World Academy of Science, Engineering and Technology
International Journal of Mechanical and Industrial Engineering
Vol:16, No:3, 2022

Anomaly Detection in a Data Center with a Reconstruction Method Using a Multi-Autoencoders Model

Victor Breux, Jérôme Boutet, Alain Goret, Viviane Cattin

Open Science Index, Mechanical and Industrial Engineering Vol:16, No:3, 2022 publications.waset.org/10012429.pdf

***Abstract*—**Early detection of anomalies in data centers is important to reduce downtimes and the costs of periodic maintenance. However, there is little research on this topic and even fewer on the fusion of sensor data for the detection of abnormal events. The goal of this paper is to propose a method for anomaly detection in data centers by combining sensor data (temperature, humidity, power) and deep learning models. The model described in the paper uses one autoencoder per sensor to reconstruct the inputs. The auto-encoders contain Long-Short Term Memory (LSTM) layers and are trained using the normal samples of the relevant sensors selected by correlation analysis. The difference signal between the input and its reconstruction is then used to classify the samples using feature extraction and a random forest classifier. The data measured by the sensors of a data center between January 2019 and May 2020 are used to train the model, while the data between June 2020 and May 2021 are used to assess it. Performances of the model are assessed a posteriori through F1-score by comparing detected anomalies with the data center's history. The proposed model outperforms the state-of-the-art reconstruction method, which uses only one autoencoder taking multivariate sequences and detects an anomaly with a threshold on the reconstruction error, with an F1-score of 83.60% compared to 24.16%.

***Keywords*—**Anomaly detection, autoencoder, data centers, deep learning.

I. Introduction

OUTAGES and malfunctions of the equipment of a data center could lead to considerable damage and to severe consequences on the integrity of the center and of the hosted services. A recent example is the case of the fire in OVH data center in Strasbourg, France in March 2021 [1]. The cause of the fire is not precisely known but the first elements of the investigation tend to a fire from Uninterruptible Power Supply (UPS) systems. This fire caused a shutdown of major services for companies hosting their websites or online services on OVH servers. This event emphasizes the need of a proper and complete maintenance program for data centers.

The three types of maintenance are commonly distinguished: reactive, preventive (or periodic) and predictive.

Reactive maintenance consists of replacing or repairing a broken part of the system after an outage or a problem happened. In the case of a critical (or costly) piece of equipment, this approach may not be satisfying as it might lead to longer outages and higher maintenance costs. In these cases, preventive maintenance, which is a way to overcome the issue from the reactive one, is often preferred. However, it involves replacing a piece according to a predetermined timeline, without considering its wear condition, which may also lead to financial and environmental costs. Therefore, a balance between reducing the number of outages and reducing the costs of the maintenance program must be found.

Predictive maintenance performs as a kind of trade-off between preventive and reactive. The key aspect of predictive maintenance is to evaluate continuously the status of the equipment. With this approach, a piece of equipment is replaced only when it is considered as abnormal or faulty and just before a critical issue happens.

Predictive maintenance has its own drawbacks, as it requires additional sensors, leading to higher costs and potential sensor failures. It also relies on the hypothesis that the failure can be predicted with a low error margin.

Outages in data centers are extremely critical, as a data center has to be available at any time to perform computation and to answer queries and as it is composed of expensive devices. Therefore, a proper maintenance program is needed for these facilities. The current state of maintenance in a data center is preventive maintenance and redundancy. Redundancy is to ensure that even if something fails, a secondary system could take over it.

In 2016, Ponemon Institute and Emerson Network Power highlighted the main causes of outages in data centers [2]: 25% are due to UPS systems, 22% to cybercrime (denial-of-service attacks), 22% to accidental human error, 11% to air conditioning systems, 10% to the weather, 6% to electrical generators and 4% to IT machines.

This paper is focused on the predictive maintenance of UPS systems, as they are the first causes of outages in a data center.

This paper will describe a new method to perform predictive maintenance on the UPS system through anomaly detection. The proposed model relies on a sensor-wise reconstruction using auto-encoders and with a data fusion block based on a random forest whose sensitivity is adjusted with a threshold on the output probabilities.

This paper is organized as follows. The related work of anomaly detection is presented in Section II. Section III describes the data used and the architecture of the proposed model. Section IV explains in detail each part of the model and

Victor Breux, Jérôme Boutet, Alain Goret and Viviane Cattin are with CEA-Leti, Université Grenoble Alpes, F-38000 Grenoble, France (phone: +33438783857; e-mail: Jerome.boutet@cea.fr)

Open Science Index, Mechanical and Industrial Engineering Vol:16, No:3, 2022 publications.waset.org/10012429.pdf

Section V highlights the results obtained with the model and the comparison with others. Finally, the goal of Section VI and Section VII is respectively to discuss about the model and about the choices made and to talk about the improvements and the future work that can be made to improve this model, and conclusions are drawn in Section VIII.

