Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 17 (2018) 126–133

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

28th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2018), June 11-14, 2018, Columbus, OH, USA

A Convolutional Autoencoder Approach for Feature Extraction in Virtual Metrology

Paper ID 259, Marco Maggipinto[a], Chiara Masiero[a], Alessandro Beghi[a,b], Gian Antonio Susto[a,b,*]

[a] *Department of Information Engineering, University of Padova, Italy*
[b] *Human Inspired Technology Research Centre, University of Padova, Italy*

Abstract

Exploiting the huge amount of data collected by industries is definitely one of the main challenges of the so-called Big Data era. In this sense, Machine Learning has gained growing attention in the scientific community, as it allows to extract valuable information by means of statistical predictive models trained on historical process data. In Semiconductor Manufacturing, one of the most extensively employed data-driven applications is Virtual Metrology, where a costly or unmeasurable variable is estimated by means of cheap and easy to obtain measures that are already available in the system. Often, these measures are multi-dimensional, so traditional Machine Learning algorithms cannot handle them directly. Instead, they require feature extraction, that is a preliminary step where relevant information is extracted from raw data and converted into a design matrix. Features are often hand-engineered and based on specific domain knowledge. Moreover, they may be difficult to scale and prone to information loss, affecting the effectiveness and maintainability of machine learning procedures. In this paper, we present a Deep Learning method for semi-supervised feature extraction based on Convolutional Autoencoders that is able to overcome the aforementioned problems. The proposed method is tested on a real dataset for Etch rate estimation. Optical Emission Spectrometry data, that exhibit a complex bi-dimensional time and wavelength evolution, are used as input.

© 2018 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/)
Peer-review under responsibility of the scientific committee of the 28th Flexible Automation and Intelligent Manufacturing (FAIM2018) Conference.

Keywords: Convolutional Autoencoder, Deep Learning, Etching, Feature Extraction, Industry 4.0, Neural Network, Optical Emission Spectroscopy, Semiconductor Manufacturing

1. Introduction

During the last years, industries have collected a huge amount of historical data from their production processes, leading to the so-called Big Data era. The challenges that Big Data pose in industrial environments are various [17] and the scientific community is in a continuous effort to propose innovative solutions to face them. One of the main

* Gian Antonio Susto
E-mail address: gianantonio.susto@dei.unipd.it

2351-9789 © 2018 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/)
Peer-review under responsibility of the scientific committee of the 28th Flexible Automation and Intelligent Manufacturing (FAIM2018) Conference.
10.1016/j.promfg.2018.10.023

problems in the Industry 4.0 [14] is how to turn available data into business valuable information. In this sense, Machine Learning (ML) technologies have become extremely popular, allowing the development of statistical models that can be employed to infer relevant knowledge of the production process, based on historical data.

In Semiconductor Manufacturing, these data-driven models are the foundation of Advanced Process Control (APC). Virtual Metrology (VM) [26], in particular, is nowadays extensively applied in the semiconductor industry[1]; its goal is to exploit the already available information (e.g. physical sensors measurements) to infer the value of a costly or even unmeasurable variable that is relevant to assess the status of the production process. Usually this goal is achieved through *supervised learning* [5] methods that leverage labeled data, i.e. physical measurements of both input and output from past process runs, to train a statistical model. Often in Semiconductor Manufacturing the input data of the VM modules exhibit complex structures. In particular, it is common to encounter data in the form of time series [24], or with a multidimensional evolution. Traditional ML techniques that are usually employed in this context are not suitable to deal with these highly structured input data; In fact, they require a preliminary operation, called feature extraction, where a set of informative values is extracted from the raw data, and then cast into a *design matrix*. The feature extraction phase can be performed in two ways:

Hand-designed feature extraction Informative peculiarities of the data are translated into features by means of visual inspection and quantitative studies; hand-designed feature extraction may require the help of process experts to exploit physical knowledge of the process under examination.

Automatic feature extraction Features are computed automatically based on statistical properties of the input variables (such as mean value and standard deviation), sub-sampling, or averaging of different portions of the input data [8, 20, 30].

