Using Machine Learning and Deep Learning Algorithms for Downtime Minimization in Manufacturing Systems: An Early Failure Detection Diagnostic Service

Mohammad Shahin
F. Frank Chen (ff.chen@utsa.edu)
University of Texas at San Antonio
Ali Hosseinzadeh
Neda Zand

Research Article

Keywords: Maintenance, Lean Manufacturing, Big Data, Waste Reduction, Machine Learning, Industry 4.0

Posted Date: May 15th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2895149/v1

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on August 23rd, 2023. See the published version at https://doi.org/10.1007/s00170-023-12020-w.

Abstract

Accurate detection of possible machine failure allows manufacturers to identify potential fault situations in processes to avoid downtimes caused by unexpected tool wear or unacceptable workpiece quality. This paper aims to report the study of more than 20 fault detection models using Machine Learning (ML), Deep Learning (DL), and Deep Hybrid Learning (DHL). Predicting how the system could fail based on certain features or system settings (input variables) can help avoid future breakdowns and minimize downtime. The effectiveness of the proposed algorithms was experimented with a synthetic predictive maintenance dataset published by the School of Engineering of the University of Applied Sciences in Berlin, Germany. The fidelity of these algorithms was evaluated using performance measurement values such as accuracy, precision, recall, and the F-Score. Final results demonstrated that Deep Forest and Gradient Boosting algorithms had shown very high levels of average accuracy (exceeded 90%). Additionally, the Multinominal Logistic Regression and Long Short Term Memory based algorithms have shown satisfactory average accuracy (above 80%). Further analysis of models suggests that some models outperformed others. The research concluded that, through various ML, DL, and DHL algorithms, operational data analytics, and health monitoring system, engineers could optimize maintenance and reduce reliability risks.

Introduction

The term “Lean” refers to the efficient usage of the resources available by cutting down the non-value-added activities or wastes [1]. Lean manufacturing includes synergistic tools that integrate to build a high-quality and streamlined system and produce finished products based on the customer’s demand. According to [2], Lean manufacturing is a multi-facet production technique consisting of several industrial practices to identify customer value-adding processes and allow the processes to flow at the customer’s pull throughout the organization. In technological terms, Lean production is considered a complement to the automation technique [3].

In the past few decades, various industrial sectors have widely adopted the concept of Lean. Initially, it was started as the Toyota production system, describing the company's manufacturing philosophy [4]. Different researchers have identified enormous benefits of Lean methodology in the service industries, like hospitals, food, education, public sector, airlines, retail banking, etc. For instance, according to [5], hospitals achieved high-quality healthcare results in the healthcare sector, like a significant reduction in 30-day mortality rates. Womack and Jones [6] clearly stated that implementing the core values of Toyota’s manufacturing system in various sectors has proved beneficial by considering Lean thinking, which refers to the thinking process of Lean within a company and its supply chain.

Moreover, it was suggested that complex production systems must have Lean practices to smooth the operations and processes of a company [7]. In general, Lean manufacturing considers waste as any activity that does not add value to the customer. It can be any service, product, action, or process which needs talent, money, time, or investment. It can also include underutilized resources, idle time, additional inventory, inefficient processes, and untapped talent potential.

Mainly, Lean manufacturing aims to minimize waste by efficient use of resources. A waste in Lean manufacturing refers to an activity that adds cost to the service or product without adding value to the customer. According to [8], waste can be classified into three main types: obvious waste, less obvious waste, and invisible (unobvious) waste. The obvious waste may include excessive setup times, rework, additional procedures, unnecessary inventory, etc. In contrast, less obvious waste is incurred for several reasons, such as demand, staffing, yield, delivery times, and so on [9]. Finally, invisible waste is the waste that can not be seen but causes high costs (for example, material that can not be recycled and ends up in a landfill or incinerators) [10, 11]. No doubt, waste elimination through Lean manufacturing enables new opportunities to build and retain much more value for the end customer, considering the core competencies and resources of the business [12]. In other words, Lean manufacturing focuses on providing a sustainable business environment in which a company operates to run smoothly and efficiently.

