Journal of Information Systems Engineering and Management
2025, 10(7s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Comparative Analysis of Predictive Monitoring Systems Utilizing Machine Learning and Deep Learning Algorithms

Pranita Bhosale [1, 2], Dr. Sangeeta Jadhav[3]
[1] Research Scholar at Department of Electronics and Telecommunication, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, INDIA
ORCID ID : 0000-0001-6796-0872, pranita.bhosale29@gmail.com,
[2] Faculty at Department of Electronics and Telecommunication, Army Institute of Technology, Dighi Hills, Pune, INDIA
ORCID ID : 0000-0001-6796-0872, pranitatambe@aitpune.edu.in
[3] Head of Department of IT Engineering, Army Institute of Technology, Dighi Hills, Pune,
ORCID ID : 0000-0002-0610-0374, hodit@aitpune.edu.in

ARTICLE INFO

Received: 10 Oct 2024

Revised: 09 Dec 2024

Accepted: 26 Dec 2024

ABSTRACT

This study offers an in-depth comparative examination of predictive monitoring systems utilizing sophisticated machine learning (ML) and deep learning (DL) algorithms. The investigation delves into the efficacy, advantages, and constraints of ML algorithms, exemplified by Random Forest and Extreme Gradient Boosting will be contrasted with Deep Learning (DL) algorithms including artificial neural networks (ANNs) and LSTM, recurrent neural networks (RNNs) in this study. By evaluating these algorithms across diverse domains, the research aims to discern optimal strategies for predictive monitoring, considering factors like efficiency, real-time processing, and adaptability. The findings contribute valuable insights for practitioners and researchers, informing the selection and deployment of algorithms in predictive monitoring systems

Keywords: Predictive Maintenance (PdM), Machine Learning (ML), Deep Learning (DL), Comparative examination, Sample Dataset

1. INTRODUCTION

People always face such kind of questions: In a machine, what is the Maintenance and What kind of Maintenance are available to us? Can I just use and throw the Machine? Or Do I need to invest to keep it running? When is Machine going to fail? I have maintained Machine, it is running nice but still I would like to know, how long it's going to live? [5] These are the serious issues and people can't afford to lose costly machinery. Maintenance will be directly proportional to the criticality of the equipment to the strategic importance of that machinery. Maintenance comprises systematic activities conducted to ensure an item remains in its optimal operational state. So there are techniques which will let us know like which kind of maintenance to be use. Once we diagnose the fault in machine, people are always interested in to know the Remaining useful life (RUL) [19] [37] of the Machine. Predictive monitoring systems play a crucial role in various domains due to their ability to anticipate issues, optimize processes, and enhance decision-making. Here are some key aspects highlighting the importance of predictive monitoring systems across different domains [39]:

Healthcare: The ongoing surveillance of patients' health parameters facilitates the early identification of abnormalities, thereby mitigating the risk of complications.

Manufacturing and Industry: Anticipating equipment failures through monitoring helps prevent unplanned downtime, reduces maintenance costs, and improves overall operational efficiency. Monitoring production processes in real-time allows for the early identification of defects, ensuring the production of high- quality goods. [13] [23]

Finance: In real-time, predictive monitoring systems can examine patterns and anomalies within financial transactions to identify and detect fraudulent activities.

Market Analysis: Forecasting market trends and analyzing financial data enables better decision- making for investment and risk management.

Information Technology: Monitoring IT infrastructure can predict potential system failures, ensuring optimal performance and minimizing downtime.

Transportation and Logistics: Predictive monitoring in vehicle maintenance enhances fleet efficiency by forecasting maintenance needs, reducing breakdowns, while in traffic management, it optimizes flow, mitigates congestion, and enhances overall transportation infrastructure.

Energy Management: Predictive monitoring contributes to power grid optimization by anticipating demand fluctuations, ensuring efficient energy distribution, and minimizing the risk of power outages; concurrently, it enhances equipment efficiency through predictive maintenance, optimizing energy consumption, and reducing operational costs.

Environmental Monitoring: Predictive monitoring systems analyze environmental data for both natural disaster predictions, including hurricanes, earthquakes, and floods, and contribute to climate change analysis by tracking environmental parameters over time, enhancing the understanding of climate patterns.

Retail and E-commerce: Predictive monitoring leverages customer behavior analysis for demand forecasting, optimizing inventory management to reduce stock outs, and utilizes monitored customer preferences to enable businesses in crafting targeted and personalized marketing strategies.

Hence, predictive monitoring systems provide valuable insights, enhance operational efficiency, and contribute to informed decision-making across a various domains, ultimately resulting in enhanced outcomes and optimize resource utilization.[9] [15][21][38]

2. LITERATURE SURVEY

Numerous researchers have established the pivotal role of maintenance in enhancing production performance. [8][11][12][23] As technological advancements continue to reshape factory landscapes, corresponding maintenance methods evolve to meet the changing demands of manufacturers. The advent of Industry 4.0 necessitates the development of new maintenance techniques, termed Maintenance 4.0 [1] [29], aligning with the paradigm shift in manufacturing.

[20] [31] Manufacturing industries hold a significant position in national economies, emphasizing the critical importance of optimizing the use of all processing equipment. The challenge lies in synchronizing activities, operations, and equipment utilization to minimize losses and defects, which, in turn, drive investments in the maintenance sector. Traditionally, the concept of maintenance has been limited to the repair of equipment, focusing on preventive, predictive, and corrective measures. [15][27] Yet, within the context of Industry 4.0, the complete value chain seamlessly integrates, exchanging digitized information to facilitate collaborative task execution and generating extensive datasets.

