Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

CHARACTERIZING CUSTOMER ORDERING BEHAVIORS IN SEMICONDUCTOR SUPPLY CHAINS WITH CONVOLUTIONAL NEURAL NETWORKS

Marco Ratusny
Alican Ay
Thomas Ponsignon

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

ABSTRACT

Advancements in the semiconductor industry have resulted in the need for extracting vital information from vast amounts of data. In the operational processes of demand planning and order management, it is important to understand customer demand data due to its potential to provide insights for managing supply chains. For this purpose, customer ordering behaviors are visualized in the form of two-dimensional heat maps. The goal is to classify the customers into predefined ordering patterns on the example of a semiconductor manufacturing, namely Infineon Technologies. Therefore, a convolutional neural network is used. By classifying the customers into preselected ordering patterns, a better understanding on how the customer demand develops over time is achieved. The results show that customers have a certain ordering pattern, but their behavior can be meaningfully classified only to a certain extend due to unidentified behaviors in the data. Further research could identify additional ordering patterns.

1 INTRODUCTION

The semiconductor industry is a competitive market with a wide range of customers from different industry sectors. Semiconductor companies not only have to put a lot of attention on designing and manufacturing innovative products to survive in this highly competitive market, but also induces the need to react to very fast changing market conditions. In the year 2009, the global semiconductor market shrank by nearly 40 percent, whereas it grew over 40 percent in 2010. Therefore, an efficient and flexible supply chain is mandatory for the semiconductor industry (Seitz et al. 2016). The fear of overproduction and excess capacity, especially after the break of the dot-com bubble in 2000 is still present. Not only does the shrinking market demand put pressure on semiconductor companies. Additional complexity arises from both market and production conditions. The semiconductor market is highly volatile with generally short product life cycles, but customers still request high quality and deliveries on time (Gupta et al. 2006). The production is cost intensive, resulting from high investment cost as well as complex and time-consuming production steps (Yagi et al. 2014). A new wafer fabrication plant can cost up to 4 billion USD of which 75% are cost for equipment (Gupta et al. 2006; Hood et al. 2003). Complexity is further increased by steep ramp-ups or -downs, price pressure from competitors and the high product variety (Chien et al. 2011). On top of that customers can adjust their initial orders, even a week before the actual delivery date, despite the fact that manufacturing cycle times of up to six months are a reality. Usually, forecasts from all customers are treated in the same way and no distinction is made, if the originally requested quantities match the ones finally being ordered, i.e. if the demand picture recurrently being communicated on a weekly basis by the customers to the semiconductor manufacturing, e.g. Infineon Technologies AG (short: Infineon), is relatively stable over a period of 26 weeks before the actual delivery happens and no significant deviations are visible. This is important for the manufacturing, as due to legally binding contracts, it has to ensure, that order confirmations are sent to its customers only if manufacturing capacities and material requirements to produce the same are checked from a planning perspective. Currently, there exists one supply chain process at Infineon for all customers and their demanded products and they are treated in the same way in the order management process. The goal is to identify, if customers in the semiconductor industry follow certain ordering patterns and if those patterns can be classified with the help of convolutional neural networks (CNN). This would lay the foundation for a significant improvement potential to avoid starting production for high volume customer orders that finally will never materialize and on the other hand decline demand from other customers due to blocked capacity. Thus, identifying and categorizing Infineon's customers and providing statistics, if an newly made order will increase or decrease over time, could give an indication whether additional customer orders can be accepted and eventually increase the bottom-line.

2 RELATED WORK

Deep learning and especially image recognition has been a huge success lately in many different areas. Also, it is becoming more popular in the semiconductor industry. In the semiconductor manufacturing, image recognition is mainly used for fault and defect pattern classification due to its potential to prevent yield loss. In this paper, we propose a method for recognizing customer ordering behavior patterns. Therefore, CNNs are used as they are the state-of-the-art architecture for image classification tasks.

In the literature, there can be found several variants of the architecture of a CNN. Still, their main components are very similar. Most of the CNN architectures consist of convolutional, pooling, flattening and a fully connected layer in the end. Within the convolutional layers, CNNs have the ability to learn local patterns and can recognize them in any location of the image. Furthermore, the pooling layer is placed between the convolutional layers. In addition, each feature map of a pooling layer is connected to its associated feature map of the preceding convolutional layer. Flattening converts both the 2- or 3-dimensional output of the previous layer into a single long and continuous linear vector. It is required before the flow goes into the fully connected or dense layer. Fully connected layers are responsible for processing simple vector data, stored in 2D tensor shape. Furthermore, every single node is directly connected to each node in both the previous and in the next layer in a fully connected layer. In the end of the architecture of a CNN, there is the last layer activation. This layer establishes useful constraints on the network's output. In most cases, a sigmoid or softmax function is used as they provide the best results. Both functions have the advantage of interpretability of the final results. By using these functions, the results are in a range between zero and one (Chollet 2018).

