Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

A SCALABLE DEEP LEARNING-BASED APPROACH FOR ANOMALY DETECTION IN SEMICONDUCTOR MANUFACTURING

Simone Tedesco
Gian Antonio Susto

Department of Information Engineering
University of Padova
Via Gradenigo 6/B, 35131 Padova, ITALY

Natalie Gentner
Andreas Kyek
Yao Yang

Infineon Technologies AG
Am Campeon 1-15
85579 Neubiberg, GERMANY

ABSTRACT

The diffusion of the Industry 4.0 paradigm lead to the creation and collection of huge manufacturing datasets; such datasets contain for example measurements coming from physical sensors located in different equipment or even in different productive manufacturing organizations. Such large and heterogeneous datasets represent a challenge when aiming for developing data-driven approaches like Anomaly Detection or Predictive Maintenance. In this work we present a new approach for performing Anomaly Detection that is able to handle heterogeneous data coming from different equipment, work centers or production sites. The proposed approach exploits Deep Learning architectures: Autoencoders are employed to derive a 'normal' behaviour of the system under exam that is then used for comparison when monitoring new data. The main strength of the proposed approach is its generality and scalability: the effectiveness of the proposed approach is demonstrated through experiments performed on real world data extracted from different workcenters in semiconductor manufacturing facilities.

1 INTRODUCTION

The discovery of new technologies has guided industry development from the early adoption of mechanical systems, to today's fourth industrial revolution. Modern factories are capable of providing a high volume of data due to large availability and affordability of sensors, data acquisition systems and computer networks. This always increasing availability of data is leading to the so called Big Data era. One main goal of Industry 4.0 is to extract value from the available data. To this aim, Machine Learning (ML) has gained growing attention by industries for extracting valuable information by means of statistical predictive models trained on historical process data. One of the most important ML tasks in the field of Industry 4.0 is anomaly detection (AD). AD (or outlier detection) consists in the process of identifying items, events or observations that deviate from what is standard, normal or expected. The importance of AD in manufacturing is due to the fact that industrial equipment and processes suffer from degradation due to the continuous and intensive usage. Unacceptable degradation need to be promptly detected to prevent further losses and defects. Due to the already mentioned huge availability of data collected in modern industries during the entire production process and the many effective unsupervised AD approaches that do not required label data, AD has enormous applicability and potential in industrial scenarios.

1.1 Autoencoder-Based Anomaly Detection

An autoencoder (AE) is a type of neural network that performs hierarchical and nonlinear dimensionality reduction of the data in order to reconstruct the input as closely as possible; the AE tries to minimize the reconstruction error between its input and the reconstructed output. Since AE creates a reduced representation of the data and reconstruct the input based on that lower dimensional space representation, it is widely adopted for discovering outliers. The key idea behind AE-based anomaly detection is to measure how far the data reconstructed by the trained AE are from the input data. Taking into consideration that the AE objective is to perfectly reconstruct the input, it is possible to use the distance between input data and reconstructed data (i.e. the reconstruction error) as anomaly score (Karadayi et al. 2020). For an effective outlier detection, the AE training data set has to be cleaned as much as possible from noise and outliers through an extensive preprocessing step. If this preprocessing step is not effective, the autoencoder learns to replicate also the outliers, and obviously this is not useful for solving the AD task. The approach to autoencoder-based anomaly detection can be summarized in the following steps and it is depicted in Fig. 1.

1. **Data Collection**: data are collected during normal operating conditions of the equipment;
2. **Preprocessing**: data cleaning process that has the goal of removing noise from data and making data ready for the Autoencoder Training process;
3. **Autoencoder Training**: the AE is trained using the cleaned data in order to learn and identify a behaviour considered normal;
4. **Anomaly Detection**: detect anomalies evaluating the reconstruction loss as anomaly score of the data on the trained AE using Statistical Process Control (SPC).

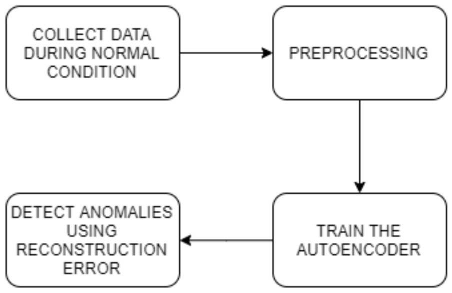
COLLECT DATA DURING NORMAL CONDITION
PREPROCESSING
TRAIN THE AUTOENCODER
DETECT ANOMALIES USING RECONSTRUCTION ERROR

Figure 1: Autoencoder-based anomaly detection approach.

In this work, we will adapt this pipeline by characterizing each step in a suitable way for dealing with semiconductor manufacturing data: preprocessing will be described in Section 2 while autoencoder training and evaluation will be discussed in Section 3 and Section 3.3.

1.2 Contribution of the Work

The key goal of the paper is to design an AD approach that could be scalable to different heterogenous conditions; this is the typical scenario for example in complex manufacturing production (like in semiconductor manufacturing) where different equipment, work centers or even production sites may be considered when deploying an AD module. The main challenges addressed here are paradigmatic of the industrial framework:

- the considered problem is a multivariate and unsupervised problem: typically in industrial dataset there is the presence of a huge number of features and the lack of labels that confirm whether the designed approach is working well or not;
- industrial data is typically very complex: the presence of several context and problems in industrial dataset, such as recipe effect or the lack of measurements, that can affect the reconstruction capabilities of an autoencoder;

- dataset extracted from different production areas are very different among them based on the performed physical process, available machine types and installed sensors besides others.

