ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Understanding wafer patterns in semiconductor production with variational auto-encoders

Tiago Santos[1] and Roman Kern[1] *

1- KNOW-CENTER GmbH - Graz - Austria

Abstract. Semiconductor manufacturing processes critically depend on hundreds of highly complex process steps, which may cause critical deviations in the end-product. Hence, a better understanding of wafer test data patterns, which represent stress tests conducted on devices in semiconductor material slices, may lead to an improved production process. However, the shapes and types of these wafer patterns, as well as their relation to single process steps, are unknown. In a first step to address these issues, we tailor and apply a variational auto-encoder (VAE) to wafer pattern images. We find the VAE's generator allows for explorative wafer pattern analysis, and its encoder provides an effective dimensionality reduction algorithm, which, in a clustering application, performs better than several baselines such as t-SNE and yields interpretable clusters of wafer patterns.

1 Introduction

The semiconductor manufacturing industry faces significant challenges with its complex production processes, consisting of up to hundreds of precise steps, and strict post-production stress tests and quality controls. These process steps may feature deviations, which impact the end-product in different ways and could ultimately lead to failed quality checks and thus production yield losses. Hence, insights into so-called wafer test data patterns, a class of analog electric measurements performed early in the production process, could help understand which shapes of patterns occur and how they are related to production process steps and production yield.

In this work, we aim to address the problem of visualizing thousands of wafer patterns and typifying them into clusters. This provides a basis for later classification of previously unseen patterns into one of the clusters, as well as for linking those wafer pattern clusters to process steps and production yield.

Related Work. In the field of semiconductor manufacturing, automated methods for the detection of manufacturing faults or costly outliers in production are of great interest. To that end, machine learning and, in particular, deep learning methods have been applied to address these issues. In a similar application to

*The work has been performed in the project Power Semiconductor and Electronics Manufacturing 4.0 (SemI40), under grant agreement No 692466. The project is co-funded by grants from Austria, Germany, Italy, France, Portugal and - Electronic Component Systems for European Leadership Joint Undertaking (ECSEL JU). The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

this paper's, Rostami et al. [1] present a machine learning pipeline, consisting of binary classifiers and projection and clustering methods, to detect and classify faults in wafer production data. More recently, Lee et al. [2] propose convolutional neural networks to extract features from multivariate time series data on a specific wafer production process step, ultimately deriving insights into root causes of production faults. Our contribution to this branch of literature lies in the novel application field of variational auto-encoders, a deep generative model, which allows us to inspect various facets of wafer production data.

Deep generative models are a class of deep neural networks, which learn to recreate their input and allow, through sampling, to generate plausibly similar output. In this context, we highlight two prominent deep architectures: variational auto-encoders, proposed by Kingma and Welling [3] and Rezende et al. [4], and generative adversarial networks, introduced by Goodfellow et al. [5]. As presented in section 2, the variational auto-encoder is an efficient deep generative model, which combines variational inference and deep neural networks. Generative adversarial networks are a framework composed of two deep neural networks trained in combination: a generator, returning samples akin to its input, and a discriminator, estimating if its input came from the generator or the true data.

Approach and Findings. For our use-case, due to the computational challenges of deriving an efficient visualization for thousands of wafer pattern images, we propose the use of a deep generative model to summarize the data and allow for interactive visualization. We focus on variational auto-encoders, not only due to their well-studied generative sampling capability, but also due to their encoder functioning as a dimensionality reduction procedure. We then cluster the low-dimensional projections formed by the variational auto-encoder's encoder. With these projections, we observe better clustering performance in comparison with dimensionality reduction methods like t-SNE [6], as well as interpretable cluster medoids in the form of wafer patterns recognizable by experts.

2 Variational Auto-Encoder Theory

The variational auto-encoder is a Bayesian deep learning technique, which learns latent data representations, even in the presence of large amounts of data and intractable posterior distributions.

In the context of variational inference, we aim to find an approximation to an intractable probability distribution in a class of tractable probability distributions. In a Bayesian setting, we assume that data $x_i, i = 1, \ldots, n$ is generated by an unobserved continuous latent random variable z via the likelihood $p_\theta(x|z)$ and the prior $p_\theta(z)$. These are assumed to be parametrized by θ and to be unknown Gaussian distributions. We also assume the posterior distribution $p_\theta(z|x)$ and the marginal likelihood $p_\theta(x)$ are intractable, so we wish to approximate them with parametric families of Gaussian probability distributions $q_\phi(z|x)$. In variational inference in general, one solves this problem by maximizing the so-called *evidence lower bound* for $p_\theta(x)$, which is given by:

$$log(p_\theta(x)) \geq -D_{KL}(q_\phi(z|x)||p_\theta(z)) + \mathrm{E}_{q_\phi}[log(p_\theta(x|z))], \tag{1}$$

