PAIS 2022
A. Passerini and T. Schiex (Eds.)

doi:10.3233/FAIA220069

Supervised Anomaly Detection in Crude Oil Stabilization

Mattia SILVESTRI [a,1] Michele LOMBARDI [a] Emiliano MUCCHI [b] Luca CADEI [c] Giovanna MAGNAGO [c] Marco PIANTANIDA [c] Valentina D'OTTAVIO [c] Nguyen VAN TU [c] Simona DUMA [c] Silvia TADDEI [c] Annagiulia TIOZZO [c] Andrea CORNEO [c] Lorenzo LANCIA [c] Laura ROCCHI [d] Pietro COFFARI DI GILFERRARO [d]

[a] *University of Bologna*
[b] *University of Ferrara*
[c] *Eni S.p.A.*
[d] *NIER Ingegneria S.p.A.*

Abstract.
In recent years, pervasive digitalization has affected the industrial world, including the oil and gas sector. With more and more data becoming available, Machine Learning algorithms have become a promising tool to improve Predictive Maintenance operations. In this work, we have designed an alerting system that notifies the site operator with an adequate advance when an anomaly is going to occur. In particular, we focus our analysis on the stabilization column of an Oil Stabilization Facility to prevent the column bottom temperature to overcome safety boundaries. The experimental analysis demonstrates that our system provides reliable results, in terms of both identified anomalies and false alarms. In addition, the system is currently under deployment on the company computing infrastructure and the first working version will be available by the end of May 2022.

Keywords. Neural Networks, Anomaly Detection, Oil and Gas industry

1. Introduction

We present a case study for a practical application of AI techniques to an industrial problem, namely *anomaly detection in an oil-stabilization facility*. The plant is operated by Eni[2], an Italian multinational energy company headquartered in Rome with operations in 69 countries.

In the considered facility, the product flowing from the wells is a high pressure/high temperature multi-phase stream of oil, water, and gas. The goal of the stabilization process is to separate water and gas from the crude oil and reduce the pressure and temperature of the latter until it is compatible with ambient conditions. Stabilization is achieved first via a sequence of gravity separators, followed by an oil-stabilization column, where the gas is stripped away and the oil becomes stable at ambient pressure and temperature.

[1]Corresponding Author: Mattia Silvestri, University of Bologna, E-mail: mattia.silvestri4@unibo.it
[2]https://www.eni.com/

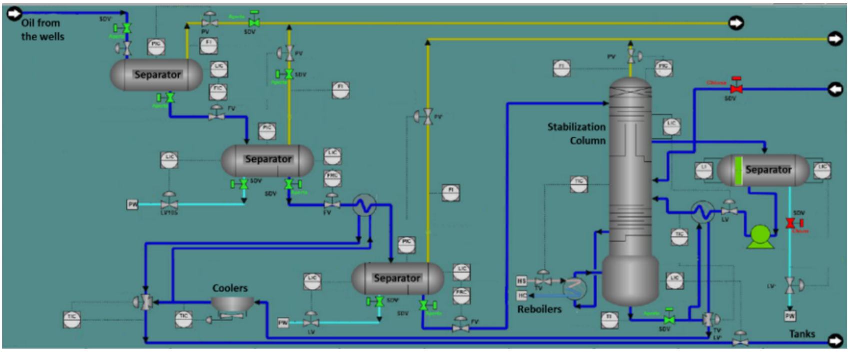

Figure 1. Schematic of the Oil Stabilization Facility.

Sub-optimal stabilization comes with *economic, and environmental costs*: an under-stabilized oil does not comply with the specifications needed to ship the oil to the refineries for processing whereas over-stabilization leads to higher costs and emissions since the process is energy-intensive. The plant control system promptly reacts to maintain the process within specifications, but an advance alert mechanism that prevents anomalous situations to occur will allow a smoother and more energy efficient control of the plant.

Our study targets the oil-stabilization column: we present the design of *an anomaly detection system capable of alerting the plant operator* before the value of a specific KPI (column bottom temperature) leaves an acceptable range. In particular, we implement a data-driven solution based on a predictive model as its core, plus pre-processing and post-processing modules.

Pre-processing transforms the input data so that it is digestible by the predictive models. In the post-processing step, raw predictions are aggregated to obtain a more robust alarm signal, and a multi-objective quality criterion allows the operator to balance the sensitivity and specificity of the system. For the predictive core, we consider a variety of solutions including regression and classification approaches, operating with both sequence and aggregated data. Models are retrained over time to account for concept drift and evolving operating conditions.

The design process relies on historical plant data, for which we report results from extensive experimentation. Our best performing solution manages to obtain solid results, with very few false negatives (missed anomalies) and an acceptable level of false positives (often less than one false alarm per day on average).