The designed method is compared with other existing methods, in order to understand its strengths and weaknesses. The designed approach reveals itself to be very robust in terms of scalability: in fact, the reconstruction of a normal behaviour works very good for data extracted from very different workcenters, confirming in this way the generalization capabilities of the developed approach.

1.3 Related Works

AS has applications in many areas, even outside manufacturing: the topic is well described in many works, such as (Chandola et al. 2009) and (Chalapathy and Chawla 2019).The use of replicator neural networks, that are networks able to reconstruct the normal behaviour of the data, in AD for multivariate frameworks has been investigated for example in (Dau et al. 2014), (Hawkins et al. 2002). Moreover, in (Zenati et al. 2018), an efficient Generative Adversarial Network-based approach for AD is described. Different type of AE are investigated for AD applications. For example, in (Zhou and Paffenroth 2017), a deep robust variant of AE is implemented and tested. The use of variational AEs (VAEs) is investigated in (An and Cho 2015), (Zhang and Chen 2019) (in particular for time-series), (Pol et al. 2019) (conditional variational AEs). In the context of semiconductor manufacturing, AD approaches have been presented in (Susto et al. 2017), (Susto et al. 2017) for tabular data and in (Puggini and McLoone 2018), (Maggipinto et al. 2019) for 2-dimensional data. To the best of our knowledge, no work focused on scalability/generalization of AD in the context of semiconductor manufacturing have been previously proposed.

2 DATASET PREPROCESSING

The goal of the preprocessing steps is to reduce the dimension of the dataset, by removing features that do not contain any information or containing only information already covered by other features present in the dataset, in order simplify the creation of a model that correctly represents the data. Three types of features are removed in the presented preprocessing steps:

1. Constant columns;
2. Highly correlated columns;
3. Noisy columns.

It is important to remark that this process has to be very general (ie. able to handle different types of variability) and highly automated in order to match the generality requirement of the designed approach. The three simple steps of pre-processing are commented in the following. Additionally a step of *feature scaling* is in place in the preprocessing phase.

2.1 Constant and NaN Columns

The first step in dataset manipulation is the removal of columns that contain only identical values or NaN values. This step is motivated by the fact that these types of columns do not contain any information that is useful in the modeling step. In fact, the key goal of an AD algorithm is to detect variations from the normal behaviour. For this reason, a constant column is not useful in order to achieve this goal.

2.2 Feature Correlation

The second step performed in dataset manipulation is the removal of highly correlated features. This step is very useful in order to reduce the data dimensionality by removing features that do not contain additional information with respect to other features that will be already present in the manipulated dataset. Reducing the number of correlated features present in the dataset can have an important impact on model accuracy by reducing the possibility to encounter overfitting. It is worth noting that the removal of one correlated feature in place of another can have an impact on the modelization step; nevertheless, it has been decided to define a selection criterion that could be exploited automatically, keeping the variable associated with the lower index and removing the correlated variables associated with higher indices. A more elaborate solution in order to perform highly correlated features removal is reported in (Yu and Liu 2003).

2.3 Noisy Features Removal

This step of the dataset manipulation is important in order to remove those features which do not contain any kind of information apart from noise. Such steo will make the creation of the monitoring model more stable and more accurate. This step is performed by using the Kolmogorov-Smirnov (KS) test named after Andrey Kolmogorov and Nikolai Smirnov and presented in (Kolmogorov 1933). The KS test is a non-parametric and distribution-free test: this means that it does not make assumptions about the probabilistic distribution of data. This fact is very important in the considered approach because, in this framework, it is not possible to know the noise distribution a priori. KS test has the goal of rejecting the null hypothesis: this means that KS test aims at rejecting the possibility that two data samples come from the same statistical distribution. The two-samples KS statistic quantifies a distance between the empirical distribution functions of two samples: in this step, the two-samples KS test is used in order to compare two data samples. The process to discard noisy columns is described as follow:

1. A reference distribution d_0 is computed from the feature data;
2. A distribution d_i is computed from a batch of N sub-sequential elements, with $i = 1, 2, \ldots, \lfloor \frac{n}{N} \rfloor$, where n is the amount of observations in the data at hand;
3. One two-samples KS test is performed between d_0 and d_i, while another one is performed between d_{i-1} and d_i;
4. If one of the two tests fail (the p-value returned by the KS test is below a given threshold t) for more than τ times, the column is discarded.

In our work we choose the value of N in order to have $\lfloor \frac{n}{N} \rfloor = 100$.

