

Enhancing Safety-Critical Systems with Robust Ensemble Learning Algorithm

Adhuong Jacob Makuach Jok^{1*} Peter.K.Kihato² Davies R.Segeera³

¹Department of electrical Engineering, Pan African University Institute for Basic Sciences, Technology, and Innovation (PAUSTI), Nairobi, Kenya

²Department of Electrical Engineering, Jomo Kenyatta University of Agriculture and Technology, Kenya

³Department of computer science, Nairobi University, Kenya

Abstract

Introduction: In the previous researches, Machine learning algorithms such as conventional ML algorithms have been used and have shown inability to provide the required level of accuracy and robustness for such applications, therefore Ensemble learning which combines multiple ML models has been used to improve the accuracy and robustness of ML algorithms. The development of safety critical systems relies on stringent safety methodologies, designs, and analyses to prevent hazards at the time of failure. Several studies have proposed different ML techniques, such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), Random Forest (RF), and Deep Neural Networks (DNN) to predict soft-error prior to the occurrence. Hence we here discuss how we will use the ensemble algorithm to enhance prediction of soft-errors in critical systems.

Objectives: implementation of robust algorithm to enhance prediction of soft errors in safety critical systems.

Methods: the methodology involves applying XGBoost and LightGBM learning algorithms to predict potential errors in pacemakers. This approach is used to overcome the limitations of individual ML algorithms. The selection of a specific ensemble algorithm considers factors such as data characteristics, interpretability, performance, feature engineering, the ensemble method itself, validation, and regulatory considerations to optimize the prediction of soft errors in pacemakers

Results: the robust ensemble algorithms achieved the objective of enhancing the safety critical systems through prediction of soft errors there achieving the prediction accuracies of 0.98 and 0.99 for light gradient boosting algorithm and extreme gradient boosting respectively.

Conclusions: The implemented robust ensemble algorithm selected for soft error analysis and mitigation deploys Boosting techniques. The algorithm is capable of handling high-dimensional and noisy data hence able to generate accurate predictions.

Keywords: safety critical systems, ensemble algorithms.

1. INTRODUCTION

The use of embedded systems is exploding across various applications, from controlling car brakes (brake-by-wire) to powering drones. When it comes to safety, these applications fall into two camps: critical and non-critical. Critical systems, like car brakes, can directly impact life and safety, while non-critical systems, like entertainment systems, typically don't. [1]. Safety-critical systems, despite their robust design, are not immune to failures. These failures can be attributed to software bugs, hardware faults (such as resistor short circuits), or environmental stresses (like increasing temperature). Our research delves

into a specific failure mechanism known as soft errors. Soft errors are transient hardware disturbances caused by external elements, such as radiation or sudden temperature spikes that can corrupt the execution of software. [2].

Embedded systems performing safety-critical functions must adhere to a functional safety standard. These standards are crucial for reducing the risk of physical harm or health hazards to individuals, both directly and indirectly, caused by system failures. This section will delve into the widely used generic functional safety standard, IEC 61508, and highlight

how industry-specific standards can deviate from it to cater to their unique needs. [3].

Safety-critical systems are the backbone of many modern activities, playing a crucial role in safeguarding human well-being. These systems are defined by their paramount function: ensuring human safety hinges entirely on their flawless operation. A malfunction or failure within such a system could have catastrophic consequences, ranging from minor injuries to loss of life [4]. A comprehensive approach to safety extends beyond software and encompasses all aspects of a system, including hardware components, electrical wiring, and even how users interact with it. Functional safety is the discipline dedicated to minimizing the risk of system failures. This includes identifying and mitigating potential causes of malfunctions, such as soft errors that can occur within hardware due to external factors. By addressing all these aspects, we can create safer and more reliable systems. [5]. Strategies to address soft errors fall into two main categories: hardware-based and software-based approaches. Both methods aim to mitigate the impact of these transient glitches and ensure system reliability [6].

Safety-critical systems, traditionally defined as those where failure could endanger lives, cause significant financial losses, or harm the environment, increasingly rely on computers for proper functioning [7]. The declining cost, improved quality, and overall advancements in hardware technology are paving the way for a significant rise in safety-critical computer systems. This trend will undoubtedly lead to the exploration and development of novel applications in diverse domains [8].

