Anticipating faults by predicting non-linearity of environment variables with neural networks: a case study in semiconductor manufacturing

Mateus Begnini Melchiades [1] **Lincoln Vinicius Schreiber** [2] **Gabriel de Oliveira Ramos** [2]
Cesar David Paredes Crovato [3] **Rodrigo Ivan Goytia Mejia** [3] **Rodrigo da Rosa Righi** [2]

Abstract

The present work proposes a neural network model capable of anticipating possible faults in a semiconductor manufacturing plant by predicting non-linearity spikes in sensor data. Early detection of significant variation can be crucial for identifying machinery degradation or issues in the process itself. We use non-linearity as it is not affected by regular process changes and autocorrelation, thus avoiding false-positives in the neural network caused by changes in demand and the presence of control systems. The developed model is able to predict up to 30min of future non-linearity with loss ≤ 0.5. Furthermore, the proposed model is flexible enough to present itself as a starting point for future work in the field of fault detection in other areas.

1. Introduction

Recent advances in computing capabilities and sensor manufacturing costs led to an increase in the utilization of sensors in manufacturing plants for process quality monitoring and insight generation. This trend is incorporated in the umbrella term Industry 4.0 and allows companies to detect deviations in environment variables that could lead to issues in the future. Deviations in a manufacturing process often occur over the span of many months and can be imperceivable for a human being. Consequently, companies have been adopting strategies for production forecasting for decades, mostly in the form of mathematical models like ARIMA as a way of reducing product cost and losses (Yaffee & McGee, 2000). Modern approaches that attempted to apply neural networks as an alternative to these mathematical models only obtained partial success, proving superior to existing strategies exclusively when handling discontinuous data (Hill et al., 1996). Moreover, work in continuous time series forecasting with neural networks is still sparse and often specific to the nature of the studied variable.

In this work, we present a methodology for identifying machine degradation-induced behavior based on the evolution of the non-linearity observed on its sensors. More specifically, our case study centers the attention to DRAM production in a semiconductor manufacturing company which has recently invested in Industry 4.0 projects with the installation of environment sensors in several areas of its production plant. We concentrated our work on forecasting significant variation in two classes of sensors: dew point and deionized water resistivity, which are considered by the company as the most critical for identifying machine degradation and process quality issues.

A system can be described as non-linear if it cannot be expressed by a linear combination of derivatives of its input and output, which is the case for most systems in the real world (De Canete et al., 2011). In other words, a system is non-linear when it is affected by external influences. In a closed loop environment such as a manufacturing plant, small changes in non-linearity can occur in the form of noise from outside sources such as temperature changes and power oscillation; however, a sudden increase usually indicates problems in the process itself. The proposed model uses an adapted calculation for non-linearity we entitled Normalized Linear Offset (NLO) to predict 30 minutes of oscillations in the future by using 30 minutes of values in the past. More importantly, the model is sensitive to unexpected variations while capable of ignoring expected changes such as the ones described in Section 2; thus, we believe it can be used in future work as a basis for fault prediction in other areas.

[1]Department of Undergraduate Studies in Computer Science, Universidade do Vale do Rio dos Sinos - UNISINOS, São Leopoldo, RS, Brazil [2]Graduate Program in Applied Computing (PPGCA), Universidade do Vale do Rio dos Sinos - UNISINOS, São Leopoldo, RS, Brazil [3]Professional Master's Program in Electrical Engineering (MPEE), Universidade do Vale do Rio dos Sinos - UNISINOS, São Leopoldo, RS, Brazil. Correspondence to: Mateus Begnini Melchiades <mateusbme@edu.unisinos.br>.

Proceedings of the LatinX in AI (LXAI) Research workshop at ICML 2021.

2. Problems with sensor data forecasting in modern manufacturing

Most modern manufacturing processes—including our case study—contain numerous elements capable of interfering with analysed data and cause unpredictable behavior. One such element is the presence of control systems, ubiquitously found in factories, whose primary purpose is to control variability to reduce waste and rework, thus resulting in higher yield (Shunta, 1997). These systems trigger actuators which directly affect the variables we intent to forecast, adding unpredictability to the training set if used "as is". Figure 1 demonstrates how a potential fault is corrected by the control system before it could demonstrate a predictable pattern.