II. Related Work

There are very few papers about anomaly detection applied to data centers. Decker et al. proposed a method to detect anomalies in the traffic of the data center by using log files and a Gaussian mixture-based rules [3]. Wang et al. described a second method based on the estimation of a density distribution [4].

Therefore, the research of related work is broadened to other application fields of predictive maintenance (turbofan engines, buildings, disk drives or elevators). The methods found in these applications can be separated in three classes: classical machine learning methods, supervised deep learning methods and unsupervised reconstruction methods.

In [5], the detection of an anomaly is done using a random forest classifier whose hyper-parameters are tuned to maximize the accuracy. The fitting of the forest is done by supervised learning. The random forest classifier is replaced by other machine learning classifiers in other papers such as Support Vector Machines (SVM), decision trees or k-nearest neighbors [6].

Other machine learning methods are trained using unsupervised learning on normal data (data labelled as normal), these methods are one-class classification methods. For example, a type of one-class classification method is density methods where the normal data are used to fit a probabilistic distribution by maximizing the likelihood as in [4], [7]. In these methods, an anomaly is detected if the likelihood of a sample with respect to the fitted distribution is low (below a fixed threshold). In [7], the one-class classification is done with a variant of SVM which is One-Class SVM. The One-Class SVM learns a boundary of the normal domain using the normal data, a sample that falls outside this domain is considered as abnormal.

The classical machine learning methods do not allow taking into account the temporal dependencies in the data.

That is why deep learning methods with more complex models are also used. In [8], [9], supervised deep learning methods are described, either using a Long-Short Term Memory (LSTM) [10] network that takes a sample and returns its status (normal or abnormal) or using another model made of convolutional layers. Both LSTM and convolutional layers allows catching the temporal information within the data. The model described in [11] is based on Convolutional Neural Network (CNN) and LSTM. The difference with other methods is that the feature extraction done by CNN is performed sensor per sensor. Meaning that we have one CNN per sensor to extract features from it. Then the extracted features of each sensor are concatenated and fed to a LSTM model. This sensor-wise approach is interesting for its modularity, one sensor can be added or removed without re-training all the CNNs.

An auto-encoder can be also used to reduce the dimension and to extract features from the data (auto-encoder trained by unsupervised learning). The supervised detection models are then fed with the latent vectors obtained with the encoder [12], [13].

Usually in anomaly detection, the dataset is highly imbalanced meaning that there are much more normal samples than abnormal one. The supervised methods need to have a balanced dataset, especially in deep learning methods that needs many data. So the imbalanced dataset has to be balanced by for example oversampling or undersampling or by attribute different weights to each class.

Finally, another kind of method for anomaly detection is reconstruction methods. The aim of reconstruction methods is to train the model to reconstruct the normal data of the dataset. The models learn the normal behavior of the system by unsupervised learning on the data labelled as normal. Once the model is fitted, a sample is detected as an anomaly if an error (based on the sample itself and its reconstruction) is above a fixed threshold. In [7], an auto-encoder is used to reconstruct the input and in [14], the auto-encoder is improved by using LSTM layers instead of fully connected layers. The error used for the detection of an anomaly is the reconstruction error, meaning either the L2-distance between the sample and its reconstruction or the Mean-Squared Error (MSE). But in [15], an error computed as a mix between the reconstruction error and an error computed in the latent space of the auto-encoder (for example Mahalanobis distance) is proposed to enhance the prediction. In [16], the authors described a reconstruction method based on Generative Adversarial Network (GAN) with convolutional layers. This model, called GANomaly, is made of a reconstructor composed of an encoder and a decoder. Another encoder is added to encode the reconstruction. The training is done as for GAN with a generator and a discriminator whose goal is to identify real sample to reconstructed ones. In this model, an anomaly is detected if the L2-distance between the outputs of the two encoders of the model is above a threshold. Finally in [17], the detection is also done using a threshold on an error computed with the reconstruction and the paper compares a Variational Auto-encoder model (VAE) with GANomaly [16].

Reconstruction methods have the benefits to be based on deep learning models such as LSTM and convolutional layers that can catch the temporal information in the time series. Furthermore, the training process using only normal data avoids the problem of the imbalance of the dataset where anomalies are far less represented than the normal behavior.

The method described in this paper belongs to the class of reconstruction methods. The reconstruction is done using auto-encoders with LSTM layers.

III. Materials and Method

The data used come from sensors set in one of the CEA's data centers in Grenoble. The measures of the sensors are taken every 5 minutes. As explained in the Ponemon Institute's study [2], most of the outages in a data center are due to UPS systems.

Open Science Index, Mechanical and Industrial Engineering Vol:16, No:3, 2022 publications.waset.org/10012429.pdf

Therefore, we decided to focus on UPS system and to perform anomaly detection only on this device.

The sensors taken from the UPS are:

- Internal temperature and battery temperature,
- Intensity on the three phases,
- Battery voltage,
- Battery capacity, and
- Total active power.

