2.4

Automated Anomaly Detection Through Assembly and Packaging Process

Saad Al-Baddai[1], Martin Juhrisch[2], Jan Papadoudis[1], Anna Renner[2], Lippmann Bernhard[1], Cristina De Luca[1], Fabian Haas[1] and Wolfgang Schober[1]

[1]Infineon Technologies AG, Germany
[2]Symate GmbH, Germany

Abstract

In the semiconductor industry the desired quality and effectiveness in the process of assembling integrated circuits is nowadays at the limit and without safety margin. To achieve important competitive advantages, this process must be continuously optimized and adjusted. Such process is indeed strongly dependent on parameters that are distributed among various control technology assemblies, materials, and the environment. However, the current inspection tools deployed for defect detection through assembly and packaging process are mainly based on rigid and simple rules. The latter are handcrafted by engineers, which can only extract shallow features. Therefore, the accuracy of classification by tools is quite low, which provides incomplete information for root cause investigation and can cause yield-loss costs due to over reject. Hence, automatic inspection tools for visual defect detection, acting as final quality gate before shipping to end customers is very demanding. Therefore, a deviation detection model based on machine learning is developed. On the other side, due to the lack of existing labelled images, an anomaly detection is proposed, in some cases as an assistant tool for collecting defect images with less effort. Results show that artificial intelligent (AI) solutions can achieve a better performance than the classical tools and overcome the human ability in detecting the deviation in the data.

Hence, AI can be used for decreasing the yield-loss, improving quality of the product and greatly reduce labour intensity.

Keywords: artificial intelligence, semiconductor industry, image classification, wirebonding, deep learning, anomaly detection.

2.4.1 Introduction and Background

Semiconductor manufacturing has an increasing complexity and demand on quality requirements, as electronics increasingly become an important part of modern society. In principle, semiconductor manufacturing is equipped with lots of sensors to monitor the processes, but it lacks a suitable way to make use of this data. Thus, new methods are needed to support quality and engineering personal at finding deviations during production to avoid costly production losses or even worse, complaints by customers. Machine learning based anomaly detection (AD) can be a powerful tool to indicate single outliers, but also systematic changes in processes and / or materials. In a next step those deviations can be analysed to label the data indicating a root cause for the different types of deviations. Therefore, one of the success factors in optimizing the industrial processes is either automatic anomaly detection, supervised learning or both, which leads to prevention of production flaws, improving the quality, increasing yields and making more benefits.

The most popular way of performing anomaly detection in many industrial applications is by adjusting digital camera parameters or sensors during the collection of either images or time series data. This is basically an image or signal anomaly detection problem that is searching for patterns that are different from normal data later on at test phase [9]. By assumption, humans can easily manage such tasks by recognizing normal patterns, but this is not as easy for machines. Unlike other classical approaches, image anomaly detection faces some of the following difficult challenges: class imbalance, quality of data, and unknown anomaly [9]. A prevalence of abnormal events is generally an exception, whereas normal events account for a significant proportion. Some techniques usually handle the anomaly detection problem as a "one-class" problem. Here models are learnt by using the normal data as truth ground and afterwards are evaluated whether the new data belongs to this ground truth or not, by the degree of similarity to the ground truth [18]. In the early applications of surface defect detection, the background is often

modeled by designing handmade features on defect-free data. For example, Bennatnoun et al. used blobs technique [5] to characterize the correct texture and to detect deviations through changes in the charter ships of generated blobs. While Amet et al. [1] used wavelet filters to extract different scales of defect-free images, then extracted the informative features of different frequency scales of images. However, most of these methods can work with homogeneous data of good quality and would require prior knowledge. But in most of real applications, this is not the case. Here, the deep learning approaches are used. One variant of common deep learning, which is used for anomaly detection, is the auto encoders (AEs), as they have unique reconstruction property.

The latter can map the input data non-linearly into a low-dimensional latent space and reconstruct it back into the data space. These models are then learned in an unsupervised fashion by minimizing input and output errors [3, 4, 12]. For time series data, the anomaly detection has a similar goal and issues alike:

- Difficulties connected to definition of normal regions, especially in regions close to boundaries.
- In many domains, normal behaviour develops gradually, and an ongoing position of normal pattern cannot guarantee its usage as sufficient proxy on another time step.
- Depending on application field, different parameter fluctuations are considered as normal, so there is no universal pattern or system, which does directly allow using techniques developed for one application to another.
- Absence of labelled data.
- Challenges connected to removing noise from data, which could be mistaken as anomalies [7].

