PREDICTIVE POWER MACHINE LEARNING MODELS FOR MACHINE FAILURE STATUS

Dr. Kavitha Subramani [1],Sophana Jennifer S[2], Abinaya S[3],Abinisheka S S[4], Darshini A[5]

[1]*Professor,*[2]*Faculty, Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India*

[3,4,5]*UG Scholar, Department of Computer Science and Engineering, Panimalar Engineering College, Chennai,India*

Abstract— **One of the most pivotal aspects of predictive maintenance is forecasting the failure of a machine, thereby minimizing downtime and improving operational agility. This project's goal is to build and deploy machine learning models that are capable of predicting failure status in industrial machinery. It has a complete workflow from data preprocessing,visualization, to model implementation. Such a dataset is clean and ready to be used as the quality and reliability of input features have already been ensured using data preprocessing techniques. Then, data visualization tools are leveraged to examine and understand patterns in the data that lead to decisions on what type of machine learning algorithm to use and how to optimize the framework. The last models, which are trained for the prediction of a machine's failure status, are bound with a Django framework that provides the user a friendly-interface of predictions and monitoring. The main aim of this integrated system is to facilitate the maintenance strategies and to enhance the predictive accuracy.**

Keywords—Predictive Maintenance, Machine Learning Models, Data Pre-processing, Data Visualization, Failure, Prediction, Django Framework, Accuracy.

I. INTRODUCTION

1.1 OVERVIEW

In Industrial environments, unexpected equipment breakdowns can lead to major disruptions, financial setbacks, and reduced operational effectiveness. Predictive maintenance has surfaced as a forward-thinking approach to tackle these issues by utilizing machine learning (ML) models to foresee failures ahead of time. The aim of creating predictive power machine learning models is to boost equipment reliability, lower maintenance expenses, and enhance operational efficiency.Machine learning methodologies like Supervised Learning, Anomaly Detection, and Time-Series Forecasting make use of past data and current sensor data to identify early indicators of possible failures. In contrast, predictive maintenance streamlines the detection of failures and refines maintenance scheduling through insights driven by AI.This research introduces a machine learning framework for forecasting machine failure status by incorporating data preprocessing, visualization, and model execution. It utilizes sophisticated ML algorithms, including Random Forest, Support Vector Machines (SVM), and Neural Networks, along with optimization methods and feature selection techniques to boost predictive precision.Furthermore, a web interface based on Django is developed to facilitate real-time monitoring and user-friendly interaction with the predictive models.

By embedding predictive power machine learning models, industries can shift from reactive to proactive maintenance approaches, thereby minimizing downtime, cutting costs, and enhancing operational performance. This study adds to the ongoing development in industrial AI and predictive analytics, providing a scalable and efficient solution for anticipating failures in industrial machinery.

1.2 PROBLEM DEFINITION

Machine failures are inevitable, with consequences ranging from minor, easily fixable issues to significant disruptions, depending on factors such as repair expenses, downtime, safety risks, and the impact on production and service delivery. Equipment failure often occurs due to common causes, and recognizing these along with implementing preventive measures is essential to minimizing the risks associated with unplanned downtime. Equipment operators play a crucial role in this process, as they receive extensive training on proper operation, basic troubleshooting, and safety protocols specific to the machinery they handle. However, situations may arise where an operator is required to use equipment they have not been adequately trained for. This issue often emerges due to workforce shortages or unexpected staff absences, which can lead to operational challenges.

II. LITERATURE REVIEW

Predictive maintenance is a crucial aspect of modern industrial operations, aimed at minimizing downtime and improving equipment reliability. Various machine learning techniques have been explored to enhance failure prediction accuracy and optimize maintenance schedules. This section provides a comprehensive review of significant contributions in the field, focusing on traditional machine learning, deep learning, and hybrid approaches.

In "Revisiting Confidence Estimation: Towards Reliable Failure Prediction" [1], the study investigated confidence estimation in deep learning models for failure prediction. It highlighted the challenges in detecting misclassifications and out-of-distribution (OOD) samples. The authors proposed a novel approach using flat minima optimization to enhance prediction reliability, bridging the gap between confidence calibration, OOD detection, and failure forecasting. Similarly, Mobley [2] discussed fundamental predictive maintenance principles and methodologies for reducing unplanned downtime.

Susto et al. [3] examined multiple classifier systems for predictive maintenance, integrating machine learning models with optimization techniques. Their findings highlighted the effectiveness of supervised learning approaches, such as support vector machines (SVM) and decision trees, in failure prediction. Additionally, Amruthnath and Gupta [4] explored unsupervised learning methods for anomaly detection, demonstrating the applicability of clustering techniques like k-means and isolation forests in industrial fault detection.

