Autoencoding Density-based Anomaly Detection for Signal Integrity Applications

Roberto Medico, Domenico Spina, Dries Vande Ginste, Dirk Deschrijver, Tom Dhaene
IDLab, Department of Information Technology, Ghent University - imec, Gent, Belgium
Email: roberto.medico@ugent.be

***Abstract*—This paper presents a machine learning-based framework for anomaly detection (AD) in signal integrity applications. The proposed approach employs autoencoders and density-based AD techniques to detect anomalies and is validated on a digital counter circuit.**

***Index Terms*—Signal integrity, Jitter management, Machine Learning, Anomaly Detection**

I. Introduction

In recent years, signal integrity (SI)-aware design methodologies have gained importance, due to the increase in signals bandwidth and the high level of integration and miniaturization of integrated circuits (ICs). This paper presents a novel machine learning (ML)-based approach for SI applications. The proposed methodology is based on anomaly detection (AD) techniques [1] to identify infrequent deviant events that do not conform to an expected behavior. In the proposed framework, such infrequent events (also called *anomalies* or *errors*) correspond to undesired behaviour of the signals under study, for example due to crosstalk effects, jitter or noise. Once an initial computational effort is made to build the ML model, AD techniques are able to individuate any anomaly in an automated way. Although AD methodologies have been successfully employed in related domains, such as electromagnetic compatibility [2], their application to SI problems is a novel contribution. The proposed framework follows a typical two-step approach for anomaly detection [1]. First, a dimensionality reduction technique is used to extract relevant features from the input data, using autoencoders [3]; next, anomaly detection is carried out on the new representation, using the Local Outlier Factor (LOF) algorithm [4]. This paper is structured as follows: Section II describes the methodology, which is validated in Section III on a digital counter circuit. Conclusions are drawn in Section IV.

II. Anomaly Detection Framework

The methodology consists of three main phases: data preprocessing, training of a ML model for feature extraction and AD based on the features learnt by the model. In this scenario, the method should be able to detect anomalies in the entire output of a digital circuit and locate where and when the anomalies occur.

a) Preprocessing: First, the signals under study (the *raw input data*) must go through a series of preprocessing steps in order to be used as input for a ML model. Since the raw data is a continuous time-series signal, it can be split into multiple subsequences of a fixed length using a sliding window approach.

b) Unsupervised Feature Extraction: Autoencoders [3] are ML models that can be used for unsupervised feature extraction, since they are able to compress the input data into a lower-dimensional representation. Specifically, an autoencoder consists of an encoder network (to compress the input data) and a decoder network (to reconstruct the original data given the latent representation). An autoencoder is typically trained by minimizing a reconstruction loss (e.g. the mean squared error between the input and reconstructed data) via gradient descent. In practice, both encoder and decoder are Neural Networks using one or more hidden layers. In its simplest form, an autoencoder has one input layer, one hidden layer (also referred to as *bottleneck* layer) that learns a compressed representation $\mathbf{h}$, and one output layer, as shown on the right side of Fig.1. After training, the encoder can be used to compress the input data and the learnt representation can be extracted from the bottleneck layer. Since the hidden layer uses a non-linear activation function (typically the sigmoid function), the learnt features are non-linear transformations of the input data. Hence, it is not always possible to associate a concrete physical meaning to the features learnt by an autoencoder. In this work, a variation called Contractive Autoencoder (CA) [5] is used, where a penalty term is added to the reconstruction loss during training to make the network robust to small changes in the input data. The objective function of the optimization during training is:

$$\min_{\mathbf{W_e},\mathbf{W_d},\mathbf{b_e},\mathbf{b_d}} L(\mathbf{x}, d(e(\mathbf{x}))) + \lambda ||\mathbf{J}_e(\mathbf{x})||_F^2. \tag{1}$$