The designed system is *currently being deployed within Eni's computing infrastructure*: the first phase of this process will be completed by May 2022, while full operation should be reached after an adjustment period of a few months.

The paper is structured as follows: in section 2 we briefly introduce the oil-stabilization process and its components. In section 3 we provide a preliminary characterization of the plant data, which is meant to provide motivations for some of our design choices. The structure of the anomaly detection systems is presented in section 4, along with potential design alternatives for all its components. In section 5 we present the results of an experimental evaluation and identify the best performing configuration. Related works are briefly covered in section 6, while concluding remarks are in section 7.

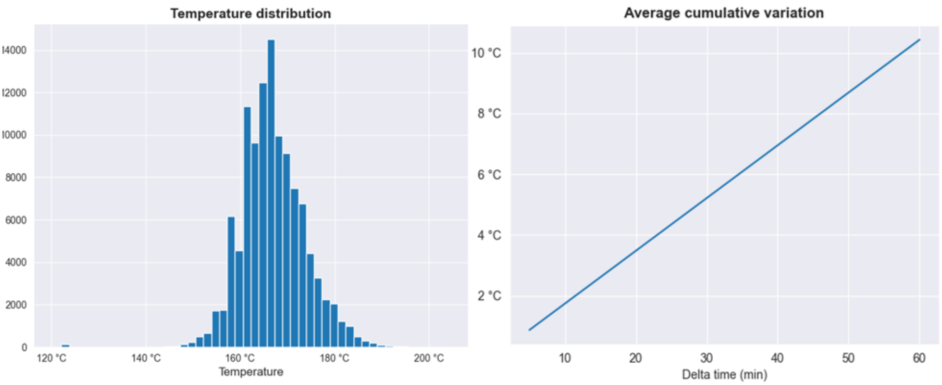

Figure 2. In the left hand-side of the figure, we see the histogram of the stabilization column bottom temperature values whereas the right hand-side shows the average cumulative variation of the temperature considering a time lag from 0 to 60 minutes.

2. Problem Description

The case study is focused on an Oil Stabilization Facility operated by Eni. The facility collects the crude oil produced by several wells drilled to exploit a subsurface reservoir. The oil flowing from the wells is a high pressure/high temperature multiphase stream of oil, water, and gas. The Oil Stabilization Facility is targeted to:

- Separate the water and the gas contained in the crude oil.
- Make the oil stable at ambient pressure and temperature.

The schematic of the facility is shown in Figure 1. The oil from the wells (shown as a dark blue line coming from the top-left corner of the figure) is sent to a sequence of three vessels, which act as gravity separators: due to the different density of oil, water and gas, a portion of the gas flows away from the top part of each separator (indicated as yellow streams in the figure), the water contained in the oil flows away from the bottom part of the separators (indicated with light blue streams) and the oil flows forward from one separator to the other. The separators act at progressively decreasing pressures, allowing additional gas to be stripped away at each following stage.

After the third separator, the oil is routed to an Oil Stabilization column: the oil enters the upper part of the column, and it flows across several plates, then it enters a further separator (shown on the right of the column), where the residual water is eliminated, and finally, it flows across further plates at the bottom of the column, where it is heated by two reboilers. The heating of the oil at the bottom of the column produces an upward flow of vapor that interacts with the oil falling across the plates, helping to strip away the residual gas from the oil. Finally, the oil is cooled down and it is stored in tanks.

The key parameter of this process is the residual vapor pressure of the oil stored in the tanks, named Reid Vapor Pressure (RVP): this must be lower than ambient pressure (1 bar) to avoid the release of gas in the tank. If the RVP is greater than 1 bar, the stabilization process does not meet the requirements. If the RVP is too low (e.g. 0.2 – 0.6 bar), an unnecessary amount of energy has been spent in the reboilers to over-stabilize the oil.

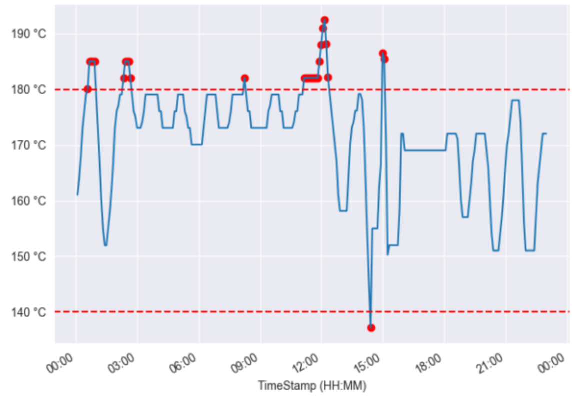

Figure 3. A representative example of anomalous events that may occur during a day.

