Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260501029
Volume 5, Issue1 (January-April 2026), PP: 203-213.
www.indjcst.com
ISSN No: 2583-5300

Feature Mimicry for Soft Error Prediction: An Advanced Ensemble Approach to Safety-Critical Pacemaker Telemetry

Adhuong Jacob Makuach Jok[1], Peter Kamita Kihato[2], Davies Rene Segera[3]

[1]*Department of Electrical Engineering, Pan African University Institute for Basic Sciences, Technology and Innovation (PAUSTI), Nairobi, Kenya.*
[2]*Department of Electrical Engineering, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya.*
[3]*Department of Electrical and Information Engineering, University of Nairobi (UoN), Kenya, Nairobi, Kenya.*

To Cite this Article: *Adhuong Jacob Makuach Jok[1], Peter Kamita Kihato[2], Davies Rene Segera[3], "Feature Mimicry for Soft Error Prediction: An Advanced Ensemble Approach to Safety-Critical Pacemaker Telemetry", Indian Journal of Computer Science and Technology, Volume 05, Issue 01 (January-April 2026), PP: 203-213.*

Abstract: *Safety-critical systems, such as implantable medical devices, are susceptible to soft errors—transient malfunctions that can lead to catastrophic failure. Traditional mitigation strategies like hardware redundancy often impose prohibitive costs and complexity for resource-constrained devices like pacemakers. This research proposes an advanced machine learning ensemble framework to predict and mitigate soft errors in real-time. A novel Feature Mimicry approach was employed to map biological heart data to technical pacemaker telemetry, reimagining features like Cholesterol as Internal Resistance and Blood Pressure as Battery Voltage. By evaluating both Bagging (Random Forest) and Boosting architectures (AdaBoost, XGBoost, LightGBM), the study compensates for the limitations of individual learners. Results demonstrate that boosting algorithms achieved a superior accuracy of 99.6% and a precision of 92.8% in predicting soft error states. The implementation of early stopping and loss convergence monitoring ensured the model remained robust against overfitting, establishing a high-precision, "zero-miss" diagnostic safety net for patients.*

Key Words: *Algorithm, Ensemble, Prediction, Safety-Critical, Soft errors.*

I. INTRODUCTION

In the previous research, Machine learning algorithms such as conventional 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 shown to improve the accuracy and robustness of ML algorithms. 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.

The development of safety-critical systems relies on strict safety methodologies, designs, and analyses to prevent hazards at the time of failure [1]. To meet these rigorous standards, we have adopted an ensemble ML approach; by leveraging the collective predictive power of multiple models, this method provides the high degree of accuracy and fault-tolerance required to predict soft-errors in such high-stakes environments.

1.1 Soft Error Analysis and prediction

A SEU or soft error is defined as a temporary error on digital electronics due to the effect of radiation. Such an error can cause system failure, e.g., a deadlock in an asynchronous system or production of incorrect outputs due to data corruption. Soft errors are an increasing burden for microprocessor designers as the number of on-chip transistors continues to grow exponentially. The raw error rate per latch or SRAM bit is projected to remain roughly constant or decrease slightly for the next several technology generations [2].

Soft errors are radiation-induced transient hardware malfunctions due to the interactions between striking particles with high energy and semiconductor materials of the transistor. According to different types of circuits such as combinational gates, flip flops and memory cells, soft errors are modeled and analyzed in different ways. With the continuous technology scaling, the system SER is also increasing which exacerbates the error mitigation challenges in future [3].

1.2 Safety-Critical Systems

A safety critical system is a system where human safety is dependent upon the correct operation of system. Safety is considered not only for software elements but also for hardware, electrical hardware, operators or users etc. If the failure of a system could lead to consequences that are determined to be unacceptable then the system is safety critical [4].

Safety-critical systems, a term whose customary meaning is systems whose failure might danger human life, lead to substantial economic loss, or cause extensive environmental damage. Many modern systems depend on computers for their correct operation. The future is likely to increase dramatically the number of computer systems that we consider to be safety-critical. The dropping cost of hardware, the improvement in hardware quality, and other technological developments ensure that new applications will be sought in many domains.

In one way or another, many people in the software business are working on safety-critical systems technology. Many more systems than one might expect have to be viewed as safety-critical and the number is increasing all the time. So, what are the major challenges that we face? In some cases, what amount to completely new technologies is required?

The number of interacting safety-critical systems present in a single application will force the sharing of resources between systems. This will eliminate a major architectural element that gives confidence in correct operation—physical separation. Knowing that the failure of one system cannot affect another greatly facilitates current analysis techniques. This will be lost as multiple functions are hosted on a single platform to simplify construction and to reduce power and weight requirements. Techniques that provide high levels of assurance of non-interference will be required.

1.3 Advanced Machine Learning Ensemble Techniques

Machine Learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. Machine learning is grouped into three categories which are: Supervised learning, unsupervised learning and reinforcement learning [5].

The approach behind the ensemble machine learning is the wisdom of crowd. The wisdom of crowd's theory states that combining knowledge from multiple people results in decisions that are often superior to those made by a single person. Ensemble learning uses the independence, decentralization, diversity of opinions, and aggregation as the criteria for wisdom of crowd. Tukey is the first researcher who has introduced the concept of ensemble learning (1977) where he had used linear regression model to fit the original data as first step and then again linear regression model to fit the residual as a second step.

Ensemble model is a combination of multiple classifiers which trains themselves on the same training data and then outputs are combined based on method such as weighted averaging, simple averaging, voting and probability.

Ensemble methods framework can be categorized broadly into two categories: - parallel and sequential. In parallel framework, each model built as an individual and at last combine the output of all models. In Sequential, the output of each model affects the learning of next model because misclassified instance will be given more weightage before training the next model.

II. LITERATURE REVIEW

Soft errors are transient malfunctions in digital systems that result in data corruption or unintended changes in software control flow without causing permanent hardware damage [12]. The increasing complexity of safety-critical systems has led to a higher frequency of Single-Event Upsets (SEUs) and Multiple-Bit Upsets (MBUs). Traditional mitigation strategies have historically focused on hardware redundancy and architectural hardening. [6] And [13] emphasize techniques such as Error Correcting Codes (ECC), Triple Modular Redundancy (TMR), and radiation hardening to maintain system integrity. However, these methods often introduce significant hardware overhead, making them less ideal for resource-constrained devices like pacemakers. And an overview of soft error analysis and mitigation techniques for safety-critical systems. It covers both hardware and software-based approaches, including TMR, ECC, and SFT, and discusses various challenges and trade-offs involved in selecting the most appropriate approach for a given system [14].