Due to these above-mentioned challenges unsupervised anomaly detection on multi-dimensional data is a very important problem in machine learning and business applications [13].
In this article we will show two examples, how we make use of AD to

1. Detect deviations and
2. Generate further benefit by applying AD such as:
    a. Setup control
    b. Material control
    c. Labelling deviations for supervised learning

d. Compare different equipment regarding process stability and matching

The first example is based on sensor data from the wire bonding process and the second is based on images of the product. For both examples, different approaches were evaluated regarding accuracy and usability in production. First implementations showed that relevant outliers can be found, labelled, and used for subsequent supervised modelling. Additionally, the anomaly detection helped the production and engineers to find systematic influences and derive process improvements based on the new data insights from the anomaly detection. The defect would happen either in early processes or after the chip completed all the process including wafer fabrication, assembly and final test. Technically, the recorded data during sequence processes is collected in a time series fashion for some process or as images for others. Such data has fluctuations, noise, bad quality and high resolution. However, the defect is relatively small and hard to detect even manually. Unfortunately, the built-in software algorithm has a poor classification performance due to rigid and simple rules. So, the specification for inspection is very tight because no defective chips are allowed to ship to customers. As a result, a huge amount of good chips is scrapped, causing unnecessary yield loss cost. Moreover, there is another challenge for defect detection in productive environment if the production environment is dynamic, which means that the data quality is always strongly inconsistent. But also, to detect new defect types which have not been seen before is challenging but important for production.

In summary, the following section will describe the development of an IT infrastructure for anomaly detection in process chains. The aim is to develop an industrialised solution for the detection and visualization of anomalies in different process – using wire bonding and optical outgoing inspection (OOI) as examples. If necessary, with subsequent notification of the user about critical analysis results via e-mail/output signals. Basis of the development and visualization in anomaly detection is the work on wire bonding and OOI image data as well as further demo data.

2.4.2 Dataset Description and Defect Types

For wire bonding data, the data consists of a set of 369 experiments, each of which is described via 432 features (coming from 3 different sensors) during 143 timestamps. However, the features are highly repetitive

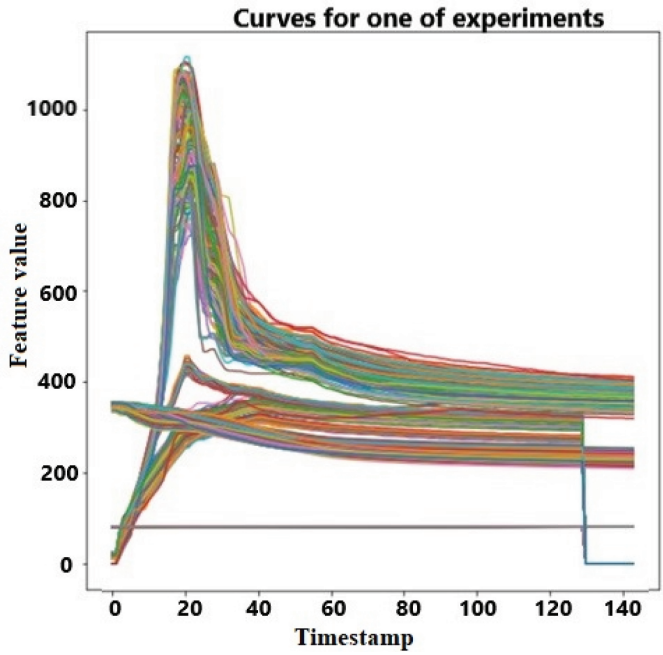

Figure 2.4.1 Curves for one experiment.

(see Figure 2.4.1). This is because there are multiple bond connections on one device, which share the same process parameters and behave quite similar. The three sensors are a current sensor, located at the transducer, a displacement sensor measuring the deformation of the wire and a frequency sensor, also located at the transducer of the wire bonder.

Changes in the raw data can have multiple reasons and are not necessarily known prior. However, most prominent are defects based on contamination of the device or a misadjusted machine, which can cause misaligned or deformed bonds. Some of the defects are shown in the following figures.

