Machine Failure Detection in Industries Using Long-Short Term Memory and Light Gradient Boosting Machine

S. Sai Nikith[1], S. Ram Charan[2], Shaik. Sher Ahamed[3], D. Tejovanth[4], S. Ramadoss[5]
[1,2,3,4] *Student/Dept of CSE, School of Computing, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu*
[5]*Asst. Professor /Dept of CSE, School of Computing, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu*

Abstract: **By analyzing real-time sensor data such as temperature, rotational speed, torque, and tool wear, our model predicts potential failures before they occur, helping reduce downtime and maintenance costs. Machine failures in automated industries result in downtime, reduced productivity, and increased costs, creating a demand for effective predictive maintenance solutions. The framework combines Long Short- Term Memory (LSTM) networks for capturing time-based patterns and LightGBM for identifying critical features to build a robust predictive maintenance system.**
The system analyzes sensor data in real-time to predict machine failures, assess their severity, and trigger alarms based on the level of urgency. The hybrid model enhances predictive accuracy, reduces false alarms, optimizes maintenance schedules, and ensures minimal downtime for seamless industrial operations. In factories and industries, machines sometimes break down unexpectedly. This leads to: Downtime (machines stop working), Lower productivity (less work gets done), Higher costs (repairs and losses in production). To prevent these failures, industries need a better way to predict when a machine about to fail.

Keywords:- **Predictive Maintenance, Real-Time sensor data, Machine failure prediction, LSTM, LightGBM, Maintenance prediction, Sensor data analysis, Time-based patterns, Alarm triggers, Predictive Accuracy**

I. INTRODUCTION

Machine failure detection is important to prevent unexpected breakdowns and reduce maintenance costs. LightGBM (Light Gradient Boosting Machine) and LSTM (Long Short-Term Memory) are two machine learning techniques that help in predicting failures before they happen. LightGBM is a fast and efficient algorithm that finds patterns in large amounts of data. It works well with structured data, like sensor readings from machines. LSTM is a type of neural network that remembers past data, making it useful for analyzing time-based information, such as temperature or vibration changes over time. By combining these two methods, we can build a system that learns from past machine behavior and predicts failures in advance. This helps industries take action before problems occur, improving efficiency and safety.

Two powerful machine learning techniques, LightGBM (Light Gradient Boosting Machine) and LSTM (Long Short-Term Memory), are particularly effective in this domain. LightGBM is a fast and efficient algorithm known for its ability to identify patterns in large datasets, making it ideal for processing structured data such as sensor readings from machines. On the other hand, LSTM is a type of neural network that excels in analyzing time-based data, allowing it to capture historical patterns such as changes in temperature or vibration over time.

- LightGBM is used to identify the most critical features that influence machine performance, allowing the system to focus on the most important data points for accurate predictions.
- LSTM is used for recording precious data and based on that data it predicts the outcomes.
- By combining these methods, the predictive maintenance system offers several key benefits.

The remainder of this paper is organized as follows: Section I Introduction Section II Related work Section III provides a background on Long-Short term memory and Light Gradient Boosting Machine, Section IV discusses data preprocessing and model optimization techniques, Section V Result and Discussion, and Section VI Conclusion.

II. RELATED WORK

As software complexity increases, predicting runtime failures before deployment becomes challenging. Machine learning (ML) models, particularly heterogeneous ensembles combining different algorithms, have shown promise in improving online failure prediction (OFP). This study explores the effectiveness of various ML techniques and combination methods, demonstrating that synergies between learners can enhance prediction accuracy, even when individual models may not be the best by [1]. This study explores the use of deep learning, specifically Multi-Layer Perceptron (MLP), for failure prediction in predictive maintenance. Trained on AI4I 2020 data, the models demonstrate high accuracy and outperform SVM methods by capturing complex patterns and temporal dependencies. However, challenges related to data quality, model interpretability, and optimization remain for further enhancement by [2]. This paper compares machine learning and deep learning algorithms for predicting machine failures, finding XGBoost and Long Short-Term Memory (LSTM) models most effective. The results highlight their potential in enhancing predictive maintenance and optimizing industrial operations by [3]. This paper reviews acoustic methods for mechanical failure detection in industrial machines, highlighting the dominance of acoustic emission and the challenges of detecting failures in noisy conditions. Despite progress, research on failure detection in noisy environments remains limited by [4]. It an addresses predicting electrical machine failures using time series analysis of vibration data, employing a hybrid CNN-LSTM model with quantile regression to handle uncertainties.