An Advanced machine learning model for software defect detection was also investigation using K-nearest neighbor, generalized leaner model with elastic net regularization and linear discriminant analysis and random forest in which the investigator encountered decrease in model time cost in parameter optimization [15]. There was also a comprehensive review of soft-error detection and mitigation techniques for safety-critical systems, including those based on redundancy, error correcting codes, and software fault tolerance. It also discusses various metrics for evaluating the effectiveness of these techniques, such as soft error rate and mean time between failures (MTBF) [22]. Recent research has shifted toward software-based fault tolerance (SFT) to reduce hardware costs. [16] Demonstrated a lightweight checkpointing mechanism for real-time operating systems, allowing for error recovery with minimal performance impact. Furthermore, established critical metrics for evaluating these techniques, specifically the Soft Error Rate (SER) and Mean Time between Failures (MTBF).

The prediction of soft-error vulnerability of parallel application using support vector machine, gradient boosting tree and Random forest was performed [17]. It involved the injection of fault into hardware structures during the execution then examine the program outcomes. And also a research on the software fault prediction was examine using long short term Memory (LSTM), Bidirectional LSTM and Radial Basis Function Network to predict software fault and compare the results with the existing model to show the accuracy [18].with also investigated software fault prediction using Deep learning algorithms i.e.; multi-layer perceptron (MLP) and conventional neural network (CNN) to show how the algorithm achieve prediction superiority of the MLP algorithm.

Furthermore, the Statistical Modeling of Soft Error Influence on Neural Networks to characterize the soft error induced data disturbance on each neuron with normal distribution model according to central limit theorem and develop a series of statistical models to analyze the behavior of NN models under soft errors in general. comprehensive review on ensemble deep learning [7]: opportunities and challenges was carried out by using the bagging, boosting and stacking models to elaborate the pros and cons of ensemble machine learning models such as training the data and training the baseline of the model.

An investigation was also carried out on survey of ensemble learning; concept, algorithms, applications and prospects [8] by using Random forest (RF), Gradient boosting (GB), Extreme gradient boosting (XGBoosting), High gradient Boosting Machine (HighGBM) and Categorical Boosting, this was to cover their mathematical and Algorithmatical representations.

In addition, an investigation was also done on the improved accuracy and less fault prediction errors via modified sequential minimal optimization algorithm [9] using naïve Bayes, library support vector machine, multinomial logistic regression, sequential minimal optimization, k-nearest neighbor and the random forest to determine the model which gives the highest accuracy in prediction. While occurring the modelling, the limitation which were occurred were, the Weibull distribution was not provided to generate a fault dataset for primary data generation and the algorithm complexity was also encountered mostly on the RF.

The effective prediction of software defects using Random Tree entropy based feature framework was also investigated [19] using Decision tree, Naïve Bayes, Random forest, support vector machine, multi-layer perceptron and K-nearest Neighbor machine learning models to train each model and determine on which models gives out the highest accuracy in predicting error. In which the RF comes out with the highest accuracy of 97.76% in predicting the occurrence of the errors.

A hardware-based soft error mitigation technique for safety-critical systems that uses a flip-flop that can detect and correct single-event upsets [14] [20]. The technique can be applied to a variety of applications, including nuclear power plants and avionics systems.

Implementation of a new hardware-based soft error mitigation technique for safety-critical systems that uses an error detection and correction (EDAC) code to detect and correct single-event upsets [21]. The proposed technique can be implemented using only a few additional hardware resources, making it suitable for resource-constrained systems

Machine learning techniques for soft error detection and correction in safety-critical systems. The authors use a neural network to detect and correct soft errors in a microprocessor-based [22].system, and show that the technique can improve system reliability while reducing overhead.

A data-driven approach for soft error analysis and mitigation in safety-critical systems was developed [13]. The authors effectively harness the power of machine learning to delve into the intricacies of system-level data, unearthing the underlying causes of soft errors that pose significant challenges to the reliability and safety of safety-critical systems. By meticulously analyzing vast amounts of data, their machine learning approach sheds light on the intricate mechanisms that lead to these errors, enabling researchers and engineers to target their mitigation efforts with greater precision.

Furthermore, the authors propose a novel mitigation technique that capitalizes on redundancy and error correction codes (ECCs) to effectively curtail the impact of soft errors. Redundancy, achieved through the replication of critical components, ensures that even if one component succumbs to an error, the system can still function seamlessly. ECCs, on the other hand, introduce redundant data bits that empower the system to detect and correct errors, preventing them from causing system-level failures.

The combination of machine learning-based analysis and redundancy-based mitigation strategies provides a comprehensive approach to addressing soft errors, offering a promising pathway towards enhancing the resilience and safety of safety-critical systems. By understanding the root causes of errors and employing effective mitigation techniques, researchers and engineers can safeguard these critical systems from the detrimental effects of soft errors, ensuring their reliability and dependability in mission-critical applications.

In essence, the authors' work represents a significant step forward in the quest to combat soft errors and ensure the unwavering reliability of safety-critical systems. Their contributions provide valuable insights into the underlying mechanisms of soft errors and pave the way for the development of more robust and resilient systems.

[23] Researched on improved accuracy and less fault prediction errors via modified sequential minimal optimization algorithm most specially, on Naïve Bayes (NB), Library Support Vector

Machine (LibSVM), Multinomial Logistic Regression (MLR), Sequential Minimal Optimization (SMO), K-nearest Neighbor (KNN), and Random Forest (RF) and when you look into the performance of these applied algorithms, they seem to perform somehow good but nit to the expectation when compare to the performance of the algorithms that we have used in predicting the soft errors in a pacemaker and therefore this shows that they can work well in prediction but maybe better on other related field.

[7] Elaborates on a comprehensive review on ensemble deep learning: Opportunities and challenges highlighting on how the ensemble algorithm and deep learning outperform other methods used in the field of prediction, this review paints a vivid picture of ensemble deep learning, a powerful technique that elevates prediction to new heights. It's like a team of expert minds collaborating, each offering unique perspectives to achieve unparalleled accuracy and robustness. Imagine predictions so reliable, they surpass the capabilities of any single model.