The data between January 2019 and May 2021 are used and will be annotated using the alarm history of the device given by the monitoring software of the data center. The history gives several time ranges during which the device sent an alarm (critical alarm or complete outages).

Finally, the dataset is highly imbalanced: during the 2.5 years measures from the UPS (corresponding to 253 968 measures), only 0.13% are abnormal.

The whole dataset is resampled by performing the following tasks for on the data of each sensor:

1. A resampling with a frequency of 5 minutes to get exactly one measure every 5 minutes for each sensor. If two measures are in the same 5 minutes time slot, only the first one is kept. Finally, if there is no measure in a 5 minutes time slot, a missing value is added.
2. Replacement of missing values by forward-fill.

With this resampling step, all sensors now have the same number of measures with the same timestamps.

The resampled data are then fed to the pipeline made of a pre-processing block and of the detection model.

The pre-processing part is made of four steps:

1. First, sensors are selected based on a correlation selection. If the correlation between the data of two sensors is above a fixed threshold (arbitrarily set to 85%), then only the first sensor is selected. This step allows to remove redundant information in the dataset.
2. After that, each sensor is scaled by a min-max normalization in order to have the same scale in the range [0, 1] for each sensor.
3. Finally, sequences are created by applying a sliding window on the data. The sliding window size (*win_size) is a parameter of the global workflow. The sliding window* moves forwards with a step of 5 minutes corresponding to a step of 1 measure. So from a dataset containing *n* measures, *n-win_size* sequences of size *win_size* are created.
4. The sequences are then labelled using the alarm history of the UPS (obtained with the monitoring software). Indeed, if in a sequence of size *win_size*, at least one measure was labelled as an anomaly in the history, then the sequence is labelled as abnormal.

After the pre-processing step, a new dataset composed of both the data sequences and the labels of each sequence is created and will be used by the detection model.

The pre-processed data passes through a sensor-wise reconstructor whose output is a reconstruction of the input sequences (based on auto-encoders). From the real data and their reconstruction, squared difference sequences are computed sensor per sensor from which statistical features are extracted. These extracted features will then be used to train in a supervised manner the prediction block that is based on random forest. A high-level view of the model is given in Fig. 1. Our method belongs to the class of reconstruction methods.

The development of the model was done with *Python 3.8*, the neural networks are made using *Keras* interface of *Tensorflow* and the random forest in the prediction block is done with *scikit-learn*. To extract some statistical features, we also used *scipy*.

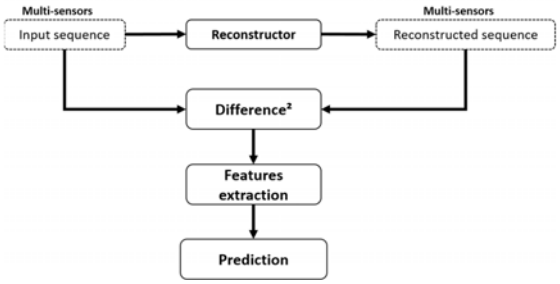

Fig. 1 Overview of the proposed model for anomaly detection

IV. Details of the Proposed Solution

A. Reconstructor Model

The goal of the reconstructor is to take the input sequence and to reconstruct it. The reconstructor is based on the auto-encoders model as in [14] with LSTM layers. Unlike other reconstruction methods, the proposed model uses one auto-encoder per sensor instead of one taking all the sensors as input, this multi auto-encoder approach is inspired by the multi-head CNN model in [11].

Each auto-encoder (AE) in the reconstructor is trained using only the data labelled as normal for the corresponding sensor in the dataset. The AEs are trained to minimize the Mean-Squared Error (MSE) between the input and the output (the reconstruction). The Adam optimizer [18] is used to learn the parameters of the AEs and the data are fed to it during training step by batches of size 64. In the Adam optimizer, the parameters are fixed to $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-7}$.

See Fig. 2, for an overview of the reconstructor model.

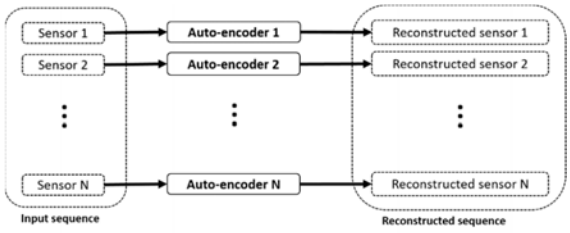

Fig. 2 Architecture of the sensor-wise Reconstructor

B. Architecture of the Auto-Encoders

Let us go further and describe the structure of the auto-encoders used in the Reconstructor.

The model needs to take into account the temporality of the sequences in input. That is why the AEs in the reconstructor are made with LSTM layers. LSTM is preferred to vanilla

Open Science Index, Mechanical and Industrial Engineering Vol:16, No:3, 2022 publications.waset.org/10012429.pdf

Recurrent Neural Network in order to avoid the problem of gradient vanishing for long sequences.

