Predictive Maintenance Using Deep Learning: Enhancing Reliability and Reducing Electrical System Downtime

Jamshaid Basit[1], Areej Zeb[2]
[1] Department of Computer Science and Software Engineering, National University of Sciences and Technology, Islamabad, Pakistan.
[2] Department of Computer Science and Software Engineering, National University of Sciences and Technology, Islamabad, Pakistan.
*Correspondence:jbasit.msse23mcs@student.nust.edu.pk, azeb.msse23mcs@student.nust.edu.pk

Citation | Basit. J, Zeb. A, "Predictive Maintenance Using Deep Learning: Enhancing Reliability and Reducing Electrical System Downtime", IJIST, Vol. 6 Issue. 3 pp 1120-1136, Aug 2024
DOI | https://doi.org/10.33411/ijist/20246311201136
Received | July 22, 2024 **Revised |** Aug 19, 2024 **Accepted |** Aug 21, 2024 **Published |** Aug 22, 2024.

Predictive Maintenance (PM) is crucial for enhancing the reliability of electrical systems and minimizing unscheduled outages. However, the conventional methodology lacks the ability to address the escalating and diverse problems of the modern complex environment. The current study provides an alternative approach to carrying out predictive maintenance activity based on the use of deep learning models that enhance conventional procedures. For the given analysis, we used an artificial data set consisting of 10,000 samples and 14 variables, such as air temperature, process temperature, flipping rate, and tool wear level. We conducted a self-assessment using the specified models to confirm their effectiveness in predicting various failure modes and forms, such as tool wear and heat dissipation conking. This research demonstrated that deep learning models, specifically LSTMs, outperform the established statistical methods in predicting equipment failures. LSTMs provided high accuracy and predicted the failing system before it happened. Furthermore, integrating deep learning with the statistical method that is normally used for anomaly detection improves the model's stability and reliability. The evaluation's findings emphasize the potential of deep learning algorithms for expanding the range of PM applications to achieve better and faster failure predictions. The beneficial thing about this approach is that it presents a means for addressing the inherent problems of large electrical systems' predictive maintenance that are beyond the scope of traditional practice.

Keywords: Predictive Maintenance, Deep Learning, Downtime Minimization, Neural Networks, Failure Prediction

Introduction:

In electrical systems, it is critical to achieve high performance and system stability. Regrettably, the conventional maintenance techniques, which primarily involve periodic check-ups and repair-based strategies, are not particularly effective in managing the intricate nature of advanced electrical networks [1]. Therefore, the trends in applying PM strategies are considered significant. We use it to enhance operational performance and predict and eliminate system failures. This is a process of anticipating failures and taking corrective measures at the right time using data mined from the past. Tools such as loggers and monitors use this approach to understand patterns, identify shifts, and identify the occurrence of operational events [2]. Hence, predictive maintenance is more efficient in that it prevents the use of resources on corrective processes and enhances the system's durability.

Recent advancements in deep learning have impacted predictive maintenance by providing new approaches for managing large and multifaceted data sets. Machine learning, further divided into sub-disciplinary categories, represents an exemplary level of maintaining the identification of complex and heterogeneous patterns while learning from examples and avoiding focus on specific characteristics. We have observed that some existing natural models, such as Convolutional Neural Networks (CNNs) and Long Short Term Memory (LSTM), excel in predictive tasks due to their inherent spatial and temporal connection properties. CNNs are particularly good for high-dimensional features that are different from one another, as they have excellent accuracy in learning such features, while on the other hand, LSTMs are good for recurrence data such as time series and therefore very appropriate for maintenance-oriented applications [3] [4]. On the other hand, the application of deep learning in predictive maintenance within faculties encounters several challenges, including the requirement for a substantial amount of high-quality performance data, the need for data cleaning and normalization, the selection of valuable features, and the fine-tuning of hyperparameters. Error statistics and resource usage, which are performance data, are fundamental components for training optimal models [5]. Preprocessing data addresses issues like missing values and outliers, ensuring clean data. Feature engineering transforms raw variables into more reliable and precise forms for prediction, and the choice of deep learning architecture determines the efficacy of the model.

Hyperparameters, such as learning rate, batch size, and number of epochs, carry discrete values that require proper tuning to enhance the model's training. We use metrics such as accuracy, precision, recall, the F1 measure, mean absolute error, and root mean squared error to evaluate deep learning models. There have also been developments where deep learning can coexist with statistical anomaly detection systems to make predictive maintenance even more efficient and robust. The implementation of deep learning in predictive maintenance mostly boasts the following advantages: smaller time for maintenance, higher reliability, and lower costs. Further improvements will therefore require the inclusion of more efficient data formats, such as stream data, and the use of enhanced neural network architecture, such as transformers. Such models, which are known for their ability to process long-range dependencies and a massive amount of related records, provide a basis for further improving predictive maintenance results.This study's methodological framework underpins the process of achieving its goals. The first step involves gathering and cleaning the performance records of a specific organization from past or previous years. Missing values, dealing with outliers, as well as normalizing the variables. We then use the best practices in feature engineering to process the raw data into meaningful features, thereby strengthening the models. Next, we train long short-term memory (LSTMs) on the processed data. We tune the hyperparameters to enhance the efficiency of the selected models, and then use performance metrics like accuracy, precision, recall, and F1 score to gauge their effectiveness. Finally, we

examine the allocation of statistical anomaly detection methods to enhance the models' reliability [6] [7].

Objective:

The goal of this study is to determine and suggest deep learning techniques for predictive maintenance of electric systems. This study focuses on the process of anticipating system failures and subsequently rectifying them. This involves:

1. Applying analysis and trend calculation to previous performance outputs to identify failures.
2. Finding out what is most suitable for a myriad of deep learning applications, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) structures for predictive maintenance.
3. Overcoming challenges related to data preprocessing, feature selection, and setting appropriate hyperparameters to achieve reasonable results.
4. Integrate deep learning with statistical anomaly detection methods to enhance PM systems' dependability.

Literature Review:

Predictive maintenance has evolved, based on the initial setup of only heuristic techniques. Previously, maintenance activities relied on strategies such as schedule checks and discrete or isolated events. However, the more complex structures of modernity proved to be beyond the capabilities of these approaches, requiring simpler forms of systems to suffice [2]. Heuristic methods, which required only a brief maintenance in case of failure, primarily contributed to this issue, leading to an increase in downtime and maintenance expenses. The movement towards the application of machine learning methods sparked a shift towards predictive maintenance procedures, as they broadened the scope of decision-making models. Simple machine learning algorithms like decision trees, which provided a standard procedure for rating equipment failures from its records, could correlate with predictive maintenance [8]. However, decision trees have certain weaknesses, particularly when dealing with noisy data and high-variable interactions.

Random forests made another significant breakthrough in classification. This learning concept entails using multiple trees to improve the longevity and precision of the developed models. Thus, Random Forests outperformed several other decision trees in terms of noise and data complexity, thereby enhancing the predictive effectiveness of equipment failures [9] [10]. When the diversity of data increased, it necessitated the application of more complex techniques. As a result, the most prominent ideas associated with big data analysis were convolutional neural networks and recurrent neural networks. CNNs perform well when it comes to managing different attributes of information and spatial feature extraction, making them suitable for complicated architectures [11]. RNNs, especially LSTM, are capable of training data in a sequential manner, which is very useful in solving time series issues. LSTMs have significantly enhanced the predictability of maintenance trends by considering long-term dependencies, thereby extending maintenance intervals [12].