Real-time RVP readings are not available, but the temperature at the bottom of the column can be used a proxy, since the two are highly correlated. Low temperature are associated to high RVP values, while high temperatures to oil over-stabilization. In the considered scenario, a temperature within the $[140, 180]$°C range is considered acceptable. The temperature at the bottom of the column can be controlled through the reboilers, which heat the oil through the thermal exchange of hot vapor and the oil across a tubing system. The amount and the temperature of the vapor can be regulated to control the value of the temperature at the bottom of the column.

3. Plant Data Characterization

As per the considerations in section 2, the temperature of the bottom of the column provides a natural target for an alarm system. As preliminary step, it is therefore important to characterize the temperature behavior in the context of the facility under investigation.

We start by inspecting the distribution of the temperature values: the histogram from fig. 2 (left) shows this is not too far from being Normal, with a mean and a standard deviation of respectively 167°C and 6.8°C and a few outlier values at around 120°C (likely related to plan maintenance events). It can also be seen that the lower bound of the safe interval (140°C) is almost never passed, whereas the upper bound (180°C) is exceeded a non-negligible number of times. The temperature very rarely reaches values close to 200°C, due to the control systems operating on the plant.

In an effort to investigate the temperature dynamics, we look at how quickly the temperature value changes over time. For this purpose, we compute the average cumulative difference between temperature values that are separated in time by a given time lag, i.e.:

$$\frac{1}{N}\sum_{i=1}^{N} |y_{\Delta t}^{(i)} - y^{(i)}| \tag{1}$$

where N is the dataset size, $y^{(i)}$ is the temperature value for the i-th sample, $y_{\Delta t}^{(i)}$ is its corresponding time-lagged value with a lag Δt. As shown in fig. 2 (right), temperature

Figure 4. Architecture of our designed alarm detection system.

changes are relatively slow, with average variations of about 10°C per-hour : this is due partly to capacity effect, and to the flowing speed of the oil through the pipes.

Finally, we attempt a characterization of anomalous events (i.e. out of bounds temperature). In fig. 3, we see an example that is representative of the issues that may affect the facility:

- Multiple anomalies may occur on the same day.
- Anomalies may have different duration.
- Multiple anomaly events are more likely when the temperature stays close to 180°C, signifying an ongoing alarm condition.
- As shown in the timeframe 00:00 AM-03:00 AM, the temperature may overcome the desired bounds in different intervals which are relatively close in time. It is important to decide when they are classified as different anomalies and when they are assigned to the same one.

Overall, this is a practical scenario that is complex to handle, thus making the design of a well-behaving alarm system far from trivial.

4. Design of the Alarm Detection System

We developed a data-driven system designed to alert the user when the oil temperature at the bottom of the stabilization column is expected to be outside a specified $[t_{min}, t_{max}]$ interval over the next τ minutes. The idea is that, if we are able to notify the plant operators with a proper advance, they may adopt suitable containment measures.

In principle, a Machine Learning model could be used to obtain predictions about the temperature behavior in the considered time frame and then to directly warn the operator. More realistically, the predictions would need to be post-processed to improve the robustness and sensitivity of the alarm signal. Since mistakes (both false alarms and missed anomalies) are unavoidable, such a post-processing step should be flexible enough to accommodate priorities determined by the plant operator.

The general architecture of our alarm detection method is depicted in fig. 4. Data (either historical or real-time) are collected from remote sensors installed on the plant and stored via an architecture that records the measurements from the field. The collected data undergo a pre-processing step to make them digestible for a data-driven approach. Finally, raw predictions require a post-processing step to generate an alarm signal that is easily interpretable by and manageable for the human operator.

In the remainder of the section, we will provide additional details about the building blocks of our method, namely data preprocessing, predictive model, and alarm signal generation.

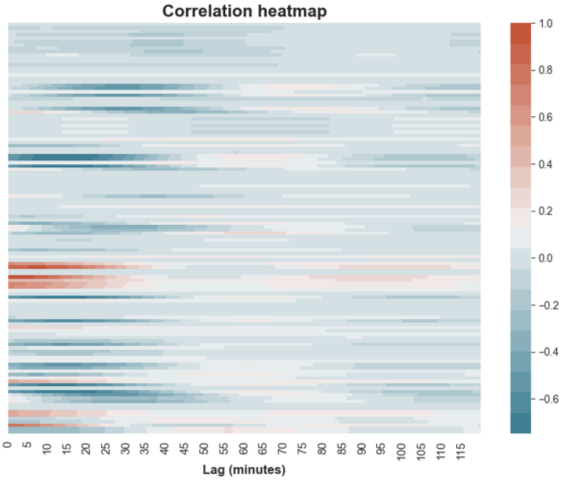

Figure 5. Time-lagged correlation analysis with lag in the interval $[0, 120]$ minutes.