2.4 Data Scaling

Feature scaling is a vital element of data preprocessing for machine learning: implementing the right scaler is equally important for precise foresight with machine learning algorithms. The necessity of feature scaling is given by the presence of values that could be of different orders of magnitude among them. This large numeric difference on the values would be a very strong impact on the performance of a machine learning algorithm because it works on the numbers and does not know what that given number represents, giving in this way more importance to large number. Hence, there is the necessity of bringing all the features to the same standing. As already stated, the action of scaling industrial data can be often very tricky, due to their complexity. There are some phenomenon, for example the Recipe Effect that will be described in Section 4, that has be to taken into account in order to perform an adequate scaling process. Hence, the scaler for this type of data has to be designed very carefully to obtain the advantages that the scaling process has on the machine learning algorithm. It is worth noting that a final clipping in the interval $[-1, 2]$ is performed in the scaler. Finally, the dataset is splitted into training set and test set in order to use theses sets for the choice of a proper Autoencoder architecture. The percentages of data used in the splitting process are 90% of data for the training set and 10% of data for the test set. The training set is then randomly shuffled in order to cancel every possible time dependency in the data.

3 AUTOENCODER ARCHITECTURE

An autoencoder (AE) is a feed-forward neural network in which the desired output is the input itself. AE is able to tackle unsupervised tasks, and for this reason are strongly used for anomaly detection. The typical AE structure is subdivided in three main parts:

1. **Encoder:** neural network that has the goal to learn how to encode the original data in a low-dimensional representation;
2. **Bottleneck Layer:** the layer that contains the coded reduced representation;
3. **Decoder:** neural network that has the goal to reconstruct from the bottleneck representation the original data in the best possible way.

The AE goal is to efficiently encode data and from this reduced representation generate a reconstruction that is as close as possible to the original input data. In other words, the key objective of an AE is typically to perform dimensionality reduction, trying in this way to discard the noise in the original signal. The milestone paper (Hinton 2006) explains how high-dimensional data can be converted to low-dimensional codes using AE. If linear activation functions are used in the AE, it becomes virtually identical to a simple linear regression model or PCA/matrix factorization model. When a nonlinear activation function, such as rectified linear unit (ReLU) or a sigmoid function, is introduced in the multi-layer AE architecture the autoencoder goes beyond linear models, capturing multi-modal aspects of the input distribution. It is shown that carefully designed autoencoders with tuned hyperparameters outperform PCA or K-Means methods in dimensionality reduction and characterizing data distribution (Vincent et al. 2010). AEs are also more efficient in detecting anomalies than linear PCAs and in computation cost than kernel PCAs (Sakurada and Yairi 2014). It is important to remark that AE have the following important properties:

- **Lossy**: the AE reconstructed output will be a degraded representation of the input;
- **Data-specific**: AEs are only able to meaningfully reconstruct data similar to the training data. Indeed AE learns latent features specific for the given training data;
- **Unsupervised**: this important property is due to the fact that for training an AE labelled training data is not necessary.

Given the goal of AE of reconstructing the input, the autoencoder training objective is to minimize the reconstruction loss. This loss allows the measure of how close the output is to the input data. The cost function used in this work for training and evaluating an AE is the Mean Squared Error (MSE):

$$\mathscr{L}(X,\hat{X}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2, \tag{1}$$

where $X = \{x_1, ..., x_n\}$ is the input vector, $\hat{X} = \{\hat{x}_1, ..., \hat{x}_n\}$ is the output vector reconstructed from the AE and n is the number of datapoints. The optimizer used in training the autoencoder is the Adam optimizer. Adam, or adaptive moment estimation, optimization algorithm is an improvement of stochastic gradient descent (SGD), introduced in (Kingma and Ba 2017). Moreover, the Early Stopping strategy (Prechelt 1996) is introduced in order to perform regularization and therefore avoid overfitting.

3.1 Activation Functions

In the presented work, the activation function implemented for all the layers of the AE with the exception of the output layer is the *LeakyReLU* activation function σ_{LR}, whose formula is reported in (2):

$$\sigma_{LR}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}, \tag{2}$$

The use of *LeakyReLU* instead of the standard *ReLU* is motivated by the fact that with *LeakyReLU* the dying ReLU problem is not present. The use of gradient-based optimization is inevitable in training deep neural networks. The dying ReLU is a kind of vanishing gradient, which refers to a problem that occurs during the optimization process when ReLU neurons become inactive and only output 0 for any input, making the training process not useful. In the output layer, a modified hyperbolic tangent activation function is used. The equation of this activation function is reported in (3):

$$\sigma_{ModHT}(x) = \begin{cases} 2 * \sigma_{HT}(x) & \text{if } x \geq 0 \\ \sigma_{HT}(x) & \text{otherwise} \end{cases}, \quad \text{with } \sigma_{HT}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{3}$$

The use of this activation function is motivated by the data scaling process.

3.2 Tournament Search Architecture Optimization

In order to make the approach general to different sizes of dataset, a tournament search AE architecture optimization is introduced. This optimization algorithm has the goal to automatically create autoencoders of different depth and size with respect to the different dataset considered. This kind of approach is very important in terms of generality because dataset extracted from different workcenters are different in terms of size. This fact is caused by the presence of a different number of sensors for each workcenter. The implemented optimization approach is a Tournament Search optimization, introduced in (Miller and Goldberg 1995) and (Blickle and Thiele 1995). The optimization of the architecture of a neural network is an integer optimization problem, in fact all the parameters, depth and number of neurons per layer, are integers. Such problems can be solved by integer programming, or by a metaheuristic approach. Genetic algorithms can be classified as metaheuristics methods, and will be used in this work to solve the architecture optimization problem presented. A key advantage of genetic algorithms is the ability to handle hard problems, problems that don't have a derivative, and the simplicity of the implementation in some cases. However, the biggest disadvantage is that genetic algorithms do not guarantee that an optimal solution is ever found. Tournament search is a genetic algorithm that consists in the generation of a set of possible solutions: each solution is tested and, taking into account a proper score, the best one is selected.