[27] The massive data generated in Industry 4.0 environments presents an opportunity for more accurate problem detection, root-cause analysis, damage prediction, effect assessment, and reliable maintenance planning. Critical elements for Maintenance 4.0 include comprehensive data coverage, high quality, and efficient utilization of data, yet addressing the challenges of handling such extensive datasets and developing tools for transforming data into actionable information poses a considerable hurdle. [1] [6] [29] the goals of Industry 4.0 are emphasized by the need for reduced time-to-market, personalized mass production, and enhanced efficiency. To sustain Industry 4.0's success, Maintenance 4.0 must exhibit rapid responsiveness to dynamic changes in operating conditions, Sustain machine quality at an economical rate to boost profitability in both maintenance and production processes, ultimately attaining superior performance in production machinery [36][38][40].

The need for advanced algorithms, particularly Machine Learning (ML) and Deep Learning (DL) [10][26][39], in enhancing predictive monitoring arises from their ability to handle complex patterns, large datasets, and nonlinear relationships more effectively than traditional methods. Here are key reasons why ML and DL are crucial in the context of predictive monitoring:

1. Pattern Recognition: ML: Machine learning algorithms are highly effective at recognizing patterns within data, enabling the recognition of subtle trends or anomalies that may be indicative of future events. DL: Deep learning, with its neural networks, can automatically learn hierarchical representations of data, capturing intricate patterns that may be challenging for traditional algorithms [32].

2. Handling Large Datasets: ML: ML algorithms can process and analyze extensive datasets with efficiency, which is common in predictive monitoring where numerous variables and parameters are involved. DL: Deep learning models, especially deep neural networks, are adept at handling massive datasets and extracting meaningful features, making them suitable for scenarios with high-dimensional input data.

3. Nonlinear Relationships: ML: Machine learning models can capture nonlinear relationships between variables, providing more accurate representations of complex systems compared to linear models. DL: Deep learning, with its ability to model intricate nonlinear relationships, is particularly effective when dealing with complex, interconnected features in data.

4. Adaptability and Learning: ML: Adapting to dynamic conditions and learning from fresh data, machine learning algorithms enable predictive monitoring systems to progressively enhance their accuracy over time. DL: Deep learning models, especially in reinforcement learning scenarios, can dynamically adjust their behavior based on feedback, making them suitable for environments with evolving patterns.

5. Feature Extraction: ML: Machine learning algorithms frequently necessitate manual feature engineering, where domain experts define pertinent features. Nevertheless, certain ML algorithms can autonomously select features. DL: In deep learning models, pertinent features can be automatically learned from raw data, alleviating the necessity for extensive manual feature engineering in numerous instances [34].

6. Complex Decision-Making: ML: Machine learning algorithms can make complex decisions based on learned patterns, making them suitable for predictive monitoring tasks that involve multifaceted decision- making processes. DL: Deep learning models, with their deep neural architectures, can learn hierarchical representations that enable them to make intricate decisions, especially in tasks with high-level abstractions.

7. Real-time Processing: ML: Many machine learning models are capable of real-time processing, allowing for timely predictions in dynamic and fast-changing environments. DL: Deep learning models can be optimized for efficient real-time processing; making them suitable for applications where quick responses are crucial. [14] [17]

In summary, the need for advanced algorithms like ML and DL in predictive monitoring stems from their ability to handle complex patterns, large datasets, nonlinear relationships, and adapt to changing conditions. These technologies empower predictive monitoring systems to provide more accurate, timely, and insightful predictions across a variety of domains.

3. SAMPLE DATA FROM MANUFACTURING INDUSTRY

This time utilizing sample data, an attempt was made to develop machine learning and deep learning algorithms. The sample data has been taken from kaggle official website [41]. The provided sample data file comprises three primary categories of information:

1. Timestamp data, including both the date and time [33].
2. Sensor data, consisting of 52 series of raw values.
3. Machine status, serving as the target label indicating the predicted occurrence of failure.

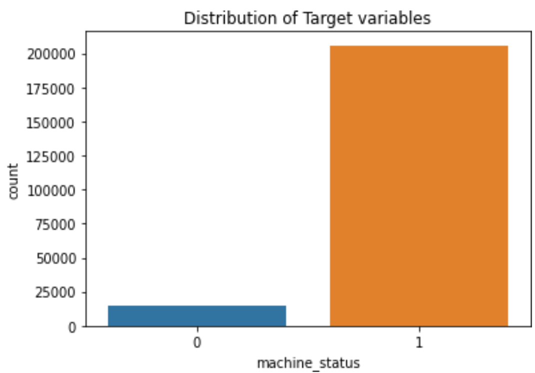

Fig. 1. Distribution of Variables

Sr. No.	Normal	Broken	Recovering
1	205836	07	14477
		14484	
Status	Normal	Worst	

Fig. 2. Distribution of Variables

This research utilizes a dataset spanning five months. Through a time-based segmentation approach, the dataset was divided into training and testing subsets. The methodology adopted for this partitioning is as follows:

a. Three months of historical data, specifically the 4^{th}, 5^{th}, and 6^{th} months, were utilized as the training dataset.

b. The remaining month, i.e., the 7^{th} month was reserved exclusively for testing purposes.

To provide a succinct overview of the dataset structure, Table 2 presents a snapshot of the data as it appears in the Excel format. The initial column denotes timestamps, depicting the date and minute-wise recording intervals. Subsequent columns encompass the sensor readings, ranging from Sensor 00 to Sensor 51, constituting a total of 52 sensors. The final column encapsulates the machine status information. This meticulously collected dataset serves as the foundational resource for evaluation of predictive maintenance algorithms and for the development.

