Impact of Soft Errors on High Performance Autoencoders for Cyberattack Detection

Impact of Soft Errors on High Performance Autoencoders for Cyberattack Detection

F. Finelli
University of Bologna
Bologna, Italy
fabrizio.finelli@studio.unibo.it

M. Omaña
Dep. of Electrical, Electronic, and Information Engineering
University of Bologna
Bologna, Italy
martin.omana@unibo.it

C. Metra
Dep. of Electrical, Electronic, and Information Engineering
University of Bologna
Bologna, Italy
cecilia.metra@unibo.it

***Abstract*—Detecting anomalous messages generated by cyberattacks is essential for IoT based critical applications (*e.g.*, finance, healthcare, manufacturing, etc.), in order to guarantee high levels of security and availability. Autoencoder-based anomaly detection has been proven effective in detecting the presence of messages altered by malicious attacks. To guarantee low detection latency, autoencoders' algorithms are typically executed by HW accelerators implemented by nanotechnology. However, such HW accelerators are susceptible to soft errors (SEs), that may occur during their in-field operation. In this paper, we analyze the impact of SEs on the effectiveness of autoencoders in detecting anomalous messages generated by cyberattacks. As an example, we consider autoencoders trained to discriminate credit card legal and illegal transactions. We show that SEs do not decrease system's security, but may reduce system's availability of the 70%, so that proper solutions to avoid the detrimental impact of SEs on system's availability should be devised for autoencoders implemented by high performance HW accelerators.**

Keywords—Autoencoders, Availability, Security, Soft Errors, Anomaly Prediction, Hardware accelerators, Nanotechnology.

I. INTRODUCTION

The advancement of nanotechnology has enabled a significant increase in Internet of Things (IoT) and big data applications using wireless communication networks [1, 2, 3]. Even though wireless networks adopt solutions to protect data against malicious attacks, they are still more vulnerable than traditional wired networks, and some malicious attacks might hack them [2, 4]. In order to guarantee the required levels of availability and security of IoT based critical applications (such as finance, healthcare, manufacturing, etc.), it is of utmost importance to detect, as soon as possible, the presence of messages that have been altered by malicious attacks. It has been shown that such altered messages usually present "anomalous" patterns, that are different from those in legal messages. Thus, messages modified by malicious attacks can be efficiently detected by anomaly detection approaches based on autoencoders that, as known [1, 5, 2, 6, 7, 8], are a class of Deep Neural Networks (DNNs). Such as any DNN, autoencoders should be properly trained with a dataset representative of legal communications among IoT devices, in order to find the optimal values for the weights of the autoencoder's nodes (neurons) that minimize the Means Square Error (MSE) of the performed predictions [6].

In order to guarantee the required level of availability and security of IoT based critical applications, the latency of the autoencoders in detecting malicious attacks should be as low as possible. Therefore, autoencoders' algorithms are typically executed by HW accelerators implemented by nanotechnology, to guarantee high performance [9, 10]. To implement the autoencoder, the weights of its nodes, generated during the training phase, are stored in the memory of the HW accelerator [9, 10]. However, HW accelerators implemented by nanotechnology are becoming increasingly susceptible to soft errors (SEs) [9, 10, 11, 12]. Such a vulnerability may be exacerbated by process parameter variations occurring during manufacturing and by degradation phenomena, such as Bias-Temperature-Instability (BTI) [13, 14]. Consequently, as highlighted in [9], standard ECCs may no longer able to protect the memory of HW accelerators implemented in nanotechnology.

Based on these considerations, in this paper we analyze the impact of single SEs on the effectiveness of autoencoders in detecting anomalous messages generated by cyberattacks. As an example, for our analyses we trained a typical autoencoder with two representative datasets containing credit card legal/illegal transactions. Similar results have been obtained also for other datasets. We realistically emulated SEs in memory elements of HW accelerators as bit flips in the weight of the trained autoencoders. We introduce two metrics to evaluate the impact of such SEs on system's availability and security. We show that SEs do not decrease system's security, but may reduce system's availability of the 70%. Our analyses have been performed considering single SEs. However similar problems are expected also for the case of multiple SEs that, as known [9], will become increasingly likely with technology scaling. Therefore, our analysis highlights the need for solutions enabling to avoid the detrimental impact on system's availability caused by SEs affecting the weights of autoencoders' algorithms executed using HW accelerators implemented by nanotechnologies.