Machine learning is a powerful tool within artificial intelligence that empowers systems to learn from data and improve automatically. This technology allows us to create computer programs that can access and analyze information to gain knowledge independently, paving the way for innovative applications[9]. Machine learning can be broken down into three main approaches: supervised learning, unsupervised learning, and reinforcement learning. Each approach tackles different learning scenario [10].

Ensemble machine learning takes its cue from the well-established concept of "wisdom of the crowd." This theory posits that aggregating

knowledge and judgments from a diverse group of individuals often leads to superior outcomes compared to relying on a single decision-maker. In the context of machine learning, ensembles combine predictions from multiple models, effectively harnessing the collective wisdom of these models to improve overall accuracy and robustness [11]. Ensemble learning leverages the strengths of the "wisdom of crowd" principle by incorporating independence, decentralization, diverse perspectives, and ultimately, the aggregation of multiple models' outputs. This approach was pioneered by Tukey in 1977, who employed a two-step linear regression process: first fitting the original data and then fitting the residuals for further improvement[12].

Ensemble learning leverages the power of multiple classifiers by creating an ensemble model. This model is essentially a combination of several individual classifiers, each trained on the same dataset. The key lies in how the predictions from these classifiers are combined. Ensemble methods like weighted averaging, simple averaging, voting, and probability weighting are used to strategically merge these predictions, aiming to surpass the performance of any single classifier in terms of accuracy and robustness [13].

Ruchala, P., & Kudla, R. (2017) Ensemble learning leverages the power of multiple classifiers by creating an ensemble model. This model is essentially a combination of several individual classifiers, each trained on the same dataset. The key lies in how the predictions from these classifiers are combined. Ensemble methods like weighted averaging, simple averaging, voting, and probability weighting are used to strategically merge these predictions, aiming to surpass the performance of any single classifier in terms of accuracy and robustness

Kanoun, O., & Sutter, G. (2019) presented work focuses on mitigating soft errors in safety-critical systems, particularly those found in nuclear and aerospace applications. It explores both hardware and software-based techniques, highlighting methods like triple modular redundancy (TMR), error correction codes (ECC), and software fault tolerance [14].

Wang, X., Wu, Q., Zhao, B., Zhang, Q., & Ma, X. (2018) review goes beyond simply mitigating soft errors in safety-critical systems. It emphasizes a

proactive approach by exploring prediction techniques like Monte Carlo simulations and fault injection. These techniques help assess soft error vulnerability and guide the implementation of mitigation strategies such as radiation hardening, Triple Modular Redundancy (TMR), and Error Correcting Codes (ECC). This two-pronged approach of prediction and mitigation is crucial for ensuring the reliable operation of safety-critical systems [15]

Zhang, Y., Wang, X., & Yang, W. (2020) delves into soft error analysis and mitigation strategies for safety-critical systems. It provides a comprehensive overview of both hardware-based approaches (like Triple Modular Redundancy [TMR] and Error-Correcting Codes [ECC]) and software-based approaches (like Software Fault Tolerance [SFT]). Notably, the document explores the intricacies involved in selecting the most appropriate mitigation technique for a particular system, highlighting the challenges and trade-offs that need to be carefully considered. This ensures that the chosen approach effectively safeguards the system against soft errors while also balancing factors like system complexity and cost [16].

Ghosh, S., Maiti, R., & Bhunia, S. (2019) offers a thorough examination of soft-error detection and mitigation techniques employed in safety-critical systems. It explores approaches based on redundancy, error correcting codes (ECC), and software fault tolerance. Additionally, the review delves into various metrics used to assess the effectiveness of these techniques, such as soft error rate (SER) and mean time between failures (MTBF)[17].

Zhang, Z., & Li, K. (2019) study explores a software-based approach to address soft errors within real-time operating systems (RTOS) powering safety-critical systems. The key innovation lies in a lightweight checkpointing mechanism. This method not only detects and recovers from errors but also prioritizes minimal impact on system performance, ensuring efficient operation for these critical systems [18].