Already here enough deviations were found and labelled to enable a supervised training, which will be tested on new and historical data. Further developments were carried out based on Outliergram. It is also based on comparing the shapes of functions. Intuitively, the idea is to inspect how much time the curves spend above and between other curves from the dataset. The outliers are detected by inspecting the relationship between those two values for each of the curves. The results are presented in Figure 2.4.3.

The methods described require the pairwise comparison of all samples in the dataset. In some cases, this may be too expensive. If those methods produce meaningful results, they can be used to filter datasets before training an outliers-sensitive model, e.g., PCA or autoencoder on the rest of the dataset. Furthermore, the reconstruction error from those models could be used to detect outliers as it is less expensive to compute than the pairwise methods.

The second example is dealing with images which are basis for decision if a product has critical deviations and should be scrapped. As the availability of labelled images in a high yield manufacturing is low, AD can help to find critical devices. The further down presented procedure is in principle the same as for the wire bonding, however the used methodologies are more adapted to image data.

The last production step before packing is always the electrical test and a final optical outgoing inspection (OOI) to check that the product is free of visible defects. In the given use-case, a semiconductor power module needs to be inspected from two sides using two monochrome cameras and multiple light sources. The task of the inspection is to check the module at three areas: Leads, mold body and heatsink. Leads and mold body are very consistent in their optical appearance and the images can be checked using classical, rule-based algorithms. These are not considered in this use-case.

The biggest challenge of the optical inspection is the defect detection on the heatsink, see Figure 2.4.4, which consists of a rough copper surface. It needs to be inspected for scratches, metal, or mold particles as well as for mechanical damage like imprints. However, this surface shows a very high variety in appearance, as it is oxidized during preceding high temperature testing steps. Hence, the inspection cannot be carried out using rule-based algorithms, as the oxidized areas cannot be distinguished clearly from true defects by a rule-based algorithm. In this context, trained personnel took care of the heatsink inspection and was used to label the image data for supervised learning. The image data consists of four images per module and side, recorded with a different combination of light sources. Coaxial and diffuse lighting are used to highlight contaminations and particles on the heatsink whereas low-angle lateral lighting is used for detecting mechanical defects such as scratches or imprints in the surface, see Figure 2.4.4.

Also, for visualization purpose, two metrics are used: modified band depth (MBD) and modified epigraph index (MEI). The outliergram visualises

the relationship between these two metrics. The normal curves define a parabola in the two-dimensional space, see Figure 2.4.3. With some thresholds regarding quantiles, some outliers, which are too far away from the parabola (see Figure 2.4.2) can be identified.

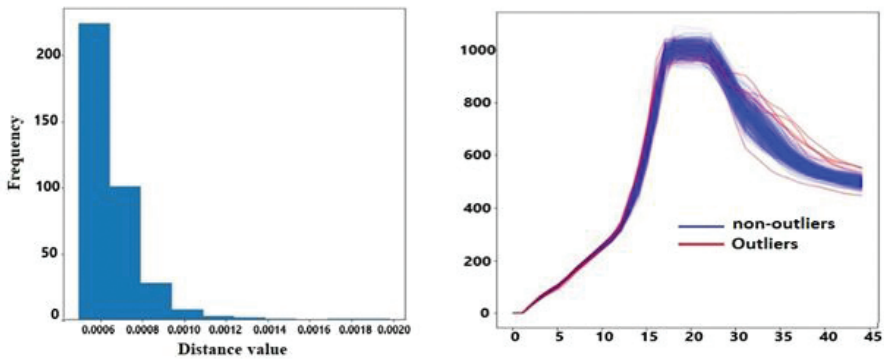

Figure 2.4.2 Left: Distribution of average curves distance to other samples. Right: the results are showed in left by using Wasserstein distance outliers.