(Nakazawa et al. 2018) presented a method for abnormal defect pattern detection and segmentation with the help of deep convolutional encoder-decoder neural network architectures. Their approach was able to detect unseen defect patterns from real wafer maps. Generally, the capability of detecting abnormal signals without using real wafers as training data, and instead using synthetic data, is useful because excursion events happen rarely. (Santos et al. 2019) presented two approaches to extract features from analog wafer maps. The first approach is a classical image processing approach based on restoration and specifically engineered features. The second approach is a convolutional auto-encoder. These researches show that image recognition and image classification is widely used in the semiconductor industry, especially for wafer defect patterns. CNNs are here the state-of-the-art technology and help a lot to improve the defect pattern classification tasks. To the authors' knowledge, image recognition is not applied for clustering customers in certain ordering patterns so far.

Furthermore, CNNs are used for time series classification tasks. (Tsantekidis et al. 2017) proposed a CNN that predicts the price movements of stocks. They trained their network on high volatile limit order book data, by directly feeding in the time series data without converting it into images. (Borovykh et al. 2018) adapted the WaveNet architecture, which is a CNN originally used for audio forecasting. To forecast the S&P500 and the volatile index (VIX), the network's structure was simplified by using a ReLU activation function instead of a gated activation function.

Overall, the application area of CNNs is widely spread and provide good results for its tasks. Its strength is definitely in image recognition and image classification tasks. But also for time series classification and regression tasks, as it has been described, the results are nowadays similar to the current state-of-the-art technologies or sometimes even better. Several approaches, like the ones from (Borovykh et al. 2018; Tsantekidis et al. 2017), proved that CNNs can handle non-image data as well. The study of related works has shown that CNNs are not used so far in the semiconductor industry for transferring demand data into images that can be interpreted as a discrete time series.

3 CUSTOMER DEMAND DATA INTO IMAGES

3.1 Heat maps

Without having a visualization of the customer ordering behaviors (COB), it is unknown, in which way the customers are forecasting. Therefore, it is decided to display their behavior. Having a visualization could lead to an increased understanding of the forecasting behavior of each customer and may result in a better forecast accuracy not only in general but also for each customer. Having gained a better understanding, inventory holding costs as well as idle and scrap costs could be decreased. Also, the Customer Logistics Management representatives (CLMs) have a visual forecast of the behavior of the customers for future Delivery Weeks Due. Future changes in the forecasting behavior can be seen in Figure 1.

For converting customer data into two-dimensional colored heat maps, three columns are of special interest. Those are "Forecasting Horizon", "Delivery Week Due" and "Forecast and Orders".

- Forecasting Horizon: Is the difference between the wish date of the delivery and the date of the creation of the forecast
- Delivery Week Due: Is the wish week, when the customer wants to receive his products.
- Forecast and Orders: Is the amount, each customer forecasts for a specific Delivery Week Due in a certain Forecasting Horizon.

Figure 1 illustrates a heat map. On the x-axis, the Delivery Week Due is plotted. On the y-axis, the Forecasting Horizon is plotted. Based on the Forecasting Horizon and Delivery Week Due the corresponding Forecast and Orders value is converted into a color gradient. Therefore, the smallest Forecast and Orders value is converted into the dark green one, the highest value into the dark red one. The value between the highest and lowest forecast is represented in yellow. All other values have a color gradient based on their Forecast and Orders amount between green, yellow and red.

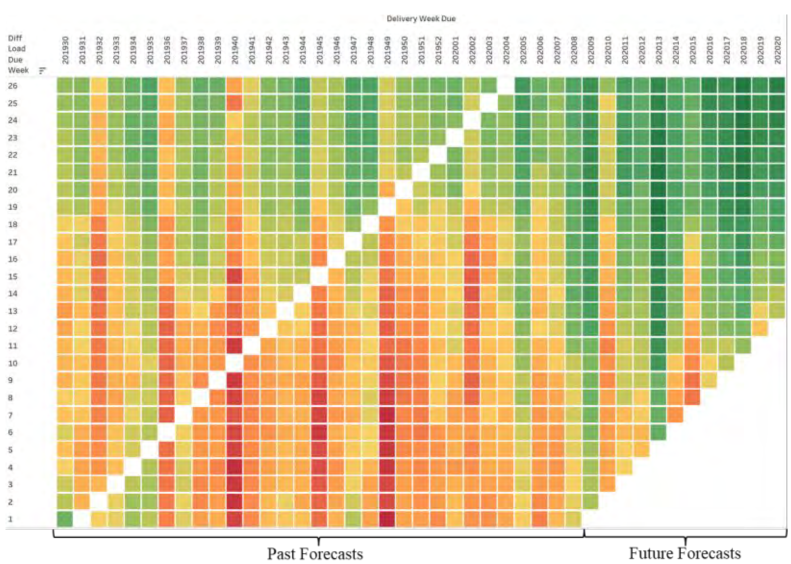

Figure 1: Example of a customer ordering behavior heat map.