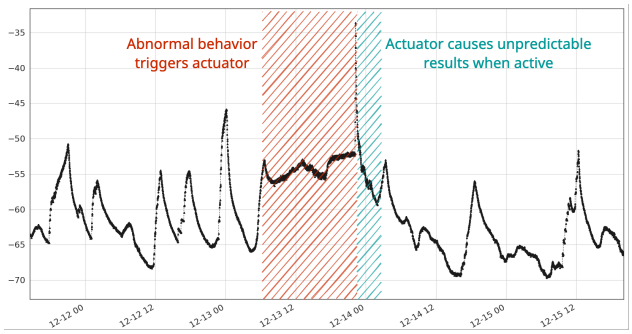

Figure 1. Example of unpredictable behavior in sensor data.

Another detrimental element for for applying fault forecasting in the observed data is the presence of autocorrelation caused by the nature of the measurement. This autocorrelation—present in Figure 1 in the form of spikes—has a short cycle and significant variability, and represents the production cycle of a specific machine. Such behavior can easily be learned by a neural network, therefore the sudden absence of spikes might be flagged as a fault by a prediction algorithm, when in most cases it only indicates a machine has finished a cycle.

One last noteworthy source of unpredictability are shifts in the data's mean as production demand increases or decreases. When a factory operates at full capacity, its associated environment measurements will behave differently than when production is slow due to lower demand. For example, a machine that idles for an hour before being used again can demonstrate a temperature curve with variability higher than one being used 100% of the time and still be perfectly normal. In summary, the process can and will change overtime and a prediction model must be able to differentiate between a cycle change and an abnormal value.

All the behaviors mentioned above are normal for most manufacturing processes, but a neural network fed unprocessed data might think otherwise. Thus, we must apply some transformation to the data capable of ignoring expected shifts in value while retaining (and emphasizing) unexpected ones.

3. Oscillation forecast with non-linearity and LSTM models

As mentioned in Section 2, attempting to forecast variation by predicting sensor data directly would not yield expected results due to production cycles and influence from the plant's control system. A suitable alternative is to feed the neural network with the sensor's non-linearity as explained in Section 1, since its presence in a time series is often a common cause of variation (Thornhill et al., 2001).

In a nutshell, our approach works as follows. Once sensor data is obtained, we estimate non-linearity by using the Linear Offset method explained in Section 3.1, which returns a time series with the same size as the original resampled data, normalized using rolling Z-Score. This non-linearity series, denominated Normalized Linear Offset, is fed into an LSTM model described in Section 3.2. Figure 2 visually demonstrates the data pipeline throughout the application.

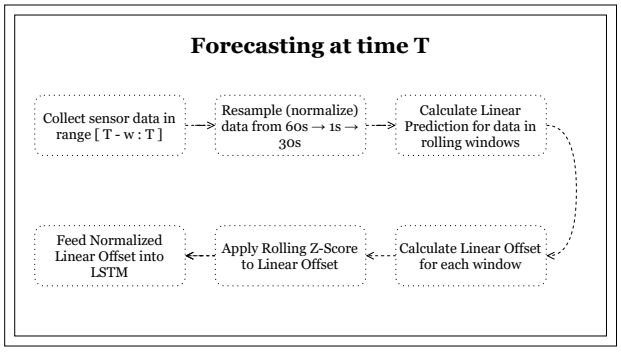

Figure 2. Data pipeline throughout prediction.

3.1. Non-linearity in time series as Linear Offset

Assuming non-linearity indicates variation, determining the predictability of the series allows us to detect deviations from the expected data. One such method to determine predictability is the average forecast error (Hegger et al., 1999). In order to obtain a series with the same size as the original data, we compute the average forecast error based on rolling windows of the sensor data with w points each. Inside each window, we calculate the root mean square error of the difference between the window's real data and its respective linear prediction:

$$E_t = \sqrt{\left(\frac{1}{w}\sum_{k=t-w}^{t} x_k - \hat{x}_k\right)^2}, \qquad (1)$$

where $x_k,\ 1 \leq k \leq w$, is the is the window's k^{th} real value, $\hat{x}_k,\ 1 \leq k \leq w$, is the window's k^{th} point predicted using linear prediction, and E_t is the forecast error for the t^{th} window.