Advancements in deep learning have further enhanced predictive maintenance capabilities. Zhao et al. [5] introduced a Transformer-driven deep reinforcement learning framework that significantly improved proactive fault mitigation. Hamaide et al. [6] compared various machine learning approaches for predictive maintenance, showcasing the benefits of hybrid models that combine deep learning with traditional statistical techniques.

Pinciroli Vago et al. [7] investigated multivariate time series analysis for machine failure prediction in industrial settings. Their study emphasized the importance of temporal dependencies and demonstrated the efficacy of long short-term memory (LSTM) networks in predictive modeling. Chevtchenko et al. [8] developed an anomaly detection model for induction motors using real-time IoT data, further improving predictive maintenance accuracy.

Maheshwari et al. [9] performed a comparative analysis of classification models and LSTM networks, illustrating the advantages of deep learning in predictive maintenance. Bidollahkhani and Kunkel [10] examined the role of artificial intelligence in enhancing system reliability, demonstrating how AI-driven predictive maintenance strategies minimize operational risks and maintenance costs.

Beyond machine learning techniques, other studies have investigated the integration of AI with real-world applications. Huang and Du [11] optimized fire detection and recognition using virtual reality-based image processing, illustrating the versatility of AI techniques in safety-critical environments. Spang [12] developed an object tracking framework using local binary descriptors, which holds potential applications in predictive maintenance for industrial automation.

Advancements in hardware technology have also played a significant role in predictive maintenance. The Raspberry Pi 4 Model B [13] has been widely adopted for edge computing applications, enabling real-time monitoring and fault detection in industrial machinery. Similarly, Bimantara et al. [14] studied the implementation of predictive maintenance strategies in maritime industries, focusing on improving vessel reliability.

Herlambang and Nurpulaela [15] analyzed the use of fire alarm systems in large-scale infrastructures, highlighting the effectiveness of predictive analytics in failure prevention. Khabib et al. [16] evaluated thermal overload relay capabilities in marine control panels, underscoring the importance of predictive maintenance in ensuring operational stability.

Several industry-specific case studies have further reinforced the importance of predictive maintenance. "Predictive Maintenance Using Machine Learning: A Case Study in Manufacturing Management" [17] applied machine learning models to manufacturing systems, demonstrating a significant reduction in maintenance costs. "Machine Learning Approach for Predictive Maintenance in Industry 4.0" [18] explored the integration of AI and IoT in modern industrial operations.

Surveys on predictive maintenance trends, such as "A Survey on Predictive Maintenance Using AI and IoT Technologies" [19], have highlighted emerging methodologies and the challenges associated with AI-driven maintenance strategies. Lastly, "A Deep Learning Approach to Predictive Maintenance in Smart Manufacturing" [20] showcased the impact of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) on predictive accuracy in smart factory environments.

III. RESEARCH METHODOLOGY

3.1 Research Design

The research follows a quantitative approach, leveraging machine learning techniques to classify and predict system failures. The methodology comprises multiple stages, starting from data acquisition to model evaluation, ensuring a structured workflow. The approach includes:

- Data collection – Acquiring relevant historical data.

- Data preprocessing – Cleaning and refining data for accuracy.

- Model selection and training – Applying machine learning algorithms.

- Model evaluation– Measuring performance using statistical metrics.

- Prediction and analysis– Interpreting results to improve decision-making.

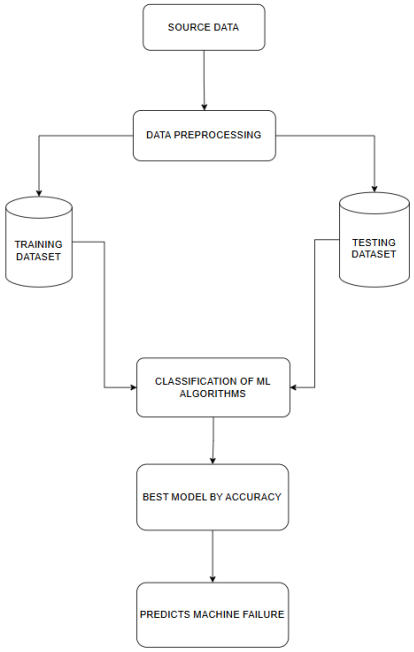

Fig.1.Workflow Diagram

3.2 Data Collection

The dataset used in this research consists of historical failure records obtained from relevant sources. The data is divided into two subsets:

- Training Set (80%) – Used for training machine learning models to learn patterns.