The approach outperforms traditional models, optimizing maintenance schedules and improving machine performance by [5]. This systematic literature review examines the use of system logs for anomaly detection and failure prediction in IT infrastructure, highlighting machine learning and deep learning approaches' superior performance over traditional methods. The study identifies research gaps and provides future directions to mitigate downtime in IT systems by [6]. This is a proposes a multi-label ensemble LSTM-Random Forest method with a GRU-based denoising autoencoder for simultaneous fault detection in Automotive Software Systems under noisy conditions. The approach achieves 99.43% detection accuracy and outperforms state-of-the-art models with a 91.2% F1-score for simultaneous faults by [7]. This evaluates classification algorithms (Naive Bayes, kNN, and ANN) for machine fault detection, showing that Naive Bayes and ANN achieve 99.9% accuracy with high AUC values, while KNN performs slightly lower. Feature selection analysis highlights key features (HDF, OSF, and PWF) that improve classification performance, contributing to more reliable fault detection systems by [8]. This article presents a LightGBM-based fault diagnosis framework for induction motors using iterative feature selection and LOLO-CV, achieving 98.55%-100% accuracy. The method maintains high performance under unseen conditions, with Bayesian optimization improving results further by [9]. In recent studies using the CWRU bearing dataset for machinery fault detection and diagnosis through deep learning algorithms. It summarizes the algorithms, results, and details to assist future research in this field by [10]. This presents an IoT machine learning and orchestration framework for real-time failure detection of surface mount devices during production. It is evaluated through a simulation of a production line, assessing software architecture, scalability, model accuracy, and production performance by [11].

The remainder of this paper is organized as follows: Section II Related work Section III provides a background on Long-Short term memory and Light Gradient Boosting Machine, Section IV discusses data preprocessing and model optimization techniques, Section V Result and Discussion, and Section VI Conclusion.

III. PROVIDES A BACKGROUND ON LONG-SHORT TERM MEMORY AND LIGHT GRADIENT BOOSTING MACHINE

1. LONG-SHORT TERM MEMORY:

Long Short-Term Memory (LSTM) is a specialized type of recurrent neural network (RNN) designed to handle the limitations of traditional RNNs, particularly their inability to learn long-term dependencies. In a standard RNN, as data is passed through the network over multiple time steps, the model struggles to retain information from earlier steps due to issues like vanishing gradients. LSTM overcomes this by introducing a more complex architecture with mechanisms, or "gates," that control the flow of information. These gates—input, forget, and output gates—regulate how much information should be remembered or forgotten at each timestep, allowing LSTMs to maintain long-term memory while processing sequential data. The core advantage of LSTM networks lies in the ability to remember relevant information over long sequences, making them particularly useful in tasks where context is spread across many steps. For example, in natural language processing (NLP), understanding a word's meaning often depends on the words that came before it. An LSTM can effectively capture and retain such context over longer sentences or paragraphs. Unlike traditional RNNs, which suffer from the vanishing gradient problem, LSTMs use a cell state that acts as a memory unit, with the ability to add, forget, or modify information as necessary. This allows them to learn dependencies that might span hundreds or even thousands of time steps. LSTMs have become the standard model for many sequence-based tasks, such as machine translation, speech recognition, and time series forecasting. In machine translation, for example, an LSTM can translate a sentence from one language to another by effectively keeping track of the sequence of words in the source language while generating the target language sentence. In time-series forecasting, LSTMs can predict future values by learning from the temporal dependencies in the data. Their ability to maintain relevant information while discarding irrelevant details makes them highly effective for tasks that involve complex, sequential patterns.

LSTM networks are increasingly used in machine failure detection by analyzing time-series data from sensors that monitor various machine parameters like temperature, vibration, and pressure. The sequential nature of LSTMs makes them ideal for identifying patterns and anomalies in the data that could indicate an impending failure. By training an LSTM model on historical data, it can learn the normal operating conditions of the machine and detect deviations that may signal wear, malfunction, or failure. The ability of LSTMs to capture long-term dependencies allows them to recognize subtle changes over time, making them effective for early detection of failures, thereby reducing downtime and preventing costly repairs or replacements.