3.2.1 Algorithmic Implementation

The algorithm designed to perform the tournament search architecture optimization takes as input the list of possible depths, the list of possible dimension of the hidden layers, the training set on which training each AE generated, and the test set, used to give the score. It is important to note that from the training set it is possible to select the number of input neurons of the autoencoder (i.e. the number of features p of the considered dataset). It is worth noting that the training process of each autoencoder is performed using Adam as optimizer, the Mean Squared Error between the input data and the reconstructed data as loss function, and early stopping, in order to prevent the overfitting phenomenon. The Tournament Search optimization algorithm is composed of the following five key steps:

1. Truncation of the list of possible dimensions to a list of values smaller than p, where p is the number of input neurons;
2. Generation of all the possible combinations of dimensions for the given depths, taking into account the number of input features and a certain degree of minimum compression (i.e. the bottleneck dimension of the Autoencoder must be smaller than $0.5 * p$);
3. For each combination, creation and training of an Autoencoder (using ADAM, Mean Squared Error, and Early Stopping) on the given training set (90% of initial data);
4. The loss on the test set (10% of initial data) is used as score to compare all the generated architectures;
5. Finally, the best combination in terms of this score and, in case of tie, the simplest one, is chosen for creating the final model.

3.3 Anomaly Score and Monitoring via Control Charts

After the normal behaviour reconstruction performed by the Autoencoder, Statistical Process Control (SPC) techniques can be used in order to understand whether a datapoint is considered anomalous or not. Key elements in SPC are the Control Limits, typically an Upper Control Limit (UCL) and a Lower Control Limit (LCL): if a data point lies within the control limits, it is considered as an inlier.

In the designed approach, a Control Chart is used to monitored the Anomaly Score computed by the AE: for each data point the Anomaly Score is defined as the reconstruction loss, i.e. the Mean Squared Error, between the real data point value and the value reconstructed by the AE is the Anomaly Score. In this context, the Upper Control Limit is also called Anomaly Threshold, while no Lower Control Limit is defined. The strategy implemented in the Control Limits computation are robust against outliers and take into account the skewness of data, improving in this way the quality of the Anomaly Detection approach. In fact, if a dataset contains outliers, the mean and the standard deviation are strongly biased. A complete description of the improvements and the strategies that SPC gives to industrial manufacturing is reported in (Montgomery 1997). The Upper Control Limit (UCL) is computed as

$$UCL = Q_{99.865\%}, \tag{4}$$

where Q_i is the i-th quantile, for a *ROBUST3SIGMA* strategy or using:

$$UCL = Q_{99.9997\%}, \tag{5}$$

for a *ROBUST4.5SIGMA* strategy. In Figure 2 the application of SPC in the definition of the Anomaly Score limits is reported.

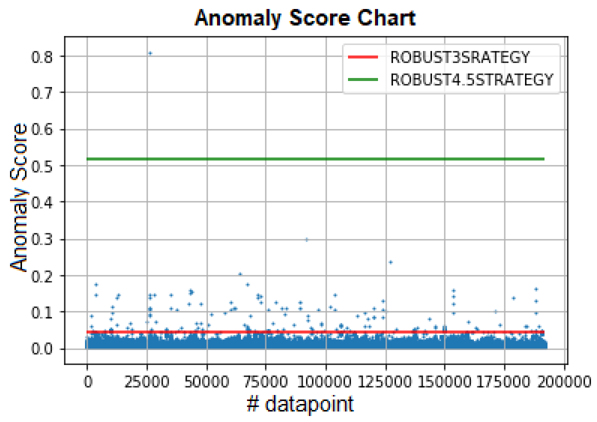

Figure 2: Anomaly Scores chart with UCLs.

4 EXPERIMENT #1: DEALING WITH RECIPE EFFECT

As a first case study we considered data coming from an ion-implantation workcenter: ion implantation is the front-end process where wafers are doped using an high energy electron beam. The considered dataset has 198434 samples, each one composed of 167 features: each feature represents the measurements performed by a given sensor present in the workcenter.

The first experiment is focused on evaluating how the proposed AD approach is able to handle the so-called *Recipe Effect*, that is related to the presence of different productive recipes in a dataset; the consequence of this fact is that the deriving heterogeneous data can make any monitoring procedure not effective. In fact, changes in values due to recipe change can be erroneously misinterpreted as an anomalous behaviour. One possible solution to this problem is the implementation of a Recipe Effect Removal process