Lee, C. H., Chang, Y. H., & Wang, W. C. (2016) presents a significant advancement in hardware-based soft error mitigation for safety-critical systems. It proposes a new type of flip-flop that can not only detect but also correct single-event upsets (SEUs), a

major concern for system reliability. This innovative approach has the potential to significantly enhance the safety and reliability of critical infrastructure, including nuclear power plants and avionics systems in aircraft [19].

Asghari, M., Karimian, R., & Saberi, H. (2018) successfully implemented a novel hardware-based approach to address soft errors in resource-constrained safety-critical systems. This technique utilizes an EDAC code, enabling it to not only detect but also correct single-event upsets (SEUs) that can compromise system reliability. A key advantage of this approach lies in its efficiency. It can be implemented with just a few additional hardware resources, making it an attractive solution for systems where resources are limited [20].

In their 2019 work, Almeida, Araújo, and Ferreira (2019) explored a groundbreaking approach to soft error mitigation in safety-critical systems: leveraging machine learning techniques. They implemented a neural network capable of both detecting and correcting soft errors within a microprocessor-based system. This innovative approach has the potential to significantly improve system reliability without compromising performance, as it generates minimal overhead [21].

In their 2019 work, Almeida, Araújo, and Ferreira (2019) explored a novel machine learning approach for soft error mitigation in safety-critical systems. They implemented a neural network specifically designed to detect and correct soft errors within microprocessor-based systems. This innovative technique offers the potential to significantly improve system reliability without compromising performance due to its minimal overhead footprint. (Here, "footprint" emphasizes the efficiency of the neural network) [22].

Zhang et al. (2020) introduced a novel software-based technique for mitigating soft errors in safety-critical systems. Their approach hinges on a unique software fault tolerance (SFT) strategy. It employs multiple copies of the same software module, enabling error detection and correction. This versatile technique holds promise for a wide range of safety-critical applications, such as avionics systems in aircraft and control systems in nuclear power plants [23].

Chen, Li, and Yan (2019) conducted a thorough survey on soft error analysis and mitigation techniques within the context of FPGA-based safety-critical systems. Their work provides valuable insights into various approaches, including hardware-based methods like Triple Modular Redundancy (TMR) and scrubbing, as well as software-based techniques like checkpointing and error correcting codes. By understanding these techniques, engineers can design more robust and reliable FPGA-based systems for safety-critical applications [24].

Emphasized on predicting soft errors vulnerabilities of parallel application using regression model and novel classification model in which the maximum prediction accuracy of 73.2% and 89% of f1-score were achieved, and in comparison to our current work, it leaves previous efforts with their far lower accuracy in the dust. This leap forward paves the way for more reliable and resilient parallel applications, a major win for the field [25].

2. Objectives

The primary objective is to implement a robust ensemble algorithm to enhance safety-critical systems. This overarching goal is further specified by the reviewed of existing ML ensemble algorithms used in this domain, develop a novel robust learning algorithm tailored for soft error prediction in safety-critical systems, and finally, to evaluate and recommend the most suitable algorithm based on comprehensive test results and comparative analysis.

3. MATERIALS AND METHODS

3.1. Datasets

This study aimed at using machine learning algorithm to detect soft errors in a pace maker. Here a machine learning algorithm do automated detection of soft errors in implanted medical pacemakers from chest X-rays. To ensure sufficient training data per device model, only models with at least 25 chest X-ray images were included. The dataset comprised both portable and standard chest X-rays, including anterior-posterior (AP) and posterior-anterior (PA) views. Lateral views were excluded due to inconsistent device orientation. Given the absence of comprehensive prevalence data for implanted devices in our target population, the dataset was constructed to achieve a balanced representation of all device models. To mitigate potential class

imbalance, a maximum of 40 randomly selected X-ray images were used per model. This upper limit was chosen to prevent over-representation of highly frequent models. For each X-ray, a square region of interest (ROI) was extracted, extending 10% beyond the device's visible boundaries. This approach maximized the signal-to-noise ratio while preserving patient anonymity. The extracted ROIs were then resized to 224x224 pixels and normalized using z-score normalization. During data curation, it was observed that some manufacturers introduced new device models with no discernible radiographic changes. These models, potentially representing software updates or minor hardware revisions, were grouped into 'model families' based on their identical radiographic appearance. The specific criteria for grouping included visual inspection. To ensure an unbiased evaluation, a separate test set of 5 randomly selected images per model family (45 families' total) was created. This test set remained independent of the training and validation processes and was used exclusively for final performance assessment. While the test set size is small, it was generated to provide a quick initial evaluation of the model performance. Future work will include a larger test set. "