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

where D_{KL} is the Kullback-Leibler divergence. In the context of the variational auto-encoder, we employ a deep auto-encoder architecture, which encodes input x to a lower-dimensional representation z and then decodes it back to x, to estimate the functional components of the right-hand side of equation 1. On the one hand, $-D_{KL}(q_\phi(z|x)||p_\theta(z))$ corresponds to optimizing the auto-encoder's encoder to provide a compact representation of input data x as latent variable z. On the other hand, in a conflicting optimization objective, we also optimize the auto-encoder's decoder to plausibly replicate the input x given z with equation 1's term $\mathrm{E}_{q_\phi}[log(p_\theta(x|z))]$. Since equation 1 represents a lower bound, we encode it as the training objective the deep auto-encoder architecture will maximize in training, with the guarantee that as long as this objective function is growing, then the parameter estimation is getting more accurate. In particular, the so-called reparametrization trick allows for end-to-end optimization of the auto-encoder via commonly used stochastic gradient descent methods, despite the probabilistic setting. For more details, we refer to Kingma and Welling [3].

3 Experiments

Dataset Description. A semiconductor manufacturer provided our dataset, which consists of 22 different post-production tests performed on the semiconductor devices of 284 wafers from one semiconductor product. Each wafer test has a correspondence to a two-dimensional ellipsoid called *wafer pattern*, which captures the test's values per (x, y)-coordinate region of the wafer. In the semiconductor production process these wafer patterns have an associated visualization, which is a two-dimensional heatmap of the test's value per wafer coordinate.
Pre-Processing. Our input wafer pattern test data exhibits many of the problems commonly encountered in real-world datasets: missing values and outliers. Furthermore, each wafer test has its unique purpose (for different electrical measurements) and thus its unique scale. We cope with these issues by first replacing a wafer test's missing values with the test's median value. Then, we scale each test's value by subtracting the test's median from it and dividing that result by the difference of the test's 75th and 25th percentiles. We found this missing value imputation and scaling function to be robust against outliers for our data. Finally, we plot the test's values on a gray-scale heatmap image with a resolution of 112x112 pixels. In total, our dataset includes 6248 pre-processed wafer pattern images, which we feed to our variational auto-encoder.
Variational Auto-Encoder Architecture. As previously discussed, our variational auto-encoder consists of the encoder and the decoder. Since we deal with image data, we employ *convolutional layers* in the encoder (and, respectively, deconvolutions in the decoder) due to their ability to extract higher level feature representations of image data. The encoder has four convolutional layers, followed by a fully-connected layer of 128 neurons into another fully-connected layer representing the two-dimensional latent variable z. The decoder then consists of two fully-connected rectified linear unit layers that upsample the latent variable, followed by three deconvolutional layers and a final convolutional layer

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

that map the activation volumes back to the original input image size. We choose the rectified linear unit activation function for every layer and the loss function is the variational auto-encoder's loss described in 2. As far as the convolutional layers are concerned, we employ 128 filters of size 3x3 with a stride of two. For visualization purposes, we set the number of dimensions of the latent variable portion to two. We deployed all code related to this work in Python v3.5.2 and the deep learning library Keras v2.0.6 with tensorflow v.1.2.1.

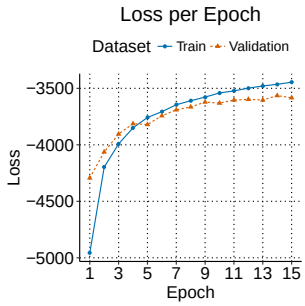

Fig. 1: **Variational auto-encoder loss function values per training epoch for the train and validation datasets.** The validation loss stabilizes at a slightly lower loss than the train loss, indicating convergence and good generalization of the trained model.

Training Performance. In our training procedure of the variational auto-encoder, we use a batch size of 10 wafer patterns and split 80% of the data into a train set and 20% into a validation set. Training the variational auto-encoder for 15 epochs yields the results of Figure 1, which plots the variational auto-encoder's loss function evaluated at the train and validation datasets separately per epoch. The validation loss starts converging after 8 epochs and remains in the vicinity of the train loss for the remaining epochs. Training the variational auto-encoder for more than 15 epochs did not improve the train set error remarkably, thus we deduce the trained model generalizes well to the validation set.

Wafer Pattern Visualization. To visualize the patterns learned, we leverage the decoder's property of having learned to generated wafer patterns from their latent two-dimensional representation. To that end, we first generate uniformly distributed samples from $[0, 1]^2$, i.e. the inverse cumulative distribution function of the two-dimensional Gaussian random variables learned in the latent layer. Then, we evaluate the decoder's output at those points, and the generator outputs the patterns we see in Figure 2. This plot serves as a visual aid for experts to glance at wafer test data patterns present in a semiconductor product.

