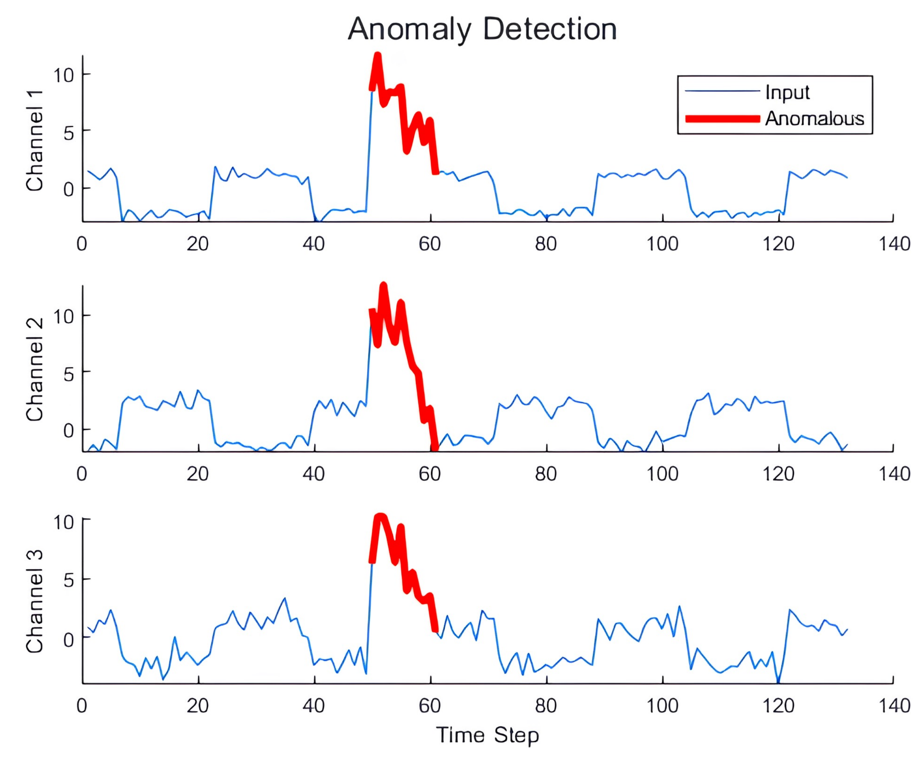
SPIKING AUTOENCODER ARCHITECTURE FOR ANOMALY DETECTION ON SATELLITE DATA WITH FPGAs

Paolo Ritirato, Giuseppe Sorrentino, Davide Conficconi
paolo.ritirato@mail.polimi.it