2. Light Gradient Boosting Machine:

LightGBM (Light Gradient Boosting Machine) is a highly efficient, scalable, and fast implementation of gradient boosting, a popular machine learning technique. Developed by Microsoft, LightGBM is designed to handle large datasets with high efficiency while providing state-of-the-art performance in terms of both speed and accuracy. It is primarily used for supervised learning tasks, such as classification and regression, and has gained popularity due to its ability to handle large-scale data with less memory usage and faster training times compared to traditional gradient boosting methods like XGBoost. One of the key features of LightGBM is its leaf-wise growth strategy, which contrasts with the level-wise growth used by other gradient boosting frameworks. Instead of splitting trees level by level, LightGBM grows trees by selecting the leaf with the maximum delta in the loss function. This approach can lead to more complex trees that are able to better capture intricate patterns in the data, resulting in higher accuracy with fewer iterations. LightGBM also uses histogram-based techniques to speed up training by converting continuous feature values into discrete bins, reducing memory usage and improving computational efficiency. LightGBM supports a variety of advanced features, such as handling categorical features directly without the need for one-hot encoding, which saves time and memory. It also supports parallel and GPU learning, making it highly scalable for large datasets and enabling faster model training on modern hardware. Additionally, it offers built-in support for handling missing values, robust regularization methods, and early stopping to prevent overfitting. Once trained, the LightGBM model can predict the likelihood of a machine failure at any given time by analyzing incoming sensor data, allowing for early detection of potential issues. This enables predictive maintenance strategies, reducing unplanned downtime, preventing catastrophic failures, and optimizing maintenance schedules. The model's speed and scalability further enhance its utility in industrial settings, where large amounts of data need to be processed in real-time.

LightGBM is highly effective in machine failure detection by analyzing sensor data such as temperature, vibration, and pressure from machines. By training on historical data with labeled instances of both normal operation and failure events, LightGBM can learn to identify patterns that signal potential machine failures. Its ability to efficiently handle large, high-dimensional datasets and deal with missing or categorical features makes it ideal for real-time monitoring systems in industrial settings. Once trained, the model can predict failures early, enabling predictive maintenance, minimizing downtime, and optimizing repair schedules, ultimately preventing costly breakdowns and improving operational efficiency.

IV. PROPOSED FRAMEWORK FOR MACHINE FAILURE DETECTION USING LONG-SHORT TERM MEMORY AND LIGHTGBM

1. Data Preprocessing:

This stage involves cleaning and preparing the raw data for analysis. Data preprocessing in machine failure detection is a crucial step in transforming raw sensor data into a format suitable for training machine learning models. The first step is data cleaning, which involves handling missing or corrupted data commonly found in sensor readings. Techniques like imputation, interpolation, or even removing rows with missing values help ensure that the dataset remains complete and usable. In addition, noise reduction is important to eliminate measurement errors and environmental factors that may distort the data. This can be achieved through methods like smoothing or outlier detection, which helps reveal true patterns in the sensor data. It includes tasks like:

- Handling missing values: Imputing or removing missing data points.
- Data transformation: Scaling, normalizing, or encoding categorical variables.
- Outlier detection and treatment: Identifying and dealing with extreme values.

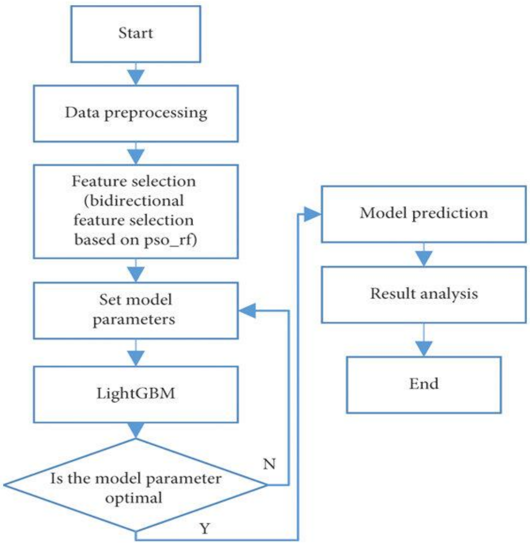

Figure 1. Architecture for Machine Failure Prediction

2. Feature Selection:

The process of selecting the most relevant features from the dataset to improve model performance and reduce complexity. This stage focuses on identifying the most relevant features from the dataset. It employs a sophisticated technique, likely combining Particle Swarm Optimization (PSO) and Random Forest (RF), to optimize feature selection.

Feature Selection contains:

- Bidirectional Feature Selection: A technique that combines forward selection (adding features incrementally) and backward elimination (removing features incrementally) to find the optimal subset.
- pso_rf: This likely refers to Particle Swarm Optimization (PSO) combined with Random Forest (RF). PSO is an optimization algorithm used to guide the feature selection process, while RF is used to evaluate the performance of different feature subsets.