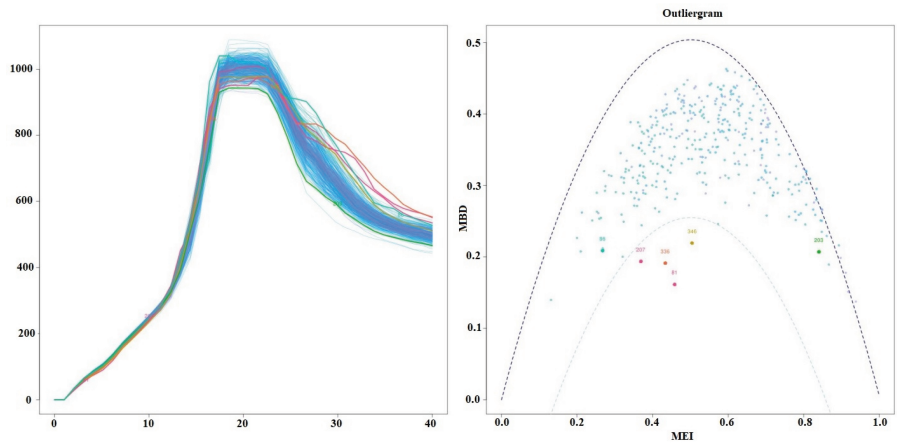

Figure 2.4.3 Outliergram, an example of feature for device current traces. Outliers are detected by inspecting the relationship between MEI and MBD.

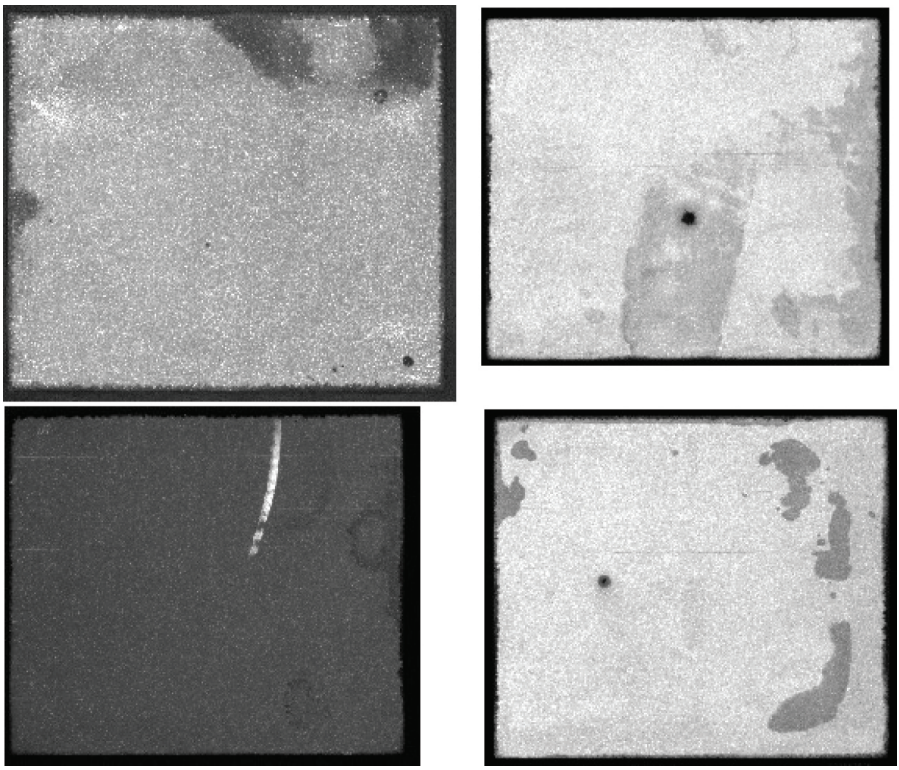

Figure 2.4.4 Shows samples of OOI use case. Top left: particle in lower right corner (bottom side). Top right: particle in centre of image (top side). Bottom left: particle in centre of image (top side). Bottom right: scratch in upper area of heatsink (top side). Note that bottom side is larger than top side.

2.4.3 Methodology

In this work, we used absolutely pure anomaly detection for the first use case and combined AD with supervised learning for the second use case. Hence, we apply the following scenarios:

- For wire bonding use case, Warstein distance outlier is applied.
- For optical outgoing inspection (OOI), two approaches are considered:
    a. Anomaly detection, using pre-trained DL algorithms, was used first in order to reduce effort of labelling data.
    b. Afterwards, the labelled data were used for training a convolutional neural network (CNN).

2.4.3.1 Anomaly Detection

Anomalies are defined as events that deviate from the standard, rarely happen, and don't follow the rest of the "pattern", see Figure 2.4.5. In general, anomaly detection algorithms (ADA) can be classified into two types:

- **Outlier detection**: In this case the dataset consists of both good and abnormal units. Here ADA tries to find the optimal region boundaries of the training data, where the good units are most concentrated and therefore isolating the abnormal units. Such algorithms are often trained using unsupervised learning [6] (i.e., without labels). This type of detection can detect global outliers [2], contextual outliers [8, 10], or collective outliers [8]. However, sometimes, such methods could be used as a pre-process for datasets before applying additional machine learning techniques [11].
- **Novelty detection**: Unlike outlier detection, which includes examples of both normal and abnormal units, novelty detection algorithms have only the normal units (i.e., no anomaly events) during training phase. These algorithms are trained with only labelled examples of good units (semi-supervised learning). At inference phase, novelty detection algorithms must detect when an input data point is far (deviate) off to the good ones.