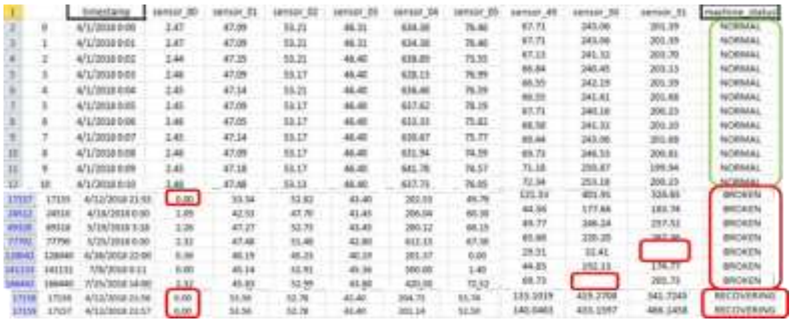

Fig. 3. Sample dataset

Deep learning, a subset of machine learning, is derived from ANN Artificial Neural Networks and is characterized by multiple nonlinear processing layers. Its primary objective is to learn hierarchical representations of data. The field of deep learning is rapidly evolving, with various architectures continually being developed. The community is highly collaborative, offering numerous high-quality tutorials and books.

[14] [21] [25] [35] Therefore, this summary offers a concise overview of prominent deep learning methods employed in machine health monitoring. [8] It specifically examines four deep architectures such as Auto-encoders etc. along with their respective variations.

[1] [9] [16] [36] Researchers have built predictive maintenance algorithms using Machine Learning. For testing purpose two machine learning algorithms have built just to compare the results with the deep learning algorithms.

The chosen MACHINE LEARNING algorithms are:

a. Random Forest Classifier (RF)

b. Extreme Gradient Boost Classifier (XGBoost)

The chosen DEEP LEARNING algorithms are:

a. Artificial Neural Network (ANN)

b. RNN-Longest short term Memory (LSTM)

4. PREDICTIVE MONITORING SYSTEMS USING MACHINE LEARNING

4.1. Mostly used ML Algorithms: 1) Random Forest Classifier

Accurately categorizing observations holds significant importance across a spectrum of business endeavors, ranging from predicting individual user purchase behavior to forecasting loan default likelihood. In the field of DS, data science, there exists a wide array of algorithms, encompassing SVM, NBC, DT and LR. Among these options, the random forest classifier stands out as one of the most prominent methods at the apex of the classifier hierarchy.

[16] Random forest, as implied by its name, comprises a multitude of individual classification/regression trees functioning collectively. In the random forest every individual tree has ability to produce a class prediction, and then it dictates the model's final prediction. The principle behind this algorithm is straightforward yet sturdy.

A collective of relatively uncorrelated models tends to outperform any individual model. The essential aspect here is the minimal correlation between models. This phenomenon arises as the trees safeguard each other from errors, provided they do not consistently commit errors in the same direction. While certain trees might produce inaccurate predictions, many others will provide accurate predictions, enabling the collective movement of the group of trees in the right direction.

To ensure the effectiveness of random forest, specific conditions need to be fulfilled:

1. The features must contain substantial signal so that models built upon them perform superior to random guessing.

2. The predictions made by individual trees, and consequently their errors, should exhibit low correlations among themselves. Random forest utilizes bagging and feature randomness during the construction of each tree to generate a diverse forest, resulting in a collective prediction that surpasses the accuracy of any single tree.

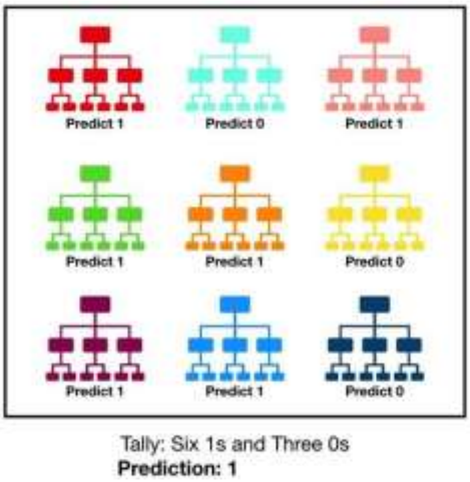

Fig. 4. Illustration of a Random Forest Model Predicting

4.2. Mostly used ML Algorithms: 2) Extreme Gradient Boosting Classifier (XGBoost)

This is machine learning algorithm, that's like a team of experts working together to make predictions. Each expert is like a small decision maker called a decision tree. XGBoost combines the predictions of many decision trees to make more accurate predictions. It's called "extreme" because it works

Really well and is often used in competitions where accuracy is crucial.

[16] Extreme Gradient Boosting is a supervised tree-based algorithm mainly used for classification tasks in Machine Learning. In contrast to traditional Gradient Boosting, XGBoost employs its own approach to constructing trees, wherein the Similarity Score and Gain are used to determine the optimal splits for nodes.

Basically, XGBoost is a supervised Machine Learning algorithm based on trees. While it can tackle classification and regression problems, the focus here is on its application for classification. Unlike traditional Gradient Boosting methods, XGBoost has its own way of constructing trees, using factors like Similarity Score and Gain to determine the best node splits. XGBoost is known for its efficiency and effectiveness, making it a popular choice in the machine learning community.

$$Similarity\ Score = (\sum n\ (Residual)^2 \ / \sum_{i=1} n\ [\text{Previous Probability} * (1 - \text{Previous Probability})] + \lambda$$

Gradient boosting, the broader category XGBoost belongs to, involves ensemble algorithms for classification or regression tasks. It utilizes decision tree models, gradually adding trees to the ensemble and adjusting them to correct prediction errors from previous models. This boosting technique involves training models using various loss functions and gradient descent optimization, which gives gradient boosting its name.