Both these approaches present significant drawbacks: Hand designed feature extraction is extremely time consuming and poorly scalable in a complex environment such as the Semiconductor Industry. On the other hand, automatic feature extraction methods are affected by information loss, and often leads to poor prediction capabilities. Recently, more sophisticated feature extraction methods have been proposed in order to overcome the aforementioned problems: in [24], a functional learning solution is presented in order to tackle the feature extraction in a supervised way for time-series data, by embedding it in the modeling phase. In [18], an approach based on regularization [5] is employed to deal with Optical Emission Spectroscopy (OES) data. OES data are paradigmatic in motivating the need of a sophisticated feature extraction mechanism due to their bi-dimensional evolution with respect to time and wavelength. In this paper, we propose a Deep Learning (DL) [6] based method for feature extraction on OES data that relies on deep *autoencoders*. An autoencoder is a particular Artificial Neural Network (ANN) that is trained to reconstruct its input. Usually, the hidden layers of the network perform dimensionality reduction on the input, learning relevant features that allow a good reconstruction. Moreover, deep autoencoders exploit multiple non-linear representational layers that learn complex hierarchical features from the data with high informative content.

The bi-dimensional structure of OES is similar to an image, suggesting the employment of Computer Vision-inspired methodologies [27]. In particular, the convolution operation is highly effective at extracting local features from images: actually, Convolutional Neural Networks (CNNs) are extensively employed for problems like object localization and recognition [11], face recognition [12], and text recognition [28]. For this reason, the proposed model will be based on CNNs.

The rest of the paper is organized as follows: in Section 2, an overview of Deep Learning for feature extraction is provided with particular attention to autoencordes, while in Section 3 the dataset at hand is described. In Section 4, the proposed architecture and the experimental results are reported, while the final remarks and future works are outlined in Section 5.

2. Deep Learning for feature extraction

In recent years, the diffusion of DL technologies has paved the way for sophisticated automatic feature extraction methods that are able to effectively compress data into a lower dimensional representations without significant loss of

[1] We underline that VM/Soft Sensing technologies are applied in many other fields beside semiconductor manufacturing like automotive [1] and chemical manufacturing [4]

information. The advancements in DL-based feature extraction are the foundation of the incredible success of these technologies in Computer Vision: Sun *et al.* [25] employed DL for face representation. The method relied on CNNs in order to reduce the dimensionality of the significant regions of the input face and thus obtain a series of "DeepID" that, combined together, provided the input for a face verification model. Romero *et al.* [21], instead, employed a stacked convolutional autoencoder for unsupervised feature learning on remote sensing images. The unsupervised pretraining of the autoencoder, combined with a supervised fine-tuning, allowed to cope with the high-dimensionality of the data and the limited dataset size. The proposed method outperformed traditional methods such as Principal Component Analysis (PCA), and its kernel counterpart version (kPCA). Moreover, deep architecture performed significantly better than shallow ones.

As for Semiconductor Industry, the diffusion of DL-based automatic feature extraction methods is still in its infancy: some recent papers employed it to deal with complex VM problems: Lin *et al.* [15] developed a single layer autoencoder for screening test escapes. In particular, the autoencoder is trained in an unsupervised fashion only on good chips, using the euclidean distance as cost function. Then, the faulty chips have been identified because of the higher reconstruction error compared to good chips. Lee *et al.* [13] designed a denoising autoencoder for wafer fault monitoring, showing the ability of the model to extract noise tolerant features from the data that led to high predictive capabilities. Yan *et al.* [29] leveraged a similar structure based on Denoising Autoencoders in order to estimate oxygen levels in flue gases.

In this paper we propose a novel method for automatic feature extraction that present concrete originality compared to the actual literature:

- While Convolutional Autoencorders are extensively employed in Computer Vision, the same thing does not apply in Semiconductor Manufacturing.
- All the previously described papers exploit only the last layer of features extracted by the network, while our method leverages all the layers of the network.
- Often, autoencoders are completed with a regression/classification layer in order to solve a supervised learning problem. Anyway, simple linear regression or softmax are usually employed. Our model exploit the powerful features extraction performed by the autoencoder, combined with complex regression algorithms such as Support Vector Regression (SVR).

As stated above, DL models have often been employed as supervised end-to-end regression or classification algorithms. However, they can also be used as feature extraction blocks, and then combined with traditional ML techniques, thanks to their inherent ability to treat complex input data automatically, i.e. without the need of time consuming and poorly scalable feature extraction. In this paper, we investigate this characteristic in order to realize an automatic feature extraction procedure based on autoencoders.