Generally speaking, outlier detection and novelty detection is a form of unsupervised learning. In this study we introduce a new version of anomaly detection called pseudo anomaly detection (PAD). The latter is indeed a supervised learning algorithm, which can be employed to do unsupervised learning (anomaly detection).

2.4.3.2 Pseudo Anomaly Detection

Following the definition of AD, the idea behind PAD is to simply follow the same definition by using an existing pre-trained algorithm like Alex [16], Resnet [17], GoogleNet [18] etc. Those pretrained algorithms are already trained on a benchmark called the ImageNet dataset [14]. The latter has labels of up to 1000 classes. To cluster the unlabelled data into different categories, under the assumption that prevalence of the defects is very low with respect to the whole population, the expected outputs is to map the good images (majority) to a specific category (one or subset of 1000 classes), within they should have some similarity. On the other hand, the scrapped images (minority) would be distributed over other categories. Such scrapped images

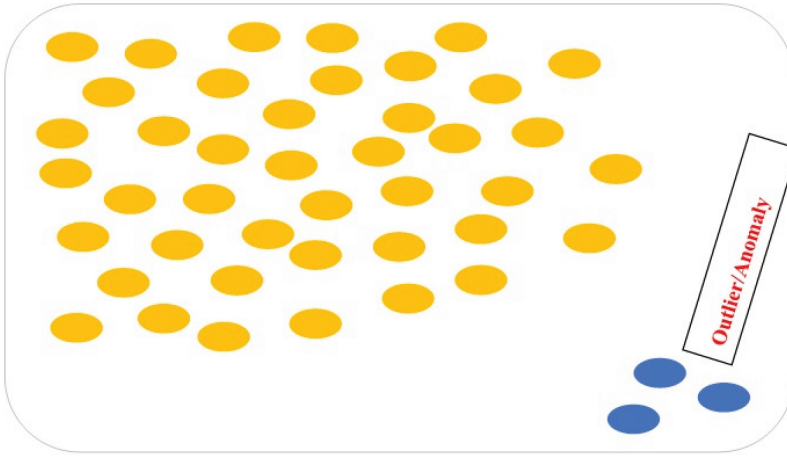

Figure 2.4.5 Show an example of outliers (anomaly) cluster which is clearly inconsistent with the rest of the dataset.

will show up but will happen with an incredibly small probability. Here, these images are reviewed by an expert. In this way the effort for labelling images was reduced by roughly 85%. Please note that names of classes as you can see in Figure 2.4.4 and Figure 2.4.5, represent the original names of the classes, which was used during training of such algorithms as supervised learning (names of real objects). However, in this work, we employ such algorithm as unsupervised algorithms for our data if they don't belong to any of these classes. As, a result, we suppose most good units have similar patterns and would map to a one or few real classes. However, from a machine learning perspective, this makes detecting anomalies hard — by definition, in case we have massive amounts of good images and few bad images of "anomaly" units, but which have a uniform distribution in our dataset. How are anomaly detection algorithms, which tend to work optimally with balanced datasets, supposed to work when the anomalies we want to detect might only 0.2% based on prevalence assumption? Luckily, in our case PAD could figure out the similarity within good images and map them to only a few categories. This is very helpful to reduce the effort for labelling defect images, see Figure 2.4.5.

For wirebonding, a method was developed for the detection of possible outliers. First attempts were done using dimensionality reduction techniques and tests of new approaches, which could smooth out possible anomalies and then, search for new approaches to analyse each feature separately.

Only the results for single feature are presented, however if the adopted approach provides meaningful results, it could be extended to the whole dataset. The planned methodological approach was to find the curves that had different shapes than the others. To compare the shapes of curves we utilised Wasserstein distance which estimates how much work should be done to transform one distribution into another. For each curve in the dataset, we computed the average of its distances to all the other curves.

Based on the histogram in Figure 2.4.2, a threshold value is selected (threshold = $1.2e^{-3}$) to detect the curves that differ much from the other.