XGBoost stands out for its dominance in competitions like Kaggle. It builds decision trees sequentially, with each tree using weights assigned to independent variables for making predictions. If a variable is predicted incorrectly, its weight is increased, and the variable is passed to the next tree. These individual trees' predictions are combined, to form a robust and accurate model.

Overall, XGBoost is a versatile algorithm capable of handling various types of problems, including regression, classification, ranking, and custom prediction tasks. Its efficient implementation and ability to produce highly accurate models have made it a go-to choice for many machine learning practitioners.

5. PREDICTIVE MONITORING SYSTEMS USING DEEP LEARNING

5.1. Mostly used DL Algorithms: 1) ANN

Artificial Neural Network (ANN) is a computer program designed to mimic the functionality of human brain, with interconnected nodes called neurons processing information in layers to generate outputs based on inputs. To illustrate, consider teaching a computer to recognize different fruits from images. These images are converted into data fragments, such as pixels, and passed through the neural network where neurons analyze patterns in the data.

[7] The ANN learns by adjusting the connections between neurons based on the patterns identified in the input data. As it encounters more examples and refines its connections, it improves its ability to recognize fruits accurately. In essence, an ANN is akin to virtual brain learning from examples to perform tasks like image recognition, outcome prediction, and decision-making.

As a vital component of Artificial Intelligence (AI), an ANN aims to replicate the complex network of interconnected neurons which found in the human brain. By programming computers to emulate the behavior of brain cells, ANNs enable computers to comprehend information and make decisions similarly to humans. The structure of an artificial neural network typically comprises three layers: input, hidden, and output.

In an ANN, inputs undergo processing through a weighted sum calculation, including a bias, followed by activation through a transfer function. The resulting output is determined by activation functions, which decide whether a node should be activated or not, with activated nodes contributing to the output layer. Various activation functions are available for implementation, allowing flexibility in modeling different types of data and tasks.

$\sum_{i=1}^{n} Wi * Xi + b$ Where Wi and Xi are the weights and inputs, respectively, and b is the bias.

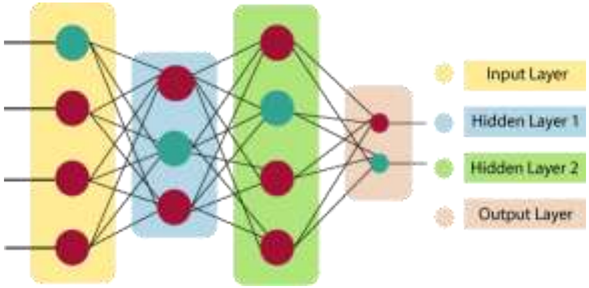

Fig. 5. Artificial Neural network structure

5.2. Mostly used DL Algorithms: 2) RNN- LSTM

Longest Short Term Memory networks, commonly referred to as LSTMs, represent a specialized form of RNNs designed specifically to capture long-term dependencies. [28] Initially proposed by Hoch Reiter Schmidhuber in 1997, LSTMs have undergone further refinement and widespread adoption. LSTMs are specifically crafted to overcome the hurdle of long-term dependency. Help in making them adept at retaining information over extended periods without encountering difficulty.

At the core of LSTM architecture lies a sequence of repeating neural network modules. While traditional RNNs feature simplistic structures like a single tanh layer within these modules, LSTMs integrate more sophisticated elements. These elements include feedback connections. The weight matrix W assigns different weights to two parameters: the current input vector along with the previous hidden state for every gate. Similar to conventional RNNs, LSTMs generate an output at each time step, and this output is subsequently utilized for training the network via gradient descent. [2] [28]

The equation is represented as:

$$ft = \sigma(wf.[ht - 1, x1] + bf) \quad (2)$$

$$it = \sigma(wi.[ht - 1, xt] + bi) \quad (3)$$

$$\mathrm{C}t = tanh\,(wc.[ht - 1, xt] + bc) \quad (4)$$

$$\mathrm{C}t = ft * Ct - 1 + it * Ct \quad (5)$$

The LSTM process begins by deciding which forget the cell state information. The forget gate layer analyzes ht1 and xt. It produces outputs within the range of 0 to 1. This is indicating whether to retain or discard each element. Subsequently, the focus shifts to determining the new information to integrate into the cell state. This entails two stages: the “input gate layer” identifies values for updating, followed by the generation of a vector of potential new values, C t, by a tanh layer. These components are then amalgamated to formulate an update to the state.

Transitioning from previous cell to new cell that is C_{t1} to C_t is the subsequent task. Guided by the directives established in prior steps, this transition involves discarding designated elements from the old state and incorporating new candidate values, scaled based on the degree of update determined for each state value. Finally, the determined output is completely based on the filtered cell state. A sigmoid layer identifies segments of the cell state for output; these are then passed through a tanh transformation to constrain the values within the range of -1 to 1. Multiplying the output of the sigmoid gate ensures that only the designated segments are output [32].