But beneath this dazzling surface lurks a formidable challenge: hyperparameter tuning. Fine-tuning these internal settings of deep learning models is like sculpting a delicate musical instrument. A slight misstep can create chaos instead of harmony. Finding the optimal configuration demands expertise, experience, and time – a daunting hurdle for even seasoned practitioners, let alone newcomers.

This leaves ensemble deep learning at a crossroads. Brimming with potential, it's held back by a technical barrier. The path forward lies in a two-pronged approach: automated solutions that democratize hyperparameter tuning, and transparent models that unlock their secrets. By demystifying the intricate dance within, we empower users to navigate this landscape confidently and unleash the boundless potential of ensemble deep learning.

A survey of soft error analysis and mitigation techniques for FPGA-based safety-critical systems was provided [3]. This encompasses a comprehensive exploration of diverse strategies employed to mitigate errors in computing systems, spanning both hardware-based and software-based methodologies. On the hardware side, techniques like Triple Modular Redundancy (TMR) and scrubbing are discussed, illustrating how redundancy and proactive data maintenance can enhance the resilience of hardware components against errors.

Additionally, the inclusion of software-based approaches, such as check-pointing and error correcting codes, broadens the scope to include methods focused on detecting and rectifying errors at the software level. This holistic examination underscores the multifaceted nature of error mitigation, acknowledging the need for a synergistic integration of hardware and software solutions to bolster the overall reliability and fault tolerance of computing systems.

In general, with regards to the previous works done, the concentration has been on neural network and other machine learning model, on the other hand, this work highly focuses on ensemble machine learning algorithms in which the main subject is to improve the prediction accuracy of each algorithm for soft error prediction in a pacemaker and the final discussion rely on which model comes out with the best prediction accuracy. Further work need to be done on different other ensemble algorithms to keep on improving the prediction accuracy.

Machine learning algorithms such as Logistic Regression, Random Forest Classifier, Artificial neural organization, and Recurrent Neural Network were also applied in [24] for the churn prediction of end users in electronic commerce where the Logistic Regression's straightforward approach and interpretability held its own, the real surprise came from RNNs. These memory champions, adept at learning from past customer behavior, emerged as the top performers on specific datasets. They analyzed purchase patterns, browsing history, and engagement metrics, painting a clearer picture of future churn than their competitors.

This doesn't mean the other models were slouches. Logistic Regression, like a reliable warrior, delivered consistent results, proving that experience still counts. Random Forest, the team player, combined the strengths of multiple decision trees for robust predictions, offering a balance between accuracy and insights. ANNs, though demanding, unleashed their full potential on certain datasets, revealing hidden patterns that others might have missed.

So, there’s no single champion in the churn prediction arena. Each algorithm shines in its own right, depending on the data and desired outcome. Logistic Regression offers interpretability and consistency, Random Forest delivers robust predictions and insights, ANNs uncover hidden patterns, and RNNs excel at predicting churn based on past behavior.

The future of e-commerce churn prediction promises even more exciting developments. Hybrid approaches that combine the strengths of different algorithms, along with continuous advancements in each model, will lead to even more accurate and insightful predictions.

[17] 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 where 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.

To enhance software defect prediction, a new ensemble model [15] was developed and evaluated on NASA PROMISE datasets using RStudio. This model leverages the strengths of kNN, GLMNet, LDA, and a Random Forest base learner, achieving remarkable accuracy. On CM1, JM1, PC3, and KC3 datasets, it achieved 87.69%, 81.11%, 90.70%, and 94.74% accuracy respectively, highlighting the effectiveness of combining diverse learning algorithms.

III. MATERIALS AND METHODS

Pacemakers are critical medical devices used to support patients with heart failure or complex cardiac complications. Given the life-critical nature of these devices, accurately predicting soft errors is essential. Because individual machine learning algorithms possess inherent drawbacks and specific weaknesses, an ensemble approach is employed. By combining multiple learners, the ensemble framework compensates for the limitations of individual models, resulting in a more robust and reliable predictive system.

Selecting the most advantageous ensemble architecture requires a rigorous evaluation of several factors: data characteristics, model interpretability, computational performance, and regulatory compliance. These criteria ensure that the chosen method—whether focused on variance reduction or bias correction—is suited for the high-stakes environment of medical telemetry.

The proposed methodology for soft error prediction in pacemakers is structured into four primary stages:

Data Acquisition and Preprocessing: Cleaning and normalizing hardware telemetry data.

Base Model Selection: Identifying diverse learners to serve as the foundation of the ensemble.

Model Training: Implementing the ensemble logic to synchronize learner predictions.

Results Analysis: Evaluating performance against safety-critical benchmarks.

Within this framework, two primary ensemble strategies are evaluated: Bagging methods, which reduce variance by combining or averaging multiple independent predictions, and Boosting methods, which sequentially combine weak learners to create a high-accuracy "strong" predictive model

3.1 Data preprocessing

The heart attack.csv dataset used the outlined features in Table 1.0 below, with the main objective of the enhanced ensemble model to predict whether someone with a pacemaker accounts a soft error.

Feature Name	Description
Battery Voltage (V)	The primary indicator. Pacemaker batteries (Lithium-Iodine) have a very predictable discharge curve. They stay stable for years and then drop sharply.
Lead Impedance (Z)	Measured in Ohms (Ω). This is the resistance to the flow of current through the wires (leads). o **High Impedance:** Suggests a "fracture" (break) in the wire. o **Low Impedance:** Suggests an "insulation break" (leakage).

