique was used to illustrate normal machine operation and three failure conditions, exploiting temperature and pressure sensor data. [Mathew et al., 2017] demonstrated how Support-Vector Machine (**SVM)** techniques obtained a Root Mean Square Error (RMSE) of 0.732 on Gas turbine engine time series sensors. In [Kanawaday and Sane, 2017], a model based on Auto Regressive Integrated Moving Average, (ARIMA), has been proposed to predict production values, which are feed to different supervised models (Classification and Regression Trees, SVM, Naïve Bayes and Deep Neural Networks).

Many solutions exploiting deep learning models relied on flatten layer architectures, not considering temporal information [Zhou et al., 2020]. However, temporal information is important since machines typically degrade over time [Lei et al., 2018]. In [Zhang et al., 2018], **Long Short-Term Memory (LSTM)** neural network has been proposed to detect the system degradation and to predict the remaining useful life. In this context, another issue is represented by the fact that raw sensor data may contain relevant amount of noise, which can badly affect performances of LSTM models. Therefore, Convolutional Neural Networks (CNN) have been combined to LSTM to support the extraction of local features, in addition to temporal information [Zhao et al., 2017]. The main task of convolutional layers is to learn features from data input; actually, CNN were successfully applied for image classification. However, recently CNN have gained an increasing research attention also in the field of time series data analysis and classification. CNN-LSTM models have been successfully applied to time series analysis in different contexts, such as stock market forecasting [Kim and Kim, 2019], photovoltaic power predictions [Tovar et al., 2020], outperforming traditional LSTM models for their roboustness to noise. In [Shao et al., 2019], a model for faults diagnosis in chemical process based on Multi-channel **CNN-LSTM** architecture is proposed. The model achieved an F1-score of 92.03% only on a simulated scenario, and not in a real test case. In [Huuhtanen and Jung, 2018], CNN are used to monitor the operation of photovoltaic panels aiming at predicting malfunctions on the panels. The approach outperformed compared approaches based on simple interpolation filters. In [Zhang and Zhao, 2017], a model exploiting a **Deep Belief Network** (DBN) has been presented for fault detection and diagnosis in a chemical process. The DBN model extracted the features from process data collected over a given time period, to classify the fault status, achieving an average fault diagnosis rate of 82.1%. In [Sánchez et al., 2016], a method based on Deep Boltzman machine has been used for learning features and fault classification from vibration measurements of a rotating machinery. The fault classification performances assessed in experiments using this method report a classification rate of 95.17%. Accuracy greater than 90% have been also obtained with the use of **Autoencoder** and **softmax regression** in [Tao et al., 2015], where bearing failures are diagnosed by receiving as input their characteristics.

The state of the art does not solve some typical problems in industrial plants. One problem is related to the amount of data describing failures. Most companies collect a huge amount of data related to the normal operating conditions of the plant. Moreover, the use of deep learning techniques creates problems of interpretability of the results.
In Table I, a comparative overview of the above mentioned Related Work is presented.

III. General Architecture

Industry 4.0 approaches expect to have a high level of digitization in the production plant. As a first step, a number of DCS (Distributed control systems) or SCADA (Supervisory Control And Data Acquisition) are devoted to control the production pipelines and machines. On top of them a higher level of control and supervision may be needed. The latter may be also used for telecontrol. All the industry subsystems are typically connected on local area networks, and a number of IoT devices may help to glue and monitor the in/out flows,

connections and areas among the machines, and production plants. On the basis of the foreseen planned production, the acquisitions of raw materials have to be scheduled for minimizing the stock. Thus the transportation of resulting products has to be carefully planned. In H24/7day active plant, any reason to stop production also impacts on delivering, ordering, transportation, etc. In addition, in chemical plants the production lines are in most cases connected each other (see Figure 1). Thus, subproducts (liquid, gasses, matter) of a production line, may be a primary matter or source of energy for the functioning of another line/transformation (ISx, input storage/source). This implies nonlinear cause effect and thus relationships among the several phases of the production lines. Moreover, multiple points may have storages and direct feeding with basic (e.g., ISx) and self-produced matter (S2), for example when the internal production is not enough or it is more expensive according to the market (for example transformation A can exploit IS3 and/or S2). In this view, the storages are important, and may create problems when they are full (no more space for producing other results/matter) or empty (not enough matter for the next process phases) since the production cannot continue. In addition, when they are full an immobilization of capital is also realized (see Figure 1).

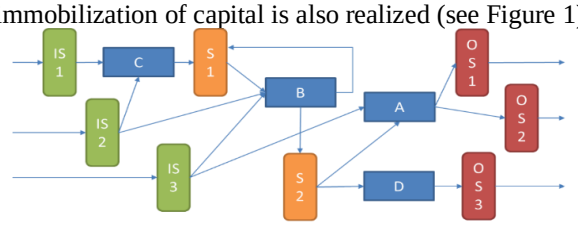

Figure 1 – Example of chemical production in plant in which storage (round squared) and matter transformations (squared) are connected in a network of consumer/producer.