3.2 Characterizing Customer Ordering Behaviors

To classify the customers in a later step, first patterns have been defined. This is necessary since classification with a CNN cannot be done without predefined patterns. Constant Planning, Over-, Underplanning, Up-Down and Random are selected as most common patterns. These patterns are certainly not all patterns, which exist in the data, but that is where the focus lies in the current approach. Constant Planning can be described as always ordering the same amount. So, the customer is not changing his forecast amount. Overplanning in general means that the customer decreases his forecasts over time. This is for example done to guarantee the needed delivery. Underplanning can be seen as the opposite of Overplanning. So the customer expects at the beginning a lower amount than he actually needs. When the customer notices that he needs a higher amount of products, he increases forecasts heavily on short notice. Moreover, steady increasing forecasts can also be declared as Underplanning. Up-Down can be described as that the customer starts with forecasting a lower amount at the beginning. He then increases his amount until a certain point. Then, he continues to decrease his forecast again. This behavior does not have to be perfectly symmetric – meaning the extreme point can be closer towards the start week of the Delivery Week Due. Random means that no pattern is directly visible. Here the customer orders in a way, which does not fit any of the predefined patterns. So all images that do not look like any of the before mentioned patterns, should be then assigned to class Random. Figure **2** shows examples of the before mentioned patterns.

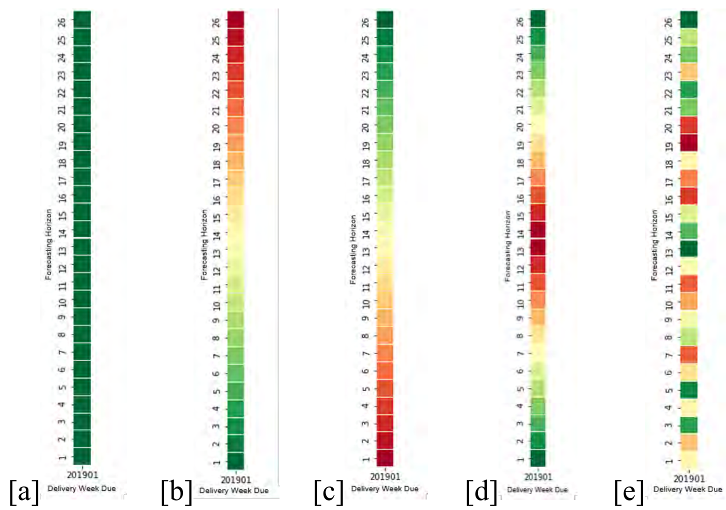

Figure 2: Classification into [a] Constant, [b] Overplanning, [c] Underplanning, [d] Up-Down and [e] Random

4 USING CNN TO RECOGNIZE CUSTOMER ORDERING BEHAVIORS

4.1 Reasons for Choosing Image Recognition and CNN

Initially customer ordering data were transformed into heat maps to provide CLMs a visualization to support their daily work. Thus, the heat maps are existing already and can be directly used to analyze the COB with an algorithm. Therefore, we use images as our input. As we have labeled training data and an image classification task to solve, CNNs are the state of the art machine learning approach to tackle this task. Furthermore, the heat maps can be interpreted as discrete time series, and as described in Section 2, CNNs do offer the possibility to handle discrete-time series, which were properly transferred into images. Provided that a large dataset is available, CNNs outperform every other ML technique in case of image classification, because of their ability to join feature and classifier learning. Additionally, since 2012, CNNs have the highest accuracy by competing in the ILSVRC-competition (Bhandare et al. 2016). However, a simple heuristic could also be used to tackle this task. Due to the reason that the CNN in the end should also be used to solve more complex tasks, like analyzing several delivery weeks at once, a simple heuristic is not enough in the long run when the patterns to identified will get more complex. Regarding those reasons, CNNs are selected as the appropriate machine learning algorithm to solve the image classification problem.

4.2 Absolute Value CNN (AV-CNN)

The input heat map image shape is 815x100x3. 815 (pixels) is the height, 100 (pixels) is the width and 3, in this case, means that the input image is a colored image with RGB coding. The network has three convolutional layers with a receptive field size of 7x7, 5x5, and 3x3. The first convolutional layer has the 32 channels, the second one has the 64 channels, and the last convolutional layer has the 128 channels. Channels, in this case, stand no longer for specific colors rather they stand for filters. Filters encode specific aspects of the input data. The rectified linear activation is used for each convolutional layer. Furthermore, in the first convolutional layer, the border mode is implemented with “same”. This means that there is some padding around the input or feature map, making the output feature map’s size equal to the input size. Each convolutional layer is followed by max pooling with a size of 2x2. After the last max pooling operation, dropout is used. This helps to prevent the CNN from overfitting. After the flattening, a fully connected layer with a size of 64 is added. Furthermore, another rectified linear activation is used before the next dropout. A dense layer is applied with the size of the number of different classes. The last layer is the softmax layer for the class probability calculation. “Categorical Crossentropy” is implemented as the loss function and “adam” as the optimizer. Figure 3 shows the architecture of the AV-CNN.