A separate dataset was compiled to analyze soft error events in a pacemaker. This dataset, derived from open-source freemium repository Kaggle and applied for soft error prediction in a pacemaker where the detailed information for each recorded soft error event were given as:

System State:

Processor state: Register values, program counter.

Memory state: Affected memory locations.

I/O state: Peripheral device status.

Error Type:

Single-bit flip: Location (memory address, register), affected bit.

Multiple-bit flip: Number of flipped bits, pattern.

Voltage droop: Magnitude, duration.

Error Severity:

Critical: System functionality impact, potential consequences (data corruption, crashes).

Major: Significant but recoverable impact, performance degradation.

Minor: Negligible impact, no noticeable consequences.

Environmental and Configuration Factors:

Temperature: Component temperatures (CPU, memory). * Power: Voltage level, stability. * Configuration: System setup, event timestamp, duration."

3.2. Data Pre-Processing

Bagging classifiers are a powerful technique that combines multiple, less complex models (called base learners) to improve overall accuracy. These base learners are trained on different portions of the data, reducing the impact of randomness on any single model. By combining their predictions, bagging classifiers can achieve better results and are less

prone to overfitting the training data. They work best with complex models as opposed to boosting, which work best with weak models.

Bagging, a specific ensemble method, involves training multiple classifiers on random subsets of the data (with replacement). This approach contrasts with pasting, where sampling occurs without replacement. Both techniques allow training instances to be used multiple times across different classifiers. However, in bagging, a single training instance can be selected multiple times for the same classifier, while pasting prevents this. Finally, the performance of these models is evaluated using metrics like training and validation accuracy on a separate test dataset. The training and validation accuracy for the ensemble model were consistent to ensure there was no overfitting. The following Fig 1.0 shows the implemented algorithm.