- Test Set (20%) – Used for evaluating the model's performance on unseen data.

The dataset includes multiple features related to system, such as types of failures that occur in a system, air temperature, process temperature, rotational speed. The reliability of the dataset plays a crucial role in building an accurate predictive model.

3.3 Data Preprocessing

Data preprocessing is a critical step in machine learning that ensures datasets are clean, structured, and suitable for model training. One of the primary aspects of this process is validation techniques, which help estimate the error rate of a model. If a dataset is large enough to accurately represent the population, validation techniques may not be necessary. However, in practical scenarios, data samples often fail to reflect the entire population, making validation crucial. Preprocessing involves detecting missing values, identifying duplicate records, and analyzing data types (such as float or integer). Properly cleaned data ensures that the model evaluation remains unbiased, particularly when tuning hyperparameters. A validation set is frequently used to assess model performance and refine configurations to improve

accuracy. Data collection, analysis, and preparation are time-intensive but essential for reliable machine learning models. Understanding data properties helps in selecting appropriate algorithms for predictive modeling. The Pandas library in Python is extensively used for data preprocessing, particularly for managing missing values. A structured approach to data cleaning minimizes the effort spent on preprocessing, allowing more focus on data exploration and model development.

3.3.1 Handling Missing Data

Missing data can arise due to various reasons, such as:

- Users neglecting to fill in required fields.
- Data loss during manual transfer from older systems.
- Programming errors causing incomplete records.
- Users deliberately omitting responses based on personal preferences.

3.3.2 Input & Expected Output

- Input**:** Raw dataset
- Output**:** Cleaned dataset with noisy data removed

3.4 Data Visualization

Data visualization is an essential technique in applied statistics and machine learning. While statistical analysis focuses on numerical measurements and estimations, visualization helps in deriving qualitative insights from data. It plays a vital role in exploring datasets, detecting trends, identifying anomalies, and spotting inconsistencies.Using domain knowledge, visual representations such as graphs, charts, and plots enable a more intuitive understanding of complex data relationships. These graphical tools make it easier for stakeholders to interpret key patterns compared to numerical summaries or statistical significance tests.The fields of data visualization and exploratory data analysis (EDA) are extensive, and a deeper understanding of these techniques can significantly enhance data interpretation. Advanced learning resources and specialized books provide further insights into these methods.Often, raw data may seem complex and difficult to interpret until it is represented visually. By applying various graphical techniques, one can better understand the structure and distribution of data, making visualization a key skill in both data science and machine learning.This process involves the use of diverse plotting techniques in Python, such as histograms, scatter plots, bar charts, and line graphs, to transform raw data into meaningful insights.