According to this view, an **integrated solution for the production plant maintenance** has to cope with a number of aspects. It has to present a **ticketing management system** addressing workflow for managing maintenance activities with teams of any kind. The teams have to be ready for planned maintenance as well as for corrective maintenance. Predictive maintenance may help them to perform preventive interventions (for example when it is possible to perform some changes into the flow from the control room, putting out of the production pipeline a part of the plant without stopping the rest). For example, to take raw materials from the same or different silos by using a secondary pump and putting a primary on maintenance. In most cases, the predictive maintenance produces probability of fault once per day/hour, since the stop of chemical plant could be very costly in terms of missed production.

The unexpected critical events and alarms that lead to some intervention and plant dysfunction may be detected in control room by the operator, by plant alarms for flooding or any other detection of dysfunction, as well as through the information of some personnel observing the inception of a problem. All these aspects may lead to create a ticket on the Maintenance Management tools. Specific problems may be detected on control room observing DCS data on dashboards. The event produced from data analysis can be automatically generated by means of a direct connection from the control room to the management department, from data flow to workflow of the maintenance teams.

The whole set of maintenance events have to be collected to be further analysed with some business intelligence tools for decision making and by some machine learning tool to perform some predictive maintenance.

A. Architecture

In this section, the general framework of the proposed solutions is presented in terms of functional architecture, as shown in **Figure 2.** The functional architecture presents several components which are described in the following by starting from left to right. The **plant is controlled by a DCS** (Distributed Control System) which produces a new status of the plat in terms of measures every minute. It is controlled on the basis of a number of set points for a large number of machines (pumps, electrolysis, heating, fans, storages, etc.) from a control room. The values of the setpoints over time is planned on the basis of the production to be performed.

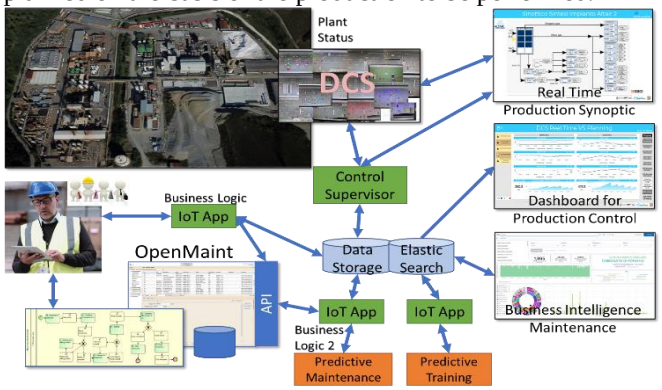

Figure 2 – Functional Architecture

The OpenMaint is an open source ticketing management system which implements: (i) the planned/scheduled maintenance activities, (ii) collect the tickets, (iii) prepare the teams, (iv) send them on the field and (v) collect the results according to a set of specific workflows.

The **Team Operators** have in their hands a tablet to follow the instructions of the maintenance tickets, and a Web App to access at higher level data to see the settings and verify the plant status. They can be also started from the Team mobile App and interface, and by the Control Supervisor, which may autonomously ask the Maintenance Team to create some Events on the basis of the data collected from DCS and/or on the basis of the Predictive Maintenance results. This last possibility is enabled by: (i) the development of a number of MicroService/Nodes for Node-RED into the Snap4City library for accessing to OpenMaint API, (ii) the Predictive Maintenance solutions presented in this paper, (iii) and by the whole architecture. Both, the data of the DCS, and those of the Ticketing system have been collected in ALTAIR for several years. As a first step, we started with the study of the relationships among the data collected from DCS and the events on Ticketing system. To this end, a **Business Intelligence Maintenance** tool based on Elastic Search and Kibana has been designed and set up. It allows to perform visual queries on the plant status and on the history of the maintenance interventions (see **Figure** 2). All the data collected by DCS and the **Control Supervisor** are also visible on a set of **Dashboards** realized by using Snap4City tool [Bellini et al., 2018]. The Dashboards allow to keep trace of the production status with respect to the historical data, and also to monitor the planned production with respect to the

actual production for each product of the whole chemical plant. The **Control Supervisor** collects in real time the DCS data and also data coming from administrative database, energy costs, etc., to compute the best plan for the next hours and days [Badii et al., 2020b]. In addition, it produces, in real time, the data driven streams towards a set of synoptics for the control room. Eventual alarms identified by the **Predictive Maintenance** are registered in the store and passed (in an event-driven manner) to the Control Supervisor and thus on the Synoptics, and also on the Ticketing via the API.

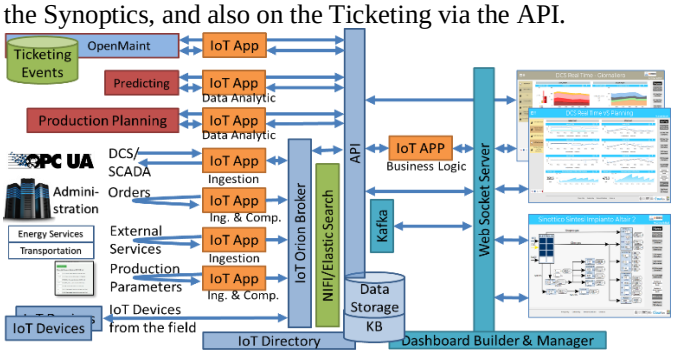

Figure 3 - technical architecture of the Data Ingestion and data flows: from data sources, to IoT App processes, storage and Dashboards with custom widgets and Synoptics.