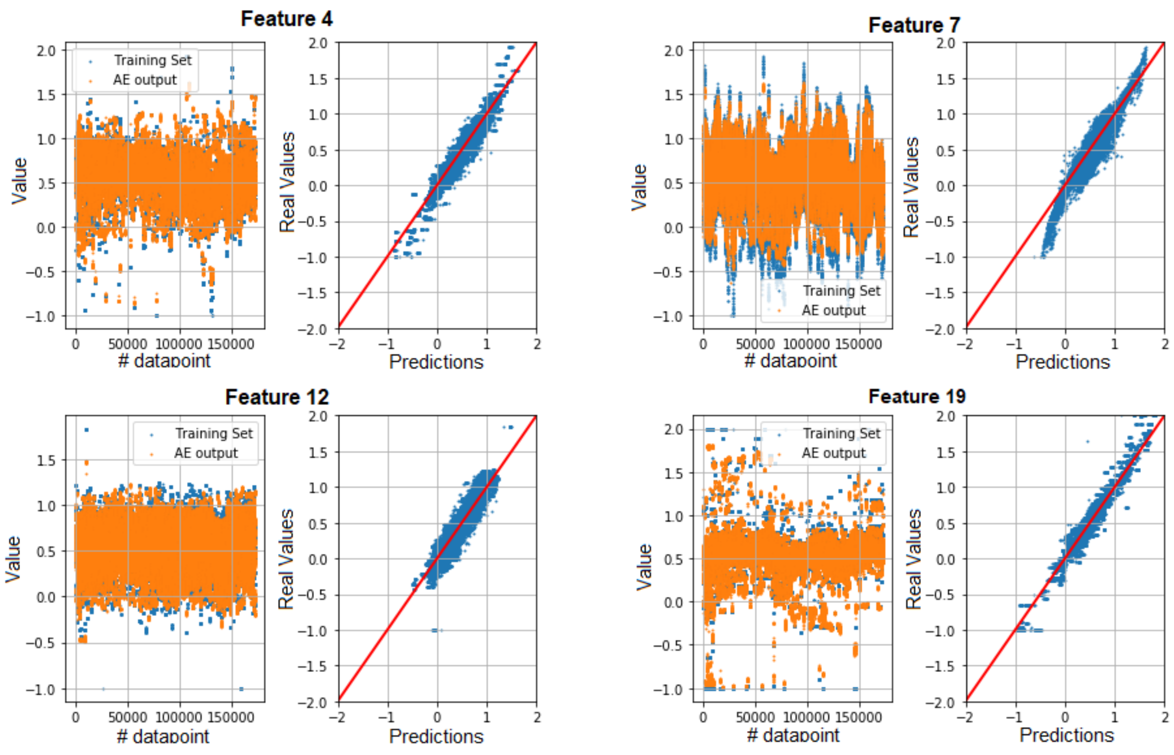

Figure 3: [Experiment #1] Examples of AE reconstructions on the training set.

during the data extraction. It is possible to deal with Recipe Effect by training the Recipe Effect Removal process for each equipment present in the workcenter, and use these results in order to remove the Recipe Effect from that single equipment. It is important to note that recipe effect often cannot be removed in the data by simply using one-hot-encoding. This fact is due to the presence of several dozens of different recipes on one equipment in volume production. In some cases (and taking into account also cleaning and testing recipes), the number of different recipes might be even greater than 100. In practice, this number is too large for applying one-hot-encoding.

4.1 Set up and Results

The preprocessing step described in Section 2 has effected the original data in the following way:

- applying the constant and NaN variables removal, the number of features removed is 7;
- after the correlation analyis, 36 features were additionally removed;
- given the aforementioned data numerosity, in the considered case we have $N = 1984$ in the noisy feature removal step. The τ value is selected to be equal to 5, while the t value is chosen to be 0.05. With these values the number of columns discarded is equal to 7, returning a dataset composed of 198434 samples each one composed of 117 features.

Regarding the AE design, the chosen architecture has two hidden layers, of 96 and 64 neurons each, and a bottleneck dimension of 32 neurons. The Leaky-ReLu parameter α is set-up equal to 0.3, after a grid search optimization. The Autoencoder reconstructions are generated from the best trained Autoencoder selected as explained in 3.2. In Figure 3, it is possible to note the good quality of the reconstructions performed by the Autoencoder on training set. On the the left side of each subfigure, it is possible to compare the real datapoint values (in blue) with the reconstructed values (in orange) in function of the time.