3.4.1 Input and Expected Output

- Input**:** Raw Data
- Output: Graphically Represented (Visualized) Data

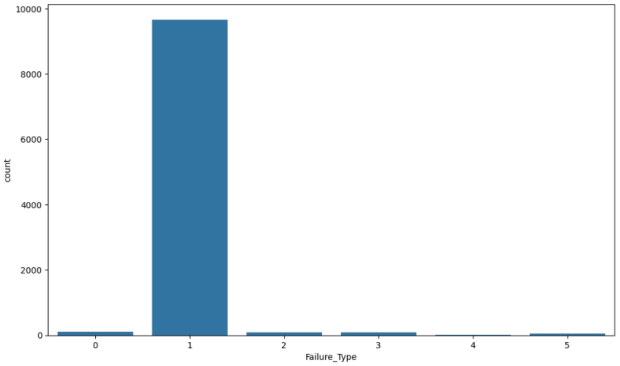

Fig.2. Count of Failure type in dataset

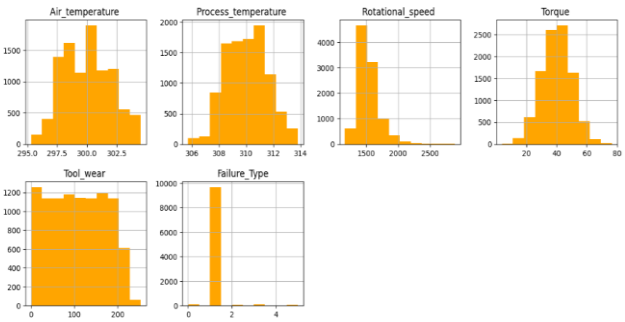

Fig.3. Histogram

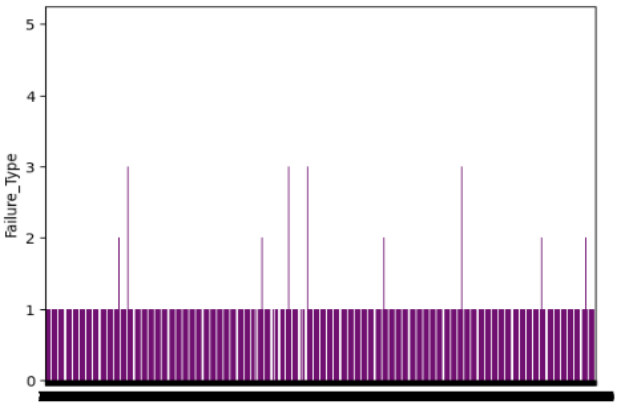

Fig.4. Barplot of Failure type

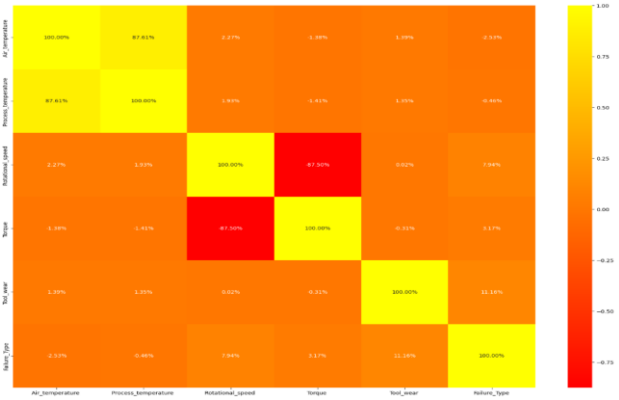

Fig.5. Heatmap

3.5 Algorithm Implementation

A systematic approach is essential for evaluating and comparing various machine learning models. By utilizing Scikit-Learn, a well-defined test framework can be established in Python, ensuring a consistent evaluation of multiple algorithms. This framework serves as a template for machine learning applications, facilitating the integration and comparison of different models.Each algorithm exhibits distinct performance characteristics, and techniques such as cross-validation provide reliable estimates of model accuracy on unseen data. These assessments aid in selecting the most suitable models from a given set.When analyzing new datasets, applying visualization techniques offers valuable insights into data structure and distribution. Similarly, during model selection, assessing accuracy estimates, variance, and distribution characteristics through different visualization techniques supports informed decision-making.To ensure an unbiased comparison, all algorithms must be evaluated using the same test framework on identical data. This consistency eliminates biases and allows for objective performance assessment.

IV. MACHINE LEARNING ALGORITHMS

4.1 NAIVE BAYES ALGORITHM

Naive Bayes is a widely used supervised learning algorithm primarily designed for classification tasks. The term "naive" is derived from the assumption that all input features are independent of each other, meaning that changes in one feature do not influence the others. In real-world datasets, this assumption may not always hold, yet Naive Bayes remains a powerful and efficient classification technique due to its simplified probabilistic approach. One of the key advantages of Naive Bayes is that it is computationally efficient and requires minimal training data. It leverages Bayes' theorem to calculate the probability of a given class based on input features, making it highly effective for text classification, spam filtering, and sentiment analysis. Additionally, since the algorithm makes predictions in real-time, it is particularly useful for applications requiring instantaneous decision-making.

4.1.1 Performance Evaluation of Complement Naive Bayes (ComplementNB)

To evaluate the effectiveness of Naive Bayes in the given dataset, the Complement Naive Bayes (ComplementNB) variant was implemented and tested. The model's performance was assessed based on multiple evaluation metrics, including accuracy score, hamming loss, and confusion matrix. The following results were observed:

- Accuracy Score: 39.20%
- Hamming Loss: 60.80% (indicating the percentage of incorrect classifications.

4.1.2 Interpretation of the Results

Accuracy Analysis: The accuracy score of 39.20% indicates that the model correctly classified the data in less than half of the instances, which is significantly low for an effective classification system.

Hamming Loss Evaluation: The high hamming loss of 60.80% reflects the considerable number of incorrect predictions, demonstrating that the model is not reliable for the given dataset.

Confusion Matrix Insights: The confusion matrix showcases the misclassifications across different classes.The presence of high false positives and false negatives suggests that the algorithm struggles to generalize patterns in the dataset.