From the technical point of view, the solution has been implemented by using Snap4Industry development environment which in turn is based on Snap4City technology [Badii et al., 2019]. In **Figure 3,** the technical implementation regarding data flow of the functional architecture of **Figure 2** is presented. In the solution, IoT Apps are Node-RED processes on Docker containers deployed on private cloud as IoT Edge. The IoT App connected with the OpenMaint ticketing system is the interface to access the new events produced by the operators, and to provide new event that may be identified by the Control system. Data are received in push and/or pull, and almost every data message with several attributes is considered an IoT Device instance. To this end, the IoT Devices have been registered, and their data structure/model formalized [Badii et al., 2020].

Once registered, the received IoT messages by an IoT App can be sent into the system via the *IoT Broker*, which saves them automatically into the *Data Storage*. The IoT Apps can collect data in Push/Pull modalities, and in some cases a preliminary computation is performed. The arrival of a new message in the Broker may provoke the sending of a new message into the data storage as well as a set of messages towards the user interface clients (dashboards and mobile Apps). In fact, each client browser showing one or more Dashboards and each mobile App connected to the Dashboard Manager need to have established a number of permanent WebSocket connections to receive in real time any change on the data.

DCS/SCADA data messages are received in push from OPC-UA. The DCS data includes measured values from productions flows, the settings planned and reached by the plant, status of the material storages, etc., such as for AcidoEsausto, FeCl2pot, FeCl3pot, FeCl3std, HCl32, HCl35, HCl36, KOH, HCl, K2CO3aq, NaOH50, that are the most important to represent the plant status and these are the values used by the planner algorithm.

B. Business intelligence for maintenance

The goal is to have a general overview of maintenance events, both planned maintenance and breakdown events that lead to downtime. It is possible to analyze the number of maintenance events per plant and per type of intervention by filtering on a chosen time frame.

The tool of **Figure 4** presents six sections. In the first section, it is possible to choose the time frame by selecting the date (absolute) or by selecting years, months, or days (relative). In the second section, the user can see the number of maintenance events in the selected period: the average and median of the number of hours needed to complete a maintenance intervention (intended as the difference between the start and end of intervention datetime). In the third section, it is possible to select data by specifying and/or the: Plant, Specialty, Work type. A graph shows the trend of the working plants (daily); if the time span is very wide the plants will be grouped by week or month. In the fourth section, through a bar graph, we keep track of the maintenance events carried out per day (weekly or monthly if the time frame is wide). It is possible to have several maintenance events for the same plant; therefore, a table at the bottom of the dashboard lists all the details by maintenance event Time, Permission List, Plant, Signature, Specialty, Status, Job Type, Air Temperature, air humidity and rain. In the fifth section, we have a visual representation of the maintenance events by Plant, Specialty, and Job Type. The font size depends on the frequency of the

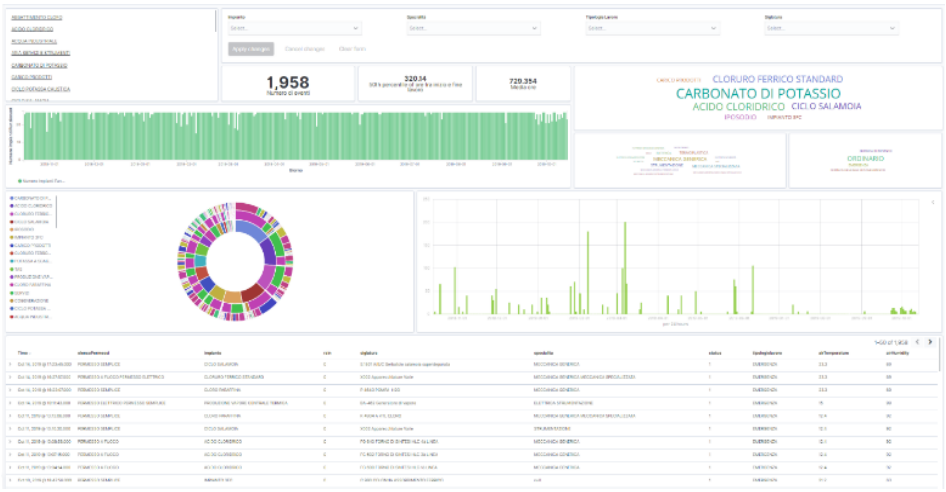

Figure 4 - Maintenance Dashboard, Business Intelligence for Maintenance

records. By choosing an item the dashboard widgets are filtered according to the choice. Finally, in the sixth section we have a pie representation of maintenance events by Plant, Specialty, and Job Type. From this dashboard it is possible to access the general management and control plant dashboards. In another section, we keep track of MTTF (mean time to Failure), MTTR (mean time to repair, and MTBF (mean time between failure, as MTTR+MTTF), current and trend values. Other information are: (i) Average temperature of the day in which failures occurred in the plant, (ii) Average Humidity for the day on which failures occurred in the system.