2.1. Artificial Neural Networks

Artificial Neural Networks are the foundation of DL technologies. ANNs resemble brain since they are obtained by the interconnection of many simple units, called *neurons*. Various ANN architectures have been developed during the years: the most simple one is the so-called Feedforward Neural Network (FNN) where each neuron is connected to all the neurons in the previous layer, defining a directed graph without any loops (Fig. 1a).

In recent years, more complex networks, called Convolutional Neural Networks (CNNs), have gained more and more popularity thanks to their achievements in Computer Vision. CNNs exploit a multilayer structure similar to FNNs but where different kind of hidden layers are alternated. In particular we can distinguish three kinds of hidden layers: (i) convolutional, (ii) pooling, (iii) fully connected.

(i) Convolutional layers differ from hidden layers in FNNs because neurons in layer l are not connected to all neurons in layer $l-1$. Instead, each neuron i in layer l applies an activation function to the convolution of the previous layer's output with a kernel W^l plus a bias term b^l, as depicted in Fig. 1b. The area of layer $l-1$ that this filter is applied to, is called the receptive field of neuron i. CNNs feature parameter sharing, because for each neuron in layer l, the same convolutional filter is applied to the corresponding receptive field in layer $l-1$.

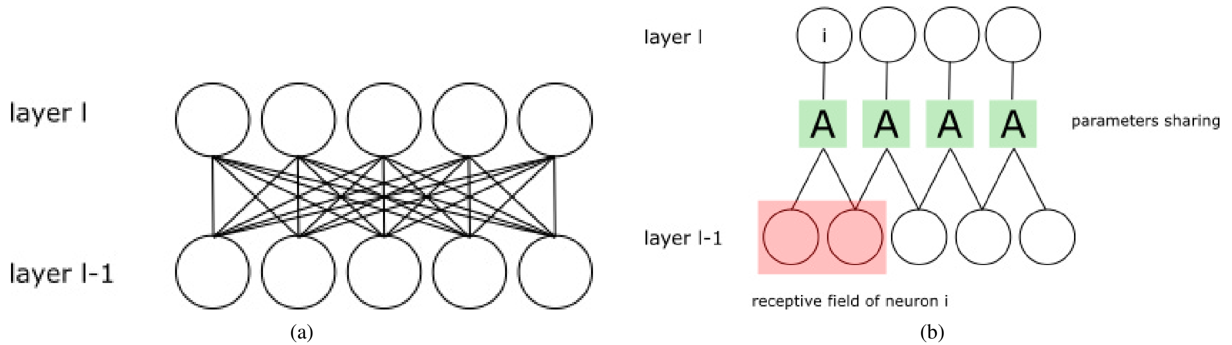


Fig. 1: (a) FNN fully connected layers; (b) CNN layers with parameter sharing

Usually, multiple kernels are employed to extract different features simultaneously. In presence of 2D data (images), it is common to preserve the original input structure by applying bi-dimensional convolutions. The different outputs obtained using different kernels are then stacked and treated as *channels* (more details available in [6]).

(ii) Pooling layers just performs a subsampling of the previous layer's output, usually averaging (*average pooling*) or taking the maximum value (*max-pooling*) over a contiguous region of values.

(iii) Fully Connected (FC) layers are actual FNNs and therefore the same description applies here. Usually FC layers are placed at the end of the network. Since the structure of FC layer is intrinsically one-dimensional, multi-dimensional data are *flattened* in a 1D vector before final FC layers.

ANNs provide an approximation function $y = f^*(x; \theta)$ of an arbitrary complex continuous function f, parametrized by a set of coefficients θ (matrices and biases in FNNs, kernel/matrices and biases for CNNs) [6]. The creation of the predictive model thus requires the estimation of the parameters θ that better approximate the desired output; this is achieved by minimizing a cost function defined according to the output layer properties; common choices are MSE for regression and cross-entropy for classification. Usually, a gradient-descent based algorithm is used, based on *backpropagation* [22].

2.2. Autoencoders

Autoencoders are generative models where an ANN is trained to reconstruct its own input in an unsupervised way. An autoencoder employs a symmetric structure composed by two main blocks: an *encoder* part that compresses the input into a low dimensional representation that contains the informative content of the data; a *decoder* part that is trained to reconstruct the input from the features extracted by the encoder. Once the unsupervised pretraining is completed, the encoder part is thus a powerful automatic feature extractor that, completed with a suitable output layer, can be then *fine-tuned* [6] in a supervised way in order to obtain the desired estimation.