According to the Lean philosophy, seven non-value-adding activities or wastes are found in manufacturing, including defective parts or products, unnecessary processing, inventory, waiting, motion, transportation, and over-production. Later, it was suggested that the services or products that do not meet customer expectations must be considered a waste [13]. Overproduction waste generally refers to manufacturing products or rendering services without any order requirement. The waste in overproduction can increase other forms of waste; it can contribute to a more significant amount of finished accumulated inventory, additional employees will be needed, and more storage facilities and warehouse operations will be required. Inventory waste is directly related to production, which comprises supplementary raw material, work-in-progress, and finished goods inventory.

On the other hand, excess inventory may reflect problems within the manufacturing system, including defective machines, long setup times, and frequent machine breakdowns. Motion waste is the unnecessary movement of materials, employees, or machinery. Waste in motion can extend production time, create an unsafe workplace environment, and many other problems. Then, the waiting waste, such as the tasks or goods that do not move, or the finished products kept at the storage facilities for a long time to get delivered. Another waste is over-processing which typically takes place because of overdoing unnecessary work that does not add additional value to the business.

Finally, the defects waste that causes rework or leads to scrap material. Usually, such defective work returns to production, continuing to higher costs that could have been otherwise eliminated according to the Lean philosophy [6]. At the same time, it is worth mentioning that such wastes are commonly believed to be inevitable when a company moves from one product to another; in that case, it will be hard or even impossible in some cases to keep the same setup time [14].

In this paper, we propose an array of ML, DL, and DHL algorithms that have the potential to perform early fault detection that would lead to future machine failure. In the next section, we will discuss the connection between Preventive Maintenance (PM) and Lean manufacturing, along with the role of industry 4.0 in PM. In addition, there will be a discussion on the part of big data in PM and its integration with Industry 4.0.

Background

2.1 Maintenance Strategies & Its Relationship with Lean Manufacturing

An effective manufacturing system generally relies on design quality and an appropriate maintenance strategy to stop system failure. A company incurs a huge cost of maintenance, which is a significant portion of the total production cost [15]. In the past few years, and due to technological advancements, there has been a tremendous increase in maintenance costs. Hence, a scientific maintenance strategy supports reducing equipment failures and eliminating the shutting down of expensive production processes [16]. For this purpose, significant attention has been devoted to maintenance strategies in manufacturing companies. This attention relies on extending the tools' useful lifespan and enhancing the system's availability and reliability. Typically, maintenance policies are divided into three classes, Preventive Maintenace (PM), Corrective Maintenance (CM), and Predictive Maintenance (PdM) [17].

CM is a reactive maintenance mode, which restores the pre-failure status of equipment after a breakdown [18]. CM can be labeled as assentive, meaning it concurs with machine breakdowns. However, this can cause severe lags, causing a halt in the production process and thus financial loss to the company. In addition, some machines may never return to their pre-failure status, thus causing more delays until an alternative replacement or option is found.

On the other hand, the PdM strategy is condition-based maintenance [19] used to maintain equipment over time, so they do not fail unexpectedly. Moreover, according to [20], PM as a model effectively reduces or prevents equipment failure and improves equipment reliability. Therefore, scholars have proposed manufacturing companies perform maintenance activities based on the prediction of the future equipment health rather than its run-time due to the precise impact of component degradation on the manufacturing system availability [15]. PM is similar to PdM, except that it is considered time-based maintenance.

Without going into more details about the three classes of maintenance, it is worth mentioning that the classification and description of CM, PM, and PdM are inconsistent in the literature, [17][19], [21], [22], [23], [24], [25], [26]. Table 1 shows a summary of the different maintenance strategies implemented in the industry.