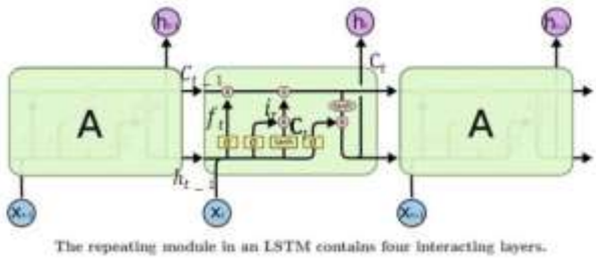

Fig. 6. The LSTM's repeating module consists of four interconnected layers

6. COMARATIVE ANALYSIS

6.1. Execution steps for Random Forest Classifier

- Import the Random Forest classifier from sklearn
- Initialize the classifier object with default parameters
- Train the object with X-train & y-train
- Predict with x-test data
- Generate metrics

6.2. Execution steps for XGBoost Classifier

- Import the XGBoost classifier from sklearn
- Initialize the classifier object with default parameters
- Train the object with X-train & y-train
- Predict with x-test data
- Generate metrics

6.3. Execution steps for ANN

- It's a 4 layer architecture
- 1st layer is fully connected dense layer with 32 neurons, Activation function as Relu
- 2nd layer is dropout layer
- 3rd layer is again a fully connected layer with 8 Neurons, Activation function as Relu
- 4th layer is dense layer with 1 neuron (Activation as Sigmoid)
- Compiled it using loss function as binary-crossentropy with optimizer as adam
- Trained this architecture with x-trian and y-train
- Predicted it on x-test
- Generate the metrics

6.4. Execution steps for LSTM

- It's a 2 layer architecture
- 1st layer is with 100 neurons
- 2nd layer is with 1 neuron with Sigmoid as Activation function
- Compiled it using loss function as binary-crossentropy with optimizer as adam
- Trained this architecture with x-trian and y-train
- Predicted it on x-test
- Generate the metrics

```
[24] print(f1_score(y_test,pred ,average="macro"))

     0.9893832335753892

[25] print(classification_report(y_test,pred))

                   precision    recall  f1-score   support

               0       0.97      0.99      0.98      5505
               1       1.00      1.00      1.00     38832

        accuracy                           1.00     44337
       macro avg       0.98      0.99      0.99     44337
    weighted avg       1.00      1.00      1.00     44337
```

Fig. 7. Performance metrics of RF Classifier

```
pred=xg.predict(x_test)
print(f1_score(y_test,pred ,average="macro"))
print(classification_report(y_test,pred))

0.991024015848327
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      5505
           1       1.00      1.00      1.00     38832

    accuracy                           1.00     44337
   macro avg       0.99      0.99      0.99     44337
weighted avg       1.00      1.00      1.00     44337
```

Fig. 8. Performance metrics of XGBoost Classifier

```
print("The F1 Score is ",f1_score(y_test,prediction ,average="macro"))
print(classification_report(y_test,prediction))

The F1 Score is  0.9913469231971253
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      5505
           1       1.00      1.00      1.00     38832

    accuracy                           1.00     44337
   macro avg       0.99      0.99      0.99     44337
weighted avg       1.00      1.00      1.00     44337
```

Figure 9. Performance metrics of ANN Classifier

```
print("The F1 Score is ",f1_score(y_test,prediction ,average="macro"))
print(classification_report(y_test,prediction))

The F1 Score is  0.8918128191929526
              precision    recall  f1-score   support

           0       0.97      0.69      0.81      5505
           1       0.96      1.00      0.98     38832

    accuracy                           0.96     44337
   macro avg       0.96      0.84      0.89     44337
weighted avg       0.96      0.96      0.96     44337
```

Fig. 10. Performance metrics of RNN LSTM Classifier

7. DISCUSSION

[26] When assessing the effectiveness of a machine learning or deep learning model, it's crucial to move beyond the simplicity of mere code execution and delve into the realm of model evaluation. Typically, the evaluation process involves the utilization of predefined metrics selected by the practitioner. In the context of classification models, the evaluation often revolves around metrics derived from a confusion matrix, which serves as a fundamental tool not only for assessing model performance but also for monitoring and managing models.

A confusion matrix, alternatively known as an error matrix, stands as a concise tabular representation employed to gauge the efficacy of a classification model. It encapsulates both the correct and incorrect predictions made by the model, providing a breakdown of counts for each class. Visualizing a 2x2 confusion matrix reveals four distinct quadrants, each offering insights into the model's predictive capabilities.

For instance, in the scenario where a classification model correctly predicts 'Yes' in ten instances where the actual value is indeed 'Yes,' these ten instances are recorded in the True Positive quadrant, situated in upper-left corner of the confusion matrix (CM). This delineates a True Positive prediction, representing cases where the model correctly identifies positive instances, contributing to an understanding of key terms crucial for model evaluation.

In essence, the confusion matrix serves as a cornerstone in the assessment of classification models, offering a comprehensive breakdown of predictive outcomes that enables practitioners to discern the model's strengths and weaknesses across different classes. Its utility extends beyond mere evaluation, encompassing model monitoring and management, thereby facilitating informed decision-making in the development and deployment of ML and DL models.

The 2x2 Confusion Matrix depicted in Figure 12 illustrates four crucial terms we need to grasp TN, TP, FN, and FP:

	0	1
0	TN	FN
1	FP	TP

Fig. 11. 2x2 Confusion Matrix

• True Positive (TP): In this case, the model accurately predicts a positive outcome. For instance, when the model accurately recognizes an image as a dog, and it truly is a dog.

• True Negative (TN): In this case, the model accurately predicts a negative outcome. An example would be when the model correctly identifies an image as not being a dog when it's actually not.