1. Motivation

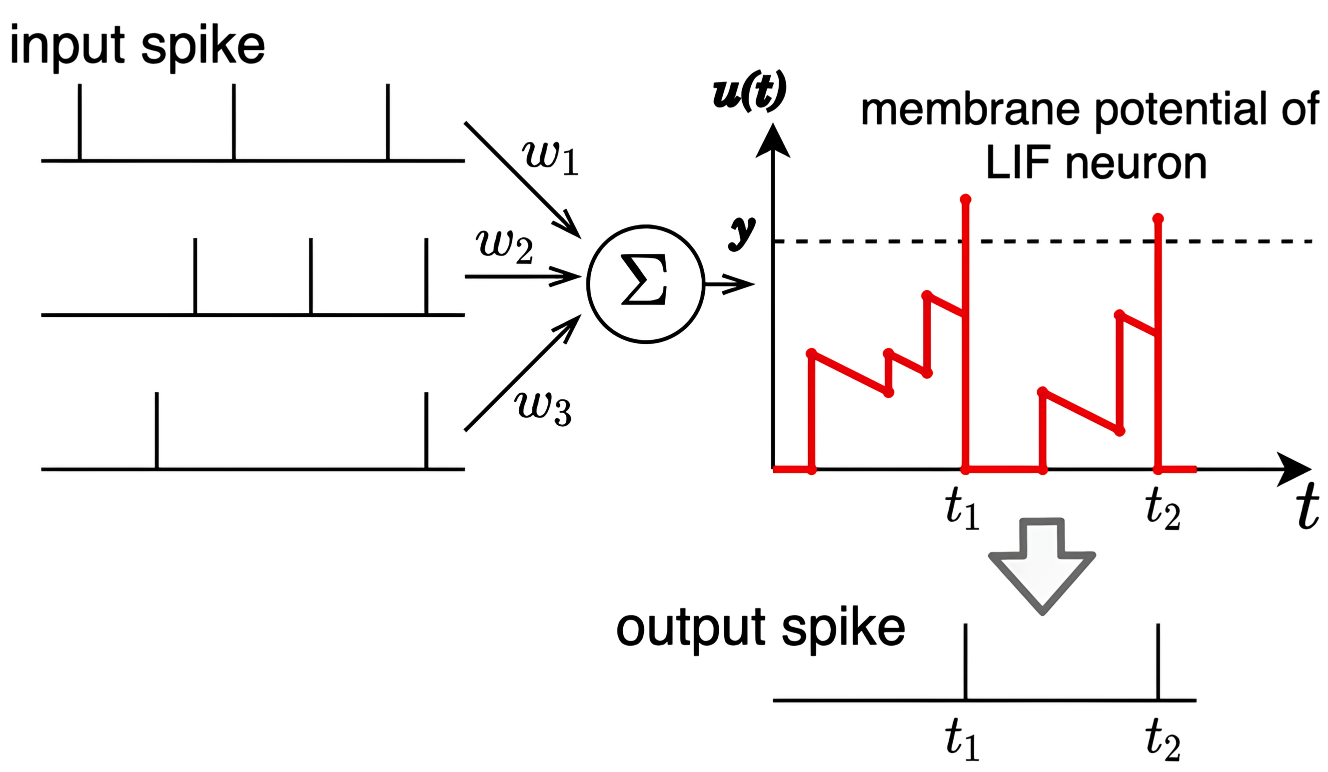
State of the Art employs different techniques for **Time Series Anomaly Detection**. **ML** based methods which require **hardware** and **power resources** not suitable for **onboard application**[1]. A specific use case is provided by **ESA-ADB Research Article**[2].



[5]