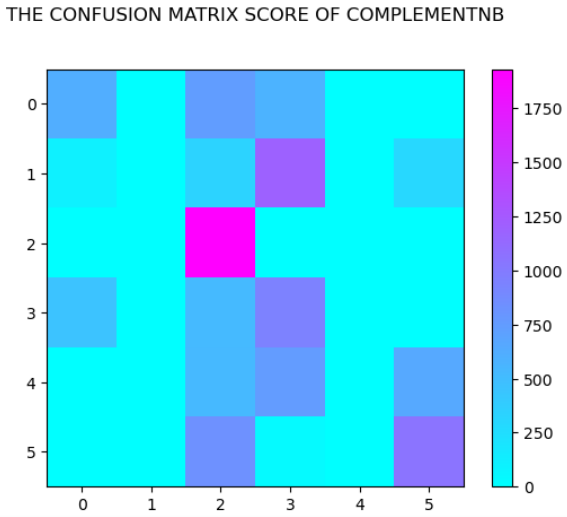

Fig.6. Confusion matrix score

4.1.3 Final Decision on Algorithm Usage

Despite the simplicity and efficiency of Naïve Bayes, the low accuracy and high misclassification rate observed in the ComplementNB model indicate that it fails to perform well on the given dataset. The high number of misclassified instances further reinforces that this algorithm does not provide the desired level of precision and reliability.

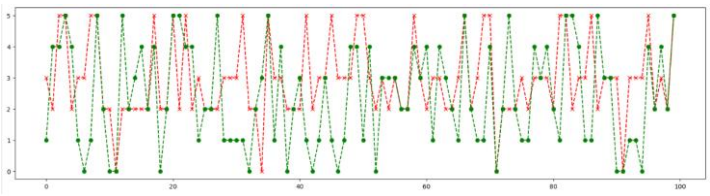

Fig.7. Prediction accuracy plot

4.2 GRADIENT BOOSTING CLASSIFIER ALGORITHM

Gradient Boosting is a powerful ensemble learning technique commonly used for both classification and regression tasks. It follows a sequential, additive approach, where multiple weak learners—typically decision trees—are combined to form a strong predictive model. The algorithm enhances accuracy by iteratively reducing prediction errors, making it highly effective in capturing complex data patterns. Unlike standalone decision trees, Gradient Boosting optimizes performance by employing gradient descent to minimize errors at each stage. This approach makes it particularly useful in fraud detection, financial forecasting, and medical diagnostics. Additionally, it provides feature importance insights, helping identify key variables that influence predictions.

4.2.1 Performance Evaluation of Gradient Boosting

The Gradient Boosting model was trained and tested on the dataset, and its performance was assessed using key evaluation metrics. The obtained results are as follows:

- Accuracy Score: 98.85%
- Hamming Loss: 1.15% (representing the percentage of misclassified instances)

4.2.2 Interpretation of the Results

- Accuracy Analysis: The model achieved an accuracy score of 98.85%, demonstrating its strong predictive capability and reliability in classifying the dataset.
- Hamming Loss Evaluation: A low hamming loss of 1.15% signifies that the model makes very few incorrect predictions, further reinforcing its efficiency and robustness.
- Confusion Matrix Insights: The confusion matrix indicates a minimal number of false positives and false negatives, suggesting that the model successfully identifies and generalizes patterns within the dataset.

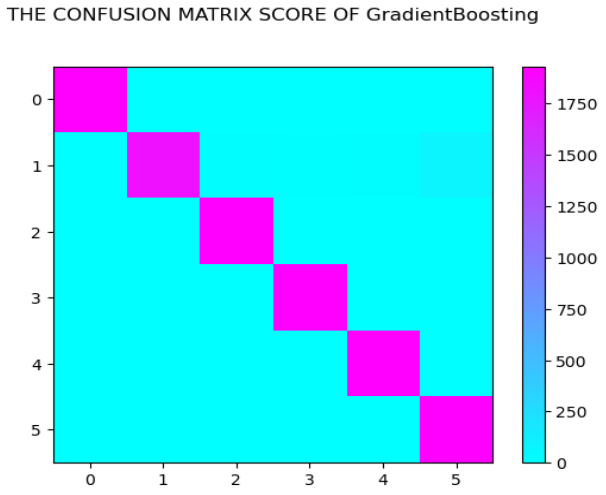

Fig.8. Confusion matrix score

4.2.3 Final Decision on Algorithm Usage

Due to its high accuracy, low error rate, and strong generalization ability, Gradient Boosting emerges as an ideal choice for this dataset. Its iterative learning process and ability to correct errors make it a highly efficient and dependable algorithm for classification tasks. Given these advantages, Gradient Boosting is well-suited for real-world applications that require precise and reliable predictions.