On the other hand, the anomaly detection for the wire bonding process was integrated into the process monitoring system from IFX with an additional visualization to quickly see the status of the machine in the anomaly detection. The machines were sending the data via the SECS/GEM interface to a central IFX system which combines different data sources to a unified format and sends the data to the IFX APC-System. The anomaly detection can access this data and calculate the anomaly score. The result of the anomaly detection system is then also stored in the IFX APC-System. This is done by creating a file with the appropriate unified format containing the anomaly detection result and storing in on a network share, where the APC-System access the data and integrates it. In this stage the visualization process can be done by accessing the data independent from the anomaly detection calculation. The data flowchart for wire bonding case can be seen in Figure 2.4.9.

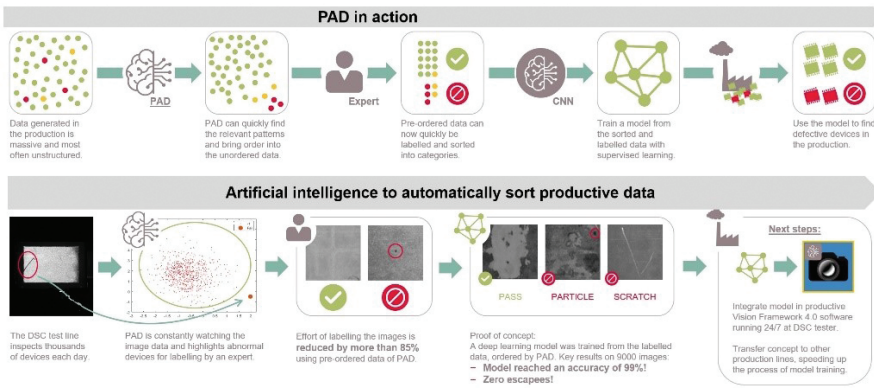

Figure 2.4.6 Shows the process flow for the whole process including PDA and supervised learning applied on optical images.

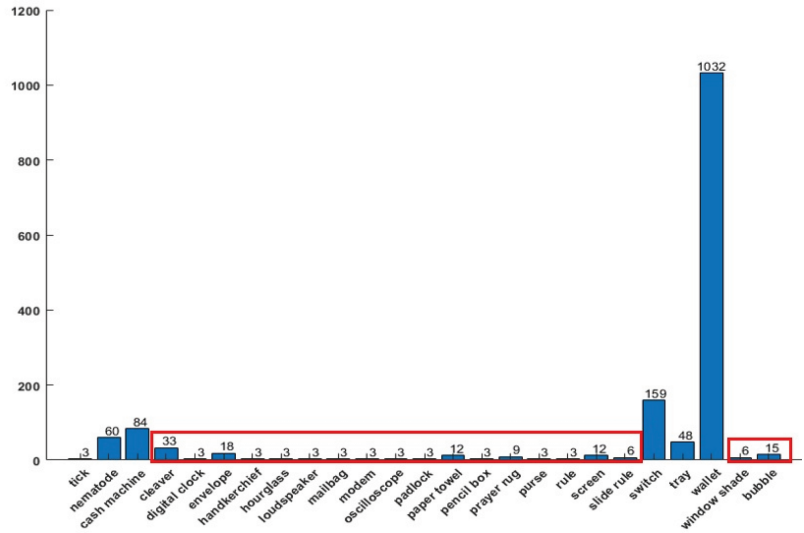

Figure 2.4.7 Anomalies exist at the marked area. In this study, anomaly detection with pre-trained algorithm Resnet was conducted.

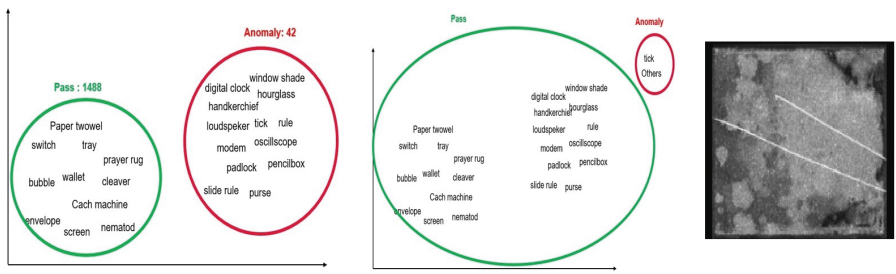

Figure 2.4.8 Shows an example of clustering anomalies units. Left: shows the clustering according to PAD. Middle: shows clustering after review process by an expert. Right: shows an example of defect units which recognized as a tick by PAD. As it shown, names represent the real names of classes of labelled images of ImageNet dataset.