2. Spiking Neural Network

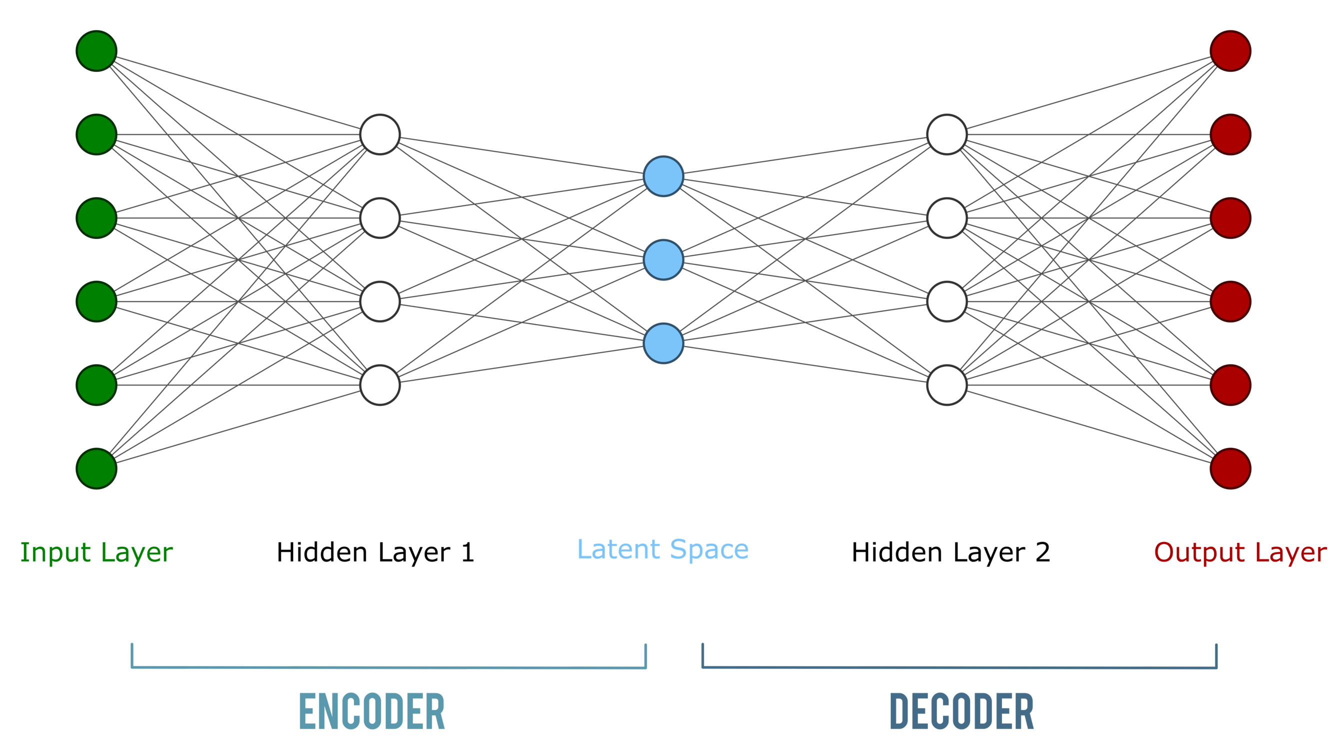
Spiking Neural Network aims to replicate brain structure and behaviour. SNNs are implemented using **spiking neurons**, which communicate through simulated «spikes» of electrical activity, showing **enhanced efficiency**, **faster processing**, **adaptive intelligence**.