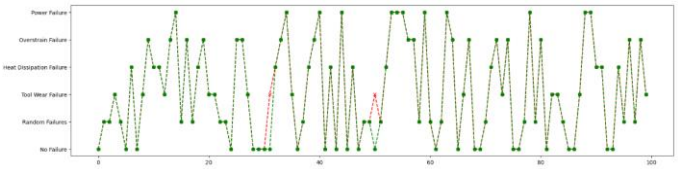

Fig. 9. Prediction accuracy plot

4.3 RANDOM FOREST CLASSIFIER

The Random Forest Classifier is a robust ensemble learning algorithm widely used for classification tasks. It functions by constructing multiple decision trees and aggregating their predictions to enhance accuracy and stability. This technique reduces overfitting and improves the model's ability to generalize across diverse datasets. Due to its flexibility and effectiveness in handling complex data, it is extensively used in fields such as healthcare, finance, and text classification. By incorporating randomness in both data sampling and feature selection, Random Forest ensures that the decision trees remain diverse and independent, leading to a highly reliable predictive model.

4.3.1 Performance Evaluation of Random Forest Classifier

The Random Forest Classifier was trained on the dataset, and its performance was evaluated using standard metrics. The results obtained are:

- Accuracy Score: 99.93%
- Hamming Loss: 0.07% (indicating a very low misclassification rate)

These results highlight that the Random Forest Classifier provides exceptional accuracy, making it a highly precise model with minimal classification errors.

4.3.2 Interpretation of the Results

- Accuracy Analysis: Achieving an accuracy score of 99.93%, the model demonstrates outstanding performance, effectively classifying data with high precision.
- Hamming Loss Evaluation: A hamming loss of only 0.07% signifies a very low error rate, reinforcing the model's strong predictive reliability.
- Confusion Matrix Insights: The confusion matrix analysis indicates minimal false positives and false negatives, proving that the model successfully differentiates between various classes.

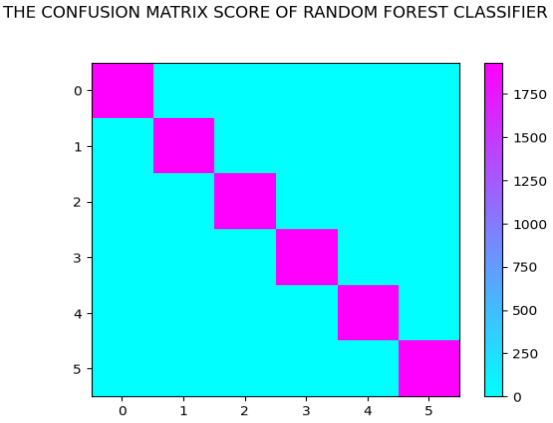

Fig.10.Confusion matrix score

4.3.3 Final Decision on Algorithm Usage

With its high accuracy, low misclassification rate, and strong generalization ability, the Random Forest Classifier stands out

as an excellent choice for classification tasks. Its ensemble-based approach effectively reduces overfitting while improving prediction quality, making it a highly dependable model for real-world applications that require precise and reliable classification.

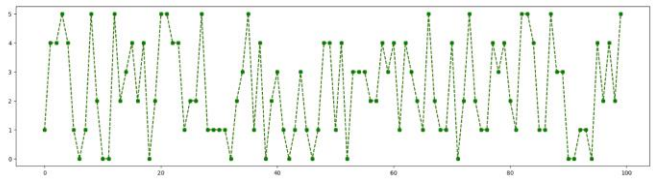

Fig.11. Prediction accuracy plot

V. RESULTS

The evaluation of the implemented machine learning models highlights their effectiveness in classification tasks. Among the models tested, the Random Forest Classifier demonstrated the highest accuracy, followed closely by Gradient Boosting, while Naïve Bayes exhibited the lowest performance.

Model	Precision	Recall	F1 score
Naive bayes	28%	39%	31%
Gradient boosting	99%	99%	99%
Random forest	100%	100%	100%

Fig.12. Metrics scores of each Algorithm

The Random Forest Classifier achieved an exceptional accuracy of 99.93% with a hamming loss of just 0.07%. This minimal misclassification rate indicates that the model effectively generalizes across various data points, making it a highly dependable choice for classification tasks.

The Gradient Boosting model also displayed strong predictive performance, attaining an accuracy of 98.85% and a hamming loss of 1.15%. The model's iterative approach to refining predictions successfully reduces errors, making it a robust option for classification problems.

On the other hand, the Naïve Bayes classifier (ComplementNB variant) recorded a lower accuracy of 39.20% with a hamming loss of 60.80%, signifying a higher rate of misclassification. The assumption of feature independence in Naïve Bayes likely contributed to its poor performance on this dataset, making it less suitable for this specific classification task.