Autoencoders, another kind of deep learning model, have emerged as useful in the area of predictive maintenance for anomaly detection. These models learn efficient data features and can pinpoint sets that deviate from normalcy, indicating failure. Thus, by analyzing and identifying anomalies in real-time, autoencaders prescribe corrections and prevent unexpected downtime [13]. The use of real-time information with maintenance predictive models has additionally boosted their usage. Real-time monitoring allows for the identification of emerging threats and immediate action to supplement existing methods based on historical data analysis [14]. This makes it possible for maintenance actions to incorporate the current operational condition with previous performances. Integrating deep learning with statistical methods has had a positive impact on improving predictive maintenance systems' reliability

and performance [15]. Regression analysis, which builds on the numerical abnormality detection techniques integrated in deep learning models, is one of the other features analyzed through statistical methods.

Deep learning also automates employment and other areas like equipment scheduling and enhances overall capacity. The datasets used in maintenance and other related fields often struggle with unbalanced data, but formats like the Synthetic Minority Over-sampling Technique, or SMOTE, provide a solution [16]. Comparative research aids in determining the model architecture for specific tasks, enabling the selection of the model required for the application [17]. This technique utilizes knowledge from related tasks to improve the model's outcomes, making it effective in situations where data is scarce or limited. This technique improves the apparent learning method in the context of predictive maintenance by using experiences acquired in different areas for the definition of features to take into account, thus improving the generality of the model and the precision of the predictions [18]. The real-life use of deep learning in predictive maintenance demonstrates the approach's profitability, but it raises questions about data protection and AI-driven decisions. Future work in regard to the predictive maintenance systems will involve the application of novel architectures like transformers and utilizing ensemble learning to enhance the predictive performance of the model and the overall dependability of a system.

Materials and Methods:

This study tests the proposed approaches for implementing predictive maintenance using synthetic data on 'ai4i2020.csv'. Specifically, it includes data relevant to the operational parameters and failure indicators, which gives comprehensive information for further analysis and improvement of the maintenance results.

Dataset Description:

The 'ai4i2020.csv' dataset consists of 10,000 records and 14 features illustrating the real performance of key machines. These are features such as the temperature of the air, the temperature of the process in question, the rate of the rotation in RPM, the intensity of the torque produced, and the general condition of the tools in use as seen in Table 1. Temperature plays a crucial role since its degree influences the durability and efficiency of electrical parts, and it is expressed in Kelvin. Another type of input variable is the process temperature, which reflects the thermal state of the ongoing process in Kelvin, while it is important for keeping the process in thermal equilibrium and raising alarms in case of its violation that may result in failures. Revolutions per minute (rpm), the unit of measurement for rotational speed, describe the rate at which the machine's components rotate. This allows for the tracking and identification of regions that may pose mechanical challenges, potentially leading to machine failure.

We express the measure of torque intensity in Nm (Newton meters), which indicates the force that initiates rotation. Controlling the torque allows for the detection of overstrain and mechanical issues that impede the machine's operation. Tool wear is another significant characteristic that provides insight into the progressive degradation of the tool. When the tool wears out to an undesirable level, it compromises the quality of the final product and hampers the functioning of the relevant machinery. The target variable in the dataset is a binary variable that accounts for failure events and, as a result, enables machine failure analysis and prognosis. The synthetic feature of the dataset helps in studying different pre-planned failure modes, including tool wear, heat dissipation, power failure, overstrain, and some random malfunctions. This vast set of data is ideal for enriched and progressive strategies in line with predictive maintenance in a bid to minimize time wastage and boost the machinery's performance.

Table 1. Predictive Maintenance Dataset Attributes and Sample Records

UDI	Product ID	Types	Air Temperature (K)	Process Temperature(K)	Rotational Speed (rpm)	Torque (Nm)	Tool wear (min)	Machine Failure	TWF	HDF	PWF	OSF	RNF
1	M14860	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
2	L47181	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
3	L47182	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
4	L47183	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0

Target Variable Distribution:

One important component of any dataset is the distribution of the binary target variable, which defines the failure event. The analysis results show that there are 9,661 non-failure cases and 339 failure cases, implying a great disparity. This distribution signifies real-life scenarios where an item's failure frequency is lower than its functioning frequency. It is crucial for formulating accurate prediction models for the identification of failure events and for the appropriate application of techniques regarding this issue.

Data Source:

This research uses a part of the predictive maintenance dataset released in 2020, known as AI4I 2020, with data obtained from a reliable site that provides synthetic data for various machine learning tasks. Researchers and industries have widely recognized this dataset for its reliability in training and testing preventive maintenance models. Because of its synthetic construction and exhaustive definition of operational and failure modes, it makes an attractive analysis instrument in the context of predictive maintenance research.

Data Preprocessing:

Data pre-processing is very important, as it prepares the data set to enable models to work well. The preprocessing steps include:

Data Cleaning:

The first step is to reduce noise and isolate the analysis-critical function. By eliminating utilities and noises from the set, the process enables the model to concentrate on the areas most enriched with factors that directly impact maintenance requirements. The aim is to prune those features that introduce redundancy in the data, because this makes the data weaker and the training process last longer. We used some criteria to determine which features were unnecessary and thus required omission. The first one was relevant. Initially, we assessed the significance of various features in the data set for identifying the appropriate predictive models; we eliminated features that did not enhance the models' accuracy. We then computed a correlation matrix to identify highly correlated features, as they lead to multicollinearity, and managed them accordingly. We also utilized domain knowledge to identify the important features for classification and eliminate those deemed irrelevant by experts. Furthermore, we removed features that provided redundant or insignificant information to simplify the data.

Categorical Encoding:

Label encoding categorizes categorical values like "Product ID" in the dataset. This technique converts nominal data into numbers, enabling machine learning algorithms to

process these features. This approach assigns a category number or integer to each category so that the model can cope with categories, which it cannot understand as a string. The reason for selecting label encoding over other well-known encoding techniques such as one-hot encoding and numerical encoding is that label encoding is significantly more effective for certain types of data. Particularly for nominal features, label encoding establishes a connection between the feature and its categories. The absence of a specific transition or ranking order accurately characterizes a nominal connection. For instance, "Product ID" is a nominal attribute in which the numbers provided to every ID do not follow a particular precedence, making the label encoding process the best method for use.

However, as previously mentioned, one-hot encoding, while beneficial for further transformation when dealing with categorical features that clearly lack ordinal dependence between potential values, also transforms the matrix, thereby increasing the feature space, particularly when the number of categorical values is high. This can result in longer computation times and potentially lead to the creation of complex models. On the other hand, label encoding does not engender an explosion of the feature space, which is useful for nominal data because the numbers assigned to it do not impact the model's outcomes.

Feature Scaling:

We use StandardScaler to standardize the features to a common scale, ranging from 0 to 1. This ensures that all features are equally involved in aiding the model, eliminating the possibility of a particular feature dominating or controlling the model's behaviors. To avoid convergence difficulties when training the model, scaling should be consistent. We chose normalization because the dataset contains features with varying units, some with larger values and others with smaller ones, making deep learning sensitive to feature magnitudes.