As in a classical auto-encoder, one part is the encoder that will match the input to the latent space. Then from the latent representation, the decoder will build a new sequence that is supposed to be close to the input.

The encoder is made of three LSTM layers, the last one returning the latent vector whose dimension (*latent_dim*) is fixed to half of the length of the input sequence (*win_size*).

As for the decoder, a RepeatVector layer is used to take the latent vector and to create a sequence of the desired length (the latent vector is repeated as many times as needed). This sequence composed of the repetition of the latent vector passes through two LSTM layers and finally through a Time Distributed Dense layer. Time Distributed Dense layer is a fully connected layer that is applied for every vector of the time series, for each timestep. Between each LSTM layer, a Dropout layer is added with a dropout parameter set to 0.2 to avoid overfitting.

The architecture is detailed in Fig. 3 (Dropout layers are not shown).

The activation function of the LSTM layers is *tanh*, and the linear activation function is used for the Time Distributed Dense layer.

In Fig. 3, the shape written just below the layers refers to the shape of the output of the corresponding layer. The first coordinate of the shape is the length of the sequence and the second one is the dimension of the vectors of the time series.

With that process, the auto-encoder is able to return a sequence having the same shape as the input and that minimizes the MSE with the input (reproduce the identity function).

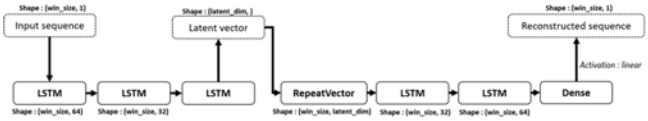

Fig. 3 Structure of the auto-encoders in the Reconstructor

1. Features Extraction

Once the Reconstructor has reconstructed the input sequence, the squared difference sequence is computed. Usually for an anomaly detection task, the Mean Square Error is computed between the input and its reconstruction. Here, we want to extract information sensor per sensor and not to have only the mean of the error sequence but other statistical features that can bring additional information.

Therefore, the aim of feature extraction and the prediction block is to do classical machine learning on the squared difference sequence using the reconstruction of the input. For each sensor, the difference sequences are just new sequences (of size *win_size)* which are the difference between the input sequence and its reconstruction (the output of the Reconstructor).

For each sensor, the following features are extracted from the squared difference sequence: min, quartiles, max, mean, standard deviation, skewness and kurtosis.

All these extracted features are then concatenated in a unique vector of size *n_features*n_sensors*, these vectors of features will then be fed to the prediction block.

So far, there is no data fusion in the model. The sensors are processed separately and no combination of sensors is used. The data fusion will be achieved within the prediction block.

2. Prediction Block

The previous block created a vector of features, which were extracted sensor per sensor from the squared difference sequence. The features vectors are used to train a Random Forest Classifier to detect whether there is an anomaly or not.

A random forest is used as it gives an insight on the features that were used to detect an anomaly and hence, we can determine the responsible sensors.

Compared to the state-of-the-art approach with a threshold set on the reconstruction error (MSE between the input and its reconstruction), here we do not only use the mean error and the thresholds on the features are not fixed manually but are found by the model to minimize the entropy of the predictions.

In the random forest, the number of features to consider at each node of the trees is set to the square root of the total number of features to add regularization. Furthermore, since the dataset is highly imbalanced, the forest is used in the "balanced_subsample" mode. It means that in each bootstrap created for the generation of the decision trees, weights will be applied on the minority class to balance the importance of the two classes in the bootstrap.

During the inference step, the random forest is used to predict the probability of a sample to be in the normal class (through the predict_proba method). If the probability of the normal class given by the forest is above a fixed threshold, then the sample is considered as normal by the model, otherwise, it is considered as an anomaly. The fixed threshold on the probability will be found to maximize the criterion used to assess the model (see the next section). This approach for the inference is equivalent to consider a sample as normal only if this sample is very likely to be normal, allowing reducing false negatives. See Fig. 4 for details on the prediction rule.

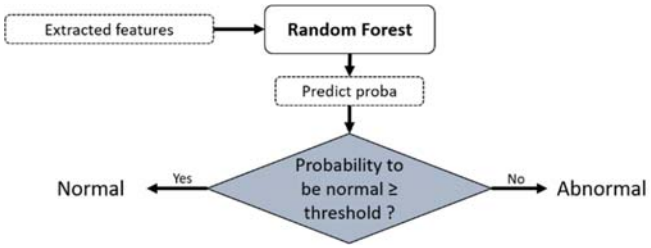

Fig. 4 Prediction block based on random forest classifier

III. Results

A. Anomaly Detection Results

To assess the performances of the model, we used the dataset presented in Section III (Materials and Method) of the paper.