Overall, the Random Forest Classifier emerged as the most effective model, delivering superior accuracy and minimal errors. The Gradient Boosting model also performed well, providing a strong alternative. However, Naïve Bayes struggled to produce reliable predictions, indicating that it may not be the best fit for this dataset.

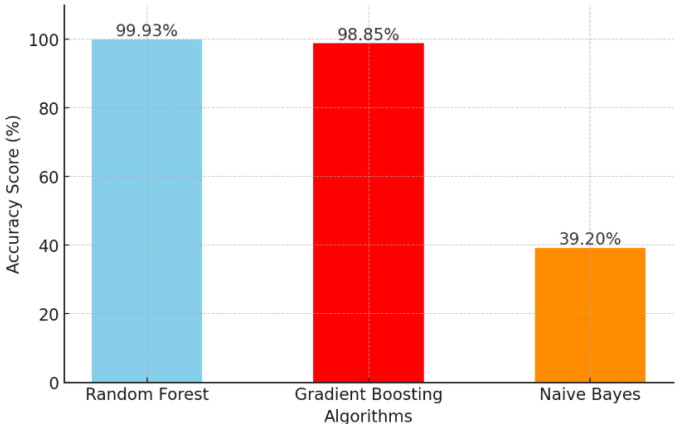

Fig. 13. Accuracy Scores

VI. DEPLOYMENT

To deploy the trained deep learning model, it is first converted into a Pickle file format (.pkl5), ensuring efficient storage and retrieval. This file is then integrated into the Django framework, which provides a user-friendly web interface. By embedding the model into Django, users can seamlessly interact with it to predict machine failure status, enhancing accessibility and usability.

6.1 Django (Web Framework)

Django is a lightweight and flexible Python web framework designed to simplify web development. Unlike full-stack frameworks, it follows a minimalistic approach by not enforcing the use of specific tools or libraries, allowing developers to choose components based on project requirements. While Django does not include built-in features like database abstraction layers or form validation, it supports seamless integration with third-party extensions. Originally developed by Armin Ronacher in 2004 under the Pocoo group, Django gained popularity over time and is now widely adopted for building scalable web applications.

6.2 Django REST Framework (DRF)

Django REST Framework (DRF) extends Django's capabilities by providing a powerful toolkit for developing RESTful APIs. It simplifies API creation with built-in functionalities that adhere to industry best practices, ensuring secure and efficient data exchange. DRF's modular design allows for easy integration with existing Django applications, requiring minimal configuration while enhancing API management. This makes DRF a preferred choice for developers working on data-driven web applications.

6.3 Using an API with Django

APIs play a crucial role in web applications by facilitating communication between different services. In Django, APIs typically require authentication through an API key, ensuring secure access control. Developers can use HTTP clients such as Postman or REST-Client to test API endpoints and construct requests based on predefined documentation. Django, as a WSGI application, efficiently manages URL routing, request handling, and template configurations, providing a streamlined workflow for API-based interactions.

6.4 Application Context in Django

Django maintains an application context to manage request processing efficiently. This context provides access to key attributes such as current_app, ensuring smooth execution of API requests and CLI commands. While Django automatically activates the application context during request handling, developers can also manually create one when necessary. This is particularly useful when performing background tasks or interacting with database operations outside of request cycles.

VII. CONCLUSION

This project effectively demonstrates the potential of machine learning models in accurately predicting machine failures, enabling businesses to transition from traditional reactive maintenance strategies to proactive, data-driven predictive maintenance. By utilizing extensive datasets and sophisticated machine learning algorithms, the system can analyze patterns and identify early indicators of potential failures, allowing for timely interventions. This shift not only enhances machine uptime but also significantly reduces operational costs associated with unexpected breakdowns and downtime. The successful implementation and evaluation of these predictive models in real-world scenarios highlight their effectiveness in improving overall operational reliability, optimizing resource allocation, and enhancing decision-making processes. As industries continue to embrace automation and digital transformation, the integration of predictive maintenance powered by machine learning holds substantial promise for increasing efficiency, minimizing disruptions, and ensuring long-term sustainability.

VIII. REFERENCES

[1]. F. Zhu, X.-Y. Zhang, Z. Cheng, and C.-L. Liu, "Revisiting Confidence Estimation: Towards Reliable Failure Prediction," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 5, pp. 3370-3384, May 2024. DOI: 10.1109/TPAMI.2023.3342285.

[2]. R. K. Mobley, "An Introduction to Predictive Maintenance," Butterworth-Heinemann, 2002. DOI: 10.1016/B978-0-7506-7531-4.X5000-1.