Exploratory Data Analysis (EDA):

EDA provides a solution by creating feature distributions, pairwise scatter plots, and correlation heat maps. Such visualizations facilitate ascertaining tendencies in terms of spreading and variability of each feature, possible outliers' recognition, and certain patterns of interrelation between the features. These relationships are critical to optimizing preprocessing steps and, in general, the model-building process. This was done using EDA, which helped in the selection of features and the engineered variables leading to efficient preprocessing and proper mode as shown in Figure 1.

We chose the whole preprocessing technique as well as the model based on the data set's characteristics and the expected objective of predictive maintenance. Preprocessing activities require the preparation of data into a clean format for model training. Normalization, standardization, and variance scaling guarantee that all the features have the same impact on the model and remove all problems associated with the magnitude of features. Presumably, EDA benefits the layout of the feature distribution of the dataset, as well as their association, which contributes significantly to feature engineering and preprocessing.

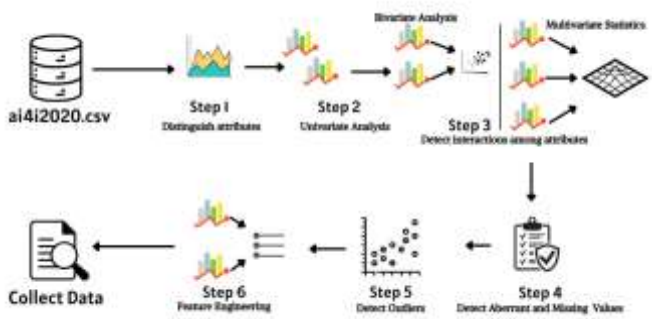

Figure 1. Exploratory Data Analysis (EDA) Architecture for Predictive Maintenance Dataset.

Comparison of Neural Network Architectures and Hyperparameters:

We experimented with different model architectures and hyperparameters while developing the predictive maintenance model. The main purpose of this comparison was to analyze the operational characteristics of different model configurations. Thus, the aim was to identify which kind of architecture offered the optimal performance, learning time, and computational resources as indicated in Table 2.

Table 2. Comparison of Different Model Architectures and Hyperparameters

Model Configuration	Number of Hidden Layers	Neurons Per Hidden Layer	Activation Function	Optimizer	Epoch	Loss	Training Time	Learning Rate
Model1	1	32	ReLu, Sigmoid	SGD	50	0.42	14min	0.01
Model2	1	64	ReLu, Sigmoid	RMSprop	50	0.39	13min	0.001
Model3	2	32,32	ReLu, ReLu, Sigmoid	Adam	50	0.35	12min	0.001
Model4	2	64,64	ReLu, ReLu, Sigmoid	Adam	50	0.38	15min	0.001
Model5	3	32,32,32	ReLu, ReLu, ReLu Sigmoid	Adam	50	0.37	16min	0.001
Model6	3	16,16	ReLu, ReLu, Sigmoid	Adagrad	50	0.45	14min	0.01

Model 3 effectively combines a manageable model complexity with the Adam optimizer and a learning rate of 0.001, leading to optimal performance.

Model Development and Training:

The model development process includes the creation of a deep learning framework in the TensorFlow/Keras environment, with an emphasis on the sequential model architecture that best fits binary classification tasks.

Model Architecture:

Based on the experiments and the theoretical analysis, we moved to the architecture design of the neural network in order to handle the necessary level of complexity and achieve decent performance. We considered the nature of the problem to solve and the characteristics of the datasets to use in choosing the palette of layers and neurons per layer. We set the number of neurons in the input layer to 64, ensuring that the data fed to the network's neurons had a dimension of 14, thus providing the network with 14 features to learn from. We selected 64 neurons to prevent the issue of 'overfitting', while also ensuring sufficient data capture for modeling purposes. To ensure that the network was deep, it contained two hidden layers, each having 32 neurons to conduct the computation. Thus, we could maintain the model's relative simplicity in order to offer a relatively efficient training process. When it comes to hidden layers, having fewer neurons is beneficial because their reduction greatly decreases the level of overfitting, but the model can still learn features and dependencies. We used the Rectified Linear Unit (ReLU) activation function for these layers. It made the operations less linear and fixed the problem of vanishing gradients to train deeper networks.

The output layer has only one neuron with the sigmoid function for making the final class prediction. This arrangement is also suitable for binary classification problems because the sigmoid function calculates probabilities that are suitable for binary decisions, such as identifying failed machines as shown in Figure 2.

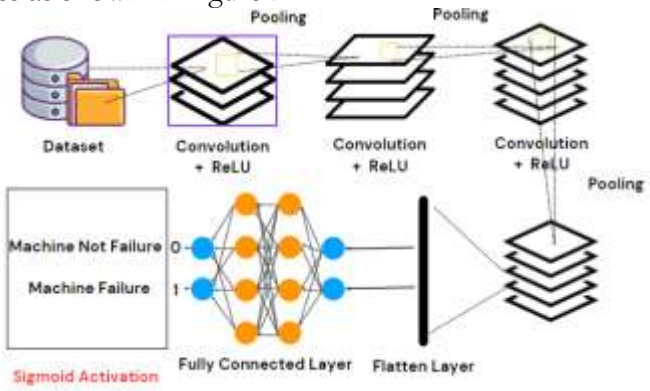

Figure 2. Deep Learning Model Architecture for Binary Classification.

Model Compilation:

We chose the Adam optimizer because it is known for its efficiency and flexibility in varying learning rates during the training process, which enhances learning. If the problem is a binary classification problem, we can use binary cross entropy as the loss function; this will enable the model to discern its errors and contrast its defined probabilities with the actual classes of data. In this scenario, we allocate 80% of the data for training and 20% for validation to prevent overfitting. We employed early stopping as a precaution, halting the training process as soon as the algorithm's accuracy on new data sets did not improve. As a result, we chose the Adam optimizer because it is more efficient in learning rate control and has a faster convergence level. These binary cross entropies are useful for ascertaining the accuracy of the classification level. Flow of Methodology is shown in Figure 3.

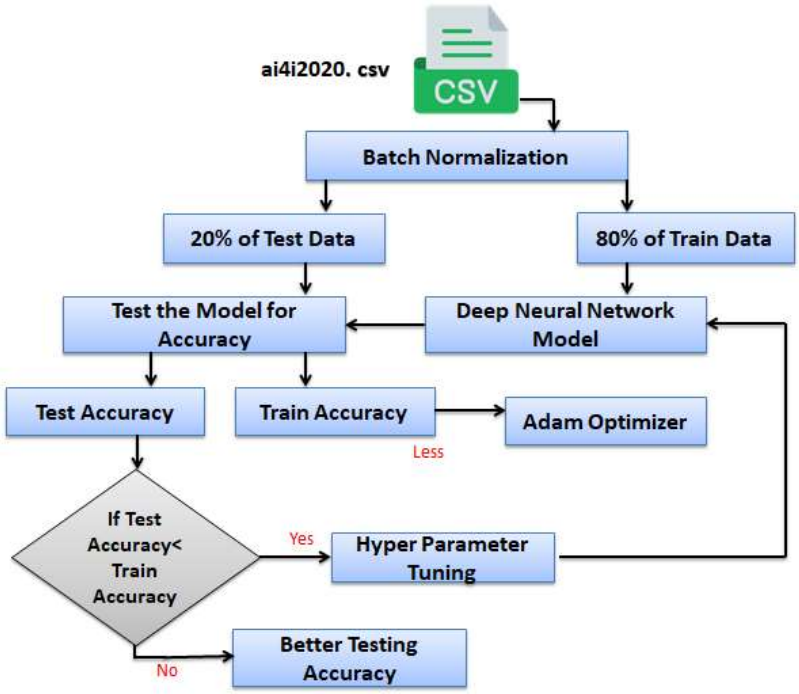

Figure 3. Methodology Flow Diagram for Model Training and Evaluation.