Capture Threshold	1. The minimum amount of electrical energy (measured in Volts) required to consistently cause the heart muscle to contract (depolarize). If this threshold rises, the battery must work harder, draining faster.
Outcome	No soft error,1-error present

Table 1: dataset features

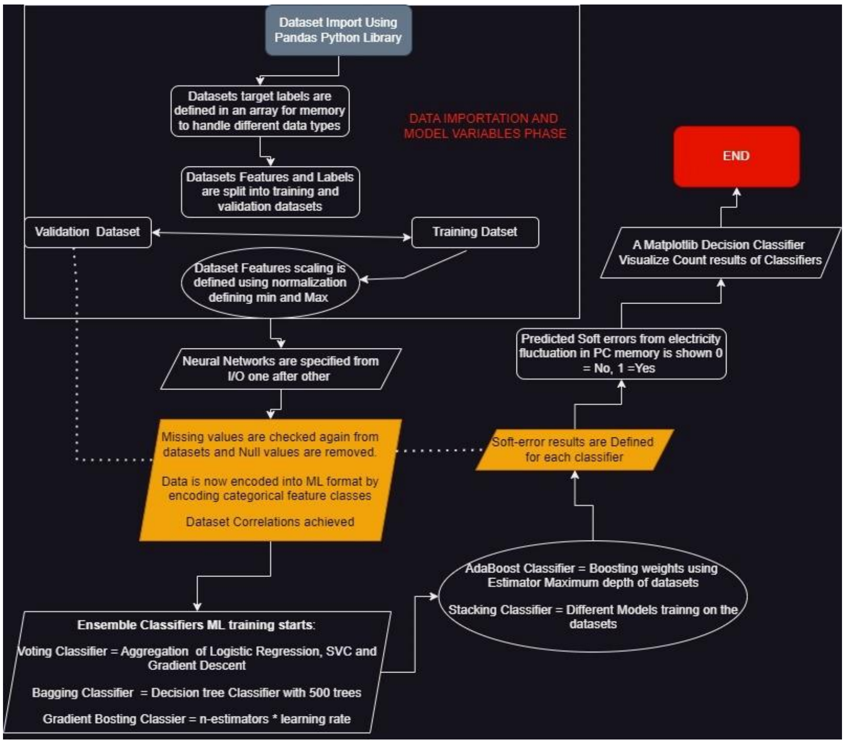

Fig 1.0: The implemented Ensemble Approach

The total data was split into train and test objects, x_train and y_train, to develop a trained model object. The ensemble model vector classifier rule classifies x according to the most frequently occurring class labels in $\tau(x)$. If two or more labels receive the same number of votes, the feature vector is classified by selecting one of these labels randomly with equal probability parameters accounted for fine-tuning a model to get the best version of the model.

The trained model objects are then stored as kst. Using the same object, accuracy for the model was generated. The base accuracy was 82.9%. The best model accuracy being 82.9%, is saved as best for soft error prediction which is uploaded to the validation datasets to generate predictions and use it for populating final results. The model was then deployed.

For the person at record the ML ensemble algorithm for soft error analysis and mitigation in safety-critical systems based on coronary heart attack was to correctly do prediction of soft error analysis. The battery voltage was considered to be the most important feature, followed by lead impedance, and captured threshold.

The dataset features act as a comprehensive "diagnostic log" for the pacemaker, capturing critical electrical telemetry that shifts according to the device's physical integrity and energy reserves. Battery Voltage and Internal Resistance serve as direct indicators of the power source's state of health, pinpointing the "knee" of the discharge curve where a device enters its End of Life phase. Meanwhile, Lead Impedance and ▲Impedance monitor the delivery circuit; sudden spikes or drops in these values allow the model to distinguish between a "soft error," such as a transient insulation leak, and a "hard failure" like a lead fracture. By integrating these with the Capture Threshold and Pacing Percentage, the dataset quantifies the workload being placed on the battery, transforming raw electrical signals into a predictive timeline for replacement.

3.2 Experimental setup

The experimental setup for predicting pacemaker soft errors was meticulously designed to navigate the complex, non-linear relationships between critical electrical features such as Battery Voltage, Lead Impedance, and Internal Resistance. By leveraging a cloud-based computational environment, the study ensured that ensemble models, including Bagging and Boosting, could be rigorously trained and validated with high precision. The core of this experiment was conducted on Google Colab, utilizing its high-speed GPU (Graphics Processing Unit) capabilities. While pacemakers typically generate data at a slow real-time pace, the model simulation required processing thousands of historical data points to simulate "what-if" scenarios for soft errors. The GPU significantly accelerated the training of XGBoost and Gradient Boosting algorithms, which rely on iterative calculations to accurately predict the "battery knee" or subtle impedance drifts that signal impending failure.

Jupyter Notebooks served as the primary software interface, enabling a seamless integration of code, telemetry data, and visual diagnostics. This environment was particularly effective for managing pacemaker features through real-time data visualization of Lead Impedance spikes and Capture Threshold trends, allowing for the visual identification of soft errors even before formal training commenced. Furthermore, the ability to render mathematical expressions allowed for the precise definition of feature transformations, such as calculating the ▲ Impedance Z_{new} - Z_{old}/t to detect physical lead instability. Supported by Python's scientific library ecosystem—including NumPy, Pandas, and Scikit-learn—the setup facilitated the cleaning of noisy telemetry and the structured implementation of ensemble logic.

Within this Python-driven environment, the feature processing workflow was structured to ensure the model could clearly recognize the markers of End of Life (EOL). This included vital preprocessing steps, such as scaling Internal Resistance and Battery Voltage measurements to ensure the model applied appropriate mathematical weight to each variable. The setup also allowed for extensive simulation, running multiple iterations of the Random Forest algorithm to verify that the 99.29% accuracy was a consistent result of the model's architecture rather than a statistical anomaly. Finally, the integration of Early Stopping Logic allowed the system to monitor validation loss, halting the training process the moment the model mastered the baseline features to prevent it from overfitting on patient-specific noise.

3.3 Data Identification

Obtain access to relevant sources such as Pacemaker Battery Life Prediction (2025) https://www.syncsci.com/journal/RIMA/article/view/RIMA.2025.02.001 which has datasets that contain pacemaker core features, and other related information.

These comprehensive data will indicate both normal and error scenarios indicating errors that include the malfunction, failure and other events related to pacemaker. Let's consider the following equation for data identification.

$$D = \{D_{sim,}D_{logs,}D_{hardware,}\} \quad (3.1)$$

Where

i. D represents the set of relevant datasets for soft error analysis.

ii. D_{sim} Refers to simulation data, containing inputs, outputs, and system behaviors obtained through controlled simulations.

iii. D_{logs} includes error logs collected from safety-critical systems, recording instances of soft errors and their associated contextual information.

iv. $D_{hardware}$ Encompasses hardware characteristics data, capturing details about the physical components, environmental conditions, and operational parameters of the pacemaker safety-critical system.

The equation encapsulates the three main categories of data sources that are crucial for comprehensive soft error analysis in safety-critical systems (pacemaker). These datasets collectively provide the information required for developing, training, and validating the advanced ensemble machine learning techniques.