The rest of this paper is organized as follows. In Section II, we present some preliminary concepts on autoencoders. In Section III, we describe the implementation and training of a typical autoencoder for two case study datasets, consisting of European credit card legal/illegal transactions. In Section IV, we report the results of our analyses on the impact of soft errors on the operation of autoencoders used to detect fraudulent credit card transactions. Finally, in Section V, we present some conclusive remarks.

II. Preliminary Concepts on Autoencoders

An autoencoder is a special type of neural network, that is trained to learn to reproduce its input vectors upon its output. Therefore, an autoencoder can efficiently learn properties of unlabeled input information [1, 5, 2, 6, 7, 8]. A schematic representation of an autoencoder is illustrated in Fig. 1. As known [6], it consists of an input and an output layer with the same number of nodes (n in Fig. 1), and k hidden layers with a smaller number of nodes than the input/output layers. A neuron in a hidden layer calculates the weighted sum of the inputs plus a bias. The result is then given to an activation function (*e.g.,* ELU, sigmoid, etc. [15]) that generates the output of the node, that is successively given as input to the nodes in the next layer.

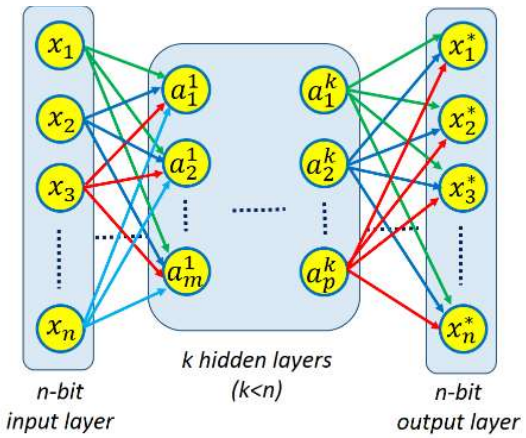

Fig. 1. Schematic representation of an autoencoder with k hidden layers.

In particular, the output a_i^l produced by a node i on a hidden layer l is given by:

$$a_i^l = f\left(\sum_j W_{i,j}^l a_j^{l-1} + b_{i,j}^l\right) \quad (1)$$

where $W_{i,j}^l$ and $b_{i,j}^l$ denote the weight and the bias parameters, respectively, associated to the *j-th* input of node i in the *l-th* layer; a_j^{l-1} is the value produced by a node j in the previous layer *l-1*; f is the activation function [6].

During the training phase, the weights and bias of each node (Eq. (1)) are properly adjusted in order to minimize the difference (error) between the inputs and the corresponding outputs of the autoencoder. For a given input $X = (x_1, x_2, \ldots, x_n)$ and the corresponding output $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$, such a difference is usually expressed in terms of the Mean Squared Error (MSE) as follows:

$$MSE = \frac{1}{n}\left(\sum_{k=1}^{n}(x_k - x_k^*)^2\right) \quad (2)$$

As shown in [1, 5, 2, 6, 7, 8], since autoencoders learn properties underlying the complex systems for which they have been trained, they can be efficiently used to detect anomalous inputs of such systems during their operation in the field. Thus, an autoencoder trained with information regarding legal transactions of credit cards will be able to reconstruct correctly (*i.e.,* with low MSE) only inputs corresponding to legal transactions of credit cards. Instead, it will reconstruct poorly (*i.e.,* with high MSE) inputs corresponding to illegal transactions of credit cards, which do not satisfy the properties of the training dataset.

III. Autoencoder Implementation and Training

For our analysis, we have implemented and trained autoencoders considering their typical architecture in Fig. 1, using the Python TensorFlow framework and the Keras open source library [16]. We considered two credit card fraud detection datasets (hereinafter referred as *Dataset 1* and *Dataset 2*) containing European credit card transactions [17].