The dataset is split in two parts, one part for training the model and the other for its evaluation. The data between January 2019 and May 2020 are used to train the model whereas

Open Science Index, Mechanical and Industrial Engineering Vol:16, No:3, 2022 publications.waset.org/10012429.pdf

the data between June 2020 and May 2021 are used for the assessment.

All steps of the pre-processing pipeline are also trained on the training set (correlation-based selection and min-max scaler). Among the sensors taken from the UPS system, six are selected by the correlation-based selector:

- Internal temperature,
- Battery temperature,
- Intensity on the first two phases,
- Battery voltage, and
- Battery capacity.

Since the dataset is highly imbalanced (much more normal samples than anomalies), the accuracy is not a relevant metrics. Indeed, even if the model classifies all the samples as normal, the accuracy will still be excellent as only a small proportion (the anomaly samples) will be misclassified.

That is why the model is assessed through F1-score metrics, which is the harmonic mean between precision and recall.

$$\text{F1-score} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

where:

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}}$$

TP refers to True Positive (anomaly well classified by the model), FP refers to False Positive (normal samples considered as abnormal by the model) and FN is False Negative (anomalies that are not detected by the model).

In the proposed model, one important parameter is the threshold that will be set on the probability of the normal class returned by the random forest. This threshold will be found by grid search on the interval [0.5, 1] to maximize the F1-score of the model on the test set.

Our model is compared with other existing reconstruction methods that were presented in the "Related work" section:

1. GANomaly [16].
2. A unique auto-encoder and the detection by a threshold on the reconstruction error as in [7], [14], [15].
3. A unique auto-encoder and the detection by a threshold on a mix between Mahalanobis distance in the latent space and reconstruction error as in [15].

The model is also compared to another model where the reconstruction is done sensor per sensor (as in ours) but where the detection is performed using a threshold on the reconstruction error (instead of machine learning on the difference sequence as in our model) (model denoted as Multi AE + reconstruction error). We add this model to the comparison to see the direct impact of the sensor-wise reconstruction on the F1-score compared to the reconstruction using only one auto-encoder.

For the four methods above, the threshold is set as the quantile of order q of the error on the training set. The order q of the quantile is found by grid search on the interval [0.99, 1] to maximize the F1-score on the test set. This interval has been chosen because an order q below 0.99 for the quantile gives a bad F1-score (near 0) as it leads to a lot of false positives.

Table I summarizes the score obtained for each method with the optimal thresholds and with the default number of trees in the random forest (100 trees) of our model.

The models have been tested for different sizes of rolling windows, since we have a measure every 5 minutes, a rolling window of size 12 represents a sequence of 1 hour. Mono-AE refers to a model where the reconstruction is done with a single autoencoder taking a multivariate time series as input, while Multi-AE refers to a reconstruction done with one auto-encoder per sensor (as in the proposed model).

TABLE I
COMPARISON OF F1-SCORE FOR DIFFERENT MODELS AND FOR DIFFERENT SIZES OF WINDOW

	6	12	24	48	72
Proposed model (ours)	**82.94%**	**65.40%**	**70.02%**	**55.56%**	**55.91%**
Multi-AE + reconstruction error	47.52%	31.90%	22.40%	22.29%	26.87%
Mono-AE + Mahalanobis distance and reconstruction error [15]	13.63%	18.93%	22.50%	15.83%	5.85%
Mono-AE + reconstruction error [7], [14]	24.16%	34.59%	21.64%	19.19%	20.36%
GANomaly [16]	38.84%	41.40%	30.80%	22.24%	17.06%
Random guess classifier	0.41%	0.47%	0.59%	0.81%	1.03%
Only majority class	0.93%	0.80%	0.65%	0.47%	0.36%

For a better reproducibility, all random seeds (tensorflow, numpy and random forest) have been set to 42.

Fig. 5 shows the impact on the threshold on the F1-score for the proposed model and for different sizes of window.

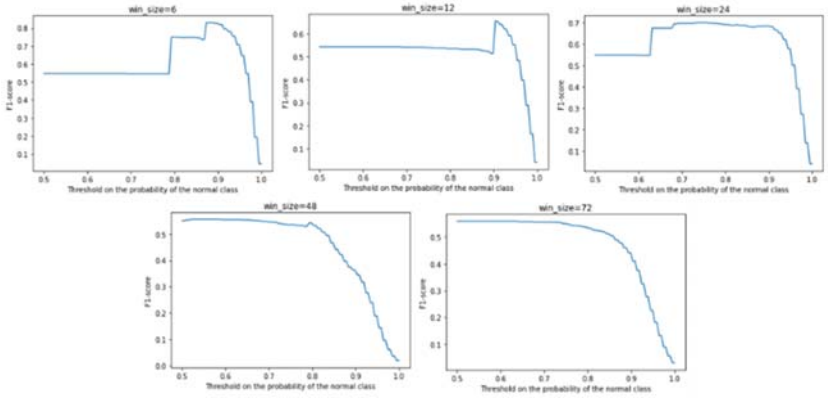

Fig. 5 Evolution of the F1-score of the proposed model for different sizes of window with respect to the value of the threshold set on the probability of the normal class

Open Science Index, Mechanical and Industrial Engineering Vol:16, No:3, 2022 publications.waset.org/10012429.pdf

The proposed model described outperforms the other reconstruction methods for every size of sliding window.