4.1. Preprocessing

During the preprocessing step, raw data from the plant are cleaned and transformed so as to make them digestible by the predictive modules. While some preprocessing steps are applied to both regression and classification models and are independent of the predictive model, others are task-specific or model-specific. We start by enumerating a list of common practices and then we proceed by describing specific preprocessing steps.

As a first step, we perform missing value imputation, so that we are not later constrained to choosing predictive models that can handle them. The specific imputation mechanism will be described in section 5.

Some Machine Learning algorithms (e.g. Neural Networks) require rescaling the input features to canonical ranges to properly work. On this purpose, we apply standardization to all the input variables. As a result, all the input features are centered on zero and have unit variance.

We then proceed by describing a task-specific pre-processing step. In section 3, we have highlighted that plant dynamics are relatively slow, and that oil takes a non-negligible time to move between components. For this reason, feeding synchronous input measurements runs the risk of missing any delayed effect due to this mechanism. Fortuntately, a preprocessing step can be used to re-align the input data so as to keep into account the delayed responses of the stabilization column temperature. Specifically, if we refer to the m-dimensional input as x, where each input feature x_j has a delayed response with lag $\Delta t'_j$ and we want to predict the stabilization column temperature with an advance τ then the result of the alignment operation will be:

$$x_j^{(t)} := x_j^{(t-\max(\tau, \Delta t'_j))} \quad \text{for } j \in \{1, \ldots, m\} \tag{2}$$

This procedure shifts each input feature in time according to the delayed response without overcoming the required prediction advance. The alignment operation is mostly relevant for regression model, since in the classification scenario our predictions refer to a rather long forward interval ($[t, t+\tau]$), which implicitly accounts for the delayed response.

The delay values are identified via a time-lagged correlation analysis between the input features and the stabilization column temperature; for simplicity, we use the (lagged) Pearson correlation coefficient:

$$\rho_{XY_{\Delta t}} = \frac{\sigma_{XY_{\Delta t}}}{\sigma_X \sigma_{Y_{\Delta t}}} \tag{3}$$

where Δt is the time lag. In fig. 5, the heatmap rows correspond to the features whereas the columns are the time lags. A subset of the cells emerge due to either their strong positive correlation (red color) or strong negative correlation (blu). This subset corresponds to the set of features and corresponding optimal lag that are fed to the predictive models.

Since we are dealing with time-series data, there is also a significant design choice concerning how to handle the predictive model input. Two natural options are:

- Feeding the predictive model with a sequence of input data, corresponding to a reasonably long time window.
- Using aggregation function (e.g. common moment statistics such as mean and standard deviation) to extract meaningful features in the pre-processing step.

Both these solutions can be effective but they require a predictive model that handles the specific data format.

4.2. Predictive Models

From a Machine Learning perspective, the predictive task at the core of the system could be framed as either one of two distinct Machine Learning problems:

- **Regression:** prediction of the temperature values τ time units in the future.
- **Classification:** prediction whether the temperature will be inside (e.g. class 0) or outside (e.g. class 1) of the feasible range in the next τ time units.

In our setup, regression is arguably harder than classification because it requires estimating specific temperature values with an advance that would allow the operator to act if needed. Additionally, such a detail in the prediction is unnecessary from a business-oriented perspective, since we only care about whether the temperature is within the safe boundaries or outside of them. Based on this motivation (and on preliminary experiments), most of our effort go into the design of classification models. For both scenarios, we adopted neural architectures due to their well-known performances, but the designed methodology is not limited to this class of algorithms.

For the regression task, the model is trained to predict the temperature value with a proper advance τ. The problem is formulated as learning a function $f : X^{(t)} \to Y^{(t+\tau)}$, where $X^{(t)} \in \mathbb{R}^m$ is the set of the input features, Y is the numerical target variable to predict with advance τ, and t is the timestep. We train the Neural Networks in a supervised fashion and thus we require the availability of a dataset $\mathscr{D} = \{x^{(i)}, y^{(i)}\}$ with $i = 0, \ldots, N$, where N is the dataset size, $x^{(i)}$ and $y^{(i)}$ are respectively the input and target features for the i-th sample.