3. Set Model Parameters:

This step involves configuring the parameters of the LightGBM model. LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework known for its efficiency and accuracy. The model's parameters which are learned during training, are set before the learning process begins and dictate how the model learns from the data. For LightGBM, this involves fine-tuning settings such as the number of boosting iterations, the learning rate, the maximum tree depth, and regularization parameters, among others. Finding the optimal combination of these hyperparameters is essential for maximizing the model's predictive accuracy and preventing issues like overfitting. The flowchart indicates this is an iterative process, where different parameter combinations are tested, and the model's performance is evaluated to determine if the parameters are optimal. This iterative tuning, often involving techniques like grid search, random search, or Bayesian optimization, continues until the model achieves satisfactory performance, as determined by the "Is the model parameter optimal?" decision point.

4. LightGBM:

The LightGBM (Light Gradient Boosting Machine) algorithm is applied to build the prediction model. This is a powerful and efficient gradient boosting framework known for its speed and accuracy.As a gradient boosting algorithm, LightGBM constructs an ensemble of decision trees sequentially, with each tree learning to correct the errors of its predecessors. Its key strengths lie in its speed and accuracy, particularly when dealing with large datasets, achieved through techniques like gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB). This framework's ability to handle categorical features directly is another significant advantage, simplifying the preprocessing of many real-world datasets. However, like most machine learning models, LightGBM's performance is highly dependent on the proper tuning of its hyperparameters, as illustrated by the iterative loop in the flowchart involving "Set Model Parameters" and the "Is the model parameter optimal?" decision point. This emphasizes the importance of carefully configuring the model to achieve the best possible predictive performance.

5. Model Prediction:

This stage involves using the trained LightGBM model (with the selected features and optimized parameters) to make predictions on new or unseen data. This is a decision point. If the parameters are deemed optimal, the process moves to the right side (model prediction and analysis). Otherwise, it loops back to "Set Model Parameters" for further tuning.

In the flowchart, 1 = optimal/ non-failure, 0 = non-optimal/failure.

In a machine state scenario, 1 = machine optimal, 0 = machine failure.

- 1 (optimal): This output signifies that, based on the evaluation criteria used, the current set of model parameters is deemed optimal. The model's performance is considered satisfactory, and the workflow progresses to the subsequent steps: model prediction and result analysis. Effectively, "1" indicates that the tuning process has achieved its goal.

- 0 (Failure): This output indicates that the model parameters are not yet considered optimal. The model's performance, as measured by this evaluation metrics, falls short of the desired level. Consequently, the workflow returns to the "Set Model Parameters" stage, triggering another iteration of parameter tuning to improve the model's performance.

V. RESULT AND DISCUSSION

By this it Implements a predictive maintenance system using LSTM and LightGBM enhances machine reliability and operational efficiency in automated industries. By analyzing real-time sensor data, the system identifies failures, and issues timely alerts. This minimizes downtime, optimizes maintenance schedules, and reduces costs, ensuring seamless industrial operations.

- Advanced AI and Deep Learning models: Enhancing LSTM networks with the transform based models for better anomaly detection.
- Automated Maintenance Execution: Integrating AI-droven robotic systems for autonomous repairs. Using predictive insights to automate part replacements before failures occur.

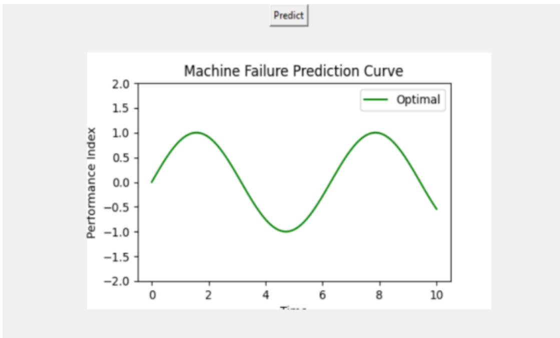

Figure 2. optimal

The machine's performance is currently at an optimal level, as indicated by the highest point on the "Optimal" curve in the figure 2. This suggests that the machine is operating at its peak efficiency and is not at immediate risk of failure. However, it is important to monitor the performance index over time to identify any emerging trends or anomalies that could indicate a potential decline in performance.