Choosing 6 as the size of the sliding windows gives the optimal results on the test set: 82.94% of F1-score (achieved with a threshold of 0.8737 and 100 trees).

Overall, a reconstruction with a multi auto-encoders model (one per sensor) seems to be more suited for anomaly detection than a reconstruction using a single auto-encoder. Even if GANomaly is supposed to deal with images, it outperforms the detection with a threshold on the reconstruction error with a single auto-encoder and competes with the multi-AE model with a threshold on the error while still being behind the proposed model (multi-AE + random forest on extracted features from the squared difference sequence).

Finally, for this dataset, the usage of Mahalanobis distance in the latent space does not improve the performances but worsen them.

So far, the random forest in our model is made of 100 trees leading to a F1-score of 82.94%. However, Fig. 6 highlights that F1-score can be slightly improved to 83.60% by using 170 trees in the random forest instead of 100. Above 170 trees, the score does not change.

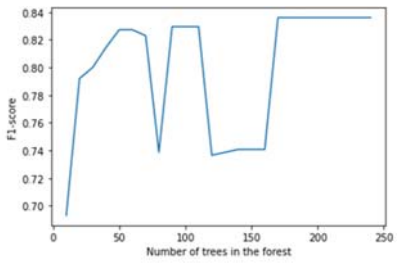

Fig. 6 Evolution of the F1-score of the model (with *win_size*=6 and *threshold*=0.8737) with respect to the number of trees in the random forest of the prediction block

Therefore, the best model for anomaly detection on this dataset is the model presented in this paper with a size of rolling window of 6, a threshold of 0.8737 on the probability of the normal class and 170 trees in the random forest classifier.

A. Comparison of Reconstruction Methods Based on AE

Finally, let us compare directly the reconstruction using a model with one auto-encoder per sensor and the reconstruction using a single auto-encoder to see the impact of the sensor-wise approach. We will compare them using the mean of the MSEs computed on the training set. The MSEs are computed on the training set because the second metrics used (Akaike Information Criterion) for the comparison has to be computed on the training set and not on the test set.

TABLE II
MEAN OF THE MEAN SQUARED ERRORS BETWEEN THE INPUTS AND THEIR RECONSTRUCTION ON THE TRAINING SET FOR DIFFERENT SIZES OF WINDOW

	6	12	24	48	72
Multi-AE	**$7.91*10^{-5}$**	**$8.16*10^{-5}$**	$4.05*10^{-4}$	$4.18*10^{-4}$	$4.11*10^{-4}$
Mono-AE	$1.09*10^{-4}$	$1.11*10^{-4}$	**$1.42*10^{-4}$**	**$2.10*10^{-4}$**	**$2.06*10^{-4}$**

Table II shows that the Multi-AE model performs a better reconstruction for small sequences while Mono-AE model seems to be better for longer sequences. However, each auto-encoder in the Multi-AE model has much fewer parameters than the model with a single auto-encoder. Therefore, we have a trade-off between performances (through MSEs) and complexity of the models (through the number of parameters).

This trade-off can be measured using Akaike Information Criterion (AIC) [19]. AIC is a criterion that considers both the performances of the model and its complexity. A model with a lower AIC is preferred.

In our case, AIC can be computed as follows:

$$AIC = n \ln(M) + 2p$$

Where n refers to the size of the training set and M to the mean of the MSEs on the training set (table) and p refers to the number of parameters.

TABLE III
AIC OF THE DIFFERENT RECONSTRUCTION METHODS BASED ON AUTO-ENCODERS FOR DIFFERENT SIZES OF WINDOW (LOWER IS BETTER)

	6	12	24	48	72
Multi-AE	**-694730.3**	**-676594.8**	**-412961.9**	**-346501.1**	**-273505.8**
Mono-AE	-517507.3	-469336.2	-325787.4	8213.9	364398.7

Open Science Index, Mechanical and Industrial Engineering Vol:16, No:3, 2022 publications.waset.org/10012429.pdf

The results of AIC are summarized in Table III. It is clear that the Multi-AE model performs a better trade-off for the reconstruction between performances (reconstruction error) and complexity (number of parameters) as it has a lower AIC. The Multi-AE model with a size of 6 for the sliding windows has the best value for AIC among all window sizes.

As a conclusion of this section, the proposed model that uses a sensor-wise reconstruction and data fusion using a random forest on the squared difference sequence outperforms the other anomaly detection models based on a reconstruction of the input. The sensor-wise approach of our model outperforms a reconstruction using a single auto-encoder in terms of balance between performances and complexity.