Alternatively, we can formulate the task as a binary classification problem, solved again via supervised Machine Learning. To collect a labeled dataset, each entry corre-

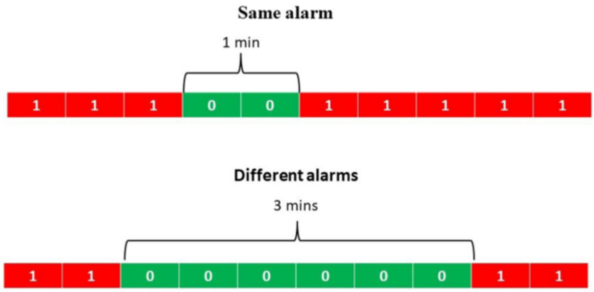

Figure 6. Assuming that the predictions frequency is 30 seconds, if we set the *alarm undershoot* equal to 3 minutes, the upper part of the figure shows an example of predictions that belong to the same alarm whereas, in the lower part, predictions belong to two different alarms.

sponding to the timestep t is labeled as anomalous if the temperature value exceeds the safe interval in at least one of the timesteps from $[t, t+\tau]$, otherwise it is considered a non-anomalous example. The binary classification problem requires learning a function $f : X \rightarrow Y$ where $X \in \mathbb{R}^m$ is the set of input features and Y is the probability value associated with the anomalous class. Also in this case we assume the availability of a dataset $\mathscr{D} = \{x^{(i)}, y^{(i)}\}$, similarly to the regression problem, except for the target which is 1 if the i-th example is labeled as anomalous and 0 otherwise.

If one opts for the sequence input, the input data will consists of $n \times w$ matrices, where w is the window length and m is the number of input features. This solution is viable with any class of predictive model, though it is especially well suited for approaches that can take advantage of the sequence information, such as 1-dimensional Convolutional Neural Networks (CNNs). For models designed to handle aggregated data, the resulting input consists of a vector $X \in \mathbb{R}^q$, where $q = m \cdot u$ and u is the number of computed statistics (e.g. mean or standard deviation) for each input feature.

4.3. Alarm Signal Generation

The overall goal of our system is to provide an alarm signal that is easily interpretable by the human operator on the field and that satisfies the business needs. An adequate alarm definition requires a few observations:

- Missed detections are expensive but also false alarms have a negative impact from a business perspective. The human operator should be alerted only when strictly required since discovering the reason for a potentially dangerous situation is an expensive operation. Moreover, frequent false alarms may lead to untrust the system.
- The technician is not continuously monitoring the alarm system hence a reasonable number of close in time predictions can be aggregated together to get a single signal.
- The anomaly detection advance should be sufficient to allow an effective repairing intervention.

We have devised an alarm generation method that transforms raw predictions in easily interpretable alarms. Despite we rely on binary classifier predictions, we can easily generalize the approach to the regression use case (e.g. applying a threshold to the model predictions). For this purpose, we introduced a set of parameters and operations:

- **Validation undershoot.** Single predictions are grouped to provide a coarser-grained *alarm* signal. This parameter defines the number of considered predictions when validating an alarm.
- **Threshold.** Required ratio of anomalous predictions in the same group (defined by the *validation undershoot*) to validate an *alarm*.
- **Anomaly undershoot.** Two anomalous temperature values that occur within this parameter are classified as the same anomaly. We can alternatively define the anomaly undershoot as the time interval that must occur between two anomalous temperature values to be classified as different anomalies.
- **Alarm undershoot.** As previously mentioned, the human operator is not continuously monitoring the system. When enough time occurs between two anomalous predictions, we assign them to different alarms. An illustrative example is shown in fig. 6.
- **Minimum advance.** An alarm is useful if the human operator has enough time to act on the facility. So we require a minimum advance between the alarm and the anomaly occurrence.
- **Blurred area.** Hard safe boundaries are required to label the data for the classification task but for the evaluation, we can be more flexible. During the evaluation, we excluded ground truth values that are within ΔT°C from the bounds because they are not serious anomalies and they may have a strong impact on the evaluation score.

5. Experimental results

We start by describing the dataset, and the training and evaluation procedures. We then show the limitations of a regression formulation of the problem that motivates the choice for a simpler classification approach. We then proceed by showing preliminary results focused on classical metrics. The section ends with the evaluation of the alarm signal using business-oriented metrics.

5.1. Dataset

We performed training and evalutation on real data obtained from sensors measurements installed on the facility. We considered historical data collected during the period 2018/10/01-2019/10/31, sampled with a frequency of 5 minutes. From the whole set, we then obtained 4 chunks of 10 months, sliding the beginning of the dataset of 1 month for each of them. This procedure allows us to assess robustness under deployment conditions: since we may observe a distributional shift in the data along time due to changes in the operational conditions of the plant, the model may need periodic retraining. Following domain experts, the plant conditions change relatively slowly and repeating the operation every month will be more than sufficient to prevent this kind of issue. For each dataset, we use the first 8 months for training and the remaining 2 months are equally and sequentially split between validation and test.