Dataset 1 contains a total of 284807 credit card transactions (each represented by a decimal number with 30 digits), out of which 190161 are legal transactions for training, 93662 are legal transactions for validation, and 984 (492 legal and 492 illegal) are transactions for testing.

Dataset 2 contains a total of 6362620 credit card transactions (each represented by a decimal number with 9 digits), out of which 4262955 are legal transactions for training, 2099665 are legal transactions for validation, and 16426 (8213 legal and 8213 illegal) are transactions for testing.

We implemented two different autoencoders of the kind in Fig. 1, hereinafter referred to as *AEnc1* and *AEnc2*, to be trained with *Dataset 1* and *Dataset 2*, respectively. *AEnc1* has a number of inputs and outputs equal to n=30 (to match the number of inputs and outputs of *Dataset 1*), while *AEnc2* has a number of inputs and outputs equal to n=9 (to match the number of inputs and outputs of *Dataset 2*).

Moreover, for both autoencoders, we considered k=5 hidden layers, such that:

1) the first layer has 20 nodes in *AEnc1*, and 8 nodes in *AEnc2*;

2) the second layer has 10 nodes in *AEnc1*, and 4 nodes in *AEnc2*;

3) the third layer has 4 nodes in *AEnc1*, and 2 nodes in *AEnc2*;

4) the fourth layer has 10 nodes in *AEnc1*, and 4 nodes in *AEnc2*;

5) the fifth layer has 20 nodes in *AEnc1*, and 8 nodes in *AEnc2*;

6) the sixth layer has 30 nodes in *AEnc1*, and 9 nodes in *AEnc2*.

The considered number of nodes for the autoencoders' hidden layers has been chosen to enable a good trade-off between accuracy and training/fault simulation times. However, results similar to those reported in this paper are expected also for autoencoders with a different number of nodes in the hidden layers.

Finally, for both autoencoders we have considered two frequently adopted activation functions (f in Eq. (1)): the ELU and the sigmoid ones [15].

We trained the two autoencoders for up to 200 epochs, using the training and validation part of the considered datasets. *AEnc1* presents a total of 1680 weights, while *AEnc2* presents a total of 224 weights.

As a result of the training phase, we obtained that, in the fault-free case, *AEnc1* and *AEnc2* feature an accuracy of 90.1% and 99%, respectively.

As an example, Fig. 2 reports the distribution of the MSE for the autoencoder *AEnc1* using the ELU function.

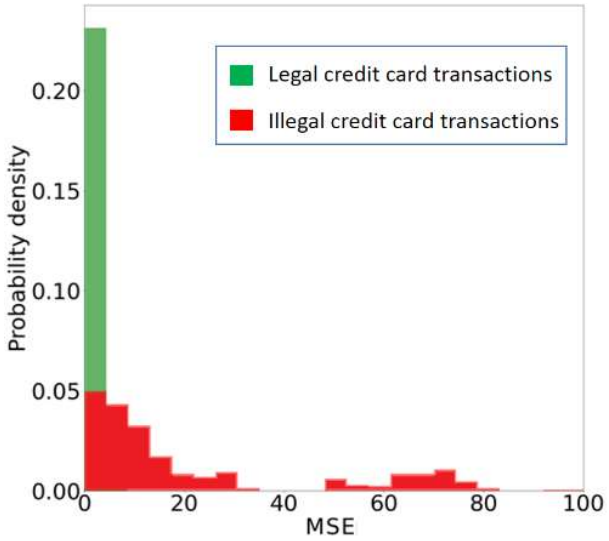

Fig. 2. Distribution of the Mean Square Error (MSE) of the autoencoder *AEnc1* using the ELU function in detecting fraudulent (anomalous) credit card transactions.

As can be seen, the distribution of the MSE for the case of legal transactions is reported in green. We can see that this distribution is completely concentrated around the 0, showing the good ability of the autoencoder in recognizing legal transactions. On the other hand, the red distribution reports the MSE for the case of illegal transactions. As can be seen, the MSE is significantly larger than 0, showing the poor ability of the autoencoder in reconstructing illegal transactions, that is its good ability to detect them.