Table 1
Summary of maintenance strategies

Maintenance Strategies		
Before Failure		After Failure
Preventive Class	Predictive Class	Corrective Class
Proactive	Proactive	Reactive
Regularly scheduled	Scheduled only when needed	Scheduled only when a machine fails
It aims to diagnose and prevent future problems	Prognostic in nature	When damage is done
Usage-based triggered by the tool's exposure to environmental conditions [27]		
Time-based maintenance (TbM), according to [28]	Condition-based Maintenance (CBM), according to [19]	
Could have more costs related to planned machine downtime, labor costs for each scheduled check, etc.	It could have a more upfront cost related to sensors, data acquisition systems, cloud storage, computing power, etc.	It can be very costly due to unexpected machine downtime and other related costs but does not require a significant upfront investment cost
In the long run, it offers a higher Return on Investment (ROI), thus, making it better suited for long-term planning		It might provide a better Return on Investment (ROI) in the short-term
High training is required on using sensors, remote health monitoring, cloud services, etc. Which contributes to higher upfront costs		No high training is required on using sensors, remote health monitoring, cloud services, etc.
Data-driven models such as physics or mathematics-based (statistical or stochastic) and ML-based [29]		
Knowledge-based models such as rule-based and case-based [29]		
Prescriptive: helps analyze and determine different options and potential outcomes to optimize maintenance and reduce reliability risks. It allows engineers to calculate the effects of varying the operating conditions to the time to failure [30–33]		While it is still corrective, engineers should plan a complete Operation and Maintenance (O&M) system that incorporates CM into it [34]
Offers the highest level of automation		
Might be able to suggest a potential solution if successfully combined with Artificial Intelligence (AI)		
More complex		Less complex
Maximum uptime		Maximum downtime

With all this in mind, a Prescriptive Maintenance (RxM) strategy [35] that incorporates both PM and PdM is far more recommended since it helps in eliminating reliability risks [35], thus, increasing the chances of having uninterrupted production. However, since no single maintenance strategy is a perfect solution for all, manufacturers should train, plan, and optimize using all maintenance strategies (CM, PM, PdM, and RxM) to increase uptime and reduce downtime. Figure 1 shows a summarized relationship between the different types of maintenance strategies and their evolution process. In RxM, sensors are used to collect data on the performance and health of equipment such as temperature, vibration, and other relevant parameters, which can then be used to build a decision model. The decision model will analyze the data collected from the sensors and generate a set of decision rules that can be used to predict equipment failures and recommend maintenance actions. Once the critical parameters that lead to component failure are inplace, then an optimal maintenance action based on the predicted failure mode would be recommended. The model could also be updated in real-time as new data is collected, allowing it to continuously improve its predictions and maintenance recommendations. This approach can help companies reduce downtime and increase equipment reliability by enabling them to identify and fix issues before they result in equipment failure.

By embracing Lean manufacturing, a company aims to enhance its operations' efficiency by maximizing its resources' utilization, cutting additional costs, and reducing lead time by preventing equipment failure [36]. For this purpose, the RxM strategy could be deployed. For instance, if a company wants to become more efficient, it will try to adopt a RxM strategy to identify the equipment that might crash to reduce the time lost in downtime while repairing the equipment. Thus, the company’s production will not be affected as much [37]. In other words, a RxM program makes machines more predictable and reliable by observing their performance and enabling the companies to determine when the problem will start. Then they can schedule repairs and replacements before any future breakdowns occur.

RxM can be considered an essential part of the Lean manufacturing process because a company cannot improve its operations and procedures if the equipment and machines are not working efficiently and reliably. Moreover, in today’s advanced technological era, companies are no longer required to adopt rudimentary approaches, such as moving data to spreadsheets and tracking the progress of operations. Instead, due to technology, companies now can leverage big data to restrict the costs and impacts of downtime by adopting a RxM approach [38]. Therefore, if a manufacturing enterprise has decided to increase the efficiency of its operations by adopting a Lean manufacturing philosophy, it must consider adopting a RxM approach due to its promising potential in reducing unexpected machine downtime and related cost [39].

The relationship between different maintenance strategies and Lean manufacturing lies in their common goal of improving operational efficiency, reducing waste, and optimizing resource utilization. Lean manufacturing focuses on eliminating non-value-adding activities, cutting costs, and minimizing waste to enhance overall efficiency. Maintenance strategies aim to maintain equipment reliability, prevent equipment failure, and minimize downtime, which in turn supports Lean manufacturing principles. By incorporating different maintenance strategies into their operations, companies can support their Lean manufacturing goals by reducing equipment downtime, optimizing resource utilization, and minimizing waste. Ultimately, this leads to increased efficiency, cost savings, and a more streamlined production process. Table 2 shows a summary of the connection between Lean manufacturing and different maintenance strategies.