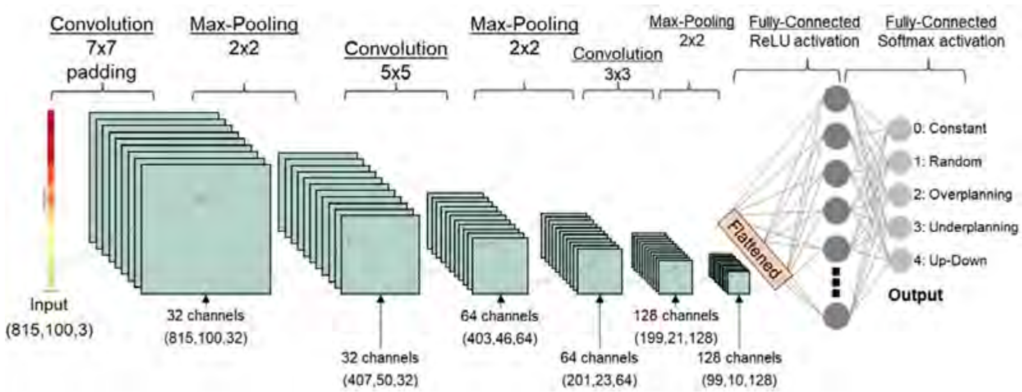

Figure 3: Architecture of the AV-CNN.

4.3 Percentage Value CNN (PC-CNN)

For this CNN, the basic architecture of the AV-CNN is used, because the input images are similar. Still, some fine tuning was conducted to get the best possible result out of the network. So, the difference in this architecture compared to the architecture of the AV-CNN is that the first max pooling operation is realized with a size of 5x5. The size of the first max-pooling operation is bigger in the PC-CNN compared to the AV-CNN, due to longer parts with the same color in the input image. Longer parts with the same color are normally not reflected in the images used by AV-CNN, but in the images that are used by the PC-CNN they are common. Figure 4 shows the architecture of the PC-CNN.

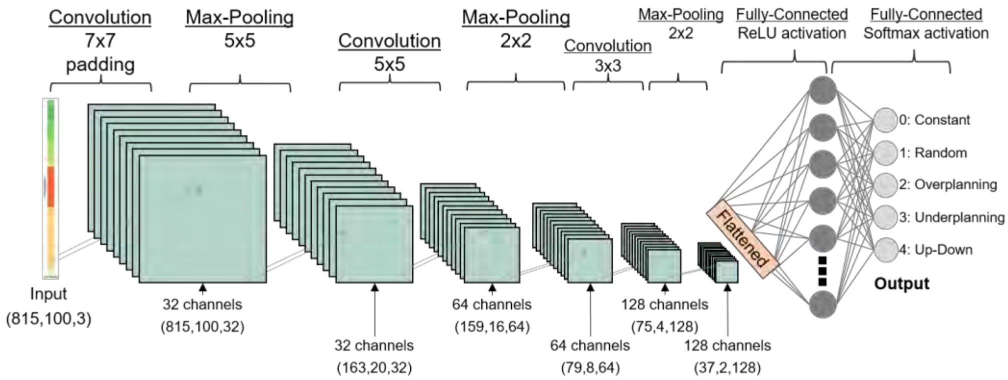

Figure 4: Architecture of the PC-CNN.

5 IMPLEMENTATION AND RESULTS

5.1 Datasets

The AV-CNN is trained with synthetic created images only and validated with real customer data. For training purposes, the whole synthetic data is split into Train and Test data. In total, 6782 synthetically created images are available. 5425 images are used for training purposes and the other 1357 images are used as testing images. For validating the AV-CNN with real customer data, the AVs of the five direct customers of major importance that are at the top level of Infineon's customer hierarchy are used. Data in the time-frame between 01.01.2018 up to 15.07.2019 represent the validation set. Firstly, each Delivery Week Due for each customer that is at the top level of Infineon's customer hierarchy is transferred into a heat map. In total 463 images are created and labeled in the next step. In total, three images are labeled as Constant Planning, 111 as Random, 141 as Overplanning, 113 as Underplanning and 95 as Up-Down. The three images that are labeled as Constant Planning represent no real customer data, instead, they are synthetic images. In real customer data, no Constant Planning is visible and therefore, synthetically created examples are used for the Constant Planning case.

In total 5760 synthetic images were created. They are also split into Train and Test data with a ratio of 80:20. So in total, 4608 images are used for training and 1152 images are used for testing. Real customer data is used for validation. Therefore, also data from 01.01.2018 up until 15.07.2019 was selected, labeled and used for validation. In total, 460 images are labeled. In this case, we have three images less compared to the first CNN, because no synthetically created data is used for validation. 27 images are labeled as Constant Planning, 71 as Random, 150 as Overplanning, 114 as Underplanning and 98 as Up-Down.

5.2 Handling of Missing Data

In contrast to the synthetically produced dataset, the real data, in the form of heat maps, may show missing cells. A missing cell represents a missing forecast and order in a particular Forecasting Horizon of a Delivery Week Due. Missing data have to be analyzed, understood and in the best case imputed in the process of data preparation in order to provide meaningful and processable datasets for our model.

In the course of analyzing missing data it is helpful to distinguish between different types of missingness to make decisions on how to handle a missing data point based on its type. In the missing data literature, there can mostly be found three to four different types of missing data that will be introduced shortly and then compared to the missing cells in heat maps.

Structurally missing data, is given when data are missing for a logical reason, i.e. the data cannot exist in the first place. Therefore, filling in imputed values into these types of missing data does not make any sense, (Lipovetsky and Nowakowska 2013). Another type of missing data is the one missing completely at random (MCAR). This occurs when there is no relationship between the fact that the data is missing and

observed or missing values. Data MCAR is not to be confused with data missing at random (MAR) which categorizes the missing data that has a systematic relationship between the propensity of missing values and observed data, in contrast to MCAR, but not the missing data.
Data missing not at random (MNAR) can, in contrast to MCAR and MAR, have a relationship between the property of a value to be missing and its values, as explained by (Graham 2009).