IV. Experimental Setup and Results

In this Section, first we describe the methodology that we have used to emulate the occurrence of SEs in the weights of the autoencoders described in Section III. Then, we present some of the results obtained by fault simulation.

A. Experimental Setup

In order to analyze the impact of SEs on the effectiveness of autoencoders in detecting anomalous messages generated by cyberattacks, we emulated the occurrence of single SEs on the weights of the autoencoders trained with the two considered datasets containing credit card legal/illegal transactions described in Section III. We realistically assumed that the weights of the autoencoders are stored in the HW accelerator memory as a 32-bit word, following the IEEE 754 single-precision floating point standard representation of decimal numbers [18].

In particular, we emulated the occurrence of single SEs as bit-flips on the MSB of the exponent part of the floating point representation of each weight of the autoencoders, since this represents the worst case scenario. In fact, SEs affecting different exponent bits will result in a lower MSE, thus in a lower impact on the effectiveness of autoencoders in detecting anomalous messages generated by cyberattacks.

We have performed fault simulation considering both the ELU and the Sigmoid activation functions. Our analyses have been carried out by injecting:

i) 3360 single SEs in the weights of the autoencoder trained with *Dataset 1* (*AEnc1*), for both the ELU (1680 single SEs) and the Sigmoid (1680 single SEs) activation functions;

ii) 448 single SEs in the weights of the autoencoder trained with *Dataset 2* (*AEnc2*), for both the ELU (224 single SEs) and the Sigmoid (224 single SEs) activation functions.

For each injected SE, we simulated the trained autoencoders (as described in Section III) by applying to their inputs the testing part of the considered datasets. They consist of 492 legal and 492 illegal credit card transactions (984 in total) for *Dataset 1*, and 8213 legal and 8213 illegal credit card transactions (16426 in total) for *Dataset 2*.

For each injected SE, one of the following conditions could occur: a) a legal transaction is correctly classified as legal ($C_{L\text{-}L}$); b) an illegal transaction is correctly classified as illegal ($C_{I\text{-}I}$); c) an illegal transaction is incorrectly classified as legal ($C_{I\text{-}L}$); d) a legal transaction is incorrectly classified as illegal ($C_{L\text{-}I}$).

In particular, SEs resulting in legal transactions incorrectly classified as illegal ($C_{L\text{-}I}$) will impact system's availability, since such transactions will not be processed by the system. In order to evaluate such an impact of SEs on system's availability, we define the following availability (AV) metric:

$$AV = 1 - \frac{C_{I-I} + C_{L-I}}{Tolal\ transactions} \tag{3}$$

Instead, SEs resulting in illegal transactions incorrectly recognized as legal ($C_{I\text{-}L}$) will impact system's security, since such illegal transactions will be processed as if they were legal. In order to evaluate such an impact of SEs on system's security, we define the following security risk (SR) metric:

$$SR = \frac{C_{I-L}}{Illegal\ transactions} \tag{4}$$

B. Results for the Autoencoder Trained with Dataset 1

The summary of the simulation results obtained for the autoencoder trained with *Dataset 1* (*AEnc1*), with the ELU and Sigmoid activation functions, are shown in Fig. 3 and Fig. 4, respectively. The figures report the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ obtained for the fault free case and for the case of SEs affecting the weights of the autoencoder. For the case of SEs, the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ have been averaged over the total number of SEs injected in the autoencoder (3360 single SEs).

As can be seen from Fig. 3, for the autoencoder *AEnc1* using the ELU function, SEs result in a reduction of approx. 60% in the number of legal transactions recognized as legal ($C_{L\text{-}L}$), and in a consequent significant increase in the number of legal transactions recognized as illegal ($C_{L\text{-}I}$), thus significantly reducing system's availability. This because SEs increase the difference between the input and output of the autoencoder (i.e., the MSE), thus increasing the number of legal transactions that are misclassified as illegal.