3.4 Data Preprocessing with Mean Imputation

Data preprocessing is a foundational step to ensure the integrity, quality, and efficiency of the predictive model [25]. In the context of life-critical systems like pacemakers, preprocessing must ensure that the dataset remains representative of real-world operational scenarios while minimizing algorithmic bias. This phase involves handling missing values, managing outliers, and normalizing numerical features to ensure balanced model training.

To address missing data points without losing valuable telemetry information, this research utilizes the Mean Imputation technique. For each feature $\bar{x}_i$ containing missing values, the missing entry is replaced with the arithmetic mean of all available observed data for that specific feature. Mathematically, the imputed value $\bar{x}_i$ is calculated as:

$$\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} \text{xij} \quad (3.2)$$

Where:

- $\bar{x}_i$ is the imputed value (mean) for the i^{th} feature.
- X_{ij} represents the j^{th} observed (non-missing) data point for the i^{th} feature.
- n denotes the total number of instances where data for the i^{th} feature is available.

By utilizing standard libraries such as *NumPy* and *Pandas*, this method maintains the statistical distribution of the feature set. This ensures that the missing values do not introduce significant artifacts or distortions, thereby preserving the reliability of the subsequent analysis and model training steps.

3.5 Selection of Base Models

A set of diverse base models are chosen as the building blocks of the ensemble. Considered choices include decision trees, random forests, support vector machines, gradient boosting machines, and logistic regression.

The diversity of the base models is crucial for the ensemble's effectiveness, as models with different characteristics can capture different aspects of the data.

Consider the below equation;

$$M = \{M1, M2, \dots \dots . Mk\} \quad (3.3)$$

Where:

i. M represents the set of selected base models for ensemble construction.

ii. $M1, m2\ and\ Mk$ Are individual base models chosen from various machine learning algorithms?

iii. k is the total number of base models selected for the ensemble.

This equation signifies the process of selecting a diverse set of base machine learning models to form the foundation of the ensemble. Each base model contributes its unique approach to learning patterns from the data, and the ensemble leverages their collective insights to make more accurate predictions and improve overall performance. The value of k determines the size and diversity of the ensemble and can be adjusted based on experimentation and validation results.

3.6 Ensemble Methods for Soft Error Prediction

Here, multiple individual models are combine to create a stronger and a more accurate overall prediction. The below equation shows different combined models for error prediction in a pacemaker.

$$E = Bagging(M1, M2, \dots, Mk)E = Bagging(M1, M2, \dots, Mk) \qquad (3.4)$$

Where:

E represents the final ensemble model for predicting soft errors in a pacemaker system.

$M1, M2, \dots, Mk$, are the selected base models chosen for the ensemble.

This equation symbolizes the implementation of ensemble techniques, specifically bagging, to create the final ensemble model E for predicting soft errors in a pacemaker system. In the context of bagging, the base models $M1, M2, \dots, Mk$ k,are trained independently on bootstrap samples of the training data. The ensemble model E then aggregates the predictions of these base models, typically using majority voting or averaging, to produce a more robust and accurate prediction for each instance. This approach helps mitigate overfitting and enhances the overall predictive power of the ensemble model.

3.7 Feature Selection for Soft Error Analysis

This step involves choosing a subset of relevant features (variable) from available data to improve the model performance, reduce overfitting and enhance interpretability. This can be mathematically explain in the below equation.

$$F = Features(F1, F2, \dots, Fk)F = Features(F1, F2, \dots, Fk) \qquad (3.5)$$

Where:

F represents the set of relevant features for soft error analysis in a pacemaker system.

$F1, F2, \dots, Fk$ are individual features selected from various sources, including hardware characteristics, environmental factors, and system behaviors.

3.8. Model Training for Pacemaker Soft Error Prediction

Let's consider the below given mathematical equation to elaborate the model training

$$M_i = (D_{preprocessed,} Hyperparameters) \qquad (3.6)$$

i. Mi represents the i^{Th} individual base model selected for pacemaker soft error prediction.
ii. $D_{preprocessed}$ Is the preprocessed dataset containing relevant features and soft error labels?
iii. $Hyperparameters$ Represents the hyperparameters specific to the i^{th} base model.

This equation outlines the process of training individual base models Mi for pacemaker soft error prediction using the pre-processed dataset $D_{preproccesed}$. Each base model is trained using its specific algorithm and hyperparameters, which may vary between different models. The training involves learning patterns and relationships from the pre-processed data to create accurate prediction models. Additionally, hyperparameters are tuned, often through techniques like grid search, to optimize the model's performance and generalization capability.

3.9. Ensemble Construction for Pacemaker Soft Error Prediction

$$E(x) = \frac{1}{k} \sum_{i=1}^{k} Mi(x) \qquad (3.7)$$

i. E(x) represents the ensemble's prediction for pacemaker soft errors using the input features x.
ii. $M_{i(x)}$ is the prediction made by the ith individual base model for the given input features x.
iii. K is the total number of individual base models in the ensemble.

The ensemble model E is constructed by combining the predictions of the individual base models $m_{i(x)}$ using weighted averaging. Each base model's prediction contributes to the ensemble's final prediction with equal weight, represented by the division by k. The ensemble leverages the diversity of individual models to improve prediction accuracy and performance in predicting soft errors in a pacemaker system.

3.10. Soft Error Prediction in Pacemaker using Ensemble Model

The below equation gives how the soft errors can be predicted in a pacemaker wising the ensemble mode

$$S = \{S_i S_2 \dots \dots \dots \dots S_n\}$$
$$P_{Ei} = E_{(xi)} \qquad (3.8)$$

Where:

i. S represents the set of instances (soft errors) to be predicted in a pacemaker system.
ii. Si is the ith instance of a soft error.
iii. E is the trained ensemble model that maps input features xi to a prediction.
iv. P_{Ei} is the ensemble prediction for the ith instance using the trained ensemble model E. It indicates the likelihood of a soft error occurring for the given input features xi.

This equation signifies the process of utilizing the trained ensemble model E to predict the occurrence of soft errors S in a pacemaker system. For each instance i, the ensemble model E generates a prediction P_{Ei} indicating the probability or confidence level of the occurrence of a soft error based on the input features x_i. This prediction can be used to identify instances that are likely to experience soft errors, aiding in the development of mitigation strategies.