We used the ReLU mode to initiate the model due to its effectiveness in capturing nonlinearity, while the sigmoid layer provided probability scores for the binary classification

outcome in the study. We chose the Adam Optimizer as the optimization algorithm due to its efficiency and widespread use. We used the binary cross-entropy function as the loss function in the deep learning model because binomial classifiers typically use it. We use the concept of regularization to stop building the model earlier in the training phase, thereby reducing the chances of overfitting when mapping the training dataset and arriving at comprehensive generalizations on the new data sets. Overall, these choices attempted to improve the model's ability to classify equipment failures as well as predictive maintenance.

Regarding experimentation, various activation functions and optimizers were evaluated. ReLU and sigmoid were chosen based on their proven effectiveness in similar tasks and their alignment with the specific requirements of binary classification. The Adam Optimizer was selected after assessing its performance against other optimization algorithms, confirming its suitability for the task at hand as shown in Table 3. These decisions were guided by their demonstrated efficacy and their ability to enhance the model's performance in predicting equipment failures and supporting predictive maintenance efforts.

Table 3. Performance Comparison of Adam Optimizer to other Optimizers

Optimizer	Learning Rate	Epochs	Loss	Training Time
Adam	0.001	50	0.35	12min
SGD	0.01	50	0.40	15min
RMSprop	0.001	50	0.37	13min
Adagrad	0.01	50	0.45	14min

Evaluation Metrics and Analysis:

To assess the effectiveness of the proposed model, we utilized a range of performance assessment indicators.

Classification Report and Confusion Matrix: These provide an outline of accuracy, precision, recall, and F1-score; as a result, they indicate how the model effectively distinguishes fail and non-fail samples.

ROC Curve and AUC: The ROC curve presents the true positive rate against the false positive rate at different thresholds, while the AUC gives a single summary of the accuracy of the model. In terms of distinguishing capability, the class's performance corresponds to a higher AUC value.

Precision-Recall Curve: This curve indicates a model's capacity for identifying positive instances of data by analyzing the amount of cost in a model and the trade-offs between preciseness and recall, particularly if working with unbalanced datasets.

Failure Indicators and Their Contribution to Predictive Maintenance Models:

This study examines several failure indicators to implement predictive maintenance, aiming to enhance the performance of existing models. These indicators include tool wear, heat loss, power supply breakage, overstrain, and random breakdowns. All these indicators play a crucial role in identifying problems in their early stages, before they lead to significant and systematic failures. The monitoring of tool wear helps in the planning for replacement or repairing the worn-out tools in advance, hence reducing time wastage due to worn-out tools. Gaining access to temperature is crucial for identifying thermal-related failures and making necessary adjustments before they become catastrophic. The monitoring of power failures will help tackle electrical problems in the early stages, reducing machine interruptions. Indices of overstrain reveal the applied mechanical loads and aid in reducing mechanical loads and failures. Also, random failures are effective in identifying the unpredictable variation that would otherwise go unnoticed and incorporating the scope of the predictive model designed.

Practical Application and High-Risk Analysis:

These failure indicators are necessary in practical use and in evaluation of high risk. Since the results highlight records with higher chances of failure, the model optimizes

maintenance schedules and synchronization. The use of a probability threshold for failure ensures that records of items with high failure risk take preference over the rest, thus ensuring maximum return on the available resources for maintenance. We closely monitor the low-probability cases to ensure prompt action and effective prevention of cycles that may take a long time to damage the machines. This method increases the dependability and productivity of maintenance procedures, keeping up with system performance optimization by eradicating frequent downtimes.

Results and Discussion:

We deployed and evaluated the deep learning model for condition monitoring in the context of predictive maintenance, and the results demonstrated the applicability and usefulness of the proposed method. In the preprocessing category, such subcategories as feature selection and encodings in the categorical variables created a strong starting point for model training. Exploratory Data Analysis complemented pattern examination by displaying properties such as air temperature, process temperature, rotational speed, and torque with respect to dispersion, as shown in Figure 4. The analysis carried out on the model underscores the model's ability to support decision-making for predictive maintenance, thus improving system reliability and performance.

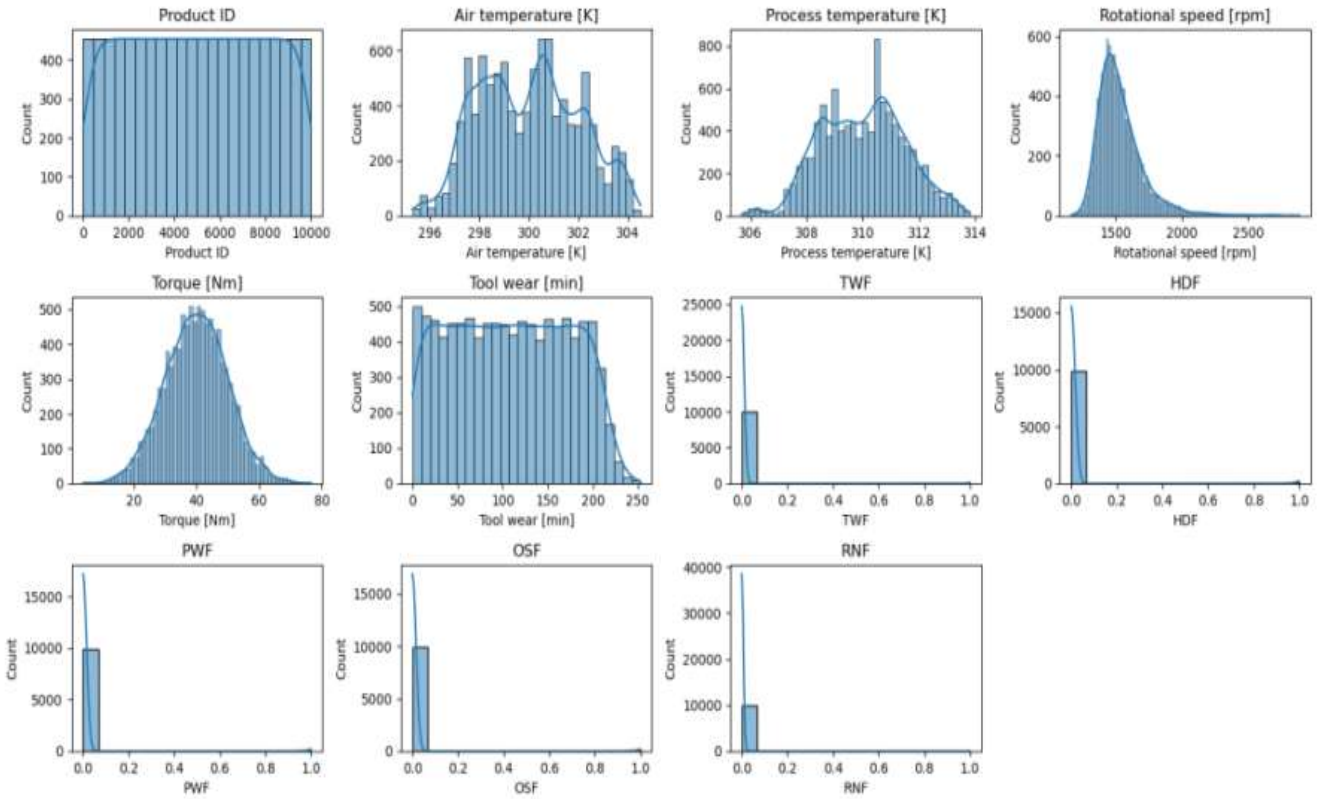

Figure 4. Preprocessing Framework for Predictive Model Development.