Moreover, we observe that SEs result in a limited increase (of approx. 13%) in the number of illegal transactions correctly classified as illegal ($C_{I\text{-}I}$), and in a consequent reduction in the number of illegal transactions erroneously recognized as legal ($C_{I\text{-}L}$), thus resulting in a reduction of system's security risk. This because some illegal transactions that are similar to legal transactions, and may be not detected by a fault-free autoencoder due to the small MSE, may become detectable if the autoencoder is affected by a SE, due to the increased MSE.

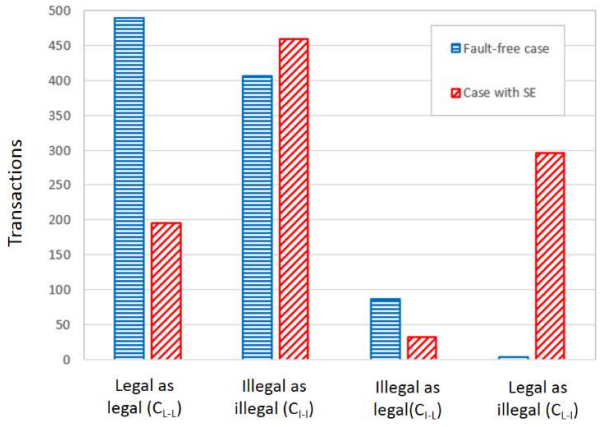

Fig. 3. Simulation results for *AEnc1* using the ELU activation function, showing the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ obtained for the fault free case and for the case of SEs in the weights of the autoencoder. For the case of SEs, the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ have been averaged over the total number of SEs.

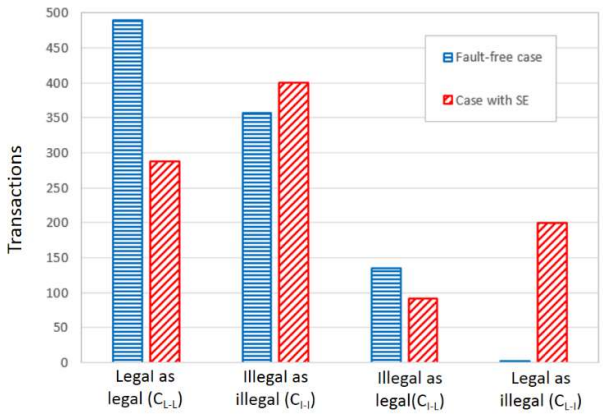

Fig. 4. Simulation results for *AEnc1* using the Sigmoid activation function, showing the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ obtained for the fault free case and for the case of SEs in the weights of the autoencoder. For the case of SEs, the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ have been averaged over the total number of SEs.

Analogous results are reported in Fig. 4 for the case of the autoencoder *AEnc1* using the Sigmoid function. In particular, SEs result in a significant reduction (of approx. 41%) in the number of legal transactions recognized as legal ($C_{L\text{-}L}$), and in a consequent significant increase in the number of legal transactions recognized as illegal ($C_{L\text{-}I}$), thus significantly diminishing system's availability.

Moreover, we also observed that SEs result in a limited increase (of approx. 12%) in the number of illegal transactions correctly classified as illegal ($C_{I\text{-}I}$), and in a consequent reduction in the number of illegal transactions recognized as legal ($C_{I\text{-}L}$), thus resulting in a reduction of system's security risk.

As for the case of the ELU function, the increase in the number of both legal transactions recognized as illegal ($C_{L\text{-}I}$) and illegal transactions correctly classified as illegal ($C_{I\text{-}I}$) can be explained by the fact that SEs increase the MSE.

C. *Results for the Autoencoder Trained with Dataset 2*

The summary of the simulation results obtained for the autoencoder trained with *Dataset 2* (*AEnc2*), with the ELU and Sigmoid activation functions, are shown in Fig. 5 and Fig. 6, respectively. As in the previous subsection, the figures report the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ obtained for the fault free case and for the case of SEs affecting the weights of the autoencoder. For the case of SEs, the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ have been averaged over the total number of SEs injected in the autoencoder (448 single SEs).