IV. First Predictive Maintenance Model

According to the previous description, the challenge was to predict the plant failure 60 minutes before it happened. This section has two subsections; the first includes some descriptive notes about the dataset, while the second describes the architecture of the predictive model exploiting the LSTM model and related validation.

A. Data Description and Engineering

The data collection has been conducted on the basis of about 300000 observations from 2020-04-28 to 2021-01-04 (nonstop for COVID-19 since the production is mainly on chemical product for food industry). Regarding production data, the collected dataset is composed by a set of data coming from the DCS (such as plant: production, storage, status, several temperatures of elements, gear plants, process/safety parameters, chemicals compounds produced etc.) measured every minute. Therefore, a multi-feature dataset composed by 1-minute observation was one of the inputs. A total of 343.183 observations for 147 features/variables were measured. Regarding the faults, we had the list all the details coming from the Business Intelligence tool for maintenance including: event datetime, Permission List, Plant, Signature, Specialty, Status, Job Type, Air Temperature, air humidity and rain. Ticket and stop classification as "GENERAL PLANT STOP", "ORDINARY", "PLANT STOP" and "EMERGENCY". The label “ORDINARY” concerns all planned maintenance operations, while the labels "PLANT STOP" and "GENERAL PLANT STOP" concern programmed machine stops and finally "EMERGENCY" concerns machine stops due to an unexpected fault in the plant. In this way, a multi-feature annotated dataset has been created, considering data with the label "EMERGENCY" as faults, while all the other above-mentioned labeled data as regular working conditions, in order to implement a predictive binary classification model (as described in details in Section IV.B).

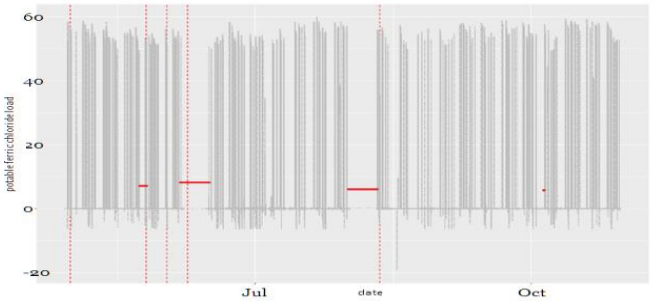

Figure 5 - Example of a variable affected by a failure in terms of percentage of plant production capability.

In total, in the selected period, there were a number of breakdowns in emergency, and they produced a wider number of time intervals in which the production has been stopped/reduced for dysfunction and repaired (see **Figure 5).** From the analysis, 28 variables have been identified removing those that replicated the same information in the production flow, as reported in **Table 2.**

Table 2. Overview of feature measured at a given time.

Feature	Plant	Description	Unit of measure
TempreactoreR4001	chlorine paraffins (CPS)	reactor temperature indication R 4001	°C
TempreactoreR4002	chlorine paraffins (CPS)	reactor temperature indication R 4002	°C
TempreactorR4003	chlorine paraffins (CPS)	reactor temperature indication R 4003	°C
S4304	chlorine paraffins (CPS)	level indication	%
standardFerric Chloride	Potable Ferric std	flow rate measurement and totalization	m3
S904C	Potable Ferric std	level indication	%
S904B	Potable Ferric std	level indication	%
S904A	Potable Ferric std	level indication	%
potFerricChloride	Potable Ferric Chloride	flow rate measurement and totalization	m3
S904E	Potable Ferric Chloride	level indication	%
S904D	Potable Ferric Chloride	level indication	%
QuantNaOHBatchNaClO_2	NaOH KOH	flow rate measure and totalization	lt
QuantNaOHperBatchNaClO	NaOH KOH	flow rate measure and totalization	m3
ConversionNaOH	NaOH KOH	electrolysis load adjustment (production)	kA
ConversionKOHlinea 1	NaOH KOH	electrolysis load adjustment (production)	kA
KOH_1_charge	NaOH KOH	flow rate measure and totalization	m3
KOH_2_charge	NaOH KOH	flow rate measure and totalization	m3
S487	NaOH KOH	level indication	%
S484	NaOH KOH	level indication	%
S5104	NaOH KOH	level indication	%
hypo sodium	sodium hypochlorite	quantity of material produced	m3
S857	sodium hypochlorite	level indication	%
S856	sodium hypochlorite	level indication	%
S851	sodium hypochlorite	level indication	%
S852	sodium hypochlorite	level indication	%
S854	sodium hypochlorite	level indication	%
S871	HCl	level indication	%
RedoxFeCl3Pot	Ferric Chloride std	potential measure redox Ferric Chloride	mV

Metrics S857, S856, S851, S852, S854, S871, S487, S484, S5104, S904E, S904D, S4304, S904C, S904B, S904A represent the level of the storages containing the chemical product. On this regard, we calculated the difference with the previous miute to highlight the total daily production of a given substance. These derived metrics were added to the above-described set of variables, thus obtaining a total of 43 features for 343183 minutes and few events of failure but a large number of minutes in which the plant have been stopped/reduced in the operative level. Then we have, on a total of 343183 1-minute observation data, 37286 minutes of failure leading to downtime.

