Title: Machine Learning-Based Error Detection and Function Approximation for Reliable Computing

Authors: Jelena Nedeljković, Goran Nikolić, Tatjana Nikolić, Marko Anđelković and Miloš Krstić

Published in: *Proceedings of the IEEE International Scientific Conference on Information, Communication and Energy Systems and Technologies* (*ICEST* 2025), 2025

DOI: 10.1109/ICEST66328.2025.11098454

Please cite the final published version:
Jelena Nedeljković, Goran Nikolić, Tatjana Nikolić, Marko Anđelković, Miloš Krstić, "Machine Learning-Based Error Detection and Function Approximation for Reliable Computing", 60th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST 2025), Ohrid, North Macedonia, June 26–28, 2025, DOI: 10.1109/ICEST66328.2025.11098454

This work has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101160293 (AIDA4Edge).

Funded by the European Union

AIDA4Edge

Machine Learning-Based Error Detection and Function Approximation for Reliable Computing

Jelena Nedeljković[1], Goran Nikolić[1], Tatjana Nikolić[1], Marko Anđelković[2] and Miloš Krstić[2]

Abstract – **Ensuring reliability in modern computing systems requires efficient and accurate error detection mechanisms. This paper explores a machine learning based approach to error detection through accurate approximation of commonly used exponential and logarithmic functions. A hybrid error detection architecture is proposed, which combines a traditional computation module and a ML-based approximator running in parallel. Several regression models are evaluated based on their approximation accuracy and complexity. Experimental results show that polynomial regression offers the best compromise between precision and complexity, while MLP and SVR provide high accuracy with increased resource requirements.**

Keywords – **Machine learning, Regression models, Error detection, System reliability.**

I. Introduction

As modern computing systems become more complex, ensuring their reliability, especially in critical fields like autonomous driving, medical devices, and industrial automation, is essential. Errors in these systems can have serious consequences, making efficient real-time error detection systems crucial. Traditional error detection methods often fall short in complex environments, particularly where system behavior is nonlinear, dynamic, or affected by multiple variables. In such cases, machine learning (ML) techniques offer a data-driven approach that can learn intricate patterns from historical data and identify with higher accuracy the anomalies or faults. In addition to fault detection, ML has shown great promise in approximating complex mathematical functions which are commonly used in embedded and signal processing applications. Accurately modeling these functions is challenging due to their nonlinear nature and the limitations of traditional methods like Look-Up Tables (LUTs), which require large memory footprints and high computational resources. In contrast, ML models can learn from data samples to approximate function outputs with high precision, while significantly reducing memory and processing requirements.

This paper investigates the application of ML models for accurately approximating exponential and logarithmic functions, with a particular focus on their integration into error detection mechanisms. The research focuses on using regression techniques to address the challenge of modeling complex, non-linear mathematical functions efficiently, balancing precision with computational and memory constraints. By comparing traditional function computational methods with ML-based approximators, the proposed approach introduces a hybrid architecture that enhances system reliability. In particular, it uses ML to detect anomalies in functional outputs through parallel execution and result comparison. The objective is to develop a methodology that ensures high accuracy and fault tolerance in mathematical computations, especially within resource-constrained environments where reliability is critical.

The paper is structured as follows: Section I provides an introduction, outlining the research context and objectives. Section II reviews related works, discussing existing approaches to error detection. Section III describes the proposed error detection system. Section IV focuses on the selection of ML models. Section V presents the experimental results and their analysis. Finally, Section VI summarizes the findings and future research directions.

II. Related Works

Machine learning is increasingly used for fault detection in various domains, improving system reliability. For instance, [1] achieves near-perfect accuracy in diagnosing induction motor faults using stator currents and vibration signals processed via matching pursuit and wavelet transform. Review [2] outlines advantages and challenges in data-driven fault detection, such as handling noisy data and selecting relevant features. In specialized systems like lighthouses [3] and building diagnostics [4], ML is shown to enhance adaptability and detect long-term faults even with incomplete sensor data.