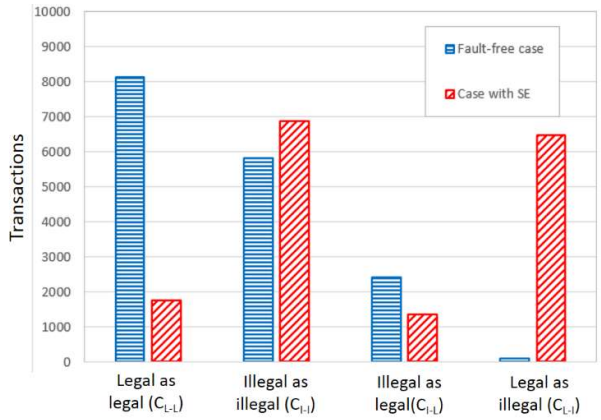

Fig. 5. Simulation results for *AEnc2* using the ELU activation function, showing the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ obtained for the fault free case and for the case of SEs in the weights of the autoencoder. For the case of SEs, the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ have been averaged over the total number of SEs.

As can be seen from Fig. 5, for the autoencoder *AEnc2* using the ELU activation function, SEs result in a reduction of approx. 79% in the number of legal transactions recognized as legal ($C_{L\text{-}L}$), and in a consequent significant increase in the number of legal transactions recognized as illegal ($C_{L\text{-}I}$), thus significantly reducing system's availability.

Moreover, we observe that SEs result in an increase of approx. 18% in the number of illegal transactions correctly classified as illegal ($C_{I\text{-}I}$), and in a consequent reduction in the

number of illegal transactions erroneously recognized as legal ($C_{I\text{-}L}$), thus increasing system's security.

The increase in the number of both legal transactions recognized as illegal ($C_{L\text{-}I}$) and illegal transactions correctly classified as illegal ($C_{I\text{-}I}$) can be explained by the fact that SEs increase the MSE.

Finally, Fig. 4 reports the results obtained for *AEnc2* using the Sigmoid activation function. We can see that SEs result in a significant reduction (of approx. 68%) in the number of legal transactions recognized as legal ($C_{L\text{-}L}$), and in a consequent significant increase in the number of legal transactions recognized as illegal ($C_{L\text{-}I}$), thus significantly diminishing system's availability.

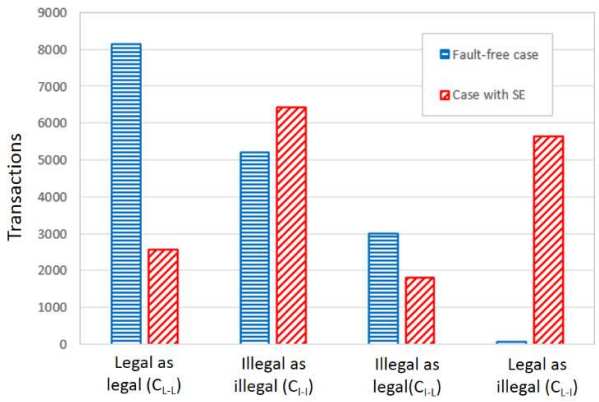

Fig. 6. Simulation results for *AEnc2* using the Sigmoid activation function, showing the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ obtained for the fault free case and for the case of SEs in the weights of the autoencoder. For the case of SEs, the values of $C_{L\text{-}L}$, $C_{I\text{-}I}$, $C_{I\text{-}L}$, and $C_{L\text{-}I}$ have been averaged over the total number of SEs.

Moreover, we can also observe that SEs result in an increase of approx. 23% in the number of illegal transactions correctly classified as illegal ($C_{I\text{-}I}$), and in a consequent reduction in the number of illegal transactions recognized as legal ($C_{I\text{-}L}$), thus increasing system's security.

Similarly to the case of the ELU function, the increase in the number of both legal transactions recognized as illegal ($C_{L\text{-}I}$) and illegal transactions correctly classified as illegal ($C_{I\text{-}I}$) can be explained by the fact that SEs increase the MSE.