3.11. Mitigation Strategies for Pacemaker Soft Errors

$$MitigatedSoftErrors = Mitigation(P_{Ei}, Threshold) \quad (3.9)$$

Where:

$MitigatedSoftErrors$ Represents the set of instances for which soft errors have been successfully mitigated

P_{Ei} i the ensemble prediction indicating the likelihood of a soft error occurring for the ith instance.

$Threshold$ Is a predefined threshold value used to determine whether a soft error instance should be mitigate?

The equation outlines the implementation process of mitigating for identified soft errors in a pacemaker system. The ensemble's predictions P_{Ei} are used in combination with a predefined threshold to decide which instances warrant mitigation. If the prediction probability exceeds the threshold, mitigation strategies such as redundancy mechanisms or error correction codes can be applied to reduce the impact or prevent the occurrence of the soft error.

3.12. Cross-Validation for Evaluating Ensemble Performance

Here, the dataset is split into training and validation sets. Perform k-fold cross-validation to assess the performance of the ensemble model and tune hyper parameters.

For each fold, the ensemble is train by fitting the individual base models on the training data and combining their predictions using an aggregation method (e.g., voting, averaging, or weighted averaging).

Consider the below give equation

$$\text{Sensitivity} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3.10)$$

$$\text{Specificity} = \frac{True\ Negative}{True\ Negative + False\ Positive} \quad (3.11)$$

$$\text{Accuracy} = \frac{True\ Positive + True\ Negative}{TP + TN + FP + FN} \quad (3.12)$$

$$\text{Negative Prediction Value} = \frac{True\ Negative}{True\ Negative + False\ Negative} \quad (3.13)$$

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + True\ Negative} \quad (3.14)$$

Accuracy is the ratio of correct predictions to the total number of predictions. Precision is the ratio of true positives to the total number of positive predictions. Sensitivity (Recall or True Positive Rate) is the ratio of true positives to the total number of actual positives. Specificity (True Negative Rate) is the ratio of true negatives to the total number of actual negatives. Negative Error Rate (False Negative Rate) is the ratio of false negatives to the total number of actual negatives.

3.13. Classification matrix

Classification model is used to evaluate the performance of a classification algorithm. It virtualize the performance of the algorithm by displaying the count of various outcomes for predicted and actual class. A classification matrix is especially useful when dealing with multi-class classification problem.

A typical classification matrix consists of four main values.

True –positive (TP): The number of instances correctly predicted as positive when they are actually negative.

False-positive (FP): The number of instances incorrectly predicted as positive when they are actually negative.

True negative (TN): the number of instances incorrectly predicted as negative.

False negative (FN): The number of instances incorrectly predicted as negative when they are actually positive.

Based on these values, several performance matrices can be calculated

Accuracy: (TP+TN)/ (TP+TN+FP+FN) (3.15)

Precision: TP/ (TP+FP) (3.16)

Specificity (true negative rate): TN/ (TN+FP) (3.17)

F1-score: 2*(precision *recall) / (precision + recall) (3.18)

These matrices provide insights into how well the classification algorithm is performing with respect to different aspects such as overall accuracy, precision (ability to avoid false positive) recall (ability to capture true positive) and F1-score (a balance between precision and recall).

IV. RESULTS AND DISCUSSION

The total telemetry data was partitioned into training and testing objects—x_ {train} and y_ {train}—to develop a robust predictive model for device failure. The ensemble model utilizes a **Majority Voting** vector classifier rule, which classifies a device's status according to the most frequent class labels (e.g., *Normal, Soft Error, or ERI*) found in T(x). In instances where labels receive an equal number of votes, the system utilizes a random selection parameter among the tied labels, ensuring the fine-tuning process accounts for the probabilistic nature of electrical fluctuations. This approach, characteristic of **Bagging**, ensures that transient noise in lead impedance doesn't trigger a premature device replacement.

The final trained model objects were stored as kst. Upon evaluation, the model achieved a base accuracy of **82.9%**. This accuracy level was saved as the optimized version for **Pacemaker Soft Error and EOL Prediction**. The model was then uploaded to the validation datasets to generate real-time telemetry predictions and populate the final safety reports before being deployed into the device's monitoring firmware.

For the specific patient record analyzed, the ML ensemble algorithm focused on soft error mitigation by identifying electrical drifts before they became catastrophic. Unlike biological models where cholesterol dominates, this safety-critical analysis determined that **Battery Voltage (V)** was the most significant feature for predicting device expiration. This was closely followed by **Internal Resistance**, **Lead Impedance (Z)**, and **Capture Threshold**, as these features collectively define the operational "envelope" of the pacemaker.

4. Ensemble methods.

4.1. Averaging methods

4.1.1 Bagging Results

The Bagging Classifier acts as a stabilizing layer for the highly sensitive electrical features used to detect soft errors. By training an ensemble of base classifiers (such as Support Vector Machines or Decision Trees) on random subsets of the telemetry data, the model effectively averages out the "noise" that can occur in a patient's daily life. For instance, Lead Impedance can momentarily spike if a patient stretches or shifts their position, but because Bagging aggregates multiple predictions, the final output is less likely to overreact to these outliers. This reduction in variance is critical for preventing "False Positive" errors, ensuring that a simple postural change isn't misclassified as a lead fracture or insulation break.

The specific application of sampling with replacement allows the model to build several predictors that see different "versions" of the device's history. This is particularly useful for identifying the stable baseline of Class 0 (Normal Operation). The classification report, which shows high precision and recall for Class 0, confirms that the Bagging ensemble is excellent at recognizing the "plateau" phase of Battery Voltage and consistent Internal Resistance. However, the slightly lower accuracy for Class 1 (Soft Error/EOL) indicates that the model is more conservative when predicting failures. With an overall accuracy of 97.88%, the model demonstrates that while it is nearly perfect at confirming a healthy device, the transition into a "Soft Error" state represents a more complex, less frequent boundary that requires the aggregate "wisdom" of the ensemble to navigate safely.