More in details, fine-tuning allows to adjust the parameters of the encoder in order to extract features that are even more effective in addressing the specific target estimation problem. This operation is performed by adding a further layer to the encoder block. Based on the problem at hand (See Fig. 2(b)), the encoder can be followed by either a classification or a regression output layer. As explained in Section 3, our problem can be formalized as a regression one. Thus, in the proposed approach, a linear output layer is added to the encoder, then the resulting network is trained in a supervised fashion. Note that previously computed encoder weights provide the starting point to the stochastic gradient descent algorithm. As a consequence, fine tuning is expected to refine the features extracted in unsupervised fashion, so that they are even more valuable in tackling the regression problem.

Autoencoders can be created using various Neural Network structures. In this paper, we propose a Convolutional Autoencoder where the underlying ANN exhibits a convolutional structure as described in Section 2.1.

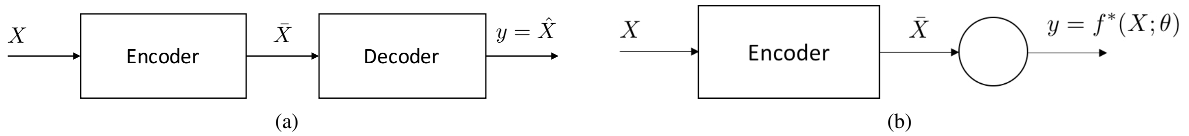


Fig. 2: (a) Structure of an autoencoder. X is the input, $\bar{X}$ is the compressed version of X and $\hat{X}$ is the reconstructed version of X. (b) Encoder employed in the final estimation. During *fine tuning* the pre-trained parameters of the encoder network are adjusted in a supervised way to achieve better performance.

3. Dataset description

In Semiconductor Manufacturing, *etching* is a key step in the realization of integrated circuits. In this process, a masked silicon wafer is hit by an high-speed stream of plasma of an appropriate ionized gas mixture in a vacuum chamber. The exposed surface is thus etched away because of the chemical and mechanical interaction with plasma. Plasma etching processes are affected by various factors that may alter the final quality of the product. In particular, chamber mismatch, non-uniformity across the wafer and within die, and surface composition/roughness have a relevant impact [3]. For this reason, in the perspective of both process control and quality assessment, it is important to understand the resolution and directionality of etching. In this sense, the *etch rate*, i.e the thickness of the eroded surface per unit of time, provides the required information. However, obtaining this measure is extremely costly, and it requires a post-processing metrology step. Thus, in semiconductor industry, it is pivotal for cost reduction and production performance to estimate the etch rate based on cheap and easy to obtain measures. Optical Emission Spectroscopy (OES) of the plasma is particularly informative for this goal[2]. It allows to observe the changes in the plasma chemistry during etching, providing the foundation to VM solutions that exploit historical data to build a predictive model for etch rate estimation.

The dataset our model will be tested on is composed by N=1747 OES data samples with the associated etch rate y that has been physically measured during historical process runs. The input OES data present a complex bi-dimensional structure, with time and wavelength evolution as depicted in Fig. 3. It is possible to notice that the measurements saturate at a value close to 4000 (normalized value), limiting the range of possible values to the interval $[0, 4000]$ (normalized values); this behaviour is resemblant of 8-bit greyscale images, where the range of possible values for each pixel is limited between 0 and 255. This property, in combination with the bi-dimensional structure of OES, suggests the employment of Computer Vision inspired technologies and justifies the choice of a non linear model because of the inherent non-linearity of saturation phenomena. The proposed autoencoder will thus be based on CNNs, that have outperformed other methods in many Computer Vision tasks [7]. Since CNNs are able to extract hierarchical sparse feature [6] from complex data like images, the proposed autoencoder is expected to provide a powerful feature extraction model for OES data.

4. Proposed Approach and Experimental Results

4.1. Convolutional Autoencoder-based Feature Extraction

The proposed feature extraction method exploits the representational power of a CNN composed of three convolutional layers alternated with average pooling layers. The employment of average pooling guarantees the extraction of smooth features that are usually suitable for regression problems as the one under study. Fig. 3 depicts in details the proposed feature extraction procedure: the aforementioned CNN is trained as described in Section 2 then, the features extracted by each average pooling layer are flattened and concatenated in order to form a final feature vector of dimension p = 38464, thus compressing the input to one third of its original size of 2014 wavelengths × 54 time

[2] For a detailed description of OES data we invite the reader to refer to [27, 24]