D. Evaluation of System's Availability and Security

In this subsection, for both *AEnc1* and *AEnc2*, and considering both the fault free case and the case of SEs, we evaluate the system's availability metric (AV in (3)) and security metric (SR in (4)), as well as their relative variation (ΔAV and ΔSR) between the fault free case and the case of SEs. In particular, such variations have been derived as follows:

$$\Delta AV(\%) = 100\frac{AV_{SE} - AV_{Faul_{free}}}{AV_{Faul_{free}}};$$

$$\Delta SR(\%) = 100\frac{SR_{SE} - SR_{Faul\_free}}{SR_{Faul\_free}}$$

Tab. 1 reports the obtained values.

As we can see from Table 1, for both autoencoders, and considered activation functions, system's availability is significantly impacted by SEs affecting the weights of the autoencoders.

In fact, for *AEnc1*, we can observe an AV reduction of approx. 60% and 26% using the ELU and Sigmoid functions, respectively. Analogously, for *AEnc2*, we can observe an AV reduction of approx. 69% and 60% using the ELU and Sigmoid functions, respectively. Therefore, our analysis highlights the need for solutions enabling to avoid the detrimental impact on system availability caused by SEs affecting the weights of autoencoders' algorithms executed using HW accelerators implemented by nanotechnologies.

Table 1.
Values of system's availability and security metrics (AV and SR), for the fault free case and the case of SEs, as well as variations (ΔAV and ΔSR) of the AV and SR values obtained for the fault free case and the case of SEs.

	AEnc1		*AEnc2*	
	ELU	Sigmoid	ELU	Sigmoid
AV Fault free case	0.584	0.635	0.614	0.678
AV with SE	0.23	0.47	0.19	0.27
ΔAV (%)	**-60.6%**	**-25.9%**	**-69.1%**	**-60.2%**
SR Fault free case	0.17	0.27	0.29	0.37
SR with SE	0.06	0.19	0.16	0.22
ΔSR (%)	**-64.7%**	**-29.6%**	**-44.8%**	**-40.5%**

From Table 1 we can also see that, for both autoencoders and both activation functions, the security risk metric SR diminishes in case of SEs on the weights of the autoencoder. For *AEnc1*, the reduction in SR is of approx. 65% and 30% for the ELU and Sigmoid functions, respectively. For *AEnc2*, the reduction in SR is of approx. 45% and 40% using the ELU and Sigmoid functions, respectively. As clarified before, the reduction in SR is associated to the fact that SEs increase the difference between legal and illegal transactions, making it easier to detect illegal transactions in case of SEs.

Results similar to those shown in Figs. 3, 4, 5, 6 and Table 1 have been obtained also for other datasets in [17], thus highlighting the need for solutions to avoid the impact on system's availability caused by SEs affecting the weights of autoencoders implemented by nanotechnology HW accelerators.

Finally, we can reasonably expect that, with nanotechnology scaling, and the consequent increased likelihood of multiple SEs, such an impact on system's availability will become even worse than that found here for the case of single SEs.

V. CONCLUSIONS

Autoencoder-based anomaly detection has recently been adopted to detect the presence of messages altered by malicious attacks in IoT based critical applications (such as finance, healthcare, manufacturing, etc.), in order to guarantee high levels of security and availability. To achieve this goal, the detection latency of autoencoders should be as low as possible. Therefore, autoencoders' algorithms are typically executed using HW accelerators, implemented by nanotechnology. However, such HW accelerators are susceptible to soft errors, which may significantly impact the autoencoder's correct operation. We have analyzed the impact of SEs on the effectiveness of autoencoders in detecting anomalous messages generated by cyberattacks. As an example, we considered autoencoders trained to discriminate credit card legal and illegal transactions. We have introduced two metrics to evaluate the impact of SEs affecting the autoencoders on system's availability and security. We have shown that such SEs may reduce system's availability of the 70%, while they do not impact system's security. Therefore, our analysis highlights the need for solutions enabling to avoid the detrimental impact on system's availability of SEs affecting the weights of autoencoders' algorithms executed by HW accelerators implemented by nanotechnologies.