• False Positive (FP): This is termed a type 1 error, this happens when the model predicts incorrectly with positive outcome. For instance, the model may wrongly classify an image as a dog when it's actually not.

• False Negative (FN): This is termed a type 2 error and arises when the model incorrectly predicts a negative outcome. An example would be when the model fails to recognize an image as a dog when it actually is.

In simpler terms, True Positive and True Negative denote accurate predictions, while False Positive and False Negative indicate errors in the model's predictions.

1) Performance Metrics:

- **Accuracy:** This indicates the rate of accurate predictions out of the total predictions.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \qquad (6)$$

- **Precision:** Precision is focuses on the accuracy of positive predictions that it concludes how many were actually correct out of all the items the model predicted as positive.

$$Precision = \frac{True\ Positives}{True\ Positives+False\ Positives} \qquad (7)$$

- **Recall:** Recall is able to find all the positive instances that it tells how many did the model manage to identify correctly.

$$Recall = \frac{True\ Positives}{True\ Positives+False\ Positives} \qquad (8)$$

- **F1 Score:** This combines aspects of precision and recall, giving us single number to gauge how well our model performs overall. It's like combining the accuracy of precision with the completeness of recall to get a balanced view of the model's effectiveness.

$$F1\ Score\ = 2 * \frac{Precision*Recall}{Precision+Recall} \qquad (9)$$

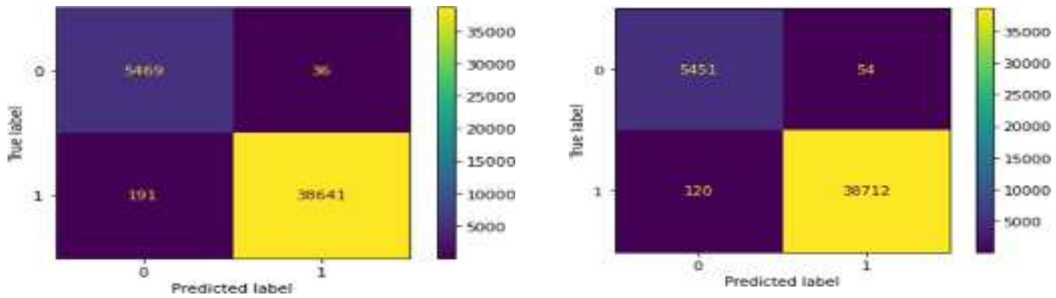

Fig. 12. RF & XGBoost Confusion Matrix

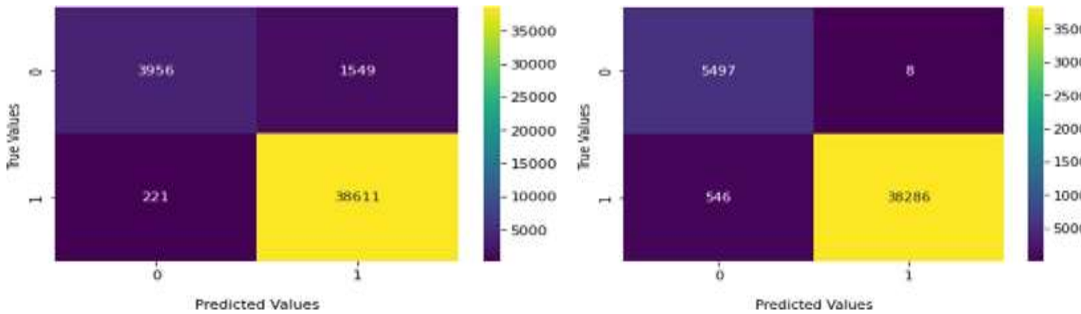

Fig. 13. ANN & LSTM Confusion Matrix

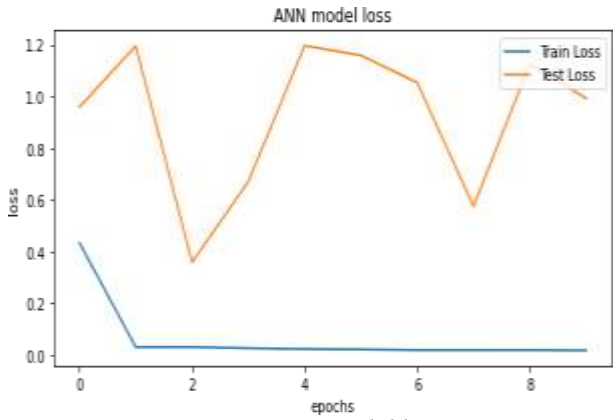

Fig. 14. ANN model loss

Binary cross entropy (BCE) is used, loss function for classification problem.

$$CE = -\sum_{i}^{C} ti \log Si \quad (10)$$

$$BCE = -[t1\, logs1 - (1 - t1)\log(1 - s1)] \quad (11)$$

Where, ti is true class distribution Si is predicted class distribution t1 is train loss and s1 is test loss

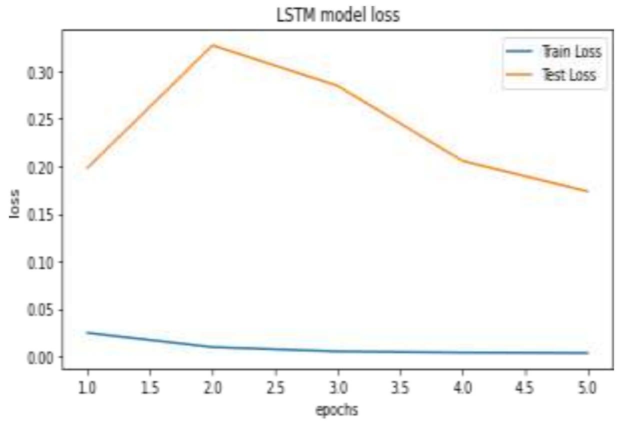

Fig. 15. LSTM model loss

Table 1. Comapative Results

		Machine Learning		Deep Learning	
Parameter	Model	RF	XGBoost	ANN	LSTM
False Negative Rate (FNR)		36	54	1549	8
F1 Score		98.8	99.10	89.73	99.13
Precision	0	97	98	95	98
	1	100	100	96	100
Recall	0	99	99	72	99
	1	100	100	99	100
Accuracy		99	100	96	100