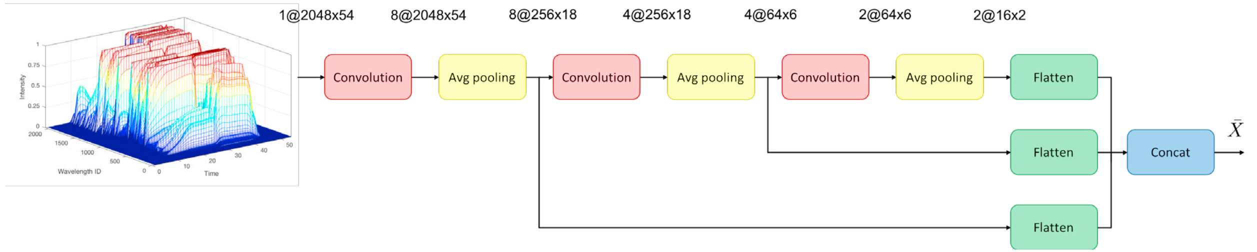


Fig. 3: Structure of the proposed feature extraction method.

stamps. The notation $C@H\times W$ defines the dimension of the input at the respective layer. In particular, C indicates the number of channels, H indicates the height and W the width.

The obtained features will be fed to a Support Vector Regression (SVR) machine that is trained in a supervised way in order to perform the final etch rate prediction. In order to asses the quality of the proposed procedure, the performance will be compared with a standard automatic feature extraction method where four statistics (average, standard deviation, minimum, maximum) on the first $k = 100$ Principal Components of the data are computed and then fed to a LASSO and SVR regression, popular approaches for VM technologies [10, 9, 23, 26].

The proposed algorithm has been implemented and trained using Keras [2] with Tensorflow [16] backend. *Adam* optimizer has been employed using Mean Squared Error (MSE) as cost function. More in details, the training operation has been performed in two steps: the unsupervised pre-training, where the MSE measures the reconstruction error of the AE and the supervised fine-tuning where the MSE measures the prediction error of the model on the targets y. It is worth remarking that the fine-tuning procedure is used only to obtain more representative features, and not to train a predictive model for the problem at hand. Indeed, etch rate is estimated based on a more advanced regression algorithm (SVR) fed with the features provided by the fine-tuned autoencoder. In particular, features are obtained by stacking the outputs of all the fine-tuned pooling layers of the network, and not only the last one.

4.2. Results

The performance of the proposed method have been assessed using 100 Monte-Carlo cross validation [19] cycles with a test set composed by the 30% of the total number of process runs available, i.e. $N_{test} = 0.3 \cdot N$. The same procedure has been employed to estimate the performance of the benchmark methods. In order to estimate the hyper-parameters of the employed regression algorithms, 10 Monte-Carlo cross validation cycles have been performed on a validation set composed by 30% of the data not used for test, i.e $N_{val} = 0.3 \cdot (N - N_{test}) = 0.21 \cdot N$. The metrics employed to quantify the models prediction capabilities are MSE (Eq. 1) and R^2 score (Eq. 1).

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N_{\text{test}}}(y_i - \hat{y}_i)^2 \qquad \mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N_{\text{test}}}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N_{\text{test}}}(y_i - \bar{y})^2} \tag{1}$$

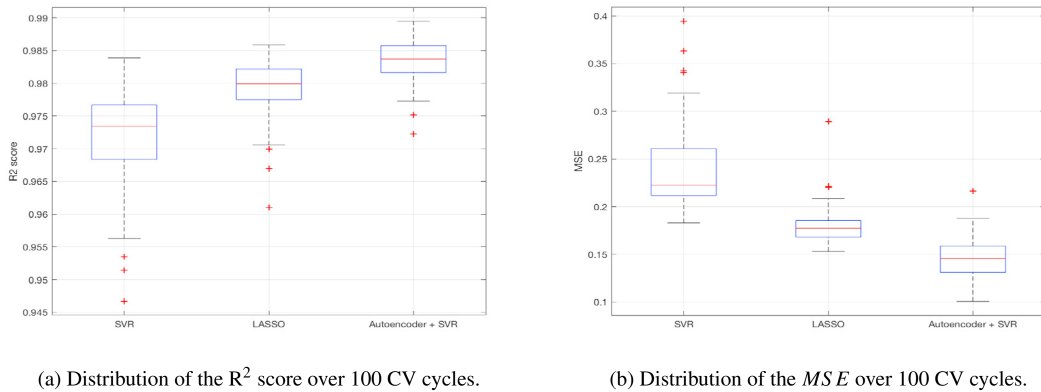
where y_i and $\hat{y}_i$ are respectively the real and predicted output of test i and $\bar{y}$ is the average of the test output values.