Table 2
Lean manufacturing connection to different maintenance strategies

Strategy	Connection
CM	CM is a reactive approach that involves repairing equipment after it has failed. This strategy aligns poorly with Lean manufacturing, as it often leads to unexpected downtime, higher costs, and production delays due to equipment breakdowns.
PM	PM is a time-based approach that involves scheduling maintenance activities at regular intervals to reduce the probability of equipment failure. This strategy supports Lean manufacturing by proactively maintaining equipment, reducing downtime, and minimizing waste associated with unexpected breakdowns.
PdM	PdM is a condition-based approach that relies on monitoring equipment performance and using data analytics to predict when maintenance is needed. This approach aligns well with Lean manufacturing, as it helps reduce downtime, optimize maintenance schedules, and improve resource utilization.
RxM	RxM is a more comprehensive approach that combines both PM and PdM strategies. It not only predicts equipment failure but also prescribes the best course of action to maintain optimal performance. This strategy supports Lean manufacturing by reducing reliability risks, minimizing downtime, and enhancing overall operational efficiency.

2.2 Link of Industry 4.0 with Prescriptive Maintenance

Toyota Motor Corporation successfully implemented Lean manufacturing and reflected a remarkable reduction in all forms of waste and a boost in productivity [40]. Today, Industry 4.0 promotes an innovative and efficient workplace environment [41]. Moreover, the revolution of Industry 4.0 fully supports a real-time RxM approach [41]. For instance, using intelligent sensors gives a complete real-time solution to monitor the system. Therefore, allowing the managers to plan maintenance activities to reduce machine downtime and promote a smoother production flow. Thus, demonstrating how an Industry 4.0 approach to RxM via big data can provide the means to realize the Extended Lean Enterprise [42, 43].

Lean manufacturing and industry 4.0 follow a similar approach [44]; Lean manufacturing makes efforts to reduce the complexity and cost by minimizing the waste and non-value-adding services from a value chain. In addition, it uses Kaizen or continuous improvement to enhance efficiency by involving all the employees across all departments. On the other hand, industry 4.0 is based on utilizing nine technologies, including simulation, industrial internet, vertical and horizontal system integration, cybersecurity, cloud computing, big data and analytics, augmented reality, advanced robotics, and additive manufacturing. Furthermore, IT systems, workpieces, machines, and sensors are all linked with a value chain that goes above a single company [45]. Such interlinked systems can allow for interaction and assess the data for predicting the operational performance levels of an enterprise. For instance, leveraging big data and ML enable a company to plan and perform RxM at the most appropriate time. Therefore, various manufacturers have formed maintenance strategies while transforming to Industry 4.0 as it allows higher efficiency and reduces the risk of equipment breakdowns or errors. For this purpose, the RxM is used to detect sources (variables) of potential failure for a machine in the future; this is done by considering possible environmental variables (for instance, torque, strength, etc.) that could trigger a failure (such as heat failure or wear failure), allowing for any sources of potential setbacks to be eliminated before an interruption occurs to the production process [46].

The relationship between RxM and I4.0 lies in their common emphasis on utilizing advanced technologies, data analytics, and automation to optimize industrial processes and improve overall efficiency. I4.0 is a term used to describe the fourth industrial revolution, characterized by the integration of cyber-physical systems, the Internet of Things (IoT), and advanced data analytics to create smart and interconnected manufacturing systems. RxM is an advanced maintenance strategy that uses data-driven insights to predict equipment failures, diagnose the root cause, and prescribe the best course of action to maintain optimal performance. This approach aligns well with I4.0 principles, as it leverages several key technologies and concepts that are fundamental to this new industrial paradigm. Table 3 shows how the concepts and fundamentals of I4.0.