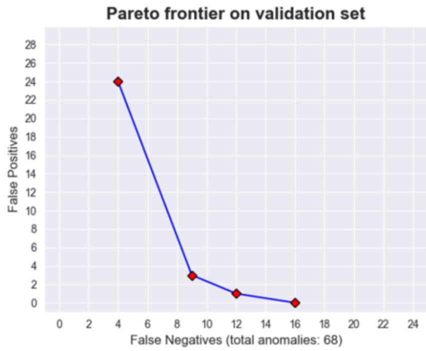

Figure 7. Example of a Pareto frontier that can be achieved during the evaluation. The red points are the Pareto optimal solutions because it does not exist any other solution that has a better value of either FP or FN.

The training set has 187 features, 65 of them are categorical and the remaining 122 are numerical. These features represent either the status or the measurements of the sensors installed on the facility. We started by dealing with missing data. We noticed that categorical features have a considerable amount of missing data. In addition, they have a quite stable trend and vary rarely in the considered period, making them practically useless for any predictive model. These reasons led us to discard them and focus on the only numerical features for the successive analysis. Since the number of missing values is just a few tens over thousands of dataset entries, we simply applied linear interpolation. During the time-lagged correlation analysis, we selected the only input features whose absolute value of the correlation is greater than 0.7 for at least one lag value. This allows us to reduce the number of input features from 122 to only 7. During the data aggregation of the preprocessing step, given the dynamics of the investigated plant and following the domain experts, we computed the mean, standard deviation, and the average of the difference between consecutive values, considering a time window of 20 minutes.

5.2. *Training and evaluation*

The evaluation relies on the alarm signal described in section 4. The validation, anomaly, and alarm undershoot are respectively 5, 10, and 5 minutes. We required a threshold of 1 and a minimum advance of 5 minutes. We excluded from the evaluation the temperature values in the range $[175, 180[$°C. In the remainder of the section, we will refer to true and false alarms, and missed anomalies respectively as True Positives (TP), False Positives (FP), and False Negatives (FN).

As previously mentioned, from a business perspective, both FP and FN are relevant and we want to jointly minimize them. A perfect model would not provide any FP and FN but this is almost impossible to achieve in practice. At the same time, it may be difficult to define apriori a good tradeoff between FP and FN and it is preferable to have a set of solutions from which the user can choose according to the business requirements. In this spirit, during the search for the best predictive model we have performed a *Pareto frontier analysis* rather than providing a unique optimal model and hyperparameter setting. An example of Pareto frontier optimal solutions is depicted in fig. 7.

Among all the hyperparameters, the threshold for binary classification has a strong impact on the Pareto analysis evaluation process. Increasing the threshold leads to a

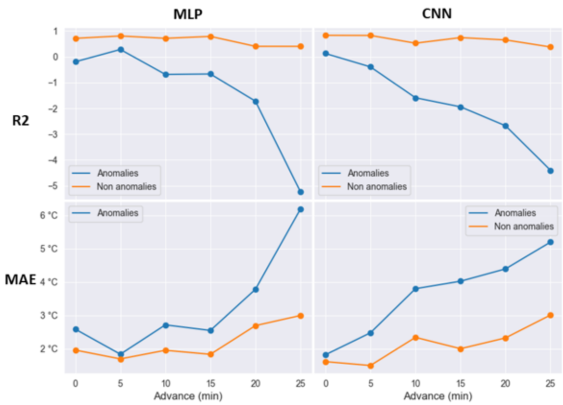

Figure 8. Average R2 score and the Mean Absolute Error for the MLP and CNN architectures on the 4 datasets.

decrease in the FP but, at the same time, we also increase the FN, so tuning the threshold value is required to achieve a good tradeoff.

We designed the training and evaluation procedure with the following steps:

1. We start by choosing a Machine Learning algorithm and identifying a set of hyperparameters to tune and their candidate values.
2. The dataset is randomly split between training, validation, and test sets.
3. For each candidate hyperparameter set, we train an instance of the model on the training set and compute the corresponding FP and FN on the validation set. The non-Pareto optimal candidate solutions are discarded.
4. For each remaining hyperparameter set, a new instance model is trained on the resulting dataset obtained appending the validation to the training set.
5. The final evaluation is performed on the test set and assesses the quality and robustness of the Pareto optimal solutions found during validation. The trained models are robust if the results on the test set are similar to the ones on the validation set.