B. Classification model LSTM

The aim of this first model is to predict the status of the plant (i.e., if the plant is properly working, under some failure for anomalies, and thus failures which may lead to downtimes/stops). This is equivalent to a multi-variate binary classification problem (considering two classes labelled as “Normality” and “Fault” for normal working conditions and failures, respectively), taking into account also temporal dependency. The prediction should be 1 hour in the future, considering the data measured in the previous 20 minutes with respect to the current observation.

Given the requirements, a deep learning model based on LSTM was adopted. The fact that we take only 20 minutes data with respect to the current is marginally relevant, since LSTM model per se keep tracks the temporal evolution for much larger number of time instants. The model architecture is composed by LSTM layers with a Rectified Linear Unit (ReLU) activation function. During the training and hyperparameter optimization phases, we noticed that adding more LSTM layers improved the quality of prediction results. To this end, 5 LSTM layers have been placed. Since our goal was a binary classification, we used a Dense output layer with a single neuron and a sigmoid activation function. The model is compiled to minimize the log loss (in our case, the binary_crossentropy metric) with an Adam optimizer.

In order to properly represent the dataset for the classification task, and to suitably prepare the input data for the LSTM layers, data have been reshaped into sequences, considering the previous 20 time-steps (i.e., 20 minutes) for each observation, and aiming at predicting the plant status 60 minute in the future (please note that the 20 minutes are just the time windows at the last time interval while the network has the capability to keep memory of a much longer time interval in its hidden layers). Therefore, the input dataset has been organized as the following time series:

$$(X_1, X_2, \dots\dots, X_{20})\ (Y_{80})$$
$$(X_2, X_3, \dots\dots, X_{21})\ (Y_{81})$$
$$\dots\dots\dots\dots$$
$$(X_n, X_{n+1}, \dots\dots, X_{n+19})\ (Y_{n+79})$$

where X_1, X_2 … X_n are multi-feature data, and Y_n the plant status measured at temporal instants (representing minutes) t=1, t=2,…,t=n. Then, we split our dataset into 70% for train data, 15% for validation data and 15% for test data.

To avoid overfitting, a dropout layer was inserted and an early stopping procedure was defined, monitoring the validation loss. This means that, if after a certain number of training iterations (set to 10 epochs in our case) the validation loss does not decrease, then the training is stopped. In this way, it is possible to prevent overfitting due to over training epochs when the validation loss is no longer improved. Finally, an automated hyperparameters optimization was performed through a Randomized Search Cross-Validation. The best model resulting from the whole process of parameters optimization and cross-validation is represented in **Figure 6.**

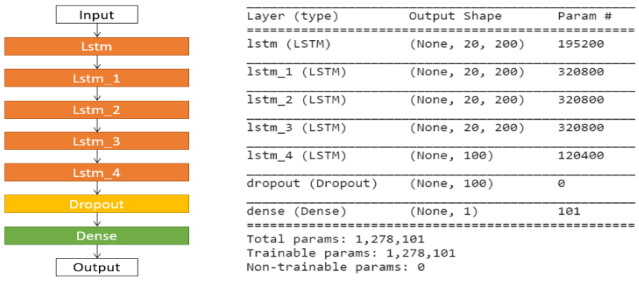

Figure 6 –LSTM Architecture Model

The above-described model has been adopted for predictions of the plant working status on unseen test data, with the capability of being executed in real-time on real production data.

C. Validation of the LSTM MODEL

The dataset used for validation is composed of 15% of the total data. Overall accuracy was calculated as the number of 1-minute observations during normal minutes working conditions, plus the number of correctly classified 1-minute observations of failures minutes as a fraction of the total number of minutes1-minute observations. The resulting accuracy was 87.40%. The ROC curve has been computed and plotted in **Figure** 7. It represents the classification performances, with True Positive Rate (TPR) on y-axis against the False Positive Rate (FPR) on x-axis. Moreover, the associated Area Under Curve (AUC) was calculated. The resulting AUC value is 0.822.

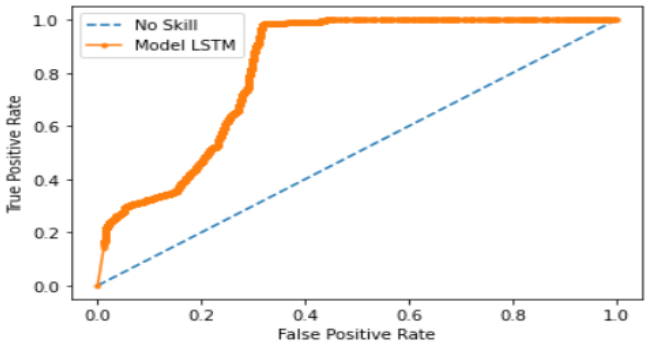

Figure 7 - ROC curve for the LSTM model

The confusion matrix is reported in **Table 3**.