2.4.3.3 Convolutional Neural Networks

Recently, deep neural networks (DNNs) have shown superior performance in a wide range of image processing tasks. We shortly summarize the most common variant of deep learning algorithms, which is called sequential convolutional neural network (SCNN): The primary purpose of the sequential convolution operation is to extract local features from the input image at various spatial scales. Convolution preserves the spatial relationship between

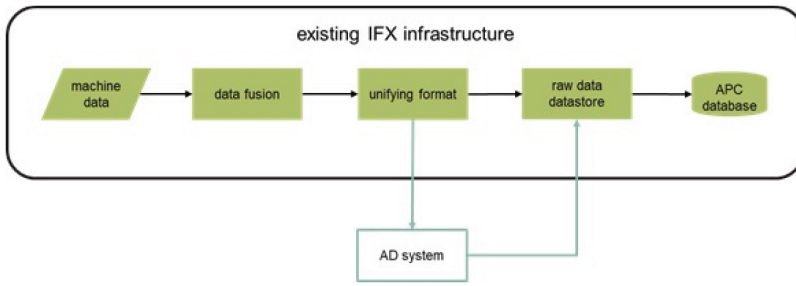

Figure 2.4.9 Shows the process flow of wire bonding use case.

pixels by learning image features using small patches of the input data. In CNN terminology, a 4×4 image patch, is called, for example, a captive field or filter kernel or feature detector, and the matrix formed by sliding the local filter over the whole image and computing the dot product of the filter weight with the input image intensity is called the convolved feature or activation map or the feature map. Each such feature map acts as input to the subsequent convolutional layer. It is important to note that filters act as feature detectors extracting various features from the original input image. As a result, the most relevant features are kept and less relevant features are suppressed. Let us suppose that an image $\mathbf{X}$ is defined by the following mapping:

$$\mathbf{X} : \{1, \ldots, M\} \times \{1, \ldots, N\} \to W \in \mathfrak{R}, (i, j) \to \mathbf{X}_{i,j} \tag{2.4.1}$$

Such an image $\mathbf{X}$ is represented by an array of size $M \times N$. Given a filter $\mathbf{F} \in \mathfrak{R}^{(2k_1+1)\times(2k_2+1)}$ the convolution of the image $\mathbf{X}$ with the filter kernel $\mathbf{F}$ is computed as:

$$(\mathbf{X}*\mathbf{F})_{r,s} := \sum_{u=-k_1}^{k_1} \sum_{v=-k_2}^{k_2} F_{u,v}\, X_{r+u,s+v} \tag{2.4.2}$$

Where the filter $\mathbf{F}$ is given by

$$\mathbf{F} = \begin{pmatrix} F_{-k_1,-k_2} & \cdots & F_{-k_1,k_2} \\ \vdots & F_{0,0} & \vdots \\ F_{k_1,-k_2} & \cdots & F_{k_1,k_2} \end{pmatrix} \tag{2.4.3}$$

However, in addition to convolution layers there are several common layers, which can be used with CNN such as rectified linear units (ReLU), pooling layers (either max or average) and fully connected layers. The latter is

corresponding to the traditional multi-layer perceptron network and is conventionally applied in the last stage of the CNN.

In this study for OOI use case, a CNN structured was created from scratch with 170 layers and 3 branches. A common set of hyperparameters as follows: number of epochs =3, initial learning rate (ILR) = 0.0001, mini-batch size = 64, and the stochastic gradient descent with momentum (SGDM) optimizer is employed.

2.4.4 Results and Discussion

For wire bonding use case, the anomaly detection system was running in parallel to production for several weeks. As it is difficult to validate the anomaly detection during production, since a difference in the raw data might result in a wide range of different impacts on the product, two different approaches to validate the system were made. The first one was to simply calculate the percentage of devices which showed an anomaly in the dataset and compare this to the process yield. If these percentages align, this is a good indicator that the anomaly detection represents the product quality. During our tests this was the case. As a second approach we gathered multiple devices which showed a high anomaly score and examined them thoroughly. In all of the cases different influences could be found on the device, like a contaminated device, reduced shear value or input material which was out of specifications. However, score indicating how different the raw data is from normal, an important aspect of the used anomaly detection was that the result is an anomaly and not a Boolean indication anomaly / no anomaly. Thus, it is necessary to find a threshold on which the difference in the raw data influences the quality of the product. It might be possible to find this threshold automatically if labelled data is available.