[3] [4] [22] [24] [30] if we compare the results LSTM works better than all among tested algorithms. Throughout all leaning algorithms, FNR is very vey less even F1 score is found to be highest among others.

8. CONCLUSION

The research has demonstrated the superior performance of LSTM in predictive monitoring systems compared to other machine learning and deep learning algorithms. With significantly lower false negative rates and the highest F1 score among tested models, LSTM emerges as a promising choice for practitioners and researchers in this field. These findings underscore the importance of considering algorithm selection carefully, with an emphasis on efficiency, real-time processing, and adaptability. Moving forward, future research could focus on further enhancing LSTM-based monitoring systems, exploring novel approaches to improve performance and scalability, and investigating the integration of multiple algorithms to enhance predictive accuracy and reliability.

REFERENCES

Journal:

[1] Angelos Angelopoulos, Emmanouel T. Michailidis, Nikolaos Nomikos, Panagiotis Trakadas, Antonis Hatziefremidis, Stamatis Voliotis, and Theodore Zahariadis. Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. Sensors, 20(1), 2020.

[2] Homa Arab, Iman Ghaffari, Romaric Mvone Evina, Serioja Ovidiu Tatu, and Steven Dufour. A hybrid lstm-resnet deep neural network for noise reduction and classification of v-band receiver signals. IEEE Access, 10:14797–14806, 2022.

[3] A. Bala, I. Ismail, R. Ibrahim, S. M. Sait, and D. Oliva. An improved grasshopper optimization algorithm based echo state network for predicting faults in airplane engines. IEEE Access, 8:159773–159789, 2020.

[4] Sergei Basovich and Shai Arogeti. Identification and robustcontrol for regenerative chatter in internal turning with simultaneous compensation of machining error. Mechanical Systems and Signal Processing, 149:107208, 2021.

[5] Javaid Butt. A conceptual framework to support digital transformation in manufacturing using an integrated business process management approach. Designs, 4(3), 2020.

[6] L. Stergioulas C. J. Turner, J. Oyekan and D. Griffin. Utilizing industry 4.0 on the construction site: Challenges and opportunities. IEEE Transactions on Industrial Informatics, 17(2):746–756, 2021.

[7] Guo G. Wang WN Chen, J.C. Artificial neural network-based online defect detection system with in-mold temperature and pressure sensors for high precision injection molding. The International Journal of Advanced Manufacturing Technology, 110:2023–2033, 2020.

[8] G. Dagnaw. Artificial intelligence towards future industrial opportunities and challenges. 2020.

9] Jovani Dalzochio, Rafael Kunst, Edison Pignaton, Alecio Binotto, Srijnan Sanyal, Jose Favilla, and Jorge Barbosa. Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. Computers in Industry, 123:103298, 2020.

10] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. An introductory review of deep learning for prediction models with big data. Frontiers in Artificial Intelligence, 3, 2020.

[11] Jonnro Erasmus, Irene Vanderfeesten, Konstantinos Traganos, Ruud Keulen, and Paul Grefen. The horse project: The application of business process management for flexibility in smart manufacturing. Applied Sciences, 10(12), 2020.

[12] A. Essien and C. Giannetti. A deep learning model for smart manufacturing using convolutional lstm neural network autoencoders. IEEE Transactions on Industrial Informatics, 16(9):6069–6078, 2020.

[13] Laurie Hughes, Yogesh K. Dwivedi, Nripendra P. Rana, Michael D. Williams, and Vishnupriya Raghavan. Perspectives on the future of manufacturing within the industry 4.0 era. Production Planning & Control, 0(0):1–21, 2020.

[14] Zhenchun Wei Xu Ding Lei Shi Juan Xu, Pengfei Xu. Dc-nnmn: Across components fault diagnosis based on deep few-shot learning. Shock and Vibration, 0:11, 2020.

[15] Christian Krupitzer, Tim Wagenhals, Marwin Z¨ufle, Veronika Lesch, Dominik Sch¨afer, Amin Mozaffarin, Janick Edinger, Christian Becker, and Samuel Kounev. A survey on predictive maintenance for industry 4.0, 2020.

[16] Yaguo Lei, Bin Yang, Xinwei Jiang, Feng Jia, Naipeng Li, and Asoke K. Nandi. Applications of machine learning to machine fault diagnosis: A review and roadmap. Mechanical Systems and Signal Processing,138:106587, 2020.

[17] Jacob Lohmer and Rainer Lasch. Blockchain in operations management and manufacturing: Potential and barriers. Computers Industrial Engineering, 149:106789, 2020.

[18] Y. Ma, D. Oslebo, A. Maqsood, and K. Corzine. Dc fault detection and pulsed load monitoring using wavelet transform-fed lstm autoencoders.IEEE Journal of Emerging and Selected Topics in Power Electronics, pages 1–1, 2020.