Table 3. Confusion Matrix for the LSTM model

Predicted Class **Actual Class**	Normality	Fault
Normality	43485	3229
Fault	3246	1436

If we consider as class" Normality" and "Fault", recalling the definitions of True Positive (TP) as an outcome where the model correctly predicts the positive class, False Positive (FP) as an outcome where the model incorrectly predicts the positive class, and False Negative (FN) as an outcome where the model incorrectly predicts the negative class, we can compute the following classification metrics:

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN}$$
$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Finally, the model predictive classification performance is reported in **Table 4**, where $weighted\ avg = Support_{Class1} * Score_{Class1} + Support_{Class2} * Score_{Class2}$, and *Support* represents the number of cases inside the test dataset**.**

Table 4. Predictive Classification Model evaluation results for the LSTM model

	Precision %	**Recall %**	**F_1 score %**
weighted avg	0.87	0.87	0.87

V. ADVANCED PREDICTIVE MODEL WITH CNN-LSTM

With the aim to get more accurate data we tried to reduce the effect of noise by added a CNN layer to our model. The CNN-LSTM approach provides the advantages of combining CNN powerful feature extraction with the capability of LSTM in capturing temporal dependencies. CNN are actually useful for learning spatial local features from input time series [Wang et al., 2017], since they have the capability of performing an optimized smoothing of the signal (through the 1D convolutional and pooling layers), while maintaining the underlying data trend. Therefore, they can extract local features of time-series data (including multi-variate time-series) more accurately, thus improving the performances of subsequent LSTM layers in learning temporal dependencies [Xie et al., 2020].

A. *Classification model CNN-LSTM*

In this case, the model architecture is composed by a one dimensional convolutional layer, followed by an Average Pooling layer that computes the average on the output of the previous convolutional layer across all time steps. Then we added LSTM layers with a Rectified Linear Unit (ReLU) activation function. During the training and hyperparameter optimization phases, we noticed that adding more LSTM layers improved the quality of prediction results. To this end, 5 LSTM layers have been placed. Since our goal was a binary classification, we used a Dense output layer with a single neuron and a sigmoid activation function. The model is compiled to minimize the log loss (in our case, the binary_crossentropy metric) with an Adam optimizer.
Finally, an automated hyperparameters optimization was performed through a Randomized Search Cross-Validation. The best model resulting from the whole process of parameters optimization and cross-validation is represented in **Figure 8.**

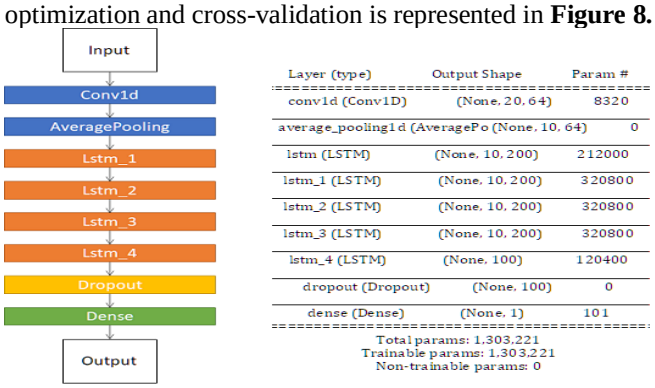

Figure 8 – CNN-LSTM Architecture Model

B. *Validation of CNN-LSTM model*

As before for the LSTM, the dataset used for validation has been composed of 15% of the total data. Overall accuracy was calculated as the number of 1-minute observations during normal minutes working conditions, plus the number of correctly classified 1-minute observations of failures minutes as a fraction of the total number of minutes1-minute observations. The resulting accuracy was 91.81%. The ROC curve is reported in **Figure** 9, and the resulting AUC value is 0.934. Showing in this cases better performance of CNN-LSTM with respect to the LSTM.

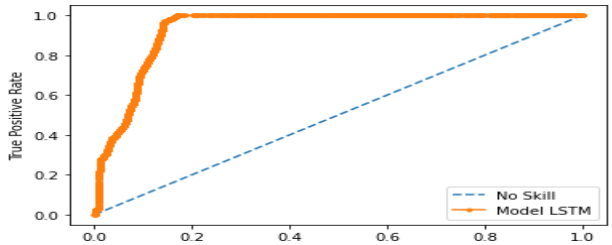

Figure 9 - ROC curve for the CNN-LSTM model

The confusion matrix is reported in **Table 5**. And the classification performance are reported in **Table 6**

Table 5. Confusion Matrix of CNN-LSTM approach

Predicted Class / **Actual Class**	Normality	Fault
Normality	45811	903
Fault	3306	1376