REFERENCES

[1] Z. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, "Autoencoder-based network anomaly detection", in Proc. of *2018 IEEE Wireless Telecommunications Symp. (WTS)*, 2018, pp. 1-5.

[2] T. -H. Lin, J. -R. Jiang, "Anomaly Detection with Autoencoder and Random Forest", in Proc. of *Int. Computer Symp.*, 2020, pp. 96-99.

[3] E. Zarepour, M. Hassan, C. T. Chou, A. Adesina, M. E. Warkiani, "Reliability analysis of time-varying wireless nanoscale sensor networks", in Proc. of *IEEE 15th Int.l Conf. on Nanotechnology (IEEE-NANO)*, 2015, pp. 63-68.

[4] D. Rossi, M. Omaña, D. Giaffreda, C. Metra, "Secure communication protocol for wireless sensor networks", in Proc. of *IEEE East-West Design and Test Symp., EWDTS'10*, 2010, pp. 17–20.

[5] X. Yang, L. Zhou, Y. Liu, Z. Tan, J. Gao, "An anomaly detection method for daily line loss rate based on deep autoencoder", in Proc. of *IEEE Int. Conf. on Power Electronics, Computer Applications (ICPECA)*, 2021, pp. 152-156.

[6] M. Sakurada, T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction", in Proc. of the *2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA), ACM*, December 2014.

[7] P. Park, P. Di Marco, H. Shin, J. Bang, "Fault Detection and Diagnosis Using Combined Autoencoder and Long-Short-Term Memory Network", *Sensors, MDPI*, 2019, 19, 4612, doi:10.3390/s19214612

[8] C. Zhou, R. Paffenroth, (2017, August). "Anomaly detection with robust deep autoencoders", in Proc. of *23rd Int. Conf. on Knowledge Discovery and Data Mining,* ACM, August 2017, pp. 665-674.

[9] A. Azizimazreah, Y. Gu, X. Gu, L. Chen, "Tolerating Soft Errors in Deep Learning Accelerators with Reliable On-Chip Memory Designs," in Proc. of *2018 IEEE Int. Conf. on Networking, Architecture and Storage (NAS)*, 2018, pp. 1-10.

[10] G. Li, S. K. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer, S. W. Keckler, "Understanding error propagation in deep learning neural network (DNN) accelerators and applications", in Proc. of the *Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC '17), ACM*, Article 8, pp. 1–12.

[11] C. Metra, D. Rossi, M. Omaña, M., Jas, A., Galivanche, R. "Function-inherent code checking: A new low cost on-line testing approach for high performance microprocessor control logic", in Proc. of *13th IEEE European Test Symp., ETS 2008*, 2008, pp. 171–176.

[12] D. Rossi, M. Omaña, F. Toma, C. Metra, "Multiple Transient Faults in Logic: an Issue for Next Generation ICs ?", in *IEEE Proceedings of The Int. Symp. on Defect and Fault Tolerance in VLSI Systems,* October 3-5, 2005, pp. 352—360.

[13] M. Omaña, T. Edara, C. Metra, "Low-Cost Strategy to Mitigate the Impact of Aging on Latches' Robustness", *IEEE Transactions on Emerging Topics in Computing*, 2018, 6(4), pp. 488–497.

[14] M. Omaña, D. Rossi, T. Edara, C. Metra, "Impact of Aging Phenomena on Latches' Robustness", *IEEE Transactions on Nanotechnology*, 2016, 15(2), pp. 129–136.

[15] K. Alrawashdeh, C. Purdy, "Fast Activation Function Approach for Deep Learning Based Online Anomaly Intrusion Detection", in Proc. of *IEEE 4th Int. Conf. on Big Data Security on Cloud,* 2018, pp. 5-13

[16] https://keras.io/

[17] https://www.kaggle.com/datasets/

[18] *IEEE Standard for Floating-Point Arithmetic. IEEE STD 754-2019, IEEE, pp. 1–84.*