In cloud infrastructure, ML models improve performance prediction and early fault detection [5], while [6] proposes fault-tolerant systems using deep and reinforcement learning for critical applications. Predictive models like Random Forests and Gradient Boosting are used in distributed systems to reduce downtime [7].

For embedded systems, [8] and [9] explore efficient deployment of DNNs and microcontroller-optimized workflows. [10] further improves performance of ML on constrained platforms. A notable reliability challenge - soft errors from deep submicron scaling - is addressed by the ML-based checker (MLC), which uses XGBoost for real-time detection of radiation-induced faults in processors like SAYAC [11]. The importance of decentralizing intelligence to edge devices is emphasized in [12] to reduce latency and enhance security.

[1]Jelena Nedeljković, Goran Nikolić and Tatjana Nikolić are with University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: {jelena.nedeljkovic; goran.nikolic; tatjana.nikolic}@elfak.ni.ac.rs

[2]Marko Anđelković and Miloš Krstić are with IHP - Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany, E-mail: {andjelkovic; krstic}@ihp-microelectronics.com. Miloš Krstić is also with University of Potsdam, Germany

While these works highlight the potential of ML in fault detection and reliability, many focus on high-resource platforms or separate fault detection from function approximation. Our approach combines ML models for both tasks, with the aim of optimizing for embedded environments. It prioritizes low complexity and high performance, aiming to find the optimal solution. Based on the principles presented in [13], this paper proposes a hybrid architecture that enables concurrent, ML-assisted fault detection with minimal hardware overhead. This direction is not sufficiently explored in the current state of the art.

III. Error Detection System

An error detection system plays a crucial role in ensuring the reliability of modern computing systems, especially in critical applications. It enables the identification and isolation of potential faults in hardware and software, preventing serious issues that could lead to a complete system failure. ML models significantly improve these systems by enabling precise modelling of the relationships between inputs and outputs, making it easier to detect deviations and anomalies that indicate faults. These models are efficient because they have a compact representation of parameters and can quickly analyse large amounts of data, which is crucial for error detection in complex systems. By integrating ML into traditional error detection systems, greater reliability is achieved. ML-based function approximation is particularly attractive for deployment in resource-constrained environments, such as embedded systems, where optimizing the trade-off between accuracy and complexity is essential.

This paper presents a ML-based error detection system for reliable computing the exponential or logarithmic function, which uses a concrete error detection approach.

The general block diagram of the proposed error detection system is shown in the Figure 1. The functional module represents part of an embedded system or even the entire system implemented as a System-on-Chip (SoC) or on a FPGA platform. This system module executes the function $Y=e^{X}$ or $Y=\log(X)$.

The proposed system consists of two independent hardware modules: a traditional computation module, which calculates the target function using a predefined mathematical algorithm, and a ML-based module, which uses a pre-trained model to approximate the same function. During runtime, both modules operate in parallel, receiving identical input values and producing corresponding outputs. A dedicated comparator unit continuously monitors the outputs and identifies any disagreements between the two results.

The proposed method operates in two phases:

1. Offline Phase (Model Training and Validation): During the offline phase, the ML model is trained using known input-output data from the functional block. Since all input-output pairs are pre-known, they are used as the training dataset. To ensure that the ML model accurately mimics the functional block, a loss function is used during training. After training, the model undergoes validation using a separate dataset that was not used during the training phase. This evaluation serves to check the generalization of the model.

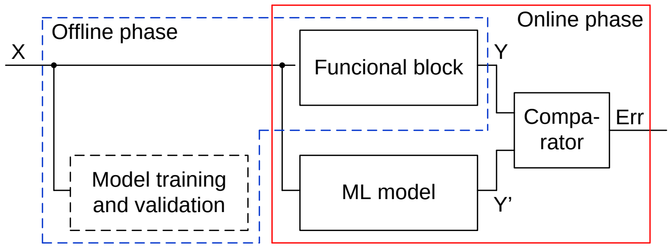

Fig. 1. Figure example

2. Online Phase (Real-Time System Operation): During real operation, the trained ML model works in parallel with the functional module. For each given input X, both modules (traditional and ML model) independently generate output values. The comparator then checks for discrepancies between the two results. If the results match, it is assumed that no error has occurred, and the system continues normal operation.