Here, $\mathbf{W_e}, \mathbf{W_d}, \mathbf{b_e}, \mathbf{b_d}$ contain the parameters (weights and biases) learnt by the autoencoder, $\mathbf{x}$ is the input data,

$$e(\mathbf{x}) = \mathbf{h} = sigmoid(\mathbf{W_e} \cdot \mathbf{x} + \mathbf{b_e}) \tag{2}$$

$$d(e(\mathbf{x})) = sigmoid(\mathbf{W_d} \cdot e(\mathbf{x}) + \mathbf{b_d}) \tag{3}$$

are the encoding and decoding function respectively. $L(\cdot)$ is the reconstruction loss and the penalty term is formed by the product of $\lambda \in \mathbb{R}_{>0}$ and the Frobenius norm of the Jacobian of the encoding function:

$$||\mathbf{J}_e(\mathbf{x})||_F^2 = \sum_{i,j} \left(\frac{\partial h_j(\mathbf{x})}{\partial x_i} \right)^2 \tag{4}$$

where $h_j(\mathbf{x})$ is the j_{th} learnt feature for the input $\mathbf{x}$. The objective function in (1) enforces the robustness of the learnt

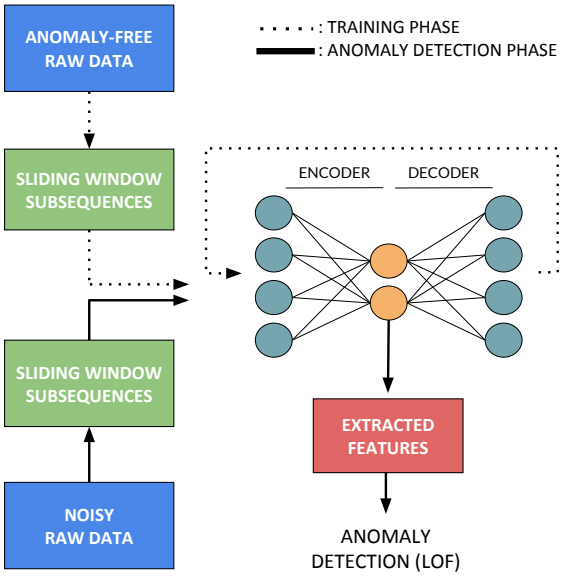


Fig. 1: Training and AD phases of the proposed framework.

representation against small perturbations in the input data, which is very important for this application considering that each extracted subsequence overlaps significantly with neighboring sequences. In the proposed modeling framework, a CA is trained to reconstruct the signal under study when no disturbances are present (error-free data), by minimizing the objective function in (1).

c) Anomaly Detection: After training, the noisy data is passed through the network and the learnt features are extracted from the bottleneck layer. In this phase only the encoder part of the network is used and no further training happens. An AD algorithm can then be used to detect anomalies in the noisy data, using the extracted features. Since these were learnt by training the model to reconstruct anomaly-free data, it can be expected that an anomalous subsequence will be represented by different feature values than a normal one. In this work, LOF [4] is used for the AD phase. LOF is a density-based approach, i.e. it assumes that normal observations lie in dense regions of the feature space, while anomalies lie in areas of lower density. The algorithm considers *local* densities, estimated by computing the density of the nearest neighbours of each data point. With this approach, it is possible to identify outliers that are close to dense areas, but whose local density is much lower than that of their neighbours. In particular, the output of LOF is a positive real number indicating the anomaly score of a subsequence of the data considered. Estimated scores close to one indicate that the subsequences do not contain anomalies, while scores significantly higher than one indicate that more or less critical errors occurred. To obtain a binary indicator 0 or 1 whether a subsequence is normal or anomalous, a suitable threshold τ is defined on the maximum score tolerated before reporting an anomaly. In practice, τ depends on the error criteria for the specific application considered, as described in the next section. A visual overview of the proposed modeling framework is shown in Fig. 1.