Table 3
Relationship between RxM and I4.0

Key Technology	**Relationship**
Big Data	RxM relies on continuous data collection from various sensors, devices, and systems, which are integral to I4.0. Data is then analyzed using advanced analytics, ML, and AI techniques to predict equipment failures, determine the root cause, and suggest optimal maintenance actions.
IoT	IoT plays a significant role in both I4.0 and RxM. In the context of RxM, IoT devices and sensors are used to monitor equipment performance, collect data, and transmit it to a central system for analysis. This enables real-time monitoring and decision-making for maintenance tasks.
Cyber-Physical Systems	I4.0 emphasizes the integration of physical systems and digital technologies to create interconnected networks. RxM benefits from these cyber-physical systems, which enable seamless communication between equipment, sensors, and data analysis platforms, allowing for real-time decision-making and improved maintenance processes.
Automation and Robotics	I4.0 encourages the use of automation and robotics to optimize processes and reduce human intervention. Similarly, RxM can leverage automation to perform certain maintenance tasks, minimizing downtime and increasing efficiency.
Digital Twin Technology	Digital twins are virtual representations of physical assets that can be used to simulate and analyze their performance. In the context of RxM, digital twins can help identify potential failure points, simulate maintenance actions, and predict their impact on equipment performance, contributing to more effective maintenance strategies.

In summary, the relationship between Prescriptive Maintenance and I4.0 is based on their shared focus on leveraging advanced technologies to optimize industrial processes. RxM is an essential component of I4.0's vision for smart, interconnected, and data-driven manufacturing systems, offering significant benefits in terms of equipment reliability, resource optimization, and operational efficiency.

2.3 Role of Big Data in Prescriptive Maintenance

The rise of computing power, the advancement in sensor technology, and big data; have allowed manufacturers to move to a “proactive” maintenance approach rather than a “reactive” approach [47]. Instead of only doing maintenance when the failure has already occurred, the core strategy is to estimate the time it takes for a tool to degrade and schedule maintenance accordingly. Thus, allowing RxM to rely on ML algorithms to detect failure instead of using traditional mathematics and physics-based tool life models [48].

This big data-powered RxM approach is mainly achieved by considering the sensors that continuously monitor the system's signals (health) and record all the varying conditions that lead to failure events over the years [49]. The data analytics algorithms are used to detect possible scenarios of failure by monitoring any input in the data that could lead to it. Therefore, at the core of this RxM approach is an ML and DL algorithm that learns from the tool’s collected data in order to extract features that would be used as inputs to detect failure.

RxM is commonly used in various sectors, such as the aircraft, automobile, and manufacturing industries. As a result, potential systems and equipment failures can be identified and predicted at a future point [36]. According to [37], big data is a critical tool for any RxM strategy. It enables data storing and monitoring, identifying the causes, and minimizing the impacts of failures to make better decisions for a company. In addition, [51] pointed out that the modern business environment requires continuous industrial equipment monitoring, creating massive amounts of information. Such data can only be processed through algorithms powered by AI via ML or DL. Therefore, illustrating the significant and critical role of ML and DL in RxM [52, 53]. Furthermore, companies can get an advantage from collecting data during a tool or machine life cycle to detect failure trends [54]. In addition, there has been a new direction in research that considers detecting possible future failures to improve system reliability so that different structures and failure mechanisms can be contained [55].

According to [56], data analytics allows to schedule the best time for performing maintenance by accurately predicting tool life or detecting potential failure due to a change in the machine’s environmental variables. Thus, machines will be offline for less time, and components are changed only when required. In other words, RxM allows for intelligent decision-making when combined with ML and DL models [57].

RxM is suitable when a company considers machine learning and regression analytics [58, 59] to evaluate the machinery condition based on available sensor data, demonstrating that maintenance operations can be done through less disruption [56]. Consequently, RxM focuses on doing something before it happens, saving considerable costs by anticipating future failures or downturns to avoid disruptions during the processing stage [56]. It is believed that the initial step in RxM is gathering relevant data from multiple sources [60]. The current technological advancements allow better data collection processes through embedded sensors, network routers, operational systems, and control systems, allowing manufacturers to analyze the big data to predict future maintenance time [47]. However, with big data comes more challenges to RxM, such as the lack of correct labels to describe the machine condition or maintenance background. Moreover, due to a large amount of heterogeneous data, it is challenging to integrate the information to obtain data-driven decisions [61]. Figure 2 shows the framework for an RxM system powered by AI. In the next section, we will discuss the dataset and the data analytics techniques that were used to produce our results.