III. Discussion

For the Reconstructor block, unlike other state-of-the-art reconstruction methods, a sensor-wise approach has been implemented. This sensor-wise approach has two benefits:

1. A higher modularity if we add a sensor or remove one.
2. Having one auto-encoder per sensor allow a more accurate reconstruction of the sensor signal with less parameters in the auto-encoders. Each auto-encoder catching the specificity of each sensor. This is illustrated by a lower value of AIC compared to a model with a single auto-encoder (see Section V, Results).

In the prediction block, the purpose of the threshold on the probability of the normal class is to mitigate the number of false negatives and false positives (false alarms) returned by the model. In the case of maintenance, false negatives are damaging as it leads to some outages not being treated. However, too many false positives come with higher costs of the maintenance program. Therefore, the number of false negatives and false positives must be balanced. This balance is done in our model with the threshold on the probability of the normal class. This threshold is meant to adjust the sensitivity of the model to anomalies. One can change the value of the threshold to change the sensitivity and to have the desired balance between false negatives and false positives.

Finally, to summarize these choices, the novelty of the approach compared to the state of the art comes from four points:

1. The sensor-wise reconstruction;
2. The sensor-wise feature extraction that is not limited to only the computation of the MSE between a sequence and its reconstruction;
3. The usage of a random forest to consolidate the classification and to perform data fusion on the features extracted from each sensor; and,
4. A prediction made by the forest based on a threshold on the probability of the normal class given by the forest to adjust the sensitivity of the model.

An important part of the proposed model is its modularity, when we want to add a sensor or to remove one for the detection task. Thanks to their sensor-wise approach, the Reconstructor and the feature extraction blocks are modular. To add a new sensor to the model, it is only required to train a single new auto-encoder while all the others auto-encoders corresponding to the other sensors do not require retraining. For the feature extraction, this remark still holds, the features are extracted on each sensor independently so there is no problem if one is added or removed. Moreover, if we have new training data for a sensor, we could retrain the auto-encoder associated to this sensor with the new training set without retraining the entire Reconstructor block.

However, while the Reconstructor does not need to be retrained entirely if a sensor is added or removed (just an auto-encoder is trained), the random forest still needs to be retrained, as it will take new features as input.

IV. Future Work and Possible Improvements

One possible improvement would be to make the model robust to the outage of a sensor. Indeed, the model can be improved by still giving a prediction even if one sensor is missing. Currently, if a sensor is missing, the model detects an anomaly but the ideal scenario would be that the model uses also the available sensors to make a prediction.

The Reconstructor part and the features extraction of the proposed model are already robust to the lack of a sensor since the reconstruction and the extraction are done sensor per sensor. However, the prediction block with the random forest is not.

One possible way to overcome this problem would be to create and train as many random forests as there are combinations of sensors (so $2^{nb\_sensors}$ forests). In case of a sensor outage, the model will make the prediction using the forest corresponding to the combination of the currently available sensors.

This model may also be applied to other applications of predictive maintenance. In this paper, we focused on the UPS system in the data center and the model was developed and tested using only the data from this device. The proposed model may be retrained and tested on the data coming from the air conditioner systems for example, as they are a great cause of outages in a data center.

V. Conclusion

In this paper, we presented a new model for anomaly detection of a UPS in a data center. The model uses a reconstruction method based on auto-encoders model. The reconstruction is done using one LSTM-auto-encoder per sensor for better modularity. The detection is then performed using a random forest trained on statistical features extracted from the squared difference sequence between a sequence and its reconstruction and using a threshold on the probability of the normal class to make the model more sensitive and avoid false negatives.

The proposed model outperforms other reconstruction methods on the dataset we created with the sensors in the CEA's data center.

In particular, our method outperforms a reconstruction method that uses a single auto-encoder for the reconstruction and a threshold on the reconstruction error for the prediction.

Open Science Index, Mechanical and Industrial Engineering Vol:16, No:3, 2022 publications.waset.org/10012429.pdf

REFERENCES

[1] Strasbourg datacentre: latest information, *Strasbourg datacentre: latest information*. https://www.ovh.com/world/news/press/cpl1787.fire-our-strasbourg-site

[2] Average Cost of a Data Center Outage, *Data Center Frontier*, janv. 25, 2016. https://datacenterfrontier.com/average-cost-of-a-data-center-outage/

[3] L. Decker, D. Leite, L. Giommi, et D. Bonacorsi, « Real-Time Anomaly Detection in Data Centers for Log-based Predictive Maintenance using an Evolving Fuzzy-Rule-Based Approach », *2020 IEEE Int. Conf. Fuzzy Syst. FUZZ-IEEE Fuzzy Syst. FUZZ-IEEE 2020 IEEE Int. Conf. On*, p. 1-8, juill. 2020, doi: 10.1109/FUZZ48607.2020.9177762.