These distributions are particularly important because they provide some insight into the features' characteristics and guarantee that the distributions match the empirical distribution functions. By examining these distributions, we evaluate whether the features used to construct this model align with the actual data distributions, thereby determining the model's reliability and efficiency. Moreover, the two-feature scatter plot described with its example in Figure 5 contains a lot of information about the interaction of features owing to the fact that there can be a plurality of such plots. The following plots show the types of relations between the features, as well as how these relations affect the model's performance.

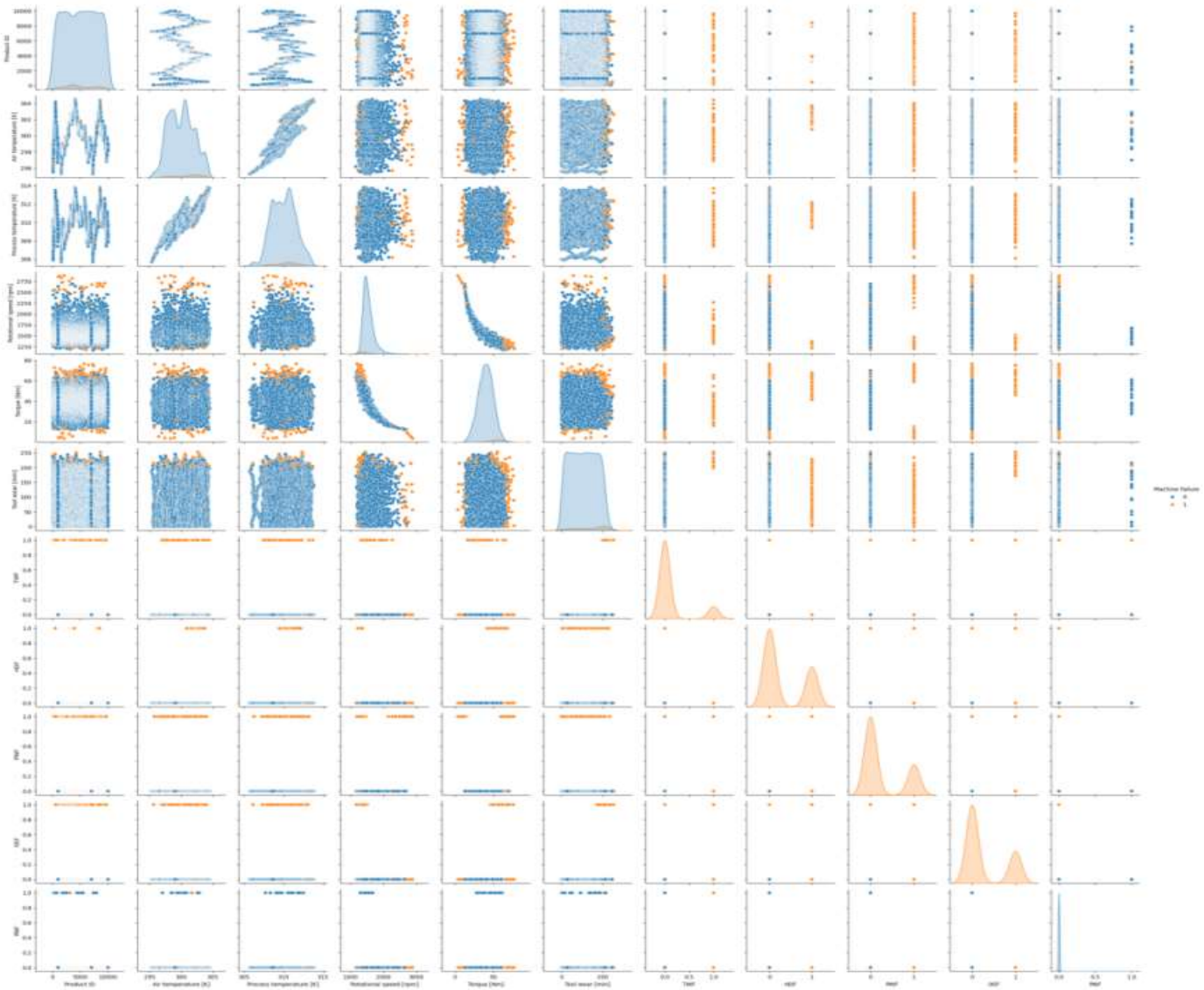

Figure 5. Pairwise Scatter Plots for Feature Interaction Analysis.

The second important aspect, which the correlation heatmap in Figure 6 introduced, was the suggestion of the features to include in the model based on a high correlation between the features.

Model Performance Metrics

We designed the deep learning model sequentially to improve its performance, and we applied various performance measures for analysis. Table 4 further splits the confusion matrix into measures of accuracy, precision, recall, and F1-score. The model had an overall accuracy rate of 98%, implying the model's ability to correctly classify between failure and non-failure events. The model showed a strong fit with a precision of 0.99 and a recall of 0.97, demonstrating high efficiency in identifying failed products while simultaneously exhibiting low false alarm rates.