Dataset and Methodology

3.1 Details of the Utilized Dataset

Since real RxM datasets are generally hard to obtain in general and even harder to publish in particular [62], we utilized the “AI4I 2020 Predictive Maintenance Dataset” synthetic dataset [63]. The dataset was published by the School of Engineering at the University of Applied Sciences in Berlin, Germany. The dataset consists of 10 000 record rows with 14 features in columns obtained from [63]. Table 4 shows a description of the dataset's salient features. Figure 3 shows a statistical summary of the failure features.

Table 4
Description of the 14 features [62]

Input Feature	**Description**
UID	Unique identifier ranging from 1 to 10000, [62]
Product ID	Consisting of a letter L, M, or H for low (50% of all products), medium (30%), and high (20%) as product quality variants followed by a variant-specific serial number, [62]
Type	A letter L, M, or H for low (50% of all products), medium (30%), and high (20%), [62]
Air temperature	Generated using a random walk process later normalized to a standard deviation of 2 K around 300 K, [62]
Process temperature	Generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K, [62]
Rotational speed	Calculated from the power of 2860 W, overlaid with a normally distributed noise and measured in rpm, [62]
Torque	torque values are distributed around 40 Nm with a σ = 10 Nm and no negative values, [62]
Tool wear (min)	The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process, and a machine failure' label that indicates whether the machine has failed in this particular data point for any of the following failure modes is true [62]
Output Feature	Description
Tool wear failure (TWF)	The tool will be replaced or fail at a randomly selected tool wear time between 200–240 mins (120 times in our dataset). At this point in time, the tool is replaced 74 times and fails 46 times (randomly assigned), [62]
Heat dissipation failure (HDF)	The heat dissipation causes a process failure if the difference between air- and process temperature is below 8.6 K, and the tool’s rotational speed is below 1380 rpm. This is the case for 115 data points [62]
Power failure (PWF)	The product of torque and rotational speed (in rad/s) equals the power required for the process. If this power is below 3500 W or above 9000 W, the process fails, which is the case 95 times in our dataset, [62]
Overstrain failure (OSF)	If the product of tool wear and torque exceeds 11,000 minNm for the L product variant (12,000 for M, 13,000 for H), the process fails due to overstrain. This is true for 98 data points [62]
Random failures (RNF)	Each process has a chance of 0,1% to fail regardless of its process parameters. This is the case for 19 data points, more frequent than could be expected for 10,000 data points in our dataset, [62]
Predicted machine failure	If at least one of the above failure modes is true, the process fails, and the machine failure label is set to 1, which is the case for 339 data points [62]

Based on Fig. 3, the dataset is heavily imbalanced, consisting of only 339 rows labeled as machine failure. At the same time, a failure rate of 3.39% (339/10,000.00) would usually make the system as similar as possible to real-world industrial control systems; at the same time, this rate can be considered alarming in mass production environments. Nevertheless, this is an inherent problem in most PdM datasets [62]. Figure 4 shows the estimated relative predictor importance of the features used in ML algorithms. Please note that we have deliberately kept our dataset imbalanced since this is the realist scenario that could happen in real-world settings.

3.2 Utilized Models and Algorithms

ML is an AI model that utilizes algorithms for the purpose of analyzing data, extracting relevant information, and using that information to make informed decisions [64]. In contrast, DL is a subdivision of ML that constructs algorithms in layers to produce an Artificial Neural Network (ANN) with the ability to learn and make intelligent decisions autonomously [65]. The way an ANN is structured is by arranging nodes into three layers: input, hidden, and output [66]. The concept of "learning" is the ability of both ML and DL models to improve their performance over time without prior knowledge. The main difference between the two is that ML models require human intervention when their AI algorithm generates inaccurate predictions. This intervention is known as feature engineering and is used to enhance the final accuracy and performance of the AI algorithm. On the other hand, DL models can assess the accuracy of their predictions using their neural network, which needs more computational power. However, DL models are susceptible to the over-fitting problem if there is a shortage of data. To resolve this issue, DHL combines the neural network of the DL algorithm for feature extraction and a standard ML algorithm for classification, resulting in high accuracy without over-fitting and without expending excessive computational resources [67–69]. Figure 5 shows the relationship between DL, ML, DHL, and AI.