Table 1: Performance comparison.

	SVR	LASSO	Autoencoder+SVR
MSE	0.238	0.18	**0.146**
R^2	0.972	0.979	**0.983**

In Table 1 the mean value of the performance indexes over 100 MC cross validation cycles is reported. The proposed feature extraction method in combination with a SVR outperformed classical automatic feature extraction procedures

justifying the employment of a complex deep architecture instead of simple shallow methods such as PCA and statistical moments. A more detailed performance comparison is reported in Fig. 4 where the boxplot of the distributions of the performance indexes over 100 MC cross validation cycles are depicted. The high values obtained for the R^2 metric highlight some limitations of the dataset at hand that appear to be easily predictable also with simple linear models, suggesting that the input-output relationship is close to linear. While the autoencoder provides better results, we expect that, employing a more complex dataset with a highly non-linear input-output relationship, this superiority would be even more evident. Moreover, Deep Neural Networks are known to perform extremely well with big amount of data it is safe to assume that our model would achieve better performance if trained with an higher number samples.



(a) Distribution of the R^2 score over 100 CV cycles.

(b) Distribution of the MSE over 100 CV cycles.

Fig. 4: Distribution of: (a) the R^2 score over 100 CV cycles; (b) the MSE over 100 CV cycles.

5. Conclusion

In this paper we proposed a novel feature extraction method based on Convolutional Autencoders which, in combination with traditional Machine Learning algorithms, is able to deal effectively with the data complexity typical of the semiconductor industry. In particular, we tested our method on OES data coming from a plasma etching process where the need of a precise etch rate estimate is of fundamental importance. The bi-dimensional structure of such input data suggested to tackle feature extraction drawing inspiration from Computer Vision algorithms. In particular, we chose to use Convolutional Neural Networks that have recently outperformed classical methods in a wide range of complex vision tasks. The proposed method outperformed classical shallow feature extraction procedures such as PCA and statistical moments, providing an accurate prediction of the required target (etch rate) leveraging the information provided by OES data. This result may have a strong impact on semiconductor manufacturing processes where the direct measurement of the etch rate is extremely resource demanding both in terms of time and money. The achieved performance gain justifies the employment of a complex DL model that, exploiting the representational power of CNNs is able to deal with the inherent bi-dimensional interdependence of OES data exhibiting a time and wavelength evolution. In addition, the proposed method presents considerable advantages when compared to the employment of hand crafted features, since it does not require any domain specific knowledge, but it treats the input complexity in a natural and scalable way. This is of paramount importance in the ever growing semiconductor industry, where the complexity of the systems keeps increasing at a very fast pace. Similarly, the availability of historical data is in continuous expansion and the proposed solution is well suited for a Big Data context. Indeed,it is in fact a well known characteristic of DL algorithms to improve their performance with the increase of data availability [6]. Thus, we are confident that with a larger training dataset our method would allow to extract even more representative features.

Finally, the presented approach provides a general feature extraction method for similar problems where the input data exhibits a complex evolution whose structure cannot be properly handled by traditional ML algorithms that require the input to be organized in a design matrix, where each row represents a single data observation. Instead,

the proposed method is can work with data where each observation exhibits a bi-dimensional evolution, and thus it is a matrix itself. While promising performance have been obtained, the presented results are still preliminary; a more extensive comparison with other autoencoder structures such Denoising Autoencoders and simple methods combined with hand designed features is left as a future development.

References

[1] Bruschetta, M., Maran, F., Beghi, A., 2017. A fast implementation of mpc-based motion cueing algorithms for mid-size road vehicle motion simulators. Vehicle system dynamics 55, 802–826.

[2] Chollet, F., 2015. Keras. URL: https://github.com/fchollet/keras.

[3] Coburn, J.W., Winters, H.F., 1979. Ion and electronassisted gassurface chemistryan important effect in plasma etching. Journal of Applied Physics 50, 3189–3196. doi:10.1063/1.326355.

[4] Facco, P., Doplicher, F., Bezzo, F., Barolo, M., 2009. Moving average pls soft sensor for online product quality estimation in an industrial batch polymerization process. Journal of Process Control 19, 520–529.