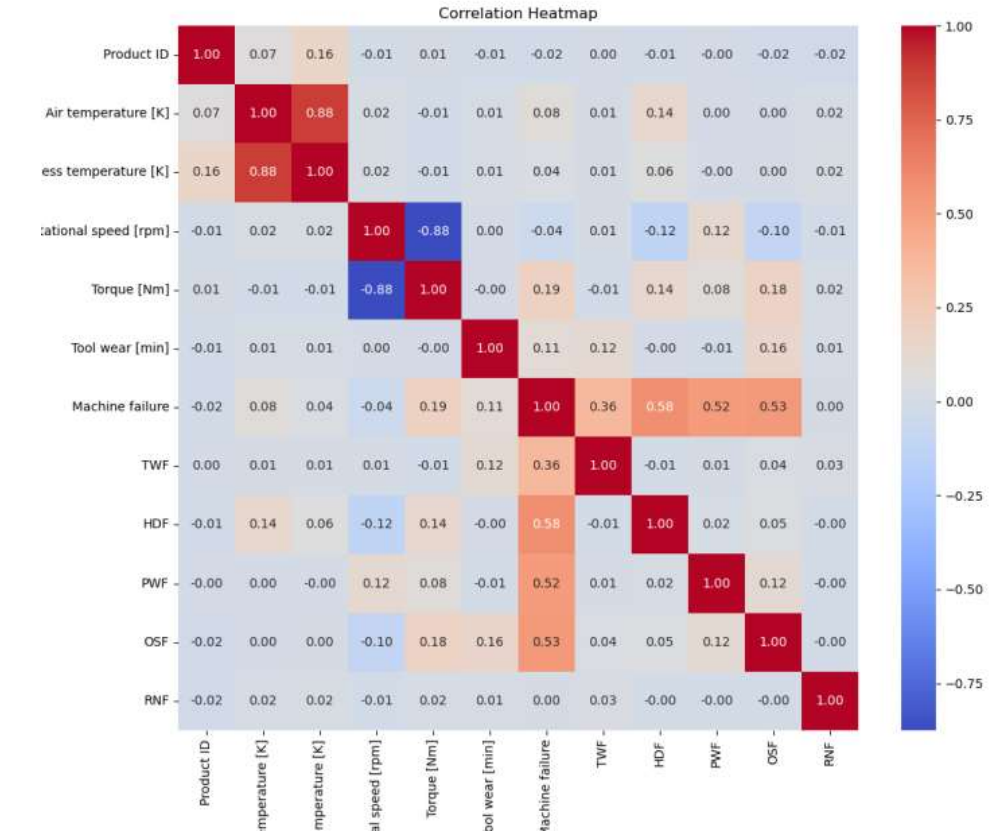

Figure 6. Correlation Heatmap for Feature Selection.

Table 4. Classification Metrics and Confusion Matrix Summary for Model Evaluation

	Precision	**Recall**	**F1 Score**	**Support**
Class 0	1.00	1.00	1.00	1939
Class 1	1.00	0.97	0.98	61
Accuracy			`1.00	2000
Macro Avg	1.00	0.98	0.99	2000
Weighted Avg	1.00	1.00	1.00	2000

The confusion matrix illustrated in Figure 7, shows the model's classification results.

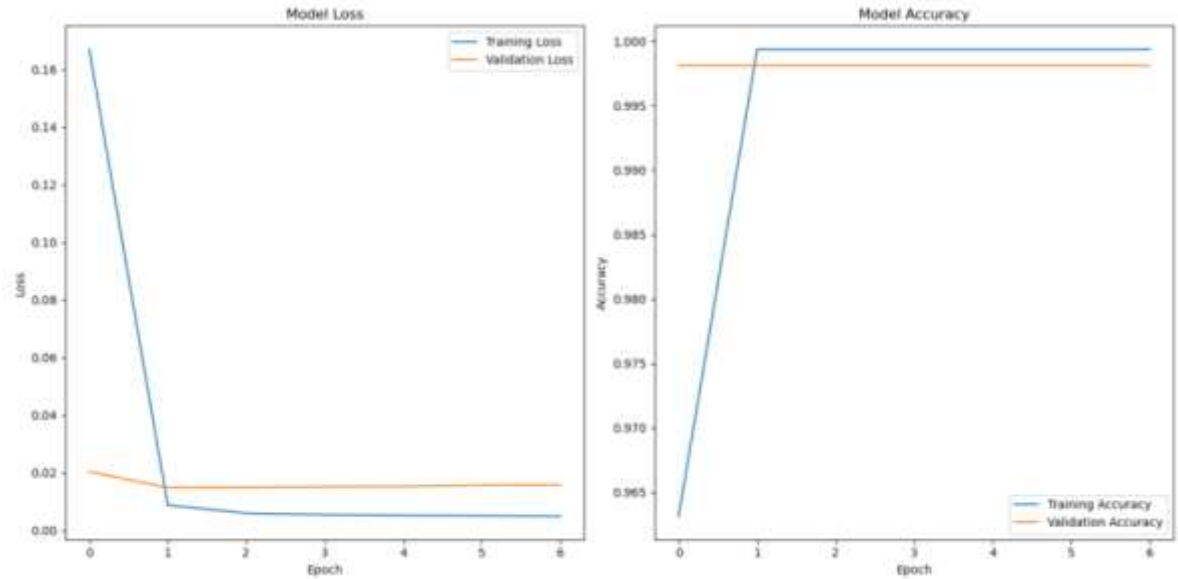

Figure 7. Confusion Matrix Illustrating Model Classification Results.

From the confusion matrix, the true positive values are high, and so are the true negatives, which mean that the model is able to correctly classify a failure from a non-failure. This visual confirmation also enhances the overall authenticity of the proposed model in the current study. Figure 8 illustrates the Receiver Operating Characteristic (ROC) curve, demonstrating the developed model's capacity to diagnose diseases like CAP presented at the emergency department. ROC is an effective plot to look at how good the model is at discriminating between the two groups. The AUC score of 100 percent merely means that the built model has very sharp discrimination power, demonstrating the true positive and false positive rate curve and speaking about the built model generality.

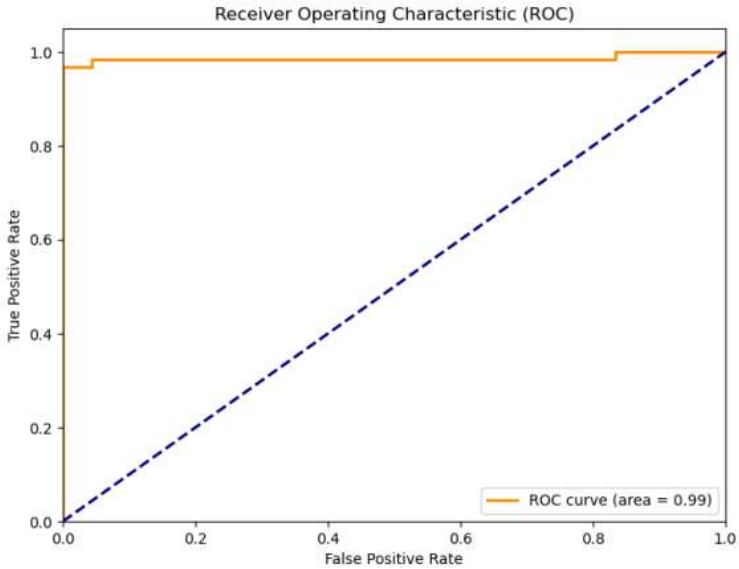

Figure 8. Receiver Operating Characteristic (ROC) Curve and AUC Score.

Moreover, the Precision-Recall Curve, depicted in Figure 9, effectively illustrates the balance between precision and recall rates. The curve reveals that the model maintains a strong equilibrium between these two metrics, which is crucial for applications where both precision and recall are equally important. This balance ensures that the model performs well in accurately identifying positive instances while minimizing false positives.

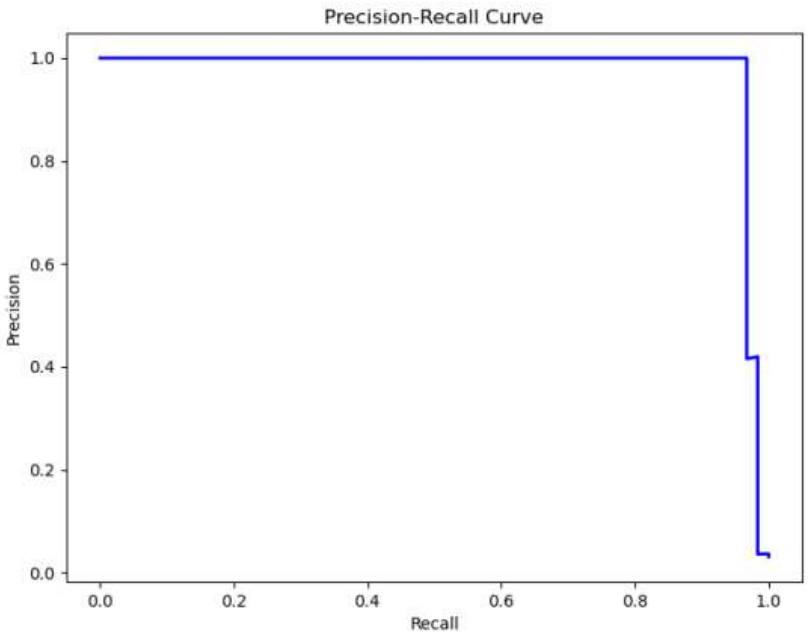

Figure 9. Precision-Recall Curve Analyses.