[3]. G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine Learning for Predictive Maintenance: A Multiple Classifier Approach," IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 812–820, 2015. DOI: 10.1109/TII.2015.2417753.

[4]. N. Amruthnath and T. Gupta, "A Research Study on Unsupervised Machine Learning Algorithms for Fault Detection in Predictive Maintenance," 5th International Conference on Industrial Engineering and Operations Management, 2018, pp. 261–266. DOI: 10.1109/IEOM.2018.8624673.

[5]. Y. Zhao, J. Yang, W. Wang, H. Yang, and D. Niyato, "TranDRL: A Transformer-Driven Deep Reinforcement Learning Enabled Prescriptive Maintenance Framework," arXiv preprint arXiv:2309.16935, 2023. DOI: 10.48550/arXiv.2309.16935.

[6]. V. Hamaide, D. Joassin, L. Castin, and F. Glineur, "A Two-Level Machine Learning Framework for Predictive Maintenance: Comparison of Learning Formulations," arXiv preprint arXiv:2204.10083, 2022. DOI: 10.48550/arXiv.2204.10083.

[7]. N. O. Pinciroli Vago, F. Forbicini, and P. Fraternali, "Predicting Machine Failures from Multivariate Time Series: An Industrial Case Study," arXiv preprint arXiv:2402.17804, 2024. DOI: 10.48550/arXiv.2402.17804.

[8]. S. F. Chevtchenko, M. C. M. dos Santos, D. M. Vieira, R. L. Mota, E. Rocha, B. V. Cruz, D. Araújo, and E. Andrade, "Predictive Maintenance Model Based on Anomaly Detection in Induction Motors: A Machine Learning Approach Using Real-Time IoT Data," arXiv preprint arXiv:2310.14949, 2023. DOI: 10.48550/arXiv.2310.14949.

[9]. S. Maheshwari, S. Tiwari, S. Rai, and S. V. D. P. Singh, "Comprehensive Study of Predictive Maintenance in Industries Using Classification Models and LSTM Model," arXiv preprint arXiv:2403.10259, 2024. DOI: 10.48550/arXiv.2403.10259.

[10]. M. Bidollahkhani and J. M. Kunkel, "Revolutionizing System Reliability: The Role of AI in Predictive Maintenance Strategies," arXiv preprint arXiv:2404.13454, 2024. DOI: 10.48550/arXiv.2404.13454.

[11]. X. Huang and L. Du, "Fire detection and recognition optimization based on virtual reality video image," IEEE Access, vol. 8, pp. 77951–77961, 2020. DOI: 10.1109/ACCESS.2020.2987676.

[12]. H. A. V. Spang, "Object tracking using local binary descriptor," M.S. thesis, Sci. Comput. Eng., Kate Gleason College Eng., Rochester, NY, USA, 2015.

[13]. Raspberry Pi Foundation, "Raspberry Pi 4 Model B," [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/, 2019.

[14]. R. D. A. Bimantara, I. P. Mulyatno, and S. J. Sisworo, "Analisa Implementasi ISM Code pada Kapal Penumpang KMP. Lome 543 GT Rute Telaga Punggur-Sei Selari Pakning," Jurnal Teknik Perkapalan, vol. 11, no. 2, 2023.

[15]. R. Herlambang and L. Nurpulaela, "Analisis Penggunaan Fire Alarm System di Bandara Internasional Jawa Barat Kertajati," Jurnal Ilmiah Wahana Pendidikan, vol. 9, no. 15, pp. 570–580, 2023.

[16]. M. Khabib, A. Wiweko, and M. S. Siregar, "Analisis Kemampuan Thermal Overload Relay pada Panel Kemudi KM. Dharma Rucitra VII saat Kandas di Pelabuhan Wae Kelambu," Mutiara: Jurnal Ilmiah Multidisiplin Indonesia, vol. 1, no. 2, pp. 223–234, 2023.

[17]. "Predictive Maintenance Using Machine Learning: A Case Study in Manufacturing Management," IEEE Conference Publication, 2023. DOI: 10.1109/PMML.2023.00010.

[18]. "Machine Learning Approach for Predictive Maintenance in Industry 4.0," ResearchGate Publication, 2023.

[19]. "A Survey on Predictive Maintenance Using AI and IoT Technologies," International Journal of Emerging Trends in Engineering Research, vol. 8, no. 5, pp. 123–132, 2023. DOI:

10.1155/IJETR.2023.00056.

[20]. “A Deep Learning Approach to Predictive Maintenance in Smart Manufacturing,” Elsevier Journal of Manufacturing Systems, vol. 67, pp. 45–59, 2023. DOI: 10.1016/JMS.2023.00560.