To deal with the different types of missing data we want to have a look whether we can, and if it makes sense to, impute missing data in the different scenarios. In the following we compare different cases in the heat maps with the different missing data types as explained. A frequently appearing case of missing cells has been when cells are missing in a diagonal pattern as seen in Figure 5. This pattern occurs most likely due to an aggregation process in a database that considers the Delivery Week Due, which produces the diagonal. Due to the randomness it may be that the missing data in this case is MCAR.

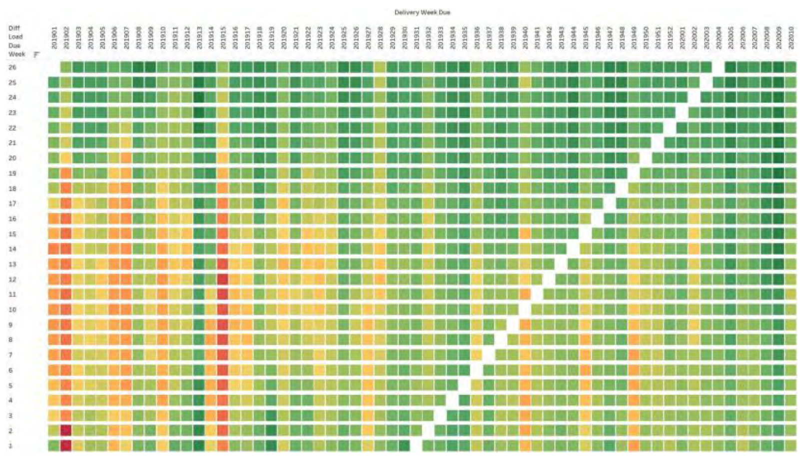

Figure 5: Diagonally missing data.

Another reason why cells may be missing can be due to a customer not reporting forecasts and orders far in the future for the desired Delivery Week Due or when the customer stops announcing forecasts and orders, sometimes very close to the Delivery Week Due. This can be seen in Figure 6 with the colored cells often starting later than the first possible Forecasting Horizon or stopping a few weeks before the corresponding Forecasting Horizon. The reason therefore is that customers do not give Forecasts and Orders very early sometimes. That means that this is a structurally missing type of data where we know why no data was given. So a possibility to impute values would be to leave the missing cells white so the model can learn this pattern or replace the missing cells with the color corresponding to zero orders.

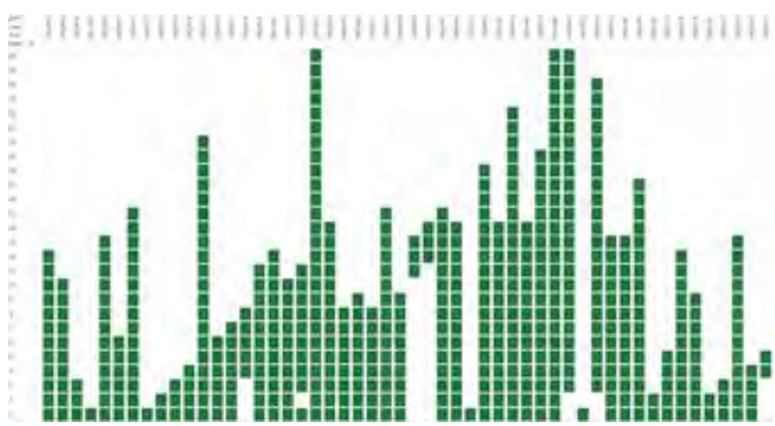

Figure 6: Structurally missing cells.

In Figure 7 we can see missing cells which possibly occur completely at random but are not in a diagonal pattern as in Figure 5.Figure 5: Diagonally missing data.

Figure 7: Data missing at random.

It is important to note that the in different granularities or aggregation levels missing data may not be visible. This is the case when for example different product lines are considered in one heat map.

5.3 Preprocessing

Before feeding the images into the CNN, some preprocessing has to be done. Therefore, the images have to be cropped to remove excess white areas. Without cropping the images the result would be larger data size for each image as well as higher computational power for the CNN and longer training time. So each image is cropped to a size of 815x100 (height x width) pixels. Figure 8Figure 8 illustrates the before mentioned method. On the left, the original image can be seen. The black frame shows the original shape. After cropping the image, less white area is visible and leads to minor computational effort and training time for the CNN.