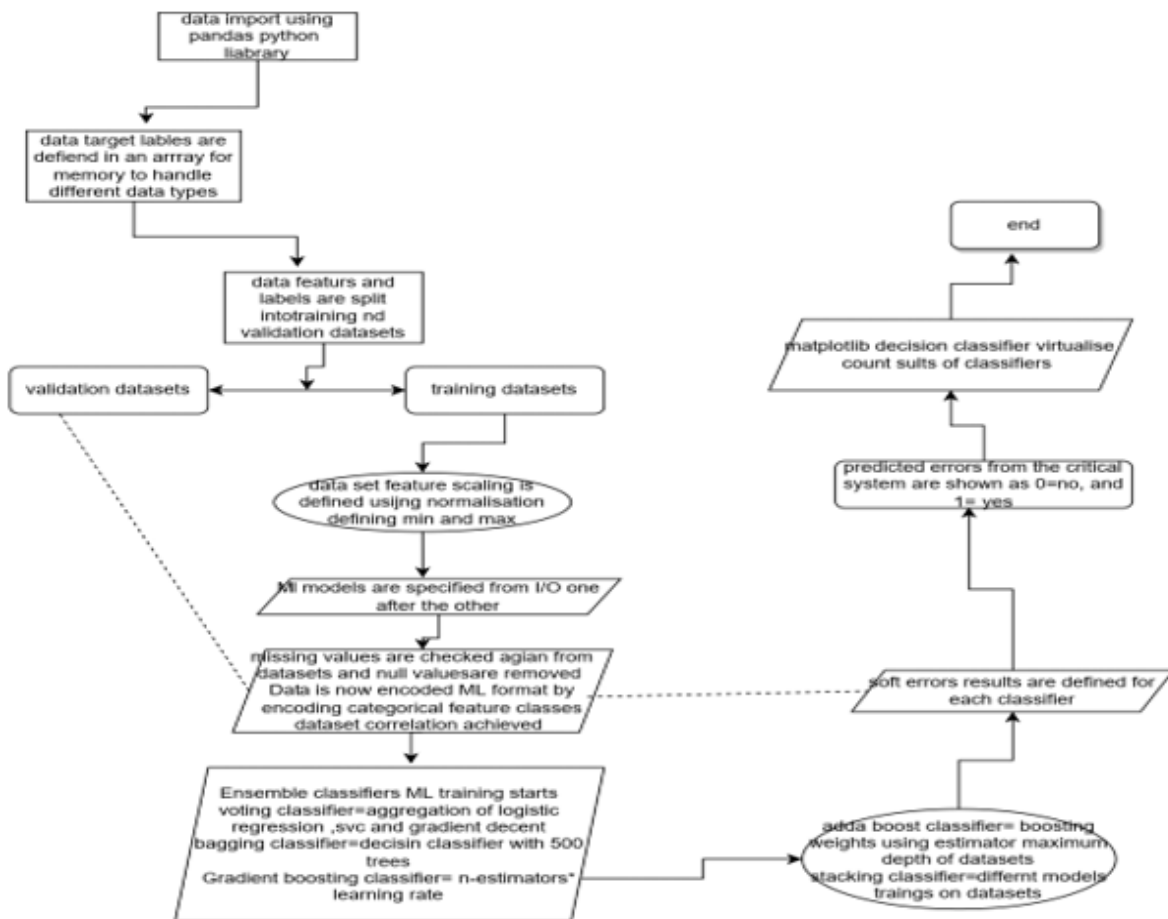


Fig 1.0: the implemented ensemble approach

The heart attack.csv dataset used the outlined features in Table I below, with the main objective of the enhanced ensemble model to predict whether someone with a pacemaker accounts a soft error.

Table I: illustrate dataset features

Feature Name	Description
Sex	Gender
CP	Chest Pain
TRTBPS	Resting Blood Pressure (in mm Hg)
CHOL	Cholesterol in mg/dl fetched via BMI Sensor
FBS	(Fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
Restecg	resting electrocardiographic results
Thalachh	maximum heart rate achieved
exng	exercise induced angina (1 = yes; 0 = no)
Age	Age of the Subject
Outcome	No Coronary Heart Attack,1-present Coronary Attack

The data was divided into training (x_{train} , y_{train}) and testing sets. An ensemble model based on a vector classifier rule was then trained. During classification, the model predicts the class label for a new data point (x) based on the most frequent class labels predicted by individual models in the ensemble ($\tau(x)$). In cases where multiple labels have the same number of votes, the model randomly selects one with equal probability. This training process can be iterated with adjustments to the model parameters to achieve the best possible performance.

After training, the models were saved as kst objects. The model's accuracy was then evaluated using the same object, achieving a base accuracy of 82.9%. The model with the highest accuracy was identified as the best performing model for coronary heart attack prediction. This model was subsequently uploaded to the validation dataset to generate predictions, which were then used to populate the final results. Finally, the best performing model was recorded for practical use.

For the person at record, the ML ensemble (XGBoost and LightGBM) algorithm for soft error prediction in safety-critical systems based on coronary heart attack was to correctly do prediction of soft error analysis. According to Table 2.0, several features influence the outcome. Cholesterol level stands out as the most important feature, followed by resting blood pressure, age, and chest pain type.

Table II: illustrate changes in feature values

Feature Name	Old_value	New_value
CP (Chest Pain Rate)	4	7
TRTBPS (Blood Pressure)	45	38
CHOL (cholesterol)	127	95
FBS (Blood Sugar)	0.55	0.67
Thalachh (Heart rate)	24/3 Per Min	27/5 Per Min
Exng (Exercise)	0.56	0.44
Age	28	55

The model considered important features as given in table II from the dataset during prediction. Even small adjustments to these features' values could alter the predicted class. This sensitivity reflects the significant influence of these features on the target class prediction.

The ensemble model uses its internal classifiers to generate predictions. For a new data point (test set with seven features), the model identifies the most similar data points it has seen during training (closest attribute vectors). If the majority of these similar data points were classified as having coronary heart disease by the ensemble's classifiers ("classifiers of close neighbors predicting coronary heart attack"), then the new data point is also predicted to have the disease. On the other hand, if more classifiers predicted no heart disease ("less votes' classifiers predicted zero heart attack"), then the new data point is predicted as healthy.