However, if any deviation is detected, an error handling routine is triggered. This routine initiates a recalculation cycle to confirm whether the error is transient or persistent. Furthermore, the system can involve additional redundancy checks or log the event for later analysis and increment an error counter. In the case of repeated or persistent mismatches, the system may automatically isolate the faulty module, reconfigure its processing route, or switch to a safe mode to prevent incorrect data propagation.

IV. Choice of Machine Learning Model

The goal of this research is to develop a ML model capable of accurately approximating exponential and logarithmic functions. These are well-known nonlinear mathematical functions frequently used in many fields, including mathematics, physics, and engineering. Given that this task involves learning a mapping from known inputs to known outputs, it is indicated as a supervised learning problem. In supervised learning, the model is trained on labelled data, where each input is associated with a corresponding output. This approach allows the model to learn patterns in the data and then use the learned information to predict outputs for new, unseen inputs.

Since the outputs to be predicted are continuous numerical values, the appropriate ML approach is regression. Regression models are specifically designed to estimate continuous outputs and can vary in complexity based on the functional relationship between input and output data.

To assess the suitability of different regression models for the task of function approximation, we examined several types of regression approaches. The model selection was made based on the characteristics of the functions being approximated and the need to balance accuracy with computational complexity. The models investigated include:

1. Linear regression: Serves as a baseline model but performs poorly with nonlinear functions.

2. Polynomial regression: Suitable for capturing nonlinear patterns, particularly effective in modeling exponential growth.

3. Multilayer perceptron (MLP) regression: A feedforward neural network capable of modeling complex nonlinear relationships.

4. Support vector regression (SVR): Employs kernel functions such as the radial basis function (RBF) to handle nonlinearities effectively.

Other regression variants such as ridge, lasso, elastic net, logarithmic, and Poisson regressions were also considered during the model selection process. However, based on empirical results and the nature of the functions studied, the focus was placed on the four core models listed above.

The training process employed batch learning, wherein the model is trained on the entire dataset at once. This approach is appropriate for scenarios involving static datasets and allows for more stable convergence of model parameters.

In addition to individual models, ensemble learning techniques were explored to further enhance predictive performance. These methods leverage the strengths of multiple models to improve robustness, reduce overfitting, and achieve higher accuracy.

V. EXPERIMENTAL RESULTS

To evaluate the suitability of different ML models for approximating mathematical functions within system blocks, a series of experiments are conducted. The focus is on two commonly encountered non-linear functions in engineering systems: the exponential and logarithmic functions. Since regression is the most appropriate approach for approximating continuous functions, several regression-based models were analyzed: linear regression, polynomial regression, multilayer perceptron (MLP) regression, and support vector regression (SVR). The experiments are executed in the Google Colab environment using Python and the scikit-learn library. All models were trained and evaluated on synthetically generated datasets representing the exponential and logarithmic functions over appropriate intervals. Each dataset consisted of input-output pairs of used functions.

To compare the performance of the models, three evaluation metrics were used:

- Mean Squared Error (MSE): Measures the average of the squares of the errors between the predicted and true values where lower values indicate better performance.
- R^2 score (coefficient of determination): Indicates how well the model explains the variance in the data. A score close to 1.0 indicates a high-quality fit.
- Complexity of model: Defined as the number of trainable parameters (or in the case of non-parametric models, an equivalent complexity measure such as support vectors), representing resource requirements and potential overfitting.

Linear regression was used as a baseline method. Due to the inherent non-linearity of exponential and logarithmic functions, this model exhibited poor performance. The results are slightly better for the logarithmic function, but still significantly lower than other models. Despite its very low complexity (2 parameters), linear regression is inadequate for accurately modeling non-linear behaviors. Note that the MSE values for the exponential function are significantly higher due to the rapid growth of the function itself.