[5] Friedman, J., Hastie, T., Tibshirani, R., 2001. The elements of statistical learning. volume 1. Springer series in statistics Springer, Berlin.

[6] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

[7] He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034.

[8] Hirai, T., Kano, M., 2015. Adaptive virtual metrology design for semiconductor dry etching process through locally weighted partial least squares. IEEE Transactions on Semiconductor Manufacturing 28, 137–144.

[9] Jia, X., Di, Y., Feng, J., Yang, Q., Dai, H., Lee, J., 2018. Adaptive virtual metrology for semiconductor chemical mechanical planarization process using gmdh-type polynomial neural networks. Journal of Process Control 62, 44–54.

[10] Kim, M., Kang, S., Lee, J., Cho, H., Cho, S., Park, J.S., 2017. Virtual metrology for copper-clad laminate manufacturing. Computers & Industrial Engineering 109, 280–287.

[11] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.

[12] Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D., 1997. Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks 8, 98–113.

[13] Lee, H., Kim, Y., Kim, C.O., 2017. A deep learning model for robust wafer fault monitoring with sensor measurement noise. IEEE Transactions on Semiconductor Manufacturing 30, 23–31.

[14] Lee, J., Kao, H.A., Yang, S., 2014. Service innovation and smart analytics for industry 4.0 and big data environment. Procedia CIRP 16, 3 – 8.

[15] Lin, F., Cheng, K.T., 2017. An artificial neural network approach for screening test escapes, in: Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific, IEEE. pp. 414–419.

[16] Martín Abadi et al., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 .

[17] Moyne, J., Samantaray, J., Armacost, M., 2016. Big data capabilities applied to semiconductor manufacturing advanced process control. IEEE Transactions on Semiconductor Manufacturing 29, 283–291.

[18] Park, C., Kim, S.B., 2016. Virtual metrology modeling of time-dependent spectroscopic signals by a fused lasso algorithm. Journal of Process Control 42, 51–58.

[19] Picard, R.R., Cook, R.D., 1984. Cross-validation of regression models. Journal of the American Statistical Association 79, 575–583.

[20] Ragnoli, E., McLoone, S., Lynn, S., Ringwood, J., Macgearailt, N., 2009. Identifying key process characteristics and predicting etch rate from high-dimension datasets, in: Advanced Semiconductor Manufacturing Conference, 2009. ASMC'09. IEEE/SEMI, IEEE. pp. 106–111.

[21] Romero, A., Gatta, C., Camps-Valls, G., 2016. Unsupervised deep feature extraction for remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing 54, 1349–1362.

[22] Rumelhart, D.E., Hinton, G.E., Williams, R.J., et al., 1988. Learning representations by back-propagating errors. Cognitive modeling 5, 1.

[23] Schirru, A., Pampuri, S., De Luca, C., De Nicolao, G., 2011. Multilevel kernel methods for virtual metrology in semiconductor manufacturing. IFAC Proceedings Volumes 44, 11614–11621.

[24] Schirru, A., Susto, G.A., Pampuri, S., McLoone, S., 2012. Learning from time series: Supervised aggregative feature extraction, in: Decision and Control (CDC), 2012 IEEE 51st Annual Conference on, IEEE. pp. 5254–5259.

[25] Sun, Y., Wang, X., Tang, X., 2014. Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1891–1898.

[26] Susto, G.A., Beghi, A., 2012. Least angle regression for semiconductor manufacturing modeling, in: Control Applications (CCA), 2012 IEEE International Conference on, IEEE. pp. 658–663.

[27] Terzi, M., Masiero, C., Beghi, A., Maggipinto, M., Susto, G.A., 2017. Deep learning for virtual metrology: Modeling with optical emission spectroscopy data, in: Research and Technologies for Society and Industry (RTSI), 2017 IEEE 3rd International Forum on, IEEE. pp. 1–6.

[28] Wang, T., Wu, D.J., Coates, A., Ng, A.Y., 2012. End-to-end text recognition with convolutional neural networks, in: Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE. pp. 3304–3308.

[29] Yan, W., Tang, D., Lin, Y., 2017. A data-driven soft sensor modeling method based on deep learning and its application. IEEE Transactions on Industrial Electronics 64, 4237–4245.

[30] Zeng, D., Spanos, C.J., 2009. Virtual metrology modeling for plasma etch operations. IEEE Transactions on Semiconductor Manufacturing 22, 419–431.