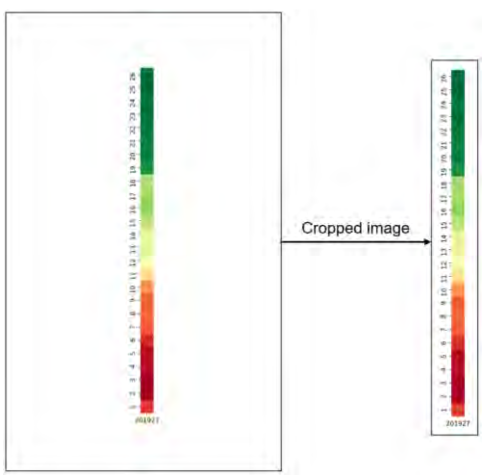

Figure 8: Image Preprocessing**.**

Furthermore, for the PC-CNN we first need to calculate the percentage change compared to the first forecast of the customer (Forecasting Horizon = 26). By applying this, it is now possible to see a constant behavior in certain Delivery Weeks Due. The forecast will now be seen in relative terms and calculated as follows:

$$Percentage\ change = \frac{Forecast\ of\ Delivery\ Week\ X}{Forecast\ of\ Delivery\ Week\ 26} - 1 \tag{1}$$

It is not possible to calculate the change based on the forecast before, because otherwise, the compounded interest would distort the result. After calculating it, each percentage change will be divided into predefined intervals. Table 1 Converting percentage change into intervalsshows the predefined intervals and their corresponding color gradient.

Table 1 Converting percentage change into intervals.

Values	Interval	Color gradient
]-∞ ; -1]	-8	
]-1 ; -0.65]	-7	
]-0.65 ; -0.40]	-6	
]-0.40 ; -0.25]	-5	
]-0.25 ; -0.20]	-4	
]-0.20 ; -0.15]	-3	
]-0.15 ; -0.10]	-2	
]-0.10 ; -0.05]	-1	
]-0.05 ; 0.05]	0	
]0.05 ; 0.10]	1	
]0.10 ; 0.15]	2	
]0.15 ; 0.20]	3	
]0.20 ; 0.25]	4	
]0.25 ; 0.40]	5	
]0.40 ; 0.65]	6	
]0.65 ; 1]	7	
]1 ; ∞[	8	

5.4 Training

For training purposes, Infineon's compute farm is used. The compute farm is based on multiple computers combined in parallel, which appear to the end user as a huge computing system with a large amount of storage. Figure 9 describes the structure of the compute farm.

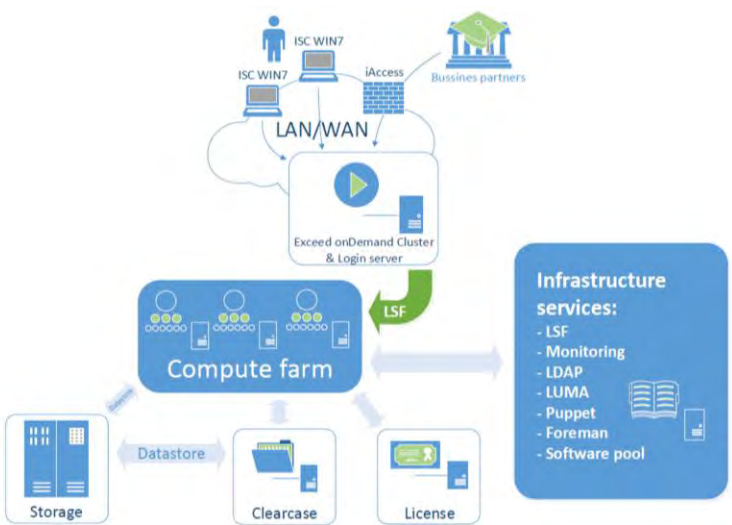

Figure 9: Structure of the compute farm.

Within the compute farm, which is based on a CentOS 7.7.1908 operation system, enough CPUs, as well as enough GPUs, are available to run the program. Because this paper focuses on a CNN based approach, GPUs are necessary to decrease the computational burden. One Tesla T4 with 16GB of GDDR6 is used for training the neural network. For training the AV-CNN and the PC-CNN, three training epochs resulted in the best results and each training epoch is about 20 seconds long.

5.5 Results

After the AV-CNN is trained, it predicts the label of each image of the Test data. The network performs well on synthetic data and has an accuracy of 98.45% and a loss of 0.04. In real customer data, no Constant Planning is visible and therefore, synthetically created examples are used for the Constant Planning case. The Constant Planning images are predicted correctly by the CNN in 100% of the cases, while the Random and Up-Down images are only predicted in a correct manner in 81%. Overplanning is predicted accurately in 87% of the cases, while Underplanning is predicted correctly in 94% of the cases. Overall, the accuracy of the real customer data was 86.39%. Figure 10 shows the results of the AV- and PC-CNN with synthetic and real data.