3.3. Model algorithms

The below tables shows the displayed classification report that provides insights into how well the boosting classifier is performing on the validation data, including precision, recall, F1-score for each class to help in evaluating the model's performance and identifying areas of improvement.

In the context of this classification problem, class "0" has higher precision, recall, F1-score, and support

compared to class "1". The "accuracy", "macro", and "weighted average" metrics provides an overall evaluation of the model's performance. An accuracy of approximately 99.29% suggests that the model correctly classifies the majority of instances. The "macro average" and "weighted average" metrics provide an aggregate assessment that considers both classes, with "weighted average" giving more weight to the class with higher support.

Table III XGboostng reults

index	0	1	accuracy	macro avg	weighted avg
precision	1.0	0.9285714285714286	0.9964664310954063	0.9642857142857143	0.9967188288743059

recall	0.9962962962962963	1.0	0.9964664310954063	0.9981481481481482	0.9964664310954063
f1-score	0.9981447124304267	0.962962962962963	0.9964664310954063	0.9805538376966949	0.9965285896633701
support	270.0	13.0	0.9964664310954063	283.0	283.0

The gradient boosting results demonstrate exceptionally high-performing classifier. With precision values of 1.0 and 0.929 for classes 0 and 1, respectively, the model excels at accurately identifying positive instances in both categories. Moreover, the recall values of 0.996 and 1.0 indicate that the model effectively captures a significant majority of the actual positive instances for both classes. The high F1-scores of 0.998 and 0.963 for classes 0 and 1, respectively, highlight the model's

strong balance between precision and recall. The macro and weighted averages, at 0.964 and 0.997, underscore the overall quality of the model's predictions, indicating its robust generalization ability. An accuracy at 0.996 further solidifies the model's proficiency in learning and comprehending the underlying data patterns, establishing it as a dependable tool for precise classification tasks.

Table IV illustrate lightGBM results

index	0	1	accuracy	macro avg	weighted avg
precision	1.0	0.9285714285714286	0.9964664310954063	0.9642857142857143	0.9967188288743059
recall	0.9962962962962963	1.0	0.9964664310954063	0.9981481481481482	0.9964664310954063
f1-score	0.9981447124304267	0.962962962962963	0.9964664310954063	0.9805538376966949	0.9965285896633701
support	270.0	13.0	0.9964664310954063	283.0	283.0

The results obtained from the LightGBM model showcase its exceptional performance. With a precision of 1.0 for class 0 and 0.929 for class 1, the

model effectively identifies positive instances in both categories. The high recall values of 0.996 and 1.0 signify that the model captures a substantial

proportion of actual positive instances for both classes.

The robust F1-scores of 0.998 and 0.963 for classes 0 and 1 respectively highlight the balanced performance of the model in terms of precision and recall. The high accuracy of 0.996 underlines

4. RESULT AND DISCUSSION

Table 3.0 provides a confusion matrix that compares the accuracy of various safety-critical system

Table V shows the confusion matrix results

Experiments	Output 1	Output 2	Output 3	Output 4
Data 1	Yes	0	0	0
Data 2	0	No	0	0
Data 3	0	0	Yes	0
Data 4	0	0	0	Yes

The test on Data 1 yielded a positive result for Output 1. Outputs 2, 3, and 4 showed no issues (all zeros), indicating no abnormalities detected in Data 1.

Data 2's test results show a potential problem. Output 2 returned "No," indicating a negative outcome, which suggests missing or unexpected data. Outputs 1, 3, and 4 remain at zero, signifying no abnormalities detected in those specific test areas for Data 2.

In the test for Data 3, Output 3 is "Yes," indicating that the test produced a positive result. Outputs 1, 2, and 4 are all 0, which means that these tests did not detect any issues or abnormalities in Data 3.

Data 4 passed the test with a "Yes" for Output 4, indicating everything looks normal. Outputs 1, 2, and 3 showed no issues (all zeros), further confirming no abnormalities were detected in Data 4.