In the practical application phase, we computed the predicted failure probabilities for all dataset records. We defined records as high-risk if their probability was greater than a certain threshold, set at 0.8. We set the significance level at 0.8 to minimize false negatives and capture the most likely failure cases. We chose this threshold to focus on the cases with high classification confidence and to minimize the number of false negative cases, while keeping the false positive rates to a reasonable level. Choosing the right threshold greatly affects this performance. By raising a bar, the model looks for a more highly probable prognosis; hence, while it may produce fewer samples of high-risk classification, they are likely to be much more accurate. This approach assists in focusing maintenance efforts on the probable failure circumstances, which enhances the use of working hours in maintenance. However, a higher threshold often results in a reduced recall value, as it classifies fewer records as high-risk, potentially overlooking less probable failures. Therefore, adjusting the threshold requires balancing the trade-off between maximizing the number of high-risk cases and reducing false alarms. For this reason, identifying the model's performance in terms of precision or recall, or better yet, the combination of both, provides a balance between performance and timeliness of the results, which is always critical in any practical setup. This classification encourages concrete maintenance action at point 0.8, as shown in Table 5.

Table 5. Detailed Product Data with Failure Probability

Product ID	Air Temperature (K)	Process Temperature(K)	Rotational Speed (rpm)	Torque (Nm)	Tool wear (min)	Machine Failure	TWF	HDF	PWF	OSF	RNF	Failure Probability
1026	298.9	309.1	2861	4.6	143	1	0	0	1	0	0	0.9987
1039	298.9	309.0	1410	65.7	191	1	0	0	1	1	0	0.9999
1046	298.8	308.9	1455	41.3	208	1	1	0	0	0	0	0.9993
1097	298.4	308.2	1282	60.7	216	1	0	0	0	1	0	0.9995

Figure 10 illustrates a scatter plot of high-risk records, with color intensity indicating the predicted failure probabilities. This visualization aids in identifying equipment that requires focused attention based on risk levels, facilitating efficient resource allocation and minimizing.

Discussion:

The evaluation of the model's characteristics and results demonstrates key benefits and suggests possible use in ongoing processes, such as predictive maintenance. The model's high accuracy, enhanced by the ROC and precision-recall graphics, demonstrates its capacity to make precise predictions about potential machine faults. All basic name transformations, such as feature scaling and encoding, were critical in properly preparing the data for training so that the model could learn properly from the input data fed to it. The centralized advantage is the ability to determine the model's accuracy based on the performance indices that characterize the model's effectiveness. The proposed approach's high AUC, along with its high rates of precision and recall, demonstrate its effectiveness in planning and conducting maintenance processes. Effective schedule management and general operational performance depend on failure predictions, as they dictate the downtime during system maintenance.

However, it's crucial to highlight some of the limitations, which include the following: The synthetic characteristics of the dataset, while providing a sample of various failure situations, could not be correlated with the real environment. The failure characteristics and distributions we uncover in the following sections might not align with practical operating conditions, which could lead to ineffective performance of the above model in a synthetic environment. To achieve this, the present study combines a real dataset with a synthetic one to account for the characteristics of real failures and improve the model's accuracy and applicability. It means that using real operating data will help to adjust the described model and make it more accurate in terms of forecasting. Moreover, a score of 0.95 in the recall aspect, calculated as one minus the recall score, is slightly less than ideal and can be further trained to capture less frequent failure events. To address this issue, we can implement preventive measures such as incorporating a variety of failure scenarios, adjusting the decision margins, and utilizing advanced techniques such as ensemble methods. As a result, these approaches can help the model capture and estimate failure cases more accurately, improving its predictability.

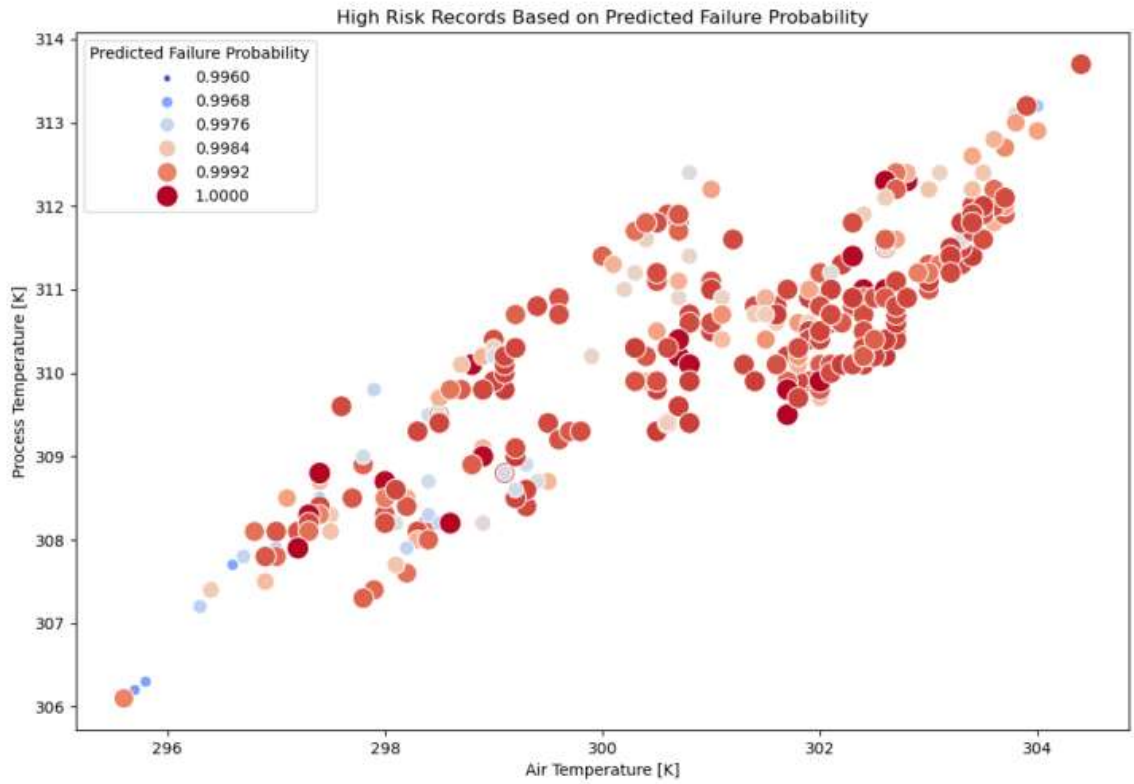

Figure 10. Scatter Plots of High-Risk Records with Predicted Failure Probabilities.