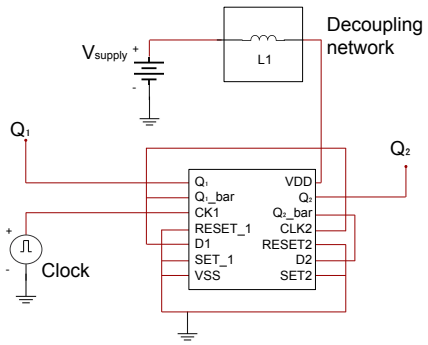
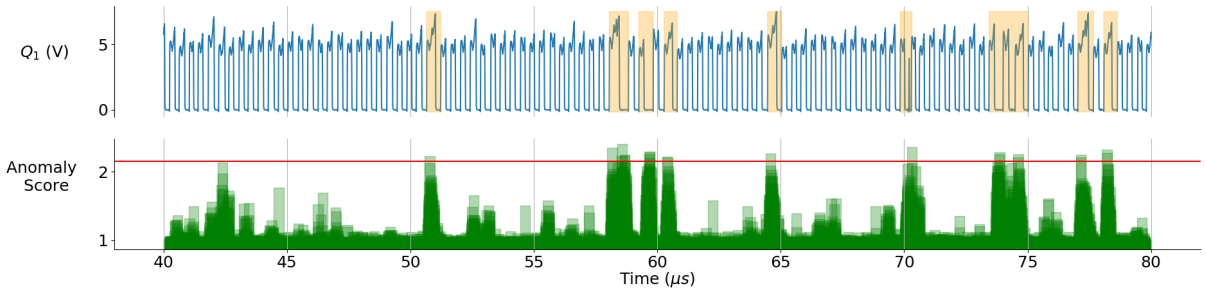


Fig. 2: The digital counter circuit under study.

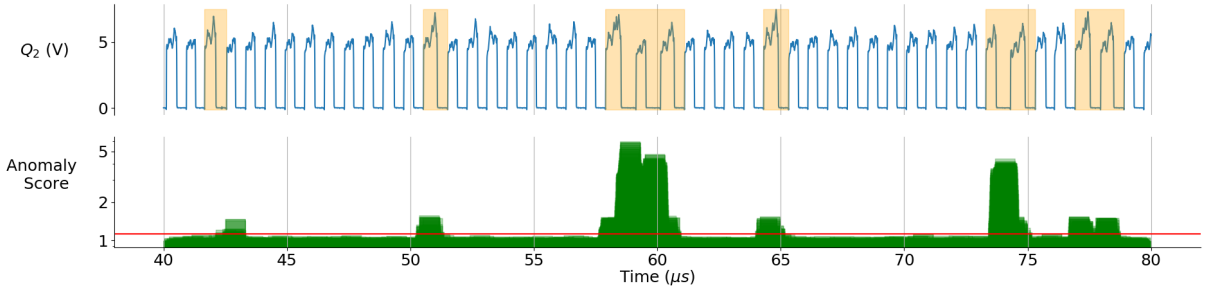
III. Numerical Example

The circuit under study is the digital counter shown in Fig. 2, using a dual D-type flip-flop based on the 74HC74 datasheet [6], and analyzed in ADS[1]. The digital counter has two output signals Q_1 and Q_2, with frequency equal to one half and one quarter of the clock frequency, respectively. The clock has a period of 200 ns, rise/fall time of 5 ns and is affected by a Gaussian jitter having a standard deviation of 20 ns. Our goal is to estimate the effect of the clock jitter on the signals Q_1 and Q_2. In particular, the digital counter is considered robust w.r.t. jitter effects if the following condition is satisfied: the variation of the period ΔT of Q_1 and Q_2 at any time does not exceed 10% of the corresponding nominal value, leading to $\Delta T_{Q_1} = \pm 40$ ns and $\Delta T_{Q_2} = \pm 80$ ns. To verify this condition leveraging AD, our first goal is to train a CA that is able to accurately reconstruct the signals Q_1 and Q_2 when no anomalies are present. This allows to extract the relevant features from those signals. Hence, a time-domain simulation of the circuit under study is performed in the range $[0, 80]$ μs in the absence of clock jitter. Both training and detection phase are carried out on data in the range $[40, 80]$ μs, i.e. once the IC is operating at steady state. First, both signals Q_1 and Q_2 are normalized to $[0, 1]$. The same scaling factors are later on re-used to normalize the signals under the effect of jitter during the AD phase. Next, multiple subsequences are extracted via the sliding window approach described in Section II. The length of such subsequences is a critical parameter for this application, since the autoencoder will try to compress and reconstruct them by learning a set of features. Moreover each subsequence should be long enough to include potential anomalies on the periodicity of the signal. For this example, the subsequences are extracted with a 99% overlap and with length equal to that of the period for both signals: 800 for Q_2 and 400 for Q_1, since both signals are sampled with a fixed timestep of $1ns$. A CA can then be trained for several epochs (one epoch corresponds to one full iteration on the training data during optimization) on the anomaly-free data –