Table 6. Predictive Classification Model evaluation results

	Precision %	**Recall %**	**F_1 score %**
weighted avg	0.90	0.92	0.90

C. *Explainable CNN-LSTM to exploit the results*

In order to interpret the results, the Shapley additive explanation (SHAP) has been used. Through this analysis it has been possible to understand how much each feature contributes (positively and negatively) to the prediction of a failure, and therefore it is possible to have both an overview on how to intervene at maintenance level on future failures and ideas to improve our model. The SHAP explanation method computes Shapley values from coalitional game theory. The feature values of a data instance act as player in a coalition. Explanations obtained by the Deep SHAP method are represented graphically. In the **Figure 10**, we have an explanation regarding the failure prediction (see Figure 10a) and the normal operation prediction (see Figure 10b) using our test data set. The SHAP results at each prediction time instant are recorded together with the Boolean output of the predictor expressing the probability of fault. The trends of the SHAP for each of the variable describe the activation of the network and thus are used to identify which are the causes of the predicted fault when it occurs. Therefore, with tune thresholds with respect to typical trends a list of critical variables are produced, and these variables correspond to DCS /IoT Devices into the plant and thus to sections of the production plant. Thus a maintenance team and the personnel in the control room are informed providing this list and area as a fundamental information at the support of their decision.

Through the image we can see how different features contribute positively to the failure output (shown to the left of the base value in red) and those that contribute negatively to the failure output (shown to the right of the base value in blue). As we can see in **Figure** 10a, the most relevant features in the fault determination are the production derived features diff_S854, diff_S904B of two different production lines (i.e., sodium hypochlorite and Potable Ferric std). Other relevant features are the RedoxFeCl3Pot and the charging features. Production anomalies are a warning symptom of a fault. Even for the plant normality situation (see **Figure** 10b), the charge features and the derived production features are the main contributors to the prediction of our CNN-LSTM model.

VI. CONCLUSIONS

In this paper, a predictive maintenance model for classification of failures in a real industrial process has been presented. The proposed solution is based on a deep learning CNN-LSTM architecture, predicting the working status of the productive process in the Altair chemical plant. The proposed model CNN-LSTM provides a one-hour prediction of the plant status and indications on the areas in which the intervention should be performed by using explainable LSTM technique. Assessing the proposed method with real production data, experimental results show an average Accuracy of 91.8% and an average F1-score of 90%, which are very good results considering that the proposed model provides predictions of the plant working status one hour in the future, and it is capable of running in real time (thus aiming at resolving some lacks found in other state-of-the art solutions). The explanation of the predictions provides suggestions for the maintenance teams. The paper also

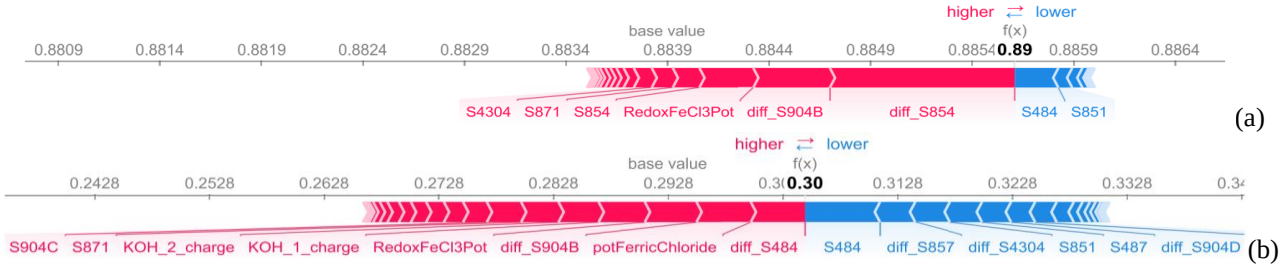

Figure 10 - Explanation of prediction generated by model for fault (a) and normality (b). Please note that this is an example of a specific event.

introduced business intelligence tools on maintenance data and the architectural infrastructure for the integration of predictive maintenance approach into the whole control and management system of ALTAIR industry 4.0 plant in the context of SODA and large renovation of the production infrastructure.

Acknowledgements

The authors would like to thank ALTAIR for which we have developed the solutions described in this paper, SODA project partners and Regione Toscana for cofounding of ALTAIR. Former technologies as Snap4City/Industry are 100% open source by DISIT Lab and can be accessed at https://www.snap4city.org.

References

[Badii et al., 2019] Badii C., P. Bellini, A. Difino, P. Nesi, G. Pantaleo, M. Paolucci, MicroServices Suite for Smart City Applications, Sensors, MDPI, 2019.

[Badii et al., 2020a] Badii C., P. Bellini, A. Difino, P. Nesi, Smart City IoT Platform Respecting GDPR Privacy and Security Aspects, IEEE Access, 2020.

[Badii et al., 2020b] C. Badii, P. Bellini, D. Cenni, N. Mitolo, P. Nesi, G. Pantaleo, M. Soderi, "Industry 4.0 Synoptics Controlled by IoT Applications in Node-RED", Proc. of the IEEE iThings, November 02-06, 2020.

[Bellini et al., 2018] Bellini P., et al. Smart City architecture for data ingestion and analytics: processes and solutions. 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, 2018.