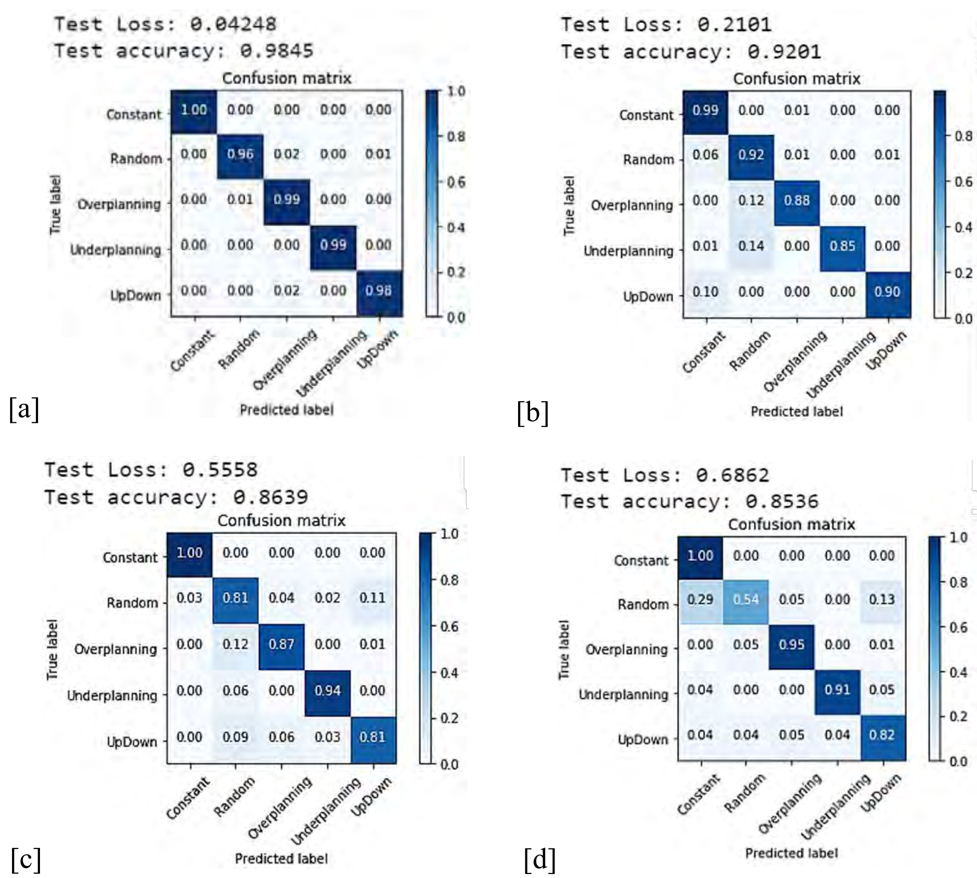

Figure 10: Confusion matrixes for [a] synthetic images with AV's, [b] synthetic images with PC's, [c] real data with AV's and [d] real data with PC's.

In total, the CNN performs also well on synthetic data. Its accuracy is 92.01% and has a loss of 0.21. The 27 Constant images are predicted correctly in 100% of the cases. Over-planning is predicted correctly in 95% of the cases, while Underplanning was predicted correctly in 91%. Up-Down was predicted rightly in 82% of the cases, whereas for Random labeled images in 54% of the cases only. In 29% of the cases, Random images are predicted as Constant Planning. This is due to the fact, that the synthetically created images for Constant Planning also contain some variances. Furthermore, wrong predicted labels contain longer periods of constant behavior. Then the net classifies them wrongly, because the highest accuracy for a class is used, and for those cases, Constant Planning has the highest accuracy. Further synthetic data have to be created so that Random images would be classified more accurately. Overall, still an accuracy of 85.37% is reached for this CNN.

It is visible, that the AV-CNN has a higher accuracy and also a lower loss. This could be due to the higher amount of training data. In total, 817 images more are used for training compared to the PC-CNN. Either by creating more training data for the PC-CNN or by using some data from the test set, the accuracy could be increased or the loss could be decreased.

6 CONCLUSION AND NEXT STEPS

While one of the most researched topics regarding image recognition in the semiconductor industry is the fault classification of wafer maps, this paper focuses on a supply chain management topic and addresses the issue of changing demand signals directly received from customers on the example of Infineon Technologies AG as semiconductor manufacturing. More precisely, it deals with the classification of the forecasting behavior into predefined patterns to gain a better understanding of the same and takes those findings into account. We have shown that CNN based approaches are able to automatically identify customer ordering behavior patterns in the heat maps. Still, there are options to improve this approach, which is discussed below. One way to enhance this approach is to retrain the network with real data. Then the network could learn some features, which are not in the synthetic data, but in the real customer data, so the CNN does not have to be completely trained again. Currently, there is a limitation on the availability of manually labeled data as the effort to this is very high.

The other way how to enhance this approach is to identify more patterns that occur in the data. During the case study, it became apparent that there have to be more patterns within the data as the Random pattern exists to a bigger extent. Therefore, an unsupervised learning (USL) approach, like image or time series clustering, could be useful to first identify the existing patterns in the data. This, in the end, leads to more possible classes wherein the customers could be classified. But, the major advantage of first selecting the possible classes with the help of a USL approach is, that Infineon can understand the customers more detailed based on real data. This approach could also be extended by analyzing a heat map that contains data for a whole year. By doing this, new patterns that are only visible by looking into several Delivery Weeks Due at once become visible. This helps to understand if certain patterns occur only at the beginning of a month for example. Furthermore, it would be then possible to produce certain amounts of products in advance. This would help in times of capacity bottlenecks. Also, other effects, like the product lifecycle, can have a high influence on the forecasting behavior of the customer. Here, it is interesting to see, how a customer behaves, shortly before the product will be replaced by a newer version. Therefore, this information is needed and could be implemented by using a second stream in the CNN, where context information is provided. Also by using a second stream, the forecast amount itself could be provided. This could increase the information gain because then it is possible to compare products based on their forecasting amount. This would help to weight customers in the factory. Customers with a higher forecasting amount, which were transformed in the end into real orders, could be the basis for a different treatment. In this approach, only colored images were used. As the training data is small and enough computational power is available, it is sufficient to use colored images for now. However, if the training data increases and needs more computational power, a comparison between colored images and greyscale images could be useful. Greyscale has the advantage that the data size of each image decreases and therefore, the performance of the CNN would increase, as less computational power is necessary. But it has to be tested, whether necessary information is lost or not. Lastly, a benchmark model needs to be developed to guarantee, that the CNN based approach performs better. For the benchmark model, a simple heuristic, a Random Forest Classifier, a Multilayer Perceptron Classifier or a Gradient Boosting Classifier could be used for analyzing heat maps that contain either one or several Delivery Weeks Due.