[1]Advanced Design System, Keysight Technologies, Santa Rosa, CA, USA.



(a) Results of the anomaly detection on Q_1.



(b) Results of the anomaly detection on Q_2.

Fig. 3: In each subfigure, the raw output signal is shown above, while the anomaly scores are shown below. The thresholds are indicated with red horizontal lines.

no clock jitter – independently on Q_1 and Q_2 to learn features describing normal behaviour of the circuit. Later, noisy data – with clock jitter – is fed to the trained network(s) to extract the features, and these are used as input for the AD based on the LOF algorithm. The hyper-parameters of the CAs were set as follows: 1 hidden layer with sigmoid activation, 8 and 16 hidden neurons in the bottleneck layer respectively for Q_1 and Q_2; 250 training epochs; $\lambda = 10^{-4}$ in equation (1). Finally, the LOF algorithm is set-up to use 200 neighbours points in the feature space and the Euclidean distance to individuate outliers. Fig. 3 shows the AD results on both outputs Q_1 (Fig. 3a) and Q_2 (Fig. 3b) in the interval $[40, 80]$ μs. The raw output signals are shown on top, and the scores are below. Each score is associated with a subsequence in the original signal, and overlaps with the neighboring scores. The regions in the outputs waveform where the error criterion is not satisfied are highlighted. As illustrated by the figure, the proposed technique successfully identifies all the critical areas for both signals Q_1 and Q_2 by assigning them higher anomaly scores. Since the method also identifies other disturbances in the signals, a threshold τ (represented as a red horizontal line) is chosen a posteriori to isolate the critical regions. In both cases, it is possible to define τ such that *all* critical regions are identified without raising any false alarm (i.e. wrongly detecting a normal subsequence as an anomaly).

IV. Conclusions

This paper proposes the application of AD techniques to detect errors in SI applications. These techniques employ ML algorithms to model the normal behavior of an electronic circuit and automatically detect deviant behaviors in new data by looking at raw output data only, without requiring any prior knowledge on the circuit properties or settings. Specifically, this contribution described an *unsupervised* approach, i.e. one that does not require any example of what an anomaly is or any knowledge on the kind of anomalies that could occur. The proposed two-steps AD framework – which employs CAs to extract features describing the *normal* behavior of a circuit and the LOF algorithm to identify the anomalies in the new extracted features space – was evaluated on a digital counter affected by clock jitter.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, July 2009.

[2] R. Medico, N. Lambrecht, H. Pues, D. Vande Ginste, D. Deschrijver, T. Dhaene, and D. Spina, "Machine learning based error detection in transient susceptibility tests," *IEEE Transactions on Electromagnetic Compatibility, in print*, 2018.

[3] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, March 2013.

[4] M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, USA, 16 - 18 May 2000.

[5] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on Machine Learning*, New York, NY, USA, June 28 - Jul 2 2011.

[6] *Dual D-type flip-flop with set and reset; positive edge-trigger*. Nexperia, 74HC74 and 74HCT74 data sheet, Rev. 5, 2015.