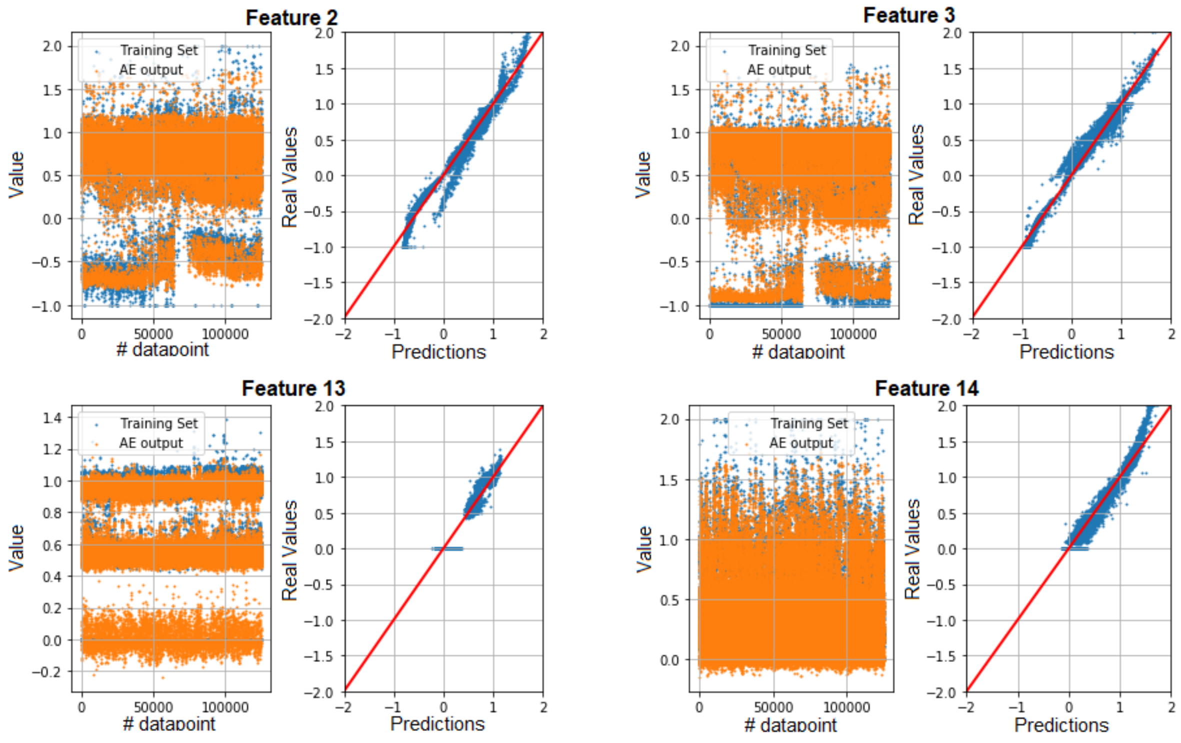

Figure 4: [Experiment #2] Examples of AE reconstructions on the dataset extracted from Workcenter B.

Moreover on the right side of each subfigure, it is reported on the x-axis the real value of the datapoint, while on the y-axis the reconstructed value is reported. Ideally the blue points in this chart have to be as close as possible to the diagonal line $x = y$, reported in red. Using the red line as reference, it is possible to confirm the good quality of the reconstructions. After having performed the reconstructions, each datapoint value is compared with its real value, computing the Mean Squared Error of each datapoint using Eq. 1.

4.2 Comparison with different Anomaly Detection Approaches

In this Section, different Anomaly Detection algorithms will be described and applied to the same dataset analyzed by the Autoencoder-based approach, comparing the results.

Isolation Forests (iForests) were introduced in (Liu et al. 2008). The main difference between iForests and the proposed model-based approach is that iForests explicitly isolates anomalies instead of profiles normal points. To achieve this, iForests takes advantage of two anomalies quantitative properties: i) they are the minority consisting of fewer instances and ii) they have attribute-values that are very different from those of normal instances. Isolation Forest builds an ensemble of iTrees for a given data set, then classifies as anomalies are those instances which have short average path lengths on the iTrees. There are only two variables in this method: the number of trees to build and the sub-sampling size. iForests detect only approximately the 2.24% of the anomalies discovered by the Autoencoder-based approach. This fact could be justified by the so-called ”masking effect”: when the number of anomalies is high it is possible that some of those are aggregated in a dense and large cluster, making it more difficult to separate the single anomalies and, in turn, to detect such points as anomalous. In fact, iForests applied to the considered dataset finds only groups of anomalies near in time.

Local Outlier Factor, or LOF, has been introduced in (Kriegel et al. 2009). LOF is a score that tells how likely a certain datapoint is considered anomalous. The LOF is a calculation that looks at the neighbours of a certain datapoint and compares this to the density of other datapoints later on. The concept of locality is given by the fact that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood. More precisely, locality is given by k-nearest neighbors, whose distance is used to estimate the local density. By comparing the local density of a sample to the local densities of its neighbors, one can identify samples that have a substantially lower density than their neighbors. These are considered outliers. LOF classifies as anomalies the 36.75% of the anomalies discovered by the Autoencoder-based approach.

5 EXPERIMENT #2: TESTS ON DIFFERENT WORKCENTERS

In this section, the detection capabilities of the given approach will be tested on workcenters belonging to different sites and areas of production. These tests will confirm the strong generalization capabilities of the proposed AE-based approach to AD.

5.1 Workcenter B

The dataset used in this test is extracted from a workcenter belonging to a plasma etching area of production. The dataset extracted from this workcenter has 138928 rows and 253 columns.After the preprocessing steps, the features to be scaled are 192. The best architecture selected by the Tournament Search Architecture Optimization algorithm is the one with 2 hidden layers, of 128 and 96 neurons each and a bottleneck with 64 neurons. Some examples of Autoencoder reconstructions are depicted in Figure 4. Also in this case the reconstructions capabilities of the approach are confirmed.

5.2 Workcenter C

The dataset used in this test is extracted from a workcenter, belonging to the implant area of production. The dataset extracted from this workcenter has 173106 rows and 230 columns. After the preprocessing steps, the features to be scaled are 151. The execution of the Tournament Search Optimization algorithm described selects as best architecture the one with 2 hidden layers, of 128 and 96 neurons each and a bottleneck with 64 neurons. Some examples of Autoencoder reconstructions are depicted in Figure 5. Also in this case the reconstruction capabilities of the approach are confirmed.