The hyperparameter search led to relatively simple architectures described in the following. The MLP has three hidden layers with 24, 12, and 6 units, ReLU activation function, and L2-regularization. The CNN has a single convolutional layer with 8 filters, a kernel of size 3, and ReLU activation function, followed by two feedforward fully-connected layers with 12 and 6 units, ReLU activation, and L2-regularization. When employed in the binary classification task, we applied the sigmoid activation function at the output layer of both the neural architectures. We trained the Neural Networks for a maximum number of 100 epochs, with a batch size of 512 and the training is stopped if the loss function computed on a separated validation set is not improved after 3 epochs. The networks parameters are optimized with Adam optimizer and a learning rate of 0.001.

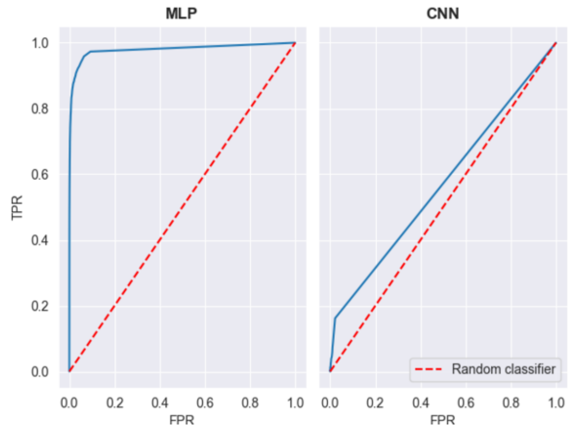

Figure 9. ROC curve for the binary classification problem.

5.3. Results

We start the section by showing preliminary results for a regression model that motivate our choice to focus on the classification approach. We evaluate the models considering predictions advances up to 25 minutes. Since the dataset is imbalanced, we compute the Mean Absolute Error (MAE) and R2-score considering separately anomalous and non-anomalous examples. The results of this evaluation are shown in fig. 8. The MLP and CNN accuracy gets quickly worse for anomaly ground truth examples when increasing the prediction advance. Since the regression model does not provide reliable estimates of the anomalous temperature values, we put our effort into the classification formulation of the problem.

A classical metric used to evaluate predictive models in binary classification problems is the Receiver Operating Characteristic (ROC) curve which reports on the x-axis the False Positive Rate (FPR) and on the y-axis the True Positive Rate (TPR). As shown in fig. 9, the MLP achieves close to ideal performances whereas the CNN is slightly better than a random classifier. Considering only this metric, one would discard the CNN as a non-viable solution because it is completely surpassed by the MLP.

The only ROC curve is not a suitable metric to evaluate the performance of the method for our use case: the user needs to find a tradeoff between reducing both the FP and FN and the solution has to satisfy the business requirements. For this reason, we designed a business-oriented validation framework that relies on the Pareto frontier analysis.

Since the threshold for binary classification has the strongest impact on FP and FN, we only focused our analysis on this parameter but, in principle, the evaluation can be extended to the whole set of hyperparameters. In the remainder of the section, we show the results for the neural architectures previously introduced, with threshold values generated in the range $[0.1, 0.9]$ with a step of 0.1. Since a unique evaluation metric is not viable for our task, we do not focus on the only FP and FN values but also the robustness of the predictive models: if the trained models are robust then the Pareto optimal solutions will provide similar results on the validation and test sets. To make a fair comparison between validation and test sets, FN and FP need normalization. We divide the FN by

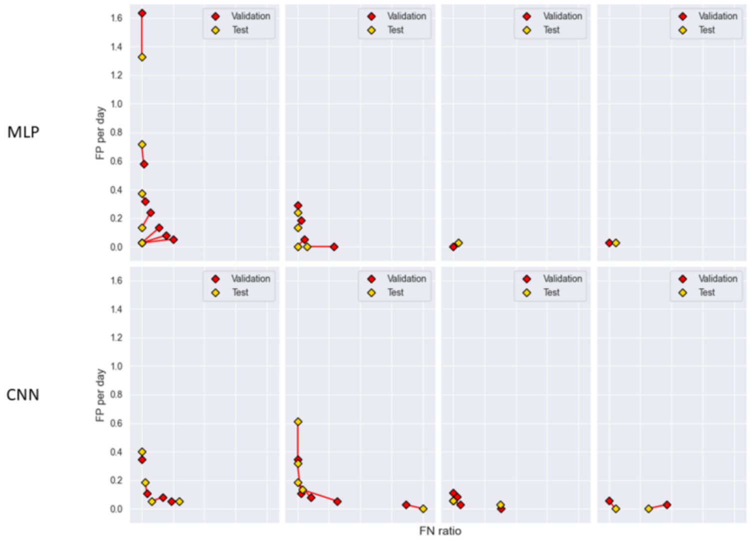

Figure 10. Pareto optimal solution computed on the validation set and then evaluated on the test set. Results are reported for 4 different datasets and for two kind of Neural architectures, namely the MLP and CNN.

the total number of anomalies and the FP by the total number of days, computed on the corresponding set, obtaining respectively the *FN ratio* and *FP per day*.