[4] L. Wang *et al.*, « Anomaly monitoring in high-density data centers based on gaussian distribution anomaly detection algorithm », *2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. AEECA Adv. Electr. Eng. Comput. Appl. AEECA 2020 IEEE Int. Conf. On*, p. 836-841, août 2020, doi: 10.1109/AEECA49918.2020.9213549.

[5] C.-J. Su et S.-F. Huang, « Real-time big data analytics for hard disk drive predictive maintenance », *Comput. Electr. Eng.*, vol. 71, p. 93-101, oct. 2018, doi: 10.1016/j.compeleceng.2018.07.025.

[6] M. Cakir, M. A. Guvenc, et S. Mistikoglu, « The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system », *Comput. Ind. Eng.*, vol. 151, janv. 2021, doi: 10.1016/j.cie.2020.106948.

[7] F. l. f. Pereira, D. n. Teixeira, J. p. p. Gomes, et J. c. Machado, « Evaluating One-Class Classifiers for Fault Detection in Hard Disk Drives », *2019 8th Braz. Conf. Intell. Syst. BRACIS Intell. Syst. BRACIS 2019 8th Braz. Conf. BRACIS*, p. 586-591, oct. 2019, doi: 10.1109/BRACIS.2019.00108.

[8] H. V. Dudukcu, M. Taskiran, et N. Kahraman, « LSTM and WaveNet Implementation for Predictive Maintenance of Turbofan Engines », *2020 IEEE 20th Int. Symp. Comput. Intell. Inform. CINTI Comput. Intell. Inform. CINTI 2020 IEEE 20th Int. Symp. On*, p. 000151-000156, nov. 2020, doi: 10.1109/CINTI51262.2020.9305820.

[9] A. P. Hermawan, D.-S. Kim, et J.-M. Lee, « Predictive Maintenance of Aircraft Engine using Deep Learning Technique », *2020 Int. Conf. Inf. Commun. Technol. Converg. ICTC Inf. Commun. Technol. Converg. ICTC 2020 Int. Conf. On*, p. 1296-1298, oct. 2020, doi: 10.1109/ICTC49870.2020.9289466.

[10] S. Hochreiter et J. Schmidhuber, « Long Short-Term Memory », *Neural Comput.*, vol. 9, nº 8, p. 1735-1780, nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[11] M. Canizo, I. Triguero, A. Conde, et E. Onieva, « Multi-Head CNN-RNN for Multi-Time Series Anomaly Detection: An industrial case study », *Neurocomputing*, janv. 2019, doi: 10.1016/j.neucom.2019.07.034.

[12] M. Alrifaey, W. h. Lim, et C. k. Ang, « A Novel Deep Learning Framework Based RNN-SAE for Fault Detection of Electrical Gas Generator », *IEEE Access Access IEEE*, vol. 9, p. 21433-21442, janv. 2021, doi: 10.1109/ACCESS.2021.3055427.

[13] S. U. Jan, Y. D. Lee, et I. S. Koo, « A distributed sensor-fault detection and diagnosis framework using machine learning », *Inf. Sci.*, vol. 547, p. 777-796, févr. 2021, doi: 10.1016/j.ins.2020.08.068.

[14] Yassine Bouabdallaoui, Zoubeir Lafhaj, Pascal Yim, Laure Ducoulombier, et Belkacem Bennadji, « Predictive Maintenance in Building Facilities: A Machine Learning-Based Approach », *Sensors*, vol. 21, nº 1044, p. 1044-1044, févr. 2021, doi: 10.3390/s21041044.

[15] F. L. F. Pereira, I. Castro Chaves, J. P. P. Gomes, et J. C. Machado, « Using Autoencoders for Anomaly Detection in Hard Disk Drives », *2020 Int. Jt. Conf. Neural Netw. IJCNN Neural Netw. IJCNN 2020 Int. Jt. Conf. On*, p. 1-7, juill. 2020, doi: 10.1109/IJCNN48605.2020.9206689.

[16] S. Akcay, A. Atapour-Abarghouei, et T. P. Breckon, « GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training », *ArXiv180506725 Cs*, nov. 2018, [En ligne]. Disponible sur: http://arxiv.org/abs/1805.06725

[17] H. Ahn, D. Jung, et H.-L. Choi, « Deep Generative Models-Based Anomaly Detection for Spacecraft Control Systems », *Sensors*, vol. 20, nº 7, avr. 2020, doi: 10.3390/s20071991.

[18] D. P. Kingma et J. Ba, « Adam: A Method for Stochastic Optimization », *ArXiv14126980 Cs*, janv. 2017, [En ligne]. Disponible sur: http://arxiv.org/abs/1412.6980

[19] H. Akaike, « Information Theory and an Extension of the Maximum Likelihood Principle », in *Selected Papers of Hirotugu Akaike*, E. Parzen, K. Tanabe, et G. Kitagawa, Éd. New York, NY: Springer, 1998, p. 199-213. doi: 10.1007/978-1-4612-1694-0_15.