For OOI use case, PDA was running on roughly 12000 images. From this historical data PAD could reduce effort for labelling by more than 85%. This enabled an expert to go through only the rest of suspicion images and categories this portion to the real defects and real over-reject (good images). Roughly 130 images were recognized as defect images. Here, the same number of good images was used for training the CNN model to avoid imbalance issues during training process. Furthermore, relative few defect images were available during the training process, a strict regularization was considered to avoid the over-fitting issue by adding a dropout layer with 0.5 parameter. However, remaining of good images were used for test purposes. But first, we split the data into 80% for training and 20% for validation.

The accuracy was 99% for sensitivity as well as for specificity. That means only 1% should be historically reviewed but also periodically during run the model in productive data. Importantly, to follow zero defect philosophy, which means that only images without any defect are sent out to a customer. The threshold of the confidence level is set higher than 95% in order to report good images. On the other hand, this leads to an increase in the over-reject rate to roughly 2.5%. In this way, the model was tested on productive data with roughly 24000 images. An expert also manually reviewed the latter. The accuracy was robust with 98% and zero escapee. Overtime, more defect images are collected, and the model is updated to reduce the over-reject. Moreover, the model was transferred to run on the BOT side of the same product. Here, no available labelled images of this side are used for training. But there is sort of similarity between both sides. Only a bit of adaptation was done as a pre-processing on BOT images due to the difference in terms of reference points and resolution. The accuracy on BOT side was 97% as well.

2.4.5 Conclusion and Outlooks

In summary an AI solution consisting of a combination anomaly detection (unsupervised learning) and supervised learning are used for detecting deviations in semiconductor processes. In this work, it was demonstrated how AI can efficiently solve real-world problems in the industrial setting. The results are promising and would be a good alternative for classical approaches. As a results yields will be increased significantly, the quality will be improved, and the effort will be reduced as well. The next steps is to monitor, optimise and validate both solutions over time, but also integration of AI models into the productive environment. Additionally, the long-term goal is not only find the deviation but also to detect the exact type of defects like scratch, particle in case of images and to point out the root cause in case of wirebonding.

Acknowledgment

This work is conducted under the framework of the ECSEL AI4DI "Artificial Intelligence for Digitising Industry" project. The project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The JU receives support from the European Union’s Horizon 2020 research

and innovation programme and Germany, Austria, Czech Republic, Italy, Latvia, Belgium, Lithuania, France, Greece, Finland, Norway.

References

[1] A. Amet, A. Ertuzun, and A. Ercil. Texture defect detection using subband domain co-occurrence matrices. pages 205–210, 05 1998.

[2] F. Angiulli and F. Fassetti. Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets. *TKDD*, 3, 01 2009.

[3] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. 04 2018.

[4] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders, pp. 372–380, 01 2019.

[5] A. Bodnarova, M. Bennamoun, and K. Kubik. Automatic visual inspection and flaw detection in textile materials: A review, pp. 194–197, 01 2001.

[6] A. Boukerche, L. Zheng, and O. Alfandi. Outlier detection: Methods, models, and classification. *ACM Computing Surveys*, 53:1–37, 06 2020.

[7] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[8] D. D. and D. Sasidhar Babu. Methods to detect different types of outliers. pages 23–28, 03 2016.

[9] T. Ehret, A. Davy, J.-M. Morel, and M. Delbracio. Image anomalies: a review and synthesis of detection methods. 08 2018.

[10] N. Liu, D. Shin, and X. Hu. Contextual outlier interpretation. pages 2461–2467, 07 2018.

[11] S. Rao, N. Shah, and H. Patil. Novel pre-processing using outlier removal in voice conversion. 09 2016.

[12] D. Zimmerer, S. Kohl, J. Petersen, F. Isensee, and K. Maier-Hein. Context-encoding variational autoencoder for unsupervised anomaly detection – extended abstract. 07 2019.

[13] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. 2018.

[14] J. Deng, W. Dong, R. Socher, L.-J Li, K. Li and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255.