- Predictive Maintenance: The "Predict" button suggests that the system is designed for predictive maintenance. This means it aims to forecast future failures based on historical and current data, allowing for proactive interventions before breakdowns occur.
- Cyclical Pattern: The sinusoidal nature of the curve indicates a recurring pattern in the machine's performance. This could be due to factors like regular maintenance cycles, temperature fluctuations, or usage patterns. Understanding this pattern is crucial for accurate predictions.
- Thresholds and Alarms: In a real-world application, the system would likely have defined thresholds for the performance index. When the index falls below a certain threshold, it could trigger alarms or alerts, indicating a

potential failure risk.

- Data-Driven Decision Making: The graph provides valuable data for decision-making regarding maintenance schedules, component replacements, and operational adjustments. By analyzing the trends and patterns, operators can optimize the machine's performance and minimize downtime.

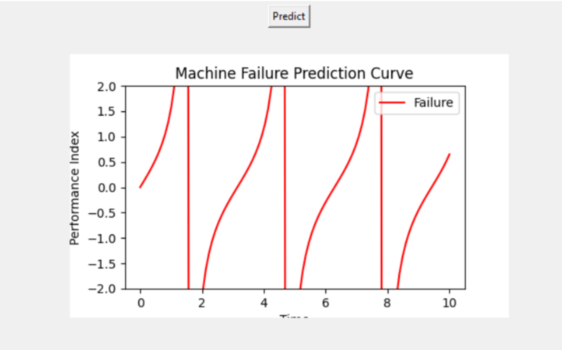

Figure 3. Non-optimal

The machine is found to be non-optimal, as evidenced by the red "Failure" curve exhibiting sharp, near-vertical fluctuations in the Performance Index across a wide range, indicating instability and a predicted failure state as determined by the "Predict" function in figure 3.

- Red "Failure" Curve: The colour and label of the curve are strong indicators of a problem.
- Sharp Vertical Lines: These represent rapid and extreme changes in the machine's performance, suggesting instability.
- Wide Performance Index Range: The fluctuations across the entire range (-2.0 to 2.0) show in significant deviations from normal operation.

To combat downtime, reduced productivity, and increased costs caused by unexpected machine failures in automated industries, a predictive maintenance system utilizing a hybrid model is proposed. This system analyzes real-time sensor data, such as temperature, rotational speed, torque, and tool wear, using Long Short-Term Memory (LSTM) networks to capture time-based patterns and LightGBM to identify critical features. By predicting potential failures, assessing their severity, and triggering alarms based on urgency, the system aims to enhance predictive accuracy, reduce false alarms, optimize maintenance schedules, and ensure minimal downtime for seamless industrial operations, ultimately resulting in cost savings and improved efficiency.

Comparison between traditional method (manual method) and our proposed system:

The manual method for machine failure detection in automated industries primarily rely on reactive or predefined threshold-based methods. These systems often depend on manual monitoring or fixed schedules for maintenance, which are insufficient for handling the complexities of the modern industrial machinery. Reactive approaches address failures only after they occur, leading to unplanned downtime and increased costs, while threshold-based methods lack the adaptability to accurately detect potential failures in dynamic environments. This highlights the limitations of traditional systems in providing timely and precise failure predictions, making them inadequate for optimizing industrial operations.

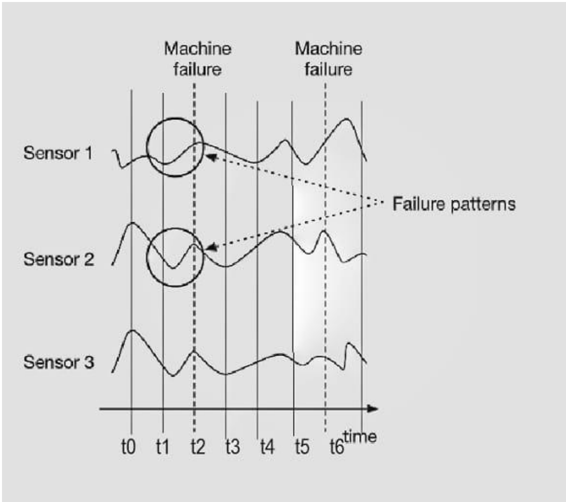

Figure 4. Traditional method

Traditional Methods:

Reliance: Often rely on fixed, rule-based systems or simple threshold checks.

Data: Primarily use basic sensor readings or manual inspections.

Analysis: Limited to static analysis, not considering temporal patterns.

Detection: Focus on immediate, obvious failures.