The linear prediction algorithm chosen is an autocorrelation method of the all-pole model, which has been widely adopted in the field of time series analysis for decades (Makhoul, 1975). This model, although intended for use in autoregressive signals, is often found in a diverse range of applications (even non-autoregressive) due to its ability to provide an adequately precise representation for a variety of time series. The autoregressive version employed in our application uses the all-pole method in a windowed signal to circumvent the issue of having a finite dataset (Hayes, 2009).

Before sending the data to the model, we apply a rolling z-score to the Linear Offset, which normalizes the data and removes any large spikes caused by periods of very small variation in the dataset. This Normalized Linear Offset series is then randomly separated into 3 series: train dataset, comprised of approximately 80% of the data; validation dataset, with approximately 10% of the data; and test dataset, with the remaining 10% of the data. The series are randomly organized due to the highly diverse nature of our data, meaning that no constant time period would be capable of fully explaining the whole dataset. In this sense, selecting random points in the dataset produces a more accurate representation of the time series.

3.2. Predicting non-linearity with LSTM

Recurrent Neural Networks (RNNs) have proven to be an effective strategy for recognizing patterns influenced by past results, such as forecasting electrical consumption (Connor et al., 1994) or changes in process quality (Pacella & Semeraro, 2007). One limitation however, is that it may not be as efficient when handling larger datasets due to memory loss. As training progresses, RNNs tend to forget past states quickly, hence losing their ability to remember beyond the immediate past. Long Short-Term Memory (LSTM) is a type of RNN with a modified neuron architecture capable of remembering patterns for several time steps by having separate pipelines for long and short term learning (Hochreiter & Schmidhuber, 1997).

Our case study, as previously demonstrated, contains long-term patterns which, if paired with a regular RNN, would not be fully retained by the model. Thus, we consider an LSTM to be the ideal basis for the construction of said model, given the fact that each training procedure receives almost 600,000 data points worth approximately a year of measurements.

Our model consists of a five layer network, where the first two are LSTM layers consisting of 240 neurons each, followed by a 60-unit Dense layer, a Dropout layer with rate 0.5 and, lastly, a copy of the previous Dense layer. This architecture better satisfied both the needs for accuracy (see Section 4) and efficiency, critical for a real-time forecasting application.

4. Results

The model was trained for 100 epochs using the train dataset explained in Section 3.2 and, in the end of each epoch, it was compared against the validation dataset for adjusting parameters for the next epoch. Finally, after the training procedure was completed, we validated the model's ability of predicting a set of values it has never seen before using the test dataset, which yielded predictions like the ones shown in Figure 3.

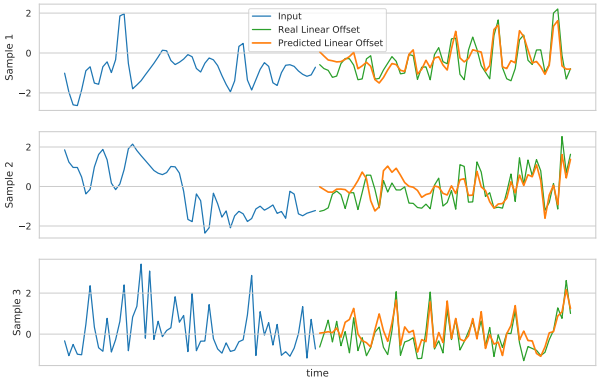

Figure 3. Prediction (orange) from test input (blue) compared to expected values (green).

As seen in Figure 3, we feed the model with 60 past readings (in blue, equivalent to 30 minutes of mono-spaced data), which then returns an array as output (in orange, with the same size as the input). The output array is then compared to the real error for that time period (in green). As can be observed, the network presents good accuracy for the type of data being predicted, presenting a loss of 0.437 and mean absolute error of 0.495. More importantly, however, the model is able to detect all significant spikes in the NLO, meaning that it is able to achieve the main goal of this paper, which is anticipating faults and significant variation.

When compared against real-time readings in a production environment, the model suffered slight hindrances in individual predictions. Figure 4 demonstrates this by comparing the prediction and NLO behaviors for a single point in time (columns 1 and 3) against the span of one hour (columns 2 an 4). As we can see, looking at individual predictions could lead to the impression that the model is making an erroneous forecast but, when we consider a larger time window, it is evident that both curves are indeed similar, if not for a slight

variation in the curve peak caused by shifts in process. Future work could focus on improving the proposed model to better detect and adjust to these seasonal variations in non-linearity.