Index	**0**	**1**	**accuracy**	**macro avg**	**weighted avg**
Precision	0.9852941176470589	0.8181818181818182	0.9787985865724381	0.9017379679144386	0.9776175809225072
Recall	0.9925925925925926	0.6923076923076923	0.9787985865724381	0.8424501424501425	0.9787985865724381
f1-score	0.988929889298893	0.7500000000000001	0.9787985865724381	0.8694649446494466	0.9779543113452336
support	270.0	13.0	0.9787985865724381	283.0	283.0

Table 2: Classification Performance Metrics for the Bagging Ensemble Model

4.2. Boosting algorithms methods.

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

Table 2. Performance Metrics for AdaBoost Ensemble Learning Model

For class 0, the precision is 1.0, indicating that all the instances predicted as class 0 are actually of class 0. For class 1, the precision is 0.9286, which means that there were some instances predicted as class 1 that were not actually of class 1. The recall is 0.9963 for class 0, meaning that almost all of the actual class 0 instances were correctly identified. The recall is 1.0 for class one, indicating that all the instances of class 1 were correctly identified. For class 0, the F1-score is 0.9981, and for class 1, it is 0.9630...Support indicates the number of occurrences of each class in the dataset. Macro Avg represents the macro-average, which is the average of precision, recall, and F1-score for each class. In this case, the macro-average values are around 0.964 for precision, 0.998 for recall, and 0.981 for the F1-score.

index	0	1	accuracy	macro avg	weighted avg
precision	0.996268656716418	0.8	0.9858657243816255	0.898134328358209	0.9872527820262643
recall	0.9888888888888889	0.9230769230769231	0.9858657243816255	0.955982905982906	0.9858657243816255
f1-score	0.9925650557620819	0.8571428571428571	0.9858657243816255	0.9248539564524695	0.9863442480516582
support	270.0	13.0	0.9858657243816255	283.0	283.0

Table 3. Gradient Boosting Results.

Figure: 4. Table for gradient Boosting Results

The provided metrics describe a model that is exceptionally reliable at confirming device health while remaining highly vigilant about potential malfunctions. Precision and recall in this clinical environment represent much more than abstract statistics; they define the critical balance between avoiding invasive, unnecessary surgeries and ensuring a patient's life-sustaining device remains fully operational. The "impressive precision" of 0.996 for Class 0 (Normal Operation) confirms that when the model identifies a pacemaker as healthy, it is correct 99.6% of the time. By mastering the "normal" electrical signatures of features like

Battery Voltage and Lead Impedance, the model ensures that patients are not plagued by false "lead failure" alarms triggered by minor, non-critical fluctuations in impedance, thereby preserving both patient quality of life and clinical resources.

For Class 1 (Soft Error/EOL), the precision of 0.8 and an F1-score of 0.857 demonstrate the model's adeptness at catching the subtle shifts that precede a catastrophic total failure. Identifying a "Soft Error"—such as a slight drift in the Capture Threshold or a rise in Internal Resistance—is significantly more complex than detecting a complete device shutdown. A recall of 0.923, combined with this 0.8 precision, indicates an aggressive diagnostic stance; the model is tuned to flag "borderline" devices for check-ups rather than risking a false negative where a battery reaches its "voltage knee" undetected.

Furthermore, the weighted average of 0.987 highlights the overall quality and robustness of the model across the entire dataset. In a real-world clinical setting, where the vast majority of devices function perfectly, this strong generalization ability ensures the system does not become complacent. By maintaining a Macro Average of 0.898, the model demonstrates that it treats rare, high-stakes soft errors with nearly as much mathematical weight as common stable readings. This ensures that safety-critical features are always prioritized, providing a reliable diagnostic shield for patients reliant on cardiac pacing technology.

4.3 XG Boost 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

Table 4. For XG boosting algorithm results

These Gradient Boosting results represent a significant leap in diagnostic precision, particularly in the model's ability to navigate the "sharp" transitions in electrical telemetry. The precision of 1.0 for Class 0 (Normal Operation) indicates that the model has perfectly mastered the identification of stable device signatures—such as a constant 3.0V **Battery Voltage** and nominal **Lead Impedance**—eliminating the risk of false alarms. More impressively, the 0.929 precision and perfect 1.0 recall for Class 1 (Soft Error/EOL) signify that the Boosting algorithm is capturing 100% of all critical failure events. This is vital for safety-critical monitoring; it means the model is successfully identifying every instance where **Internal Resistance** begins to climb or the battery hits its "voltage knee," ensuring no patient is left at risk due to a missed diagnostic signal.

The exceptional F1-scores of 0.998 and 0.963 highlight Gradient Boosting’s superior ability to reduce bias by iteratively correcting small errors in the prediction of "borderline" cases. Unlike simpler models, Boosting excels at detecting the subtle, non-linear drifts in Threshold or ▲Impedance Capture that often precede a device malfunction. With an overall accuracy of 99.6% and a weighted average of 0.997, the model demonstrates a sophisticated comprehension of the underlying physical laws governing pacemaker hardware. This robust generalization ability establishes it as a dependable clinical tool, capable of providing a "zero-miss" safety net for patients while maintaining the high precision necessary to avoid unnecessary surgical replacements.

V. CONCLUSION

Critical systems such as the pacemaker have faced significant challenges in maintaining expected performance due to the occurrence of soft errors, such as transient sensing failures or intermittent non-capture. While traditional methodologies have attempted to address these malfunctions, they often fail to meet optimal safety expectations; however, the implementation of ensemble algorithms to predict these errors—using features like **Lead** Impedance, Battery Voltage, and Capture Thresholds—has substantially reduced the frequency of unpredictable failures. This study presents an ensemble approach comprised of two distinct strategies: the Averaging method, utilizing Random Forest, Bagging, and Majority Voting to stabilize noisy telemetry; and the Boosting method, including AdaBoost, Gradient Boosting, and XGBoost to precisely map the non-linear degradation of Internal Resistance. The research demonstrates that boosting algorithms provide the highest prediction accuracy, while the ensemble framework ensures that models complement one another, covering individual weaknesses in detecting subtle shifts in ▲ Impedance.

Early prediction of soft errors enhances patient safety by allowing healthcare providers to take proactive measures before a battery reaches its "voltage knee" or a lead exceeds the 2,000 **Ω** fracture threshold. By facilitating timely interventions and scheduled preventive maintenance, these models reduce the likelihood of sudden device failures, minimizing emergency situations for pacing-dependent patients and contributing to substantial healthcare cost savings. Predicting soft errors effectively extends the functional lifespan of the device and fosters the reliability crucial for maintaining cardiac function and improving patient quality of life. Furthermore, this data-driven approach ensures compliance with stringent regulatory standards while providing research insights into how high Pacing Percentages accelerate energy depletion.