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

The presented Autoencoder-based approach is able to reconstruct in a good manner the normal behaviour of data, detecting in this way the anomalies. The implementation of very general preprocessing steps enforces the generality of the approach. Moreover, these steps allow a very accurate modelization process, using the AE. The generality of the implemented modelization step is given by the presented Tournament Search Optimization algorithm. This approach is able to deal with the main problem of data extracted from industrial workcenters, that is the Recipe Effect. Moreover, the designed approach for Anomaly Detection has been compared with different Anomaly Detection approaches in order to understand its strengths and weaknesses. Once again, the main strength of the designed approach is the strong generality and adaptability: as seen in this work, this approach works very well on dataset extracted from different workcenters. It is important to remark the fundamental importance of the Anomaly Detection process in every type of manufacturing. This process allows the detection and the correction of damages due to continuous and intensive usage of the industrial equipment. Some envisioned future improvements are:

- the implementation of a proper sampling strategy in order to deal with bigger dataset;
- implementation of the designed approach with different type of data in semiconductor manufacturing;
- investigation of the anomalies discovered, taking into account dataset containing different information, giving a ”supervised” nature to the problem.

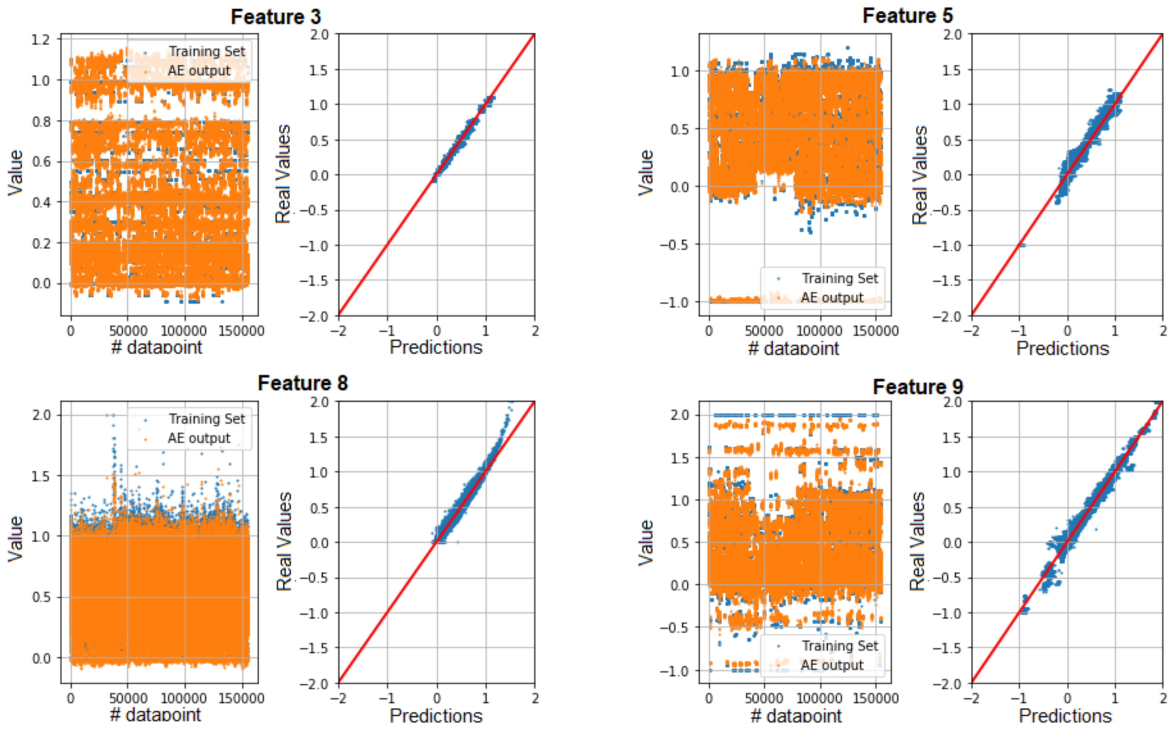

Figure 5: [Experiment #2] Examples of AE reconstructions on the dataset extracted from Workcenter C.

REFERENCES

An, J., and S. Cho. 2015. "Variational autoencoder based anomaly detection using reconstruction probability". *Special Lecture on IE* 2(1):1–18.

Blickle, T., and L. Thiele. 1995. "A Mathematical Analysis of Tournament Selection.". In *ICGA*, Volume 95, 9–15. Citeseer.

Chalapathy, R., and S. Chawla. 2019. "Deep learning for anomaly detection: A survey". *arXiv preprint arXiv:1901.03407*.

Chandola, V., A. Banerjee, and V. Kumar. 2009. "Anomaly Detection : A Survey". *ACM Computing Surveys*:1–72.

Dau, H. A., V. Ciesielski, and A. Song. 2014. "Anomaly detection using replicator neural networks trained on examples of one class". In *Asia-Pacific Conference on Simulated Evolution and Learning*, 311–322. Springer.

Hawkins, S., H. He, G. Williams, and R. Baxter. 2002. "Outlier detection using replicator neural networks". 170–180.

Hinton, G. E. 2006. "Reducing the Dimensionality of Data with Neural Networks". *Science* 313:504–507.

Karadayi, Y., M. Aydin, and A. Öğrenci. 2020. "A Hybrid Deep Learning Framework for Unsupervised Anomaly Detection in Multivariate Spatio-Temporal Data". *Applied Sciences* 10.