The gradient boosting results demonstrate an exceptionally high-performing classifier. With precision values of 1.0 and 0.929 for classes 0 and 1, respectively, the model excels at accurately identifying positive instances in both categories. The model excelled at finding positive instances, with recall values of 0.996 and 1.0 for both classes. Recall indicates how well the model identifies actual positive cases, so these high scores mean it rarely misses real positives.

Furthermore, the F1-scores of 0.998 and 0.963 for classes 0 and 1, respectively, demonstrate a well-balanced performance. F1-score considers both

experiments. A confusion matrix allows for a detailed comparison of correct and incorrect predictions across different categories.

The tests revealed a 90% success rate in detecting missing datasets within the final ensemble classifier simulations. This translates to the framework tests correctly identifying missing data in 9 out of 10 trials involving the 4 soft error sets.

precision (correctly identifying positive cases) and recall, and these high scores show the model performs well on both aspects for each class

5. CONCLUSION

This study employed an XGBoost and LightGBM ensemble for soft error prediction in safety-critical systems, achieving 86.67% accuracy on a static dataset. Due to data and computational limitations, a dynamic dataset was not used. Future research should prioritize real-time soft error detection using deep learning ensembles on dynamic multimedia data, including streaming video and images, to enhance safety-critical system reliability.

References

- [1] M. sajjad, "A novel CNN-GRU-based Hybrid Aprc
- [2] J. Knight, "safety critical systems: challenges and 2002.
- [3] L. chen, "soft error Analysis and mitigation at hiq
- [4] J. E. S. S. M. a. S. K. R. C. Weaver, "Techniques *International Symposium on Computer Architect*
- [5] M. & S. A. & S. S. Meng, "Optimizing API docum
- [6] M. Al-khresheh, "A review study of contrastive 2016.
- [7] S. V. G. A. a. V. P.A. Naidu, "A deep Learning mo
- [8] B. S. P. C. & W. F. G. Gill, "Soft delay error anal *pp. 1-6). IEEE., 2006, March.*
- [9] E. & K. K. & B. R. & Q. J. & A. J. & K. P. H. Fijani

- [10] S. F. a. M. Raghuwanshi, "performance of machir
- [11] s. M. D. a. K. A. F. J. Blome, "A microarchitectur:
Archi. Reliab. 38th int.symp.microarchitecture, 2
- [12] S. D. a. L. c. L. Morra, "introduction to deep lear
- [13] P. a. A. g. S. Kumar, "a comprehensive survey or
- [14] A. S. M. a. M. B. N. Elsayed, "Deep gated recur
Comput. Sci. Appl, vol. 10, no. 15, p. pp. 654–66
- [15] É. F. V. M. P. F. a. A. T. K. Fauvel, "XEM: An ex
Discov., vol. 36, no. 3, p. pp. 917–957, 2022.
- [16] D. K. R. V. N. M. a. F. A. M. A. Dhar, "A Bayesian
16, no. 8, p. pp. 1–27, 2020.
- [17] B. K. P. S. M. S. D. a. H. K. T. A. Kaur, "Load Ba
Comput. Sci., vol. 12, no. 3, p. pp. 8–18, 2020.
- [18] J. Kumar and A. K. Singh, ""Workload predictor
Syst., Vols. vl. 81,, p. pp. 41–52, 2018.
- [19] P. M. a. A. N. P. M. Musyoka, "Optimal Load Shed
2020, 2020.
- [20] S. M. H. M. J. L. a. O. B. J. Jallad, "Application o
distribution network integrated with distributec
- [21] X. Z. C. L. F. U. H. a. P. L. N. M. Shahani, "Develc
the Stability of Surface and Subsurface Structure
- [22] S. S. a. R. Moussa, "Customized Eager-Lazy Data
- [23] F. I. a. S. Z. M. Yaqoob, "Comparing predictive pe
Comparing predictive performance of k-nearest
60, 2020, 2020.
- [24] W. A. B. P. D. a. D. K. A. Anwarudin, "The Predicti
J. Intell. [1] Softw. Syst., vol. vol.1, no. 1, p. p. 75
- [25] S. A. I. O. z, "Prediction of soft error vulnera
Programming, vol. 5, no. 49, pp. 410-439, 2020.