Real-World Implications and Error Analysis:

In the real world, the use of predictive maintenance models that categorize critical components according to the chances of failure is greatly beneficial. It helps organizations move from a reactive paradigm to a predictive one, thus cutting down on costs associated with unplanned equipment breakdowns and increasing the equipment's useful life cycle. By concentrating maintenance efforts on the most crucial areas, organizations can gain guidance, establish appropriate resource priorities, prevent failures, and ultimately decrease their impact on the organization, thereby enhancing organizational effectiveness. However, we should not omit the error analysis, as it aids in refining the presented model. Key issues include:

- **False Positives:** Misclassifying certain components as high-risk when, in reality, they are not, leading to erroneous maintenance procedures. We can minimize this by fine-tuning threshold values and improving the model's overall calibration.
- **False Negatives:** We exacerbate the issue by failing to identify components that fail, thereby missing out on maintenance opportunities. The following can help solve this problem: We can increase the rewards for model recall and include a variety of data.
- **Data Quality:** Biases in the data set, such as incorrect entries or missing values, can significantly impact the model. Critical features such as data cleaning and validation must be followed.
- **Synthetic Data Limitations:** At times, one may argue that the synthetic data does not accurately represent real-life situations. To get the best results, it is also important to assess the model with real operating data.

Conclusion:

The deep learning solution's predictive maintenance model has improved the application's predictive capability and operational performance. Aggressive preprocessing procedures were used during training on the synthetic data set, which contributed to the achievement of genuine indicators within the model. While analyzing the results of the machine failure prediction, an accuracy of 98%, as well as good precision, recall values, and AUC score, demonstrate the model's competency. These outcomes support the model's applicability in realistic maintenance tasks, where failure prognosis is critical to optimizing maintenance intervals and, consequently, the overall mean time to failure. The model provides insights into potential failures of records, enabling maintenance actions to align with the forecasted failure risks. Because the methodology targets records with high failure probabilities, it contributes to maintenance preventative work, resource optimization, and reduction of failure likelihood.

However, there is still opportunity for improved results and additional research, as discussed below. On the downside, the comprehensive nature of the dataset may be synthetic and not closely aligned with real-life operational settings. Future research on this model should also include the creation of assessments for actual datasets to assess the model's generality. We can conduct further research on more complex model variations or combine multiple models (ensemble or hybrid) to enhance their effectiveness. Using domain knowledge and other unconsidered operation characteristics could potentially enhance the model and improve the quality of its predictions.

References:

[1] T. Zhu, Y. Ran, X. Zhou, and Y. Wen, "A Survey of Predictive Maintenance: Systems, Purposes and Approaches," Dec. 2019, Accessed: Aug. 12, 2024. [Online]. Available: https://arxiv.org/abs/1912.07383v2

[2] "(PDF) Intelligent Maintenance Systems and Predictive Manufacturing." Accessed: Aug. 12, 2024. [Online]. Available: https://www.researchgate.net/publication/343121327_Intelligent_Maintenance_Systems_and_Predictive_Manufacturing

[3] O. Asif, S. A. L. I. Haider, S. R. Naqvi, J. F. W. Zaki, K. S. Kwak, and S. M. R. Islam, "A Deep Learning Model for Remaining Useful Life Prediction of Aircraft Turbofan Engine on C-MAPSS Dataset," IEEE Access, vol. 10, pp. 95425–95440, 2022, doi: 10.1109/ACCESS.2022.3203406.

[4] H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang, "Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders," ISA Trans., vol. 77, pp. 167–178, Jun. 2018, doi: 10.1016/J.ISATRA.2018.04.005.

[5] T. Fawcett, "An introduction to ROC analysis," Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/J.PATREC.2005.10.010.

[6] D. Lee and R. Pan, "Predictive maintenance of complex system with multi-level reliability structure," Int. J. Prod. Res., vol. 55, no. 16, pp. 4785–4801, Aug. 2017, doi: 10.1080/00207543.2017.1299947.

[7] M. J. Scott, W. J. C. Verhagen, M. T. Bieber, and P. Marzocca, "A Systematic Literature Review of Predictive Maintenance for Defence Fixed-Wing Aircraft Sustainment and Operations," Sensors 2022, Vol. 22, Page 7070, vol. 22, no. 18, p. 7070, Sep. 2022, doi: 10.3390/S22187070.

[8] M. Sisode and M. Devare, "A Review on Machine Learning Techniques for Predictive Maintenance in Industry 4.0," pp. 774–783, 2023, doi: 10.2991/978-94-6463-136-4_67.

[9] S. Kaparthi and D. Bumblauskas, "Designing predictive maintenance systems using decision tree-based machine learning techniques," Int. J. Qual. Reliab. Manag., vol. 37, no. 4, pp. 659–686, Mar. 2020, doi: 10.1108/IJQRM-04-2019-0131/FULL/XML.

[10] Y. H. Hung, "Improved Ensemble-Learning Algorithm for Predictive Maintenance in the Manufacturing Process," Appl. Sci. 2021, Vol. 11, Page 6832, vol. 11, no. 15, p. 6832, Jul. 2021, doi: 10.3390/APP11156832.

[11] M. Sohaib, S. Mushtaq, and J. Uddin, "Deep Learning for Data-Driven Predictive Maintenance," Intell. Syst. Ref. Libr., vol. 207, pp. 71–95, 2021, doi: 10.1007/978-3-030-75490-7_3.

[12] D. Bruneo and F. De Vita, "On the use of LSTM networks for predictive maintenance in smart industries," Proc. - 2019 IEEE Int. Conf. Smart Comput. SMARTCOMP 2019, pp. 241–248, Jun. 2019, doi: 10.1109/SMARTCOMP.2019.00059.

[13] A. L. Alfeo, M. G. C. A. Cimino, G. Manco, E. Ritacco, and G. Vaglini, "Using an autoencoder in the design of an anomaly detector for smart manufacturing," Pattern Recognit. Lett., vol. 136, pp. 272–278, Aug. 2020, doi: 10.1016/J.PATREC.2020.06.008.

[14] "Advanced Artificial Intelligence Techniques for Real-Time Predictive Maintenance in Industrial IoT Systems: A Comprehensive Analysis and Framework | Journal of AI-Assisted Scientific Discovery." Accessed: Aug. 12, 2024. [Online]. Available: https://scienceacadpress.com/index.php/jaasd/article/view/83

[15] S. Cho et al., "A hybrid machine learning approach for predictive maintenance in smart factories of the future," IFIP Adv. Inf. Commun. Technol., vol. 536, pp. 311–317, 2018, doi: 10.1007/978-3-319-99707-0_39.

[16] S. Sridhar and S. Sanagavarapu, "Handling Data Imbalance in Predictive Maintenance for Machines using SMOTE-based Oversampling," Proc. - 2021 IEEE 13th Int. Conf. Comput. Intell. Commun. Networks, CICN 2021, pp. 44–49, Sep. 2021, doi: 10.1109/CICN51697.2021.9574668.

[17] R. Naren and J. Subhashini, "Comparison of deep learning models for predictive maintenance," IOP Conf. Ser. Mater. Sci. Eng., vol. 912, no. 2, Sep. 2020, doi: 10.1088/1757-899X/912/2/022029.

[18] H. Hesabi, M. Nourelfath, and A. Hajji, "A deep learning predictive model for selective maintenance optimization," Reliab. Eng. Syst. Saf., vol. 219, p. 108191, Mar. 2022, doi: 10.1016/J.RESS.2021.108191.