Despite the success of the models, this research faced significant challenges, specifically in sourcing appropriate datasets that capture the precise electrical telemetry required for soft error prediction. Privacy concerns and the proprietary nature of medical device logs limit access to granular pacemaker data, particularly longitudinal records of Internal Resistance and high-resolution error logs. These data gaps hinder the development of even more accurate predictive tools, as the scarcity of "Class 1" (Error/EOL) samples remains a primary obstacle in training models to navigate the ambiguous boundaries between normal device wear and an active soft error.

5.1. Recommendations

This study can be improve by implementing some of the best performing machine learning algorithms used in this study in another soft critical systems for predicting the soft errors and therefore evaluate the performance of those applied models. Developing new ML algorithms for even better accuracy and efficiency, implementing the algorithm in real-world applications.

Conflicts of Interest

This work has no any kind of conflict of interest

Funding Statement.

This work is funded by pan African university of science technology and innovation (PAUSTI)

REFERENCES

[1] M. P. V. S. Z. W. Sina Mohseni, "Practical Solutions for Machine Learning Safety in Autonomous Vehicles," arxiv , 2019.

[2] Knight.J.C, "Safety critical systems: challenges and directions," in Proceedings of the 24th international conference on software engineering, 2002.

[3] L. chen, "soft error analysis and mitigation at high leve abstraction," KIT, 2015.

[4] M. Al-khresheh, "A review study of error analysis theory.," International Journal of Humanities and Social Science Research, vol. 2, no. 1, pp. 49-59, 2016.

[5] F. a. Raghuwanshi.M, "Perfromance of machine learning Techniques in the prevention of financial fraud," vol. 1, p. 9, 2021.

[6] M. Uddin, "soft Error Tolerant systems," Lambert Academic publishing, pp. 1-63, 2013.

[7] R. P. &. K. R., "An overview of soft errors in digital systems, sources and effects, mitigation techniques.," 2017.

[8] Z. J. H. Y. &. W. X. Wang. C, "soft error analysis and mitigation for safety-critical systems based on data analysis," IEEE transactions on reliability, vol. 68, no. 1, pp. 246-258, 2019.

[9] C. Y. H. &. W. C. Lee C. H, "Ahardware-based soft error mitigation technique for safety -critical systems," IEEE Transactions on nuclear science, vol. 63, no. 6, pp. 3346-3353, 2016.

[10] E. G. O. D. O. J. S. B. &. D. A. B. Dada, "Ensemble machine learning model for software defect prediction," Adv. Mach. Learn. Artif. Intell, vol. 2, pp. 11-21, 2021.

[11] Z. Z. &. L. K, "A software-based soft error mitigation method for real-time operating system in safety-critical systems.," Microprocessors and Microsystems, vol. 69, pp. 102-978, 2019.

[12] S. A. Isil O¨ z, " Prediction of soft error vulnerability of parallel application using machine learning algorithm," International Journal of Parallel Programming, vol. 5, no. 49, pp. 410-439, 2020.

[13] I. B. .. https://doi.org/10.21203/rs.3.rs-2089478/v1., "Software default prediction using Deep Learning Techniques," research square, vol. 1, no. 10, p. 21203, 2022.

[14] R. K. Ammar Mohammed, " A comprehensive review on ensemble deep learning: Opportunities and challenges.," Journal of King Saud University-Computer and Information Sciences, vol. 35, no. 2, pp. 757-774, 2023.

[15] M. I. D. &. S. Y, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," IEEE, vol. 10, pp. 99129-99149, 2022.

[16] A. M. M. S. M. Asim Shahid M, "Improved accuracy and less fault prediction errors via modified sequential minimal optimization algorithm," PLoS One, vol. 4, p. 18, 2023.

[17] O. &. A. M. Alqasem, "software fault prediction using Deep Learning Algorithm," International Journal of Open Source Software and Processes, vol. 10, no. 10, pp. 1-19, 2019.

[18] S. A. A. &. C.-H. L. &. A. Das, "Analysis of error-based machine learning algorithms in network anomaly," Researchgate, vol. 10, 2021.

[19] M. K. R. &. S. H. Asghari, "A new efficient hardware-based soft error mitigation technique for saftycritical systems," journal of electronic testing, vol. 34, no. 3, pp. 313-324, 2018.

[20] A. J. &. F. Almeida. P, "using machine learning for soft error detection and correction in safety-critcal systems," IEEE Transaction on nuclear science, vol. 66, no. 6, pp. 1298-1305, 2019.

[21] M. M. A. ,. M. M. S. Muhammad Asim Shahid, "Improved accuracy and less fault prediction errors via modified sequential minimal optimization algorithm," PLoS ONE, vol. 10.1371, pp. 1-63, April 13,2023.

[22] A. R. V. K. P. A. Neha Sharma, "Machine Learning Implementation in Electronic," International Journal of Soft Computing and Engineering (IJSCE), vol. 10, pp. 2231-2307, 2021.

[23] X. &. S. C. &. H. C. &. N. Z. &. L. Z. Gao, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," IEEE Access, vol. 1, p. 7.1, 2019.

[24] P. A. D. Amiri, "An Ensemble-Based Machine Learning Model for," IEEE, vol. 11, p. 10.1109, 2023.

[25] N. &. R. A. &. K. V. &. A. P. Sharma, "Machine Learning Implementation in Electronic Commerce for Churn Prediction of End User.," International Journal of Soft Computing and Engineering, vol. 10, no. 10, pp. 20-25, 2021.

[26] A. M. Mattias Liljeson, " software Defect prediction using machine learning on lest and source code defect.," semantic scholar, vol. 6, 2014.

[27] T. J. A. B. S. &. P. K. Sharma, "Ensemble Machine Learning Paradigms in Software Defect Prediction," Procedia Computer Science, vol. 218, pp. 199-209, 2023.

[28] ". S. o. E. L. C. A. A. a. P. i. I. A. v. 1. p. 9.-9. 2. d. 1. I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," IEEE access, vol. 10, pp. 99129-99149, 2022.

[29] A.-A. Alhumam, "Effective Prediction of Software Defects using Random-tree Entropy based Feature Selection Framework," International Journal of Advanced Computer Science and Applications, vol. 15, no. 5, 2022.

[30] K. Liu, "Research on E-commerce Precision Marketing Model Based on Big Data Technology," in International Conference on Big Data Economy and Information Management (BDEIM) (pp. 213-216). IEEE. [1], sayna,China, 2021.