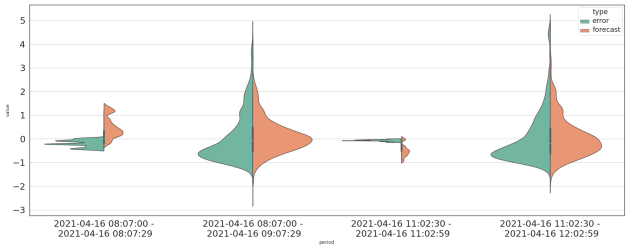

Figure 4. Normalized Linear Offset and prediction dispersion for single-point (columns 1 and 3) and 120-point (columns 2 and 4) intervals.

5. Related work

As mentioned in Section 1, very few papers focus specifically on predicting sensor data with non-linearity and neural networks. However, the literature presents relevant studies which apply NNs to predict non-linearity in other fields such as cholesterol estimation (Sahu et al., 2019) and rainfall forecasting (Chattopadhyay, 2007). These papers shall be discussed below.

In (Sahu et al., 2019), Sahu et al. create an ANN for estimating the amount of Cholesterol oxidase (COX) in a species of Streptomyces by using pH, cholesterol concentration, 4-aminoan-tipyrine, crude COX volume and horseradish peroxidase as input. Despite good results, the work is not focused towards live varying data like manufacturing sensors. Therefore, the created model may not be extended to other application domains.

Chattopadhyay (Chattopadhyay, 2007) used ANNs to forecast average rainfall in India during summer-monsoon. The author developed an accurate network for weather prediction, which, by the nature of the problem it is attempting to solve, only outputs a single value. This contrasts to what would be expected for a time series.

6. Conclusion

In this work, we presented a deep learning model capable of utilizing non-linearity associated with sensor data in order to predict spikes possibly related to early signs of degradation or human error in the context of a semiconductors factory. The proposed network is robust enough to adapt to process cycles and avoid false-positives originated from natural process variation, proving to be a reliable backend tool for the the company used as case study.

Future work in the area should be focused towards better detection of long-term process shifts that can cause behavior changes in the non-linearity, as well as the application of our model beyond semiconductors manufacturing. In conclusion, we believe the proposed model serves as a contribution for future work in fault detection in high-variability areas similar to the one in our case study.

References

Chattopadhyay, S. Feed forward artificial neural network model to predict the average summer-monsoon rainfall in india. *Acta Geophysica*, 55(3):369–382, 2007.

Connor, J. T., Martin, R. D., and Atlas, L. E. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

De Canete, J. F., Galindo, C., and Garcia-Moral, I. *System Engineering and Automation: An Interactive Educational Approach*. Springer Science & Business Media, 2011.

Hayes, M. H. *Statistical digital signal processing and modeling*. John Wiley & Sons, 2009.

Hegger, R., Kantz, H., and Schreiber, T. Practical implementation of nonlinear time series methods: The tisean package. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 9(2):413–435, 1999.

Hill, T., O'Connor, M., and Remus, W. Neural network models for time series forecasts. *Management science*, 42(7):1082–1092, 1996.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Makhoul, J. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.

Pacella, M. and Semeraro, Q. Using recurrent neural networks to detect changes in autocorrelated processes for quality monitoring. *Computers & Industrial Engineering*, 52(4):502–520, 2007.

Sahu, S., Shera, S. S., Banik, R. M., et al. Artificial neural network modeling to predict the non-linearity in reaction conditions of cholesterol oxidase from streptomyces olivaceus mtcc 6820. *Journal of Biosciences and Medicines*, 7(04):14, 2019.

Shunta, J. P. *Achieving world class manufacturing through process control*. Prentice Hall PTR, 1997.

Thornhill, N., Shah, S., and Huang, B. Detection of distributed oscillations and root-cause diagnosis. *IFAC Proceedings Volumes*, 34(27):149–154, 2001.

Yaffee, R. A. and McGee, M. *An introduction to time series analysis and forecasting: with applications of SAS® and SPSS®*. Elsevier, 2000.