[Bellini et al., 2019] E. Bellini, L. Coconea, P. Nesi, "A Functional Resonance Analysis Method driven Resilience Quantification for socio-technical System", IEEE Systems Journal, pp.1-11, 2019

[Binding et al., 2019] A. Binding, N. Dykeman and S. Pang, "Machine Learning Predictive Maintenance on Data in the Wild," *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 507-512.

[Blanchard et al., 1995] B.S. Blanchard, D. Verm, E.L. Peterson Maintainability: A key to effective and maintenance management John Wiley & Sons, New York, NY (1995)

[Gentles, 2020] William M. Gentles,Chapter 42 - HTM best practice guidelines and standards of practice around the world, 2020.

[Huuhtanen and Jung, 2018] T. Huuhtanen and A. Jung, "PREDICTIVE MAINTENANCE OF PHOTOVOLTAIC PANELS VIA DEEP LEARNING," *2018 IEEE Data Science Workshop (DSW)*, 2018, pp. 66-70.

[Jardine et al., 2006] Jardine et al., 2006 A.K.S. Jardine, D. Lin, D. Banjevic A review on machinery diagnostics and prognostics implementing condition-based maintenance Mechanical System and Signal Process, 20 (2006), pp. 1483-1510.

[Kanawaday and Sane, 2017] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using IoT sensor data," *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2017, pp. 87-90.

[Kim and Kim, 2019] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data", PLoSONE, vol. 14, no. 2 (2019). DOI: https://doi.org/10.1371/journal.pone.0212320.

[Lei et al., 2018] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, "Machinery health prognostics: a systematic review from data acquisition to RUL prediction", Mechanical Systems and Signal Processing, Vol. 104, pp. 799–834, Elsevier (2018).

[Liao and Kottig, 2016] L. Liao and F. Köttig. "A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction". Applied Soft Computing Journal, Vol. 44, pp. 191–199 (2016).

[Mathew et al., 2017] J. Mathew, M. Luo and C. K. Pang, "Regression kernel for prognostics with support vector machines," *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1-5, doi: 10.1109/ETFA.2017.8247740.

[Sánchez et al., 2016] MDPI and ACS Style Li, C.; Sánchez, R.-V.; Zurita, G.; Cerrada, M.; Cabrera, D. Fault Diagnosis for Rotating Machinery Using Vibration Measurement Deep Statistical Feature Learning. Sensors 2016, 16, 895.

[Shao et al., 2019] B. Shao, X. Hu, G. Bian and Yu Zhao, "A Multichannel LSTM-CNN Method for Fault Diagnosis of Chemical Process", Math Problems in Engineering (2019).

[Tao et al., 2015] S. Tao, T. Zhang, J. Yang, X. Wang and W. Lu, "Bearing fault diagnosis method based on stacked autoencoder and softmax regression," *2015 34th Chinese Control Conference (CCC)*, 2015, pp. 6331-6335, doi: 10.1109/ChiCC.2015.7260634.

[Tovar et al., 2020] M. Tovar, M. Robles and F. Rashid, "PV Power Prediction, Using CNN-LSTM Hybrid Neural Network Model. Case of Study: Temixco-Morelos, México", Energies, Vol, 13, 6512 (2020). DOI:10.3390.

[Uhlmann et al, 2018] Eckart Uhlmann, Rodrigo Pastl Pontes, Claudio Geisert, Eckhard Hohwieler, Cluster identification of sensor data for predictive maintenance in a Selective Laser Melting machine tool, Procedia Manufacturing, Volume 24, 2018, Pages 60-65, ISSN 2351-9789.

[Wang et al., 2017] Z. Wang, W. Yan and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline", in Proc. of 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1578-1585 (2017). DOI: 10.1109/IJCNN.2017.7966039.

[Xie et al., 2020] H. Xie, Li Zhang and C. P. Lim, "Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer", IEEE Access, vol. 8, pp. 161519-161541, 2020, DOI: 10.1109/ACCESS.2020.3021527.

[Yam et al., 2001] Yam, R., Tse, P., Li, L., & Tu, P. (2001a). Intelligent predictive decision support system for condition-based maintenance. International Journal of Advanced Manufacturing Technology, 17(5), 383–391.

[Zhang and Zhao, 2017] Zhanpeng Zhang, Jinsong Zhao, A deep belief network based fault diagnosis model for complex chemical processes, Computers & Chemical Engineering, https://doi.org/10.1016/j.compchemeng.2017.02.041.

[Zhang et al., 2018] Jianjing Zhang, Peng Wang, Ruqiang Yan, Robert X. Gao, Long short-term memory for machine remaining life prediction, Journal of Manufacturing Systems, Volume 48, Part C, 2018, Pages 78-86, ISSN 0278-6125.

[Zhao et al., 2017] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks". Sensors, Vol. 17, 273 (2017).

[Zhou et al., 2020] Y. Zhou, M. Hefenbrock, Y. Huang, T. Riedel and M. Beigl, "Automatic Remaining Useful Life Estimation Framework with Embedded Convolutional LSTM as the Backbone". In: Lecture Notes in Computer Science, vol 12460. Springer (2020).