Polynomial regression was tested with increasing degrees, corresponding to increasing model complexity. As expected, accuracy improved significantly with higher-degree polynomials. For the exponential function, the R^2 score reached 1.0 with an MSE of 315.10 and a complexity of 9 parameters. For the logarithmic function, similarly high accuracy was achieved (R^2 = 0.9975, MSE = 0.0030) with 16 parameters. This model demonstrated excellent approximation capabilities with relatively low complexity.

The MLP regressor is evaluated using three configurations with increasing hidden layer sizes, which directly affected the number of model parameters. With increased complexity, performance improved consistently. For the exponential function, the best configuration (261 parameters) yielded an MSE of 24606.26 and an R^2 score of 0.9985. For the logarithmic function, this configuration achieved an R^2 of 0.9722 and an MSE of 0.0335. However, even with comparable accuracy to polynomial regression, the MLP model requires significantly higher complexity.

SVR is evaluated with different kernel parameters. For the logarithmic function, it achieves exceptional results with the lowest observed MSE (0.0010) and highest R^2 score (0.9992), using 80 support vectors as a measure of complexity. For the exponential function, the R^2 score reached 1.0 with an MSE of 2.9017 and a complexity of 79 support vectors.

A comparison of the results is shown in Table 1. Among the evaluated models, polynomial regression consistently demonstrated high accuracy with low complexity, making it a practical choice for approximating both exponential and logarithmic functions. MLP regression provided competitive performance but at the cost of significantly increased complexity, which may not be ideal for real-time or resource-constrained applications. SVR achieved the best results for logarithmic approximation but requires careful tuning and may not generalize well to all function types. These findings highlight the importance of considering both accuracy and model complexity when selecting ML models for function approximation in embedded or adaptive control systems.

The trade-off between complexity and accuracy is significant in critical systems where reliability is key. In such systems, although execution speed and memory usage are important, accuracy must be at the highest possible level to avoid errors that can have serious consequences. In such cases, choosing a model that provides an extremely low MSE, even if it requires slightly more hardware overhead, makes perfect sense as it provides greater reliability in predictions.

In this context, hardware duplication, where one module uses the traditional approach and the other uses the ML approach, represents an optimal solution for error detection systems. It provides a balance between occupied resources and accuracy.

TABLE I
RESULTS OF MSE, R^2 SCORE AND COMPLEXITY

Model	Exponential			Logarithm		
	MSE	R^2	Complexity	MSE	R^2	Complexity
Linear regression	9047468.2615	0.4474	2	0.3384	0.7198	2
Polynomial regression	694150.6038	0.9576	5	0.0445	0.9632	6
	29291.6024	0.9982	7	0.0058	0.9952	11
	315.0951	1.0	9	0.0030	0.9975	16
MLP regression	505615.97877	0.96912	12	0.2058	0.8296	81
	107757.5216	0.9934	32	0.1069	0.9114	221
	24606.2648	0.9985	261	0.0335	0.9722	381
SV regression	5956.7997	0.9996	39	0.0241	0.9801	23
	5945.8868	0.9996	44	0.0028	0.9977	32
	2.9017	1.0	79	0.0010	0.9992	80

VI. CONCLUSION

This research demonstrates the potential of ML models to improve the reliability of mathematical calculations in embedded systems. By approximating exponential and logarithmic functions using different regression models and comparing their outputs with traditional computational methods, the proposed system introduces a hybrid architecture for real-time fault detection. Experimental evaluation showed that polynomial regression provides a highly accurate and computationally efficient solution, making it ideal for embedded applications. MLP regression and SVR also showed strong performance, especially for complex data patterns, but with significantly higher complexity. The obtained results indicate that the proposed ML-based methodology is suitable as a verification method, as it allows for independent verification of function outputs and detection of deviations caused by hardware or computational errors. The use of regression models for validation allows identification of faults, offering reliability at relatively low cost, especially when lightweight models such as polynomial regression are used.