Additionally, an extrapolation of the existing figures (see Figure 1) could be used to add another dimension to improve customer forecasts.

REFERENCES

Bahandare, A., M. Bhide, P. Gokhale, and R. Chandavarkar. 2016. "Applications of Convolutional Neural Networks". *International Journal of Computer Science and Information Technologies* 7(5):2206-2215.

Borovykh, A., S. Bohte, and C. W. Oosterlee. 2018. "Dilated Convolutional Neural Networks for Time Series Forecasting". *Journal of Computational Finance* 22(4):73-101.

Chien, C.-F., S. Dauzère-Pérès, H. Ehm, J. W. Fowler, Z. Jiang, S. Krishnaswamy, T. E. Lee, L. Mönch, and R. Uzsoy. 2011. "Modelling and Analysis Of Semiconductor Manufacturing in a Shrinking World: Challenges and Success". *European Journal of Industrial Engineering* 5(3):254-271.

Chollet, F. 2018. *Deep Learning with Python*. Shelter Island New York: Manning Publications Co.

Graham, J. W. 2009. "Missing Data Analysis: Making It Work in the Real World". *Annual Review of Psychology* 60(1):549-576.

Gupta, J. N. D., R. Ruiz, J. W. Fowler, and S. J. Mason. 2006. "Operational Planning And Control Of Semiconductor Wafer Production". *Production Planning & Control* 17(7):639-647.

Hood, S. J., S. Bermon, and F. Barahona. 2003. "Capacity Planning Under Demand Uncertainty For Semiconductor Manufacturing". *Institute of Electrical and Electronics Engineers Transactions on Semiconductor Manufacturing* 16(2):273-280.

Lipovetsky, S., and E. Nowakowska. 2013. "Modeling with Structurally Missing Data by OLS and Shapley Value Regressions". *International Journal of Operations and Quantitative Management* 19:169-178.

Nakazawa, T., and D. V. Kulkarni. 2018. "Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network". *Institute of Electrical and Electronics Engineers Transactions on Semiconductor Manufacturing* 31(2):309-314.

Santos, T., S. Schrunner, B. C. Geiger, O. Pfeiler, A. Zernig, A. Kaestner, and R. Kern. 2019. "Feature Extraction From Analog Wafermaps: A Comparison of Classical Image Processing and a Deep Generative Model". *Institute of Electrical and Electronics Engineers Transactions on Semiconductor Manufacturing* 32(2):190-198.

Seitz, A., H. Ehm, R. Akkerman, and S. Osman. 2016. "A Robust Supply Chain Planning Framework For Revenue Management In The Semiconductor Industry". *Journal of Revenue and Pricing Management* 15(6):523-533.

Tsantekidis, A., N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis. 2017. "Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks". In *2017 IEEE 19th Conference on Business Informatics: Proceedings: Volume 1: Conference Papers*, edited by P. Loucopoulos, Y. Manolopoulos, O. Pastor, B. Theodoulidis, and J. Zdravkovic, 7-12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. .

Yagi, D., K. Nagasawa, T. Irohara, H. Ehm, and G. Yachi. 2014. "Semiconductor Supply Planning By Considering Transit Options To Take Advantage Of Pre-Productions And Order Cancellations". *Simulation Modelling Practice and Theory* 41:46-58.

AUTHOR BIOGRAPHIES

MARCO RATUSNY is a PhD candidate in the Corporate Supply Chain organization of Infineon Technologies AG in Munich, Germany. He received his B. Sc. (2018) in Information Systems and Management and his M. Sc. (2020) in Stochastic Engineering in Business and Finance at the University of Applied Sciences Munich, Germany. His research interest is in demand planning and machine learning. His email address is marco.ratusny@infineon.com.

ALICAN AY is a working student in the Corporate Supply Chain organization of Infineon Technologies AG in Munich, Germany. He earned his B.Sc in Mathematics at Technical University of Munich and is currently pursuing a Master's degree in Mathematics at Technical University of Munich. His research interests focus on statistical learning theory and stochastic analysis. His email address is alican.ay@infineon.com.

THOMAS PONSIGNON is as a Senior Staff Engineer in the Corporate Supply Chain organization of Infineon Technologies AG in Munich, Germany. He obtained masters degrees in Industrial Engineering from the EPF-Ecole d'Ingénieurs, Sceaux, France and the University of Applied Sciences, Munich, Germany and a Ph.D. in Mathematics and Computer Science from the University of Hagen, Germany. His research interests include production planning and supply chain simulation. His email address is thomas.ponsignon@infineon.com.