[6]

3. Spiking Autoencoder

The proposed architecture combines **Autoencoder** structure with **SNNs**.
The idea is to exploit Autoencoder ability to **compress data width** and SNNs' intrinsic capability to **handle temporal dynamics**.

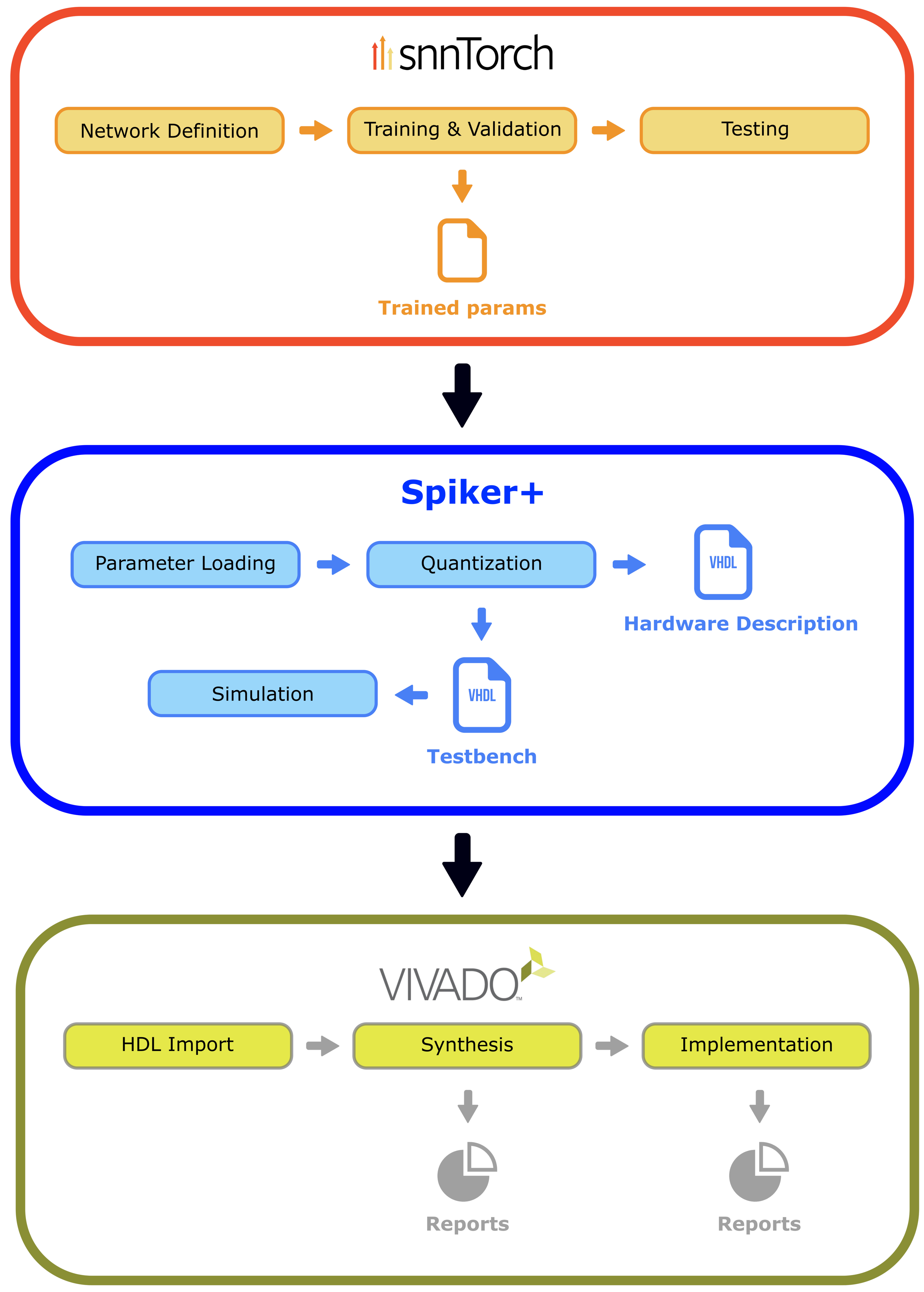


References

[1] Paul Boniol et al. "Dive into time-series anomaly detection: A decade review". In: arXiv preprint arXiv:2412.20512 (2024).
[2] Krzysztof Kotowskietal "European space agency benchmark for anomaly detection in satellite telemetry". In: arXiv preprint arXiv:2406.17826 (2024).
[3] Jason K Eshraghian et al. "Training spiking neural networks using lessons from deep learning". In: Proceedings of the IEEE 111.9 (2023), pp. 1016–1054.
[4] Alessio Carpegna, Alessandro Savino, and Stefano Di Carlo. "Spiker+: a framework for the generation of efficient Spiking Neural Networks FPGA accelerators for inference at the edge". In: arXiv preprint arXiv:2401.01141 (2024).
[5] Time Series Anomaly Detection Using Deep Learning. MathWorks.
[6] Hiromichi Kamata, Yusuke Mukuta, and Tatsuya Harada. Fully Spiking Variational Autoencoder. Sept. 2021. doi: 10.48550/arXiv.2110.00375.

4. Methodology

1. **snnTorch**[3] Python library is used to design, train and test the proposed SNN architecture.
2. **Spiker+** [4] generates an HDL description of the network and simulates its behaviour.
3. **Vivado IDE** runs synthesis and implementation on the target board.

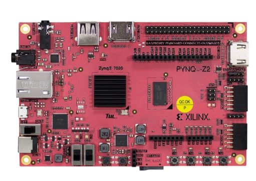


5. Results

Software detection performances are comparable with ESA proposed approaches[1] using **1/28 of training data**;
Hardware FPGA validation shows timing performances compatible with **sampling frequency** of the dataset using **1%** of total available FF, LUT and BRAM and **120 mW** of power requirement.

	F0.5				
Phase	1	2	3	4	5
PCC/HBOS/iForest/Window iForest/KNN			<0.001		
Global STD3	<0.001				**0.001**
Global STD5	0.041	0.037	0.104	0.217	**0.253**
DC-VAE-ESA STD3	<0.001		0.007	**0.009**	0.003
DC-VAE-ESA STD5	0.007	0.0012	**0.085**	0.030	0.075
Teleman-ESA	0.059	0.058	0.122	**0.309**	0.178
Teleman-ESA-Pruned	0.227	0.311	0.776	0.776	**0.786**
ADStrobot	**0.924**	**0.924**	**0.924**	**0.924**	**0.924**

670 ns	7 ns	1%	120 mW
Simulation time	Clock period	Area utilization	Power consumption



Target board: PYNQ Z2 - XC7Z020

Acknowledgement

The authors want to thank ESA Academy and AMD University Program.
We also thank Alessio Carpegna for his support as Spiker+ developer and main author.