Limitations: High rates of false alarms (reacting to normal fluctuations). Difficulty predicting subtle, gradual failures. Inability to adapt to changing machine behaviours. Inefficient maintenance schedules (either too frequent or too late).

Cost: higher cost due to more down time, and possibly more wasted parts.

LSTM-Light GBM Methods:

Reliance: Uses a hybrid machine learning approach. LSTM for capturing time-series dependencies in sensor data (e.g., how temperature changes over time). LightGBM for identifying the most critical features that contribute to failures.

Data: Processes large volumes of real-time sensor data, including temperature, vibration, pressure, etc.

Analysis: Employs sophisticated algorithms to detect complex patterns and predict future failures.

Detection: Can predict both sudden and gradual failures, even those with subtle precursors.

VI. CONCLUSION

The proposed system introduces a Machine Failure Detection In Industries Using LSTM and LightGBM a predictive maintenance system using LSTM and LightGBM enhances machine reliability and operational efficiency in automated industries. By analyzing real-time sensor data, the system identifies failures, and issues timely alerts. This minimizes downtime, optimizes maintenance schedules, and reduces costs, ensuring seamless industrial operations.

In conclusion, the integration of advanced predictive maintenance solutions, leveraging both Long Short-Term Memory (LSTM) networks and LightGBM, offers a transformative approach to managing machine reliability in automated industries. By analyzing real-time sensor data, such as temperature, rotational speed, torque, and tool wear, this hybrid model can accurately predict machine failures before they occur. This predictive capability enables proactive maintenance, reducing unexpected downtime, minimizing costly repairs, and ensuring continuous production flow. The combination of LSTM's ability to capture time-based patterns and LightGBM's feature selection strengthens the system's precision, making it an invaluable tool for enhancing operational efficiency in industrial settings.

Ultimately, adopting such predictive maintenance frameworks leads to significant cost savings and improved productivity. By assessing potential failures and their severity in advance, industries can optimize maintenance schedules, reduce unnecessary interventions, and prioritize urgent repairs, thereby ensuring that resources are utilized more effectively. This technology not only streamlines operations but also contributes to the overall sustainability of industrial systems, allowing companies to maintain high levels of efficiency and reduce the risk of unforeseen disruptions.

VI. REFERENCES

[1] *B. Dhanalaxmi,* G. A. Naidu, and K. Anuradha, ''A review on software fault detection and prevention mechanism in software development activities,'' *IOSR J. Comput. Eng., vol. 17, no. 6, pp. 661–2278, 2015.*

[2] C. B. R. Ed Targett. *Amazon Outage*: *Estimated $99 Million Lost.* Accessed: Oct. 10, 2019. [Online]. Available: https://www.cbronline.com/ news/amazon-outage-lost-sales

[3] B. C. Baraniuk. *Gps Error Caused '12 Hours of Problems' for Companies.* Accessed: Oct. 10, 2019. [Online]. Available: https://www.bbc.com/ news/technology-35491962

[4] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, ''Basic concepts and taxonomy of dependable and secure computing,'' *IEEE Trans. Depend. Sec. Comput., vol. 1, no. 1, pp. 11–33, Jan./Mar. 2004.*

[5] F. Salfner, M. Lenk, and M. Malek, ''A survey of online failure prediction methods*,'' ACM Comput. Surv., vol. 42, no. 3, pp. 1–42, 2010.*

[6] R. Polikar, ''Ensemble based systems in decision making,'' *IEEE Circuits Syst. Mag., vol. 6, no. 3, pp. 21–45, Sep. 2006.*

[7] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms.* Boca Raton, FL, USA: CRC Press, 2012.

[8] J. R. Campos, M. Vieira, and E. Costa, ''Exploratory study of machine learning techniques for supporting failure prediction,'' in *Proc. 14th Eur. Dependable Comput. Conf. (EDCC), Sep. 2018, pp. 9–16.*

[9] J. R. Campos, M. Vieira, and E. Costa, ''Propheticus: Machine learning framework for the development of predictive models for reliable and secure software,'' *in Proc. IEEE 30th Int. Symp. Softw. Rel. Eng. (ISSRE), Oct. 2019, pp. 173–182.*

[10] [F. Salfner and M. Malek, ''Proactive fault handling for system availability enhancement,'' in Proc. *19th IEEE Int. Parallel Distrib. Process. Symp., Apr. 2005, pp. 1–7. [12] F. Salfner and M. Malek*