[19] Ryoji Tani Hiroki Fujiwara Masayuki Numao Ken-ichi Fukui Masashi Kitai, Yoshinobu Akamatsu. Remaining useful life curve prediction of rolling bearings under defect progression based on hierarchical bayesian regression. 325:2986 – 2992, 2020.

[20] Ishan Mistry, Sudeep Tanwar, Sudhanshu Tyagi, and Neeraj Kumar. Blockchain for 5g-enabled iot for industrial automation: A systematic review, solutions, and challenges. Mechanical Systems and Signal Processing, 135:106382, 2020.

[21] Srikanth Namuduri, Barath Narayanan Narayanan, Venkata Salini Priyamvada Davuluru, Lamar Burton, and Shekhar Bhansali. Review—deep learning methods for sensor based predictive maintenance and future perspectives for electrochemical sensors. Journal of The Electrochemical Society, 167(3):037552, jan 2020.

[22] Haiyang Pan, Jinde Zheng, Yu Yang, and Junsheng Cheng. Nonlinear sparse mode decomposition and its application in planetary gearbox fault diagnosis. Mechanism and Machine Theory, 155:104082, 2021.

[23] Sudip Phuyal, Diwakar Bista, and Rabindra Bista. Challenges, opportunities and future directions of smart

manufacturing: A state of art review. Sustainable Futures, 2:100023, 2020.
[24] Jamal S. Rahhal and Dia Abualnadi. Iot based predictive maintenance using lstm rnn estimator. In 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pages 1–5, 2020.
[25] Zugasti E. Rodriguez J. et al. Serradilla, O. Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects. Appl Intell 52, page 10934–10964, 2022.
[26] T. A. Shifat and J. W. Hur. An effective stator fault diagnosis framework of bldc motor based on vibration and current signals. IEEE Access, 8:106968–106981, 2020.
[27] Mushtaq S. Uddin J. Sohaib, M. Deep learning for data-driven predictive maintenance. in: Ahad, m.a.r., inoue, a. (eds) vision, sensing and analytics: Integrative. Intelligent Systems Reference Library,Springer, Cham, 207:1–19, 2021.
[28] Christian Vennerød, Adrian Kjærran, and Erling Bugge. Long short-term memory rnn, 05 2021.
[29] K. Vinitha, R. Ambrose Prabhu, Radhika Bhaskar, and R. Hariharan. Review on industrial mathematics and materials at industry 1.0 to industry 4.0. Materials Today: Proceedings, 33:3956–3960, 2020. International Conference on Nanotechnology: Ideas, Innovation and Industries.
[30] Q. Wang, S. Bu, and Z. He. Achieving predictive and proactive maintenance for high-speed railway power equipment with lstm- rnn. IEEE Transactions on Industrial Informatics, 16(10):6509– 6517, 2020.
[31] Zhibin Zhao Jinxin Liu Chuang Sun Ruqiang Yan Weixin Xu, Huihui Miao. Multi-scale convolutional gated recurrent unit networks for tool wear prediction in smart manufacturing. Research Square, 0:0, 2020.
[32] Huang Aihua Sutherland John W. Wu, Haiyue. Layer-wise relevance propagation for interpreting lstm-rnn decisions in predictive maintenance. In The International Journal of Advanced Manufacturing Technology, volume 118, pages 963–978, 2022.
[33] Kelin Wu, Yun Xing, Ning Chu, Peng Wu, Linlin Cao, and Dazhuan Wu. A carrier wave extraction method for cavitation characterization based on time synchronous average and time- frequency analysis. Journal of Sound and Vibration, 489:115682, 2020.
[34] Q. Yang, T. Liu, X. Wu, and Y. Deng. Gear backlash detection and evaluation based on current characteristic extraction and selection. IEEE Access, 8:107161–107176, 2020.
[35] Zhibin Zhao, Tianfu Li, Jingyao Wu, Chuang Sun, Shibin Wang, Ruqiang Yan, and Xuefeng Chen. Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study. ISA Transactions, 107:224–255, Dec 2020.
[36] Zeki Murat C¸ ınar, Abubakar Abdussalam Nuhu, Qasim Zeeshan, Orhan Korhan, Mohammed Asmael, and Babak Safaei. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability, 12(19), 2020.
Conference:
[37] Nicolas Meier, Jan Papadoudis, and Anthimos Georgiadis. Prediction of the tool change point in a polishing process using a modular software framework. Procedia CIRP, 88:341–345, 2020. 13th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 17-19 July 2019, Gulf of Naples, Italy.
[38] Dr. Sangeeta Jadhav Pranita Bhosale. Implementation of predictive oee (poee): System engineering approach. In Application of Communication Computational Intelligence and Learning: Proceeding of National Con-ference on Communication, Computational Intelligence and Learning (NCCC IL 2021), number 1-12, 2022.
[39] Dr. Sangeeta Jadhav Pranita Bhosale. An insight on methodologies used for predictive machine health monitoring: A review. In Application of Communication Computational Intelligence and Learning: Proceeding of National Conference on Communication, Computational Intelligence and Learning (NCCC IL 2021), number 23-34, pages 1–5, 2022.
[40] Dr. Sangeeta Jadhav Pranita Bhosale. Overall equipment effectiveness (oee): Methodologies used for implementation, detection and prognostics a literature review on case studies. In Application of Communication Computational Intelligence and Learning: Proceeding of National Con-ference on Communication, Computational Intelligence and Learning (NCCC IL 2021), number 309-319, pages 1–5, 2022.
[41] Sample data set : https ://www.kaggle.com/code/irwanjunardi/pumpsensor-data-lstm-vae/notebook.