Kingma, D. P., and J. L. Ba. 2017. "ADAM: a Method for Stochastic Optimization".

Kolmogorov, A. 1933. "Sulla determinazione empirica di una legge di distribuzione". *Inst. Ital. Attuari, Giorn.* 4:83–91.

Kriegel, H.-P., P. Kröger, E. Schubert, and A. Zimek. 2009. "LoOP: local outlier probabilities". In *Proceedings of the 18th ACM conference on Information and knowledge management*, 1649–1652.

Liu, F. T., K. M. Ting, and Z.-H. Zhou. 2008. "Isolation forest". In *IEEE international conference on data mining*, 413–422.

Maggipinto, M., A. Beghi, and G. A. Susto. 2019. "A Deep Learning-based Approach to Anomaly Detection with 2-Dimensional Data in Manufacturing". In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Volume 1, 187–192.

Miller, B. L., and D. E. Goldberg. 1995. "Genetic algorithms, tournament selection, and the effects of noise". *Complex systems* 9(3):193–212.

Montgomery, D. 1997. *Introduction to statistical quality control*. 3. ed ed. New York, NY [u.a.]: Wiley.

Pol, A. A., V. Berger, C. Germain, G. Cerminara, and M. Pierini. 2019. "Anomaly detection with conditional variational autoencoders". In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 1651–1657.

Prechelt, L. 1996. "Early Stopping-But When?". In *Neural Networks: Tricks of the Trade*.

Puggini, L., and S. McLoone. 2018. "An enhanced variable selection and Isolation Forest based methodology for anomaly detection with OES data". *Engineering Applications of Artificial Intelligence* 67:126–135.

Sakurada, M., and T. Yairi. 2014. “Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction”. 4–11.

Susto, G. A., A. Beghi, and S. McLoone. 2017. “Anomaly detection through on-line isolation forest: an application to plasma etching”. In *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 89–94. IEEE.

Susto, G. A., M. Terzi, and A. Beghi. 2017. “Anomaly detection approaches for semiconductor manufacturing”. *Procedia Manufacturing* 11:2018–2024.

Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. 2010, December. “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion”. *J. Mach. Learn. Res.* 11:3371–3408.

Yu, L., and H. Liu. 2003. “Feature selection for high-dimensional data: A fast correlation-based filter solution”. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, 856–863.

Zenati, H., C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar. 2018. “Efficient gan-based anomaly detection”. *arXiv preprint arXiv:1802.06222*.

Zhang, C., and Y. Chen. 2019. “Time series anomaly detection with variational autoencoders”. *arXiv preprint arXiv:1907.01702*.

Zhou, C., and R. Paffenroth. 2017. “Anomaly Detection with Robust Deep Autoencoders”. *KDD’17*:665–674.

AUTHOR BIOGRAPHIES

SIMONE TEDESCO is a graduate engineer from the Department of Information Engineering. He earned his M.Sc. degree in Automation Engineering from the University of Padova. He was an intern student at Infineon Technologies AG. His email address is tedescosim@gmail.com

NATALIE GENTNER is currently an industrial Ph.D. student with the University of Padova and Infineon Technologies AG (Neubiberg, Germany). She earned her M.Sc in Mathematics from the Technical University in Munich. Her research interests include machine and deep learning, automatization in semiconductor manufacturing, system theory and applied analysis. Her email address is natalie.gentner@studenti.unipd.it / natalie.gentner@infineon.com.

ANDREAS KYEK is Data Scientist at Infineon Technologies AG (Neubiberg, Germany). He started his career in industry as a unit process development engineer at Infineon in 2001. From the beginning he was involved in Advanced Process Control (APC) and took over the development department for APC in 2007. During this time he also co-initiated the ENIAC funded project IMPROVE and was speaker of the executive board. In the years 2009 and 2010 he was member of the steering committee of the AEC/APC conference Europe. Later he joined a Manufacturing Excellence group, where he got involved in Factory Physics. Today he is heading projects where the usage of Big Data and Artificial Intelligence methods on manufacturing data is investigated. He holds a Ph.D. in Physics from the Technical University in Munich. His email address is andreas.kyek@infineon.com.

YAO YANG is Data Scientist at Infineon Technologies AG (Neubiberg, Germany). She earned her Ph.D. from the University of Mannheim with OR (operations research) related topics. She started her career as a scientist solving supply chain optimization problems in BASF SE (Ludwigshafen, Germany). After joining Infineon in 2017, she was involved in a large variety of semiconductor supply chain projects and is now focusing on data science topics, applying Big Data and Artificial Intelligence methods in manufacturing. Her email address is yao.yang@infineon.com.

GIAN ANTONIO SUSTO is currently an Associate Professor with the University of Padova. He earned his Ph.D. from University of Padova and he held visiting positions at University of California at San Diego, Maynooth University and Infineon Technologies Austria AG. He is co-founder of Statwolf LtD. His current research interests include machine and deep learning, industry 4.0 and natural language processing. He is an Associate Editor of the IEEE Transactions on Semiconductor Manufacturing for the area of Process Modeling. His email address is gianantonio.susto@unipd.it. His website is http://automatica.dei.unipd.it/susto.html.