Wafer Pattern Clustering Application. Figure 2 shows a smooth transition between optically different types of wafer patterns, with some patterns appearing to be rings and others best described as waves. We aim to derive these pattern types in a clustering experiment. Therefore, we interpret the encoder as a dimensionality reduction algorithm, which maps the wafer pattern images

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

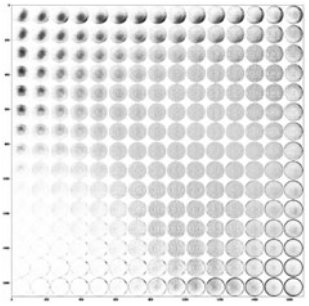

Fig. 2: **Patterns generated from the decoder.** Samples from the decoder allow for visualization of all learned patterns at once and hint at qualitatively different pattern types.

Projection	K	Avg. Silh.
Variational Auto-Encoder	5	0.57
PCA	4	0.48
T-SNE	10	0.43
None	10	0.13

Table 1: **Clustering performance comparison.**

to the two-dimensional latent space. We then cluster the two-dimensional point cloud with k-Medoids, searching for a value of k (between one and ten), which maximizes the average silhouette coefficient per cluster. Given a point i in a cluster, the silhouette coefficient, a measure combining cluster cohesion (a) and cluster separation (b), is defined as $s_i = (b_i - a_i)/max(a_i, b_i)$, where $-1 \leq s_i \leq 1$ and where values closer to 1 are better.

We build a comparison of dimensionality reduction algorithms using, as a comparison measure, the clustering performance as given by the average silhouette coefficient criterion per cluster. Thus, we project the images to a two-dimensional space with Principal Component Analysis (PCA) and t-SNE [6] and, to establish a comparison baseline, we also consider no projection at all. We summarize the results of the k-Medoid clustering search for the number of clusters k and the optimal average silhouette coefficient value in Table 1. We highlight the strong performance in clustering the encoder's output with its average silhouette coefficient of 0.57 for $k = 5$ clusters: the best result among all considered methods, especially with respect to the no-projection case ("None").

Hence, we visualize the point cloud generated by the encoder and clustered into five clusters of wafer patterns by k-Medoids in Figure 3. These families of wafer patterns exhibit visually distinct characteristics. They correspond to different types of wafer test patterns, interpretable by an expert in this semiconductor product and its wafer patterns. As an example of an expert interpretation, Figure 3 depicts a contrast between wafer production issues caused by testing or by other reasons: The pattern of cluster medoid 5 (middle left) represents test

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

issues induced by regular movement of the test needle along curved lines.

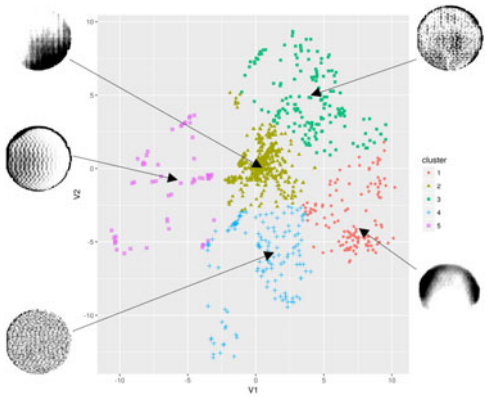

Fig. 3: **Clustered point cloud and medoids.** We depict the two-dimensional clustering performed on the encoder's output. The five medoids shown in the figure correspond to wafer test data patterns interpretable by an expert.

4 Conclusions

In this work, we set out to visualize and cluster wafer patterns, images from tests performed after semiconductor device production. To that end, we chose to learn this so-called wafer test data with a variational auto-encoder, since its benefits are two-fold: Its generator allows for a compact visualization of the input data, and its encoder can be used as a dimensionality reduction method. In a comparison of the encoder's output with other dimensionality reduction algorithms, the former performed best in a clustering experiment of reduced-dimensional wafer pattern representations and yielded interpretable wafer pattern clusters.

References

[1] Hamideh Rostami, Jakey Blue, and Claude Yugma. Equipment condition diagnosis and fault fingerprint extraction in semiconductor manufacturing. In *Machine Learning and Applications (ICMLA), 15th IEEE International Conference on*, pages 534–539, 2016.

[2] Ki Bum Lee, Sejune Cheon, and Chang Ouk Kim. A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, 30(2):135–142, 2017.

[3] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations (ILCR)*, 2014.

[4] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv:1401.4082*, 2014.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.