We report the results for the MLP and CNN on all the datasets in fig. 10. Red points are the Pareto optimal solutions values computed on the validation set whereas the gold ones are the corresponding values computed on the test set. Corresponding solutions values on validation and test sets and connected by a red line. The shorter the line the more robust are the results. As opposite to ROC curve, the results are quite similar, demonstrating that traditional metrics are not reliable for our problem. For both the models, accepting as few as 4 FP every 10 days we can obtain almost 0 FN. Despite this fact, the MLP is slightly better than the CNN: in almost all the cases the solution value on the test set is comparable to the one of the validation set, whereas for the CNN there are a few cases in which the solution is worsened from validation to test.

6. Related Works

Anomaly Detection is a research area that studies how to detect data points that are unlikely according to the data distribution. Several Machine Learning algorithms can solve Anomaly Detection tasks and with the recent outbreak of Deep Learning, Deep Neural Networks have provided state-of-the-art results in the field [1].

In recent years, the digital transformation and the availability of a large amount of data have favored the adoption of data analytic tools in the oil and gas industry [2]. For example, [3] highlights the advantages of employing data-driven methods to detect and prevent undesired operational conditions in an oil and gas producing facility. Since energy efficiency and environmental impact are key issues in the oil and gas industry, [4] shows the benefit of the adoption of a predictive model in forecasting the Stationary

Combustion CO2 Emission and supporting the site operators in taking the optimal actions. As we have pointed out in our work, to maximize the effectivenss of the method, a business-oriented evaluation is required. In [5] the authors make similar considerations and propose an evaluation procedure for Machine Learning models trained to predict peaks in the SO2 emissions of an oil and gas treatment plant.

7. Conclusions

In this work, we have developed a data-driven method to notify the site operator of anomalous operational conditions in an Oil Stabilization Facility. In particular, the user must be alerted if we expect the temperature oil at the bottom of the stabilization column to overcome safety boundaries over a defined time interval. We provided a detailed characterization of the plant data and designed an alarm detection system built on top of predictive models. The alarm is generated by transforming raw predictions and takes into account the final user requirements. Finally, we provided a business-oriented evaluation procedure that proved to be more reliable than traditional metrics.

Preliminary results demonstrated that Neural Networks, namely MLP and CNN, can be successfully applied to solve the task. From the Pareto frontier analysis, we obtained a set of optimal solutions that allows the final user to find the best tradeoff between missed and false detections. Moreover, we are currently evaluating and prototyping the overall pipeline of the method for the deployment on the real-world computing infrastructure.

Acknowledgements

The paper reports a subset of the results of the KINEMA project (Knowledge Integration in Neural networks for E-Maintenance), cofunded by the BI-REX Consortium within the Competence Center Program of MISE. The participants to the project include Bonfiglioli Riduttori S.p.A., Eni S.p.A., Aetna Group S.p.A., I.M.A. Industria Macchine Automatiche S.p.A., Marposs S.p.A., MindIT S.r.l., NIER Ingegneria S.p.A., ALASCOM srl, University of Bologna and the MechLav department of the University of Ferrara. The paper focuses on the results achieved for an Eni case study. The content, opinions and data reflect the point of view of the authors and do not necessarily reflect those of the BI-REX Consortium or MISE.

References

[1] Ruff L, Kauffmann JR, Vandermeulen RA, Montavon G, Samek W, Kloft M, et al. A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE. 2021.

[2] Pandey RK, Dahiya AK, Mandal A. Identifying applications of machine learning and data analytics based approaches for optimization of upstream petroleum operations. Energy Technology. 2021;9(1):2000749.

[3] Cadei L, Corneo A, Milana D, Loffreno D, Lancia L, Montini M, et al. A Business-Oriented Framework to Evaluate Advanced Analytics for Predictive Maintenance: Measuring Benefits-Effort Tradeoff. In: Abu Dhabi International Petroleum Exhibition & Conference. OnePetro; 2019. .

[4] Cadei L, Loffreno D, Camarda G, Montini M, Rossi G, Fier P, et al. A data driven management strategy to reduce the environmental impact of upstream production plants. In: SPE Norway One Day Seminar. OnePetro; 2019. .

[5] Cadei L, Montini M, Landi F, Porcelli F, Michetti V, Origgi M, et al. Big data advanced anlytics to forecast operational upsets in upstream production system. In: Abu Dhabi International Petroleum Exhibition & Conference. OnePetro; 2018. .