Future work will focus on extending this method to other mathematical functions and applying it to real-world embedded systems.

ACKNOWLEDGEMENT

This research has been supported in part by the European Union, within the program HORIZON-WIDERA-2023-ACCESS-02, under Grant Agreement No. 101160293, and in part by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia [Grant Number 451-03-137/2025-03/ 200102].

REFERENCES

[1] M. Z. Ali, M. N. S. K. Shabbir, X. Liang, Y. Zhang and T. Hu, "Machine Learning-Based Fault Diagnosis for Single- and Multi-Faults in Induction Motors Using Measured Stator Currents and Vibration Signals," *IEEE Transactions on Industry Applications*, 55(3), pp. 2378-2391, 2019, doi: 10.1109/TIA.2019.2895797

[2] D. Neupane, M. Reda Bouadjenek, R. Dazeley, S. Aryal, "Data-driven machinery fault diagnosis: A comprehensive review," *Neurocomputing*, Vol. 627, 2025, 129588, doi:10.1016/j.neucom.2025.129588

[3] M. Kampouridis, N. Vastardis, G. Rayment. "Using machine learning for fault detection in lighthouse light sensors." *arXiv preprint arXiv:2409.05495* (2024)

[4] W. Nelson, C. Dieckert, "Machine Learning-Based Automated Fault Detection and Diagnostics in Building Systems" *Energies*, 17, no. 2: 529, 2024, doi:10.3390/en17020529

[5] C. Kalaskar, T. Somasundaram, "Fault Tolerance of Cloud Infrastructure with Machine Learning", *Cybernetics and Information Technologies*, 23(4): 26-50, 2023, DOI:10.2478/cait-2023-0034

[6] N. T. R. Chillapalli, S. Murganoor, "AI and Machine Learning Solutions for Real-Time Fault-Tolerant System Design in Critical Applications", *Cybernetics and Information Technologies*, 28(1), 2025, DOI:10.52783/anvi.v28.2239

[7] M. Haroon, Z. A. Siddiqui., M. Husain,, A. Ali, T. Ahmad, "A Proactive Approach to Fault Tolerance Using Predictive Machine Learning Models in Distributed Systems", *Intern. Journ. of Experim. Research and Review*, 44, 208-220, 2024, doi:10.52756/ijerr.2024.v44spl.018

[8] B. Taylor, V. S. Marco, W. Wolff, Y. Elkhatib, Z. Wang, "Adaptive deep learning model selection on embedded systems", 19th ACM SIGPLAN/SIGBED Int. Conf. on Lang., Comp., and Tools for Emb. Syst. (LCTES), 2018, USA, 31-43, doi:10.1145/3211332.3211336

[9] S. S. Saha, S. S. Sandha, M. Srivastava, "Machine Learning for Microcontroller-Class Hardware: A Review," in *IEEE Sensors Journal*, 22(22), 2022, pp. 21362-21390, doi: 10.1109/JSEN.2022.3210773

[10] Haigh, K.Z., Mackay, A.M., Cook, M.R., Lin, L.G., "Machine Learning for Embedded Systems: A Case Study", 2015, https://api.semanticscholar.org/CorpusID:16033444

[11] N. Nosrati, M. Jenihhin, Z. Navabi, "MLC: A Machine Learning Based Checker for Soft Error Detection in Embedded Processors", IEEE 28th Intern. Symp. on On-Line Testing and Robust System Design (IOLTS), 2022, https://doi.org/10.1109/IOLTS56730.2022.9897309

[12] Branco S, Ferreira AG, Cabral J. "Machine Learning in Resource-Scarce Embedded Systems, FPGAs, and End-Devices: A Survey", Electronics, 2019; 8(11):1289. https://doi.org/10.3390/electronics8111289

[13] M. Gossel, V. Ocheretny, E. Sogomonyan, D. Marienfeld, "New Methods of Concurrent Checking", Springer, 2008.