In this paper, we have utilized a wide range of models to detect machine failure. Table 5 shows a summary of all the models that were deployed. More details on the models are introduced later throughout this section of the paper. Table 6 shows an overview of the ML and DL models that were utilized in previous data-driven maintenance models. Based on the information provided in both tables, this research will evaluate an array of different and unique algorithms on a single dataset. Thus, allowing a deeper understanding of the efficiency and fidelity of each model that could be utilized in an RxM strategy.

Table 5
Summary of deployed models

ML models	DL models
Decision Tree (DT)	Multilayer Perceptron (MLP)
Multinominal Random Forest (MRF)	Long Short Term Memory (LSTM)
Support Vector Machines (SVM)	Attention based Long Short Term Memory (ALSTM)
Stochastic Gradient Descent (SGD)	Convolutional Neural Network (CNN)
Multinomial Logistic Regression (MLR)	Fully Convolution Network (FCN)
K-Nearest Neighbors (KNN)	Learning Vector Quantization (LVQ)
Multinomial Naïve Bayes (MNB)	DHL models
Extreme Gradient Boosting (XGBoost)	ALSTM-FCN with XGBoost
Adaptive Boosting (AdaBoost)	ALSTM-FCN with AdaBoost
Light Gradient Boosted Machine (LightGBM)	CNN with XGBoost
Linear Discriminant Analysis (LDA)	Deep Forest (DF)
Quadratic Discriminant Analysis (QDA)	

Table 6
Overview of identified RxM models via ML and DL in the literature

Task	Models	References
RUL prediction, Fault diagnostic	DNN, RNN, LSTM, CNN, LSTM-RNN, CNN-LSTM, LR	[70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80]
Prognostics	DNN, RNN, LSTM, LR, SVM	[70], [71], [80], [81]
Fault identification	MLP, RBFN, SVM	[82], [83], [84], [85], [86]
Degradation modeling	SOM, SVM	[87], [88]
Fault detection	SOM, SVM, LSTM	[86], [89], [90], [91], [92]
Early failure detection	DT, MRF, SVM, SGD, MLR, KNN, MNB, GB, LDA, QDA, DF, MLP, LSTM, ALSTM, CNN, FCN, LVQ, DHL	Presented work in this paper
DNN: Deep Neural Network, RNN: Recurrent Neural Network, RBFN: Radial Basis Function Network, SOM: Self-Organizing Maps, LR: Logistic Regression		

In the case of models that demand numeric features, categorical features with no ordinal relationship can be problematic if encoded using integers. Assigning an integer to each category may mislead the model, leading to unexpected outcomes or reduced performance if it assumes a natural ordering between categories. One potential issue is predictions that fall halfway between categories. To address this issue, one-hot encoding is utilized to binarize categorical inputs for a more accurate representation [93]. One-Hot Encoder was used to encode categorical features for all ML, DL, and DHL models [93], and SKlearn Robust Scaler was used to scale the features and make them robust to outliers [94]. A split of 90/10 was used for training and testing, respectively.

3.3 Machine Learning (ML) Models

Cross-Validation (CV) has been used on all ML models. We choose a value of k = 5, which is very common in ML [95, 96]. GridSearch CV or RandomizedSearch CV for hyperparameters tuning for each ML model. The difference between both approaches is in GridSearch we define the combinations and do training of the model, whereas in RandomizedSearchCV the model selects the combinations randomly.

3.3.1 Decision Tree (DT)

A DT is a type of supervised machine learning algorithm that is used for solving classification or regression problems. It is a graphical representation of all the possible solutions to a decision, with each branch of the tree representing a possible decision or occurrence. Each decision or branch in the tree is based on a specific feature or attribute of the data, and the tree's structure is created by recursively partitioning the data into subsets based on the value of the selected feature until a decision is reached. The final result of a DT algorithm is a tree with decision nodes and leaf nodes, where the decision nodes contain the conditions that decide which branch to follow, and the leaf nodes provide the outcome or prediction of the model. DT is easy to interpret and can handle both categorical and numerical data, making them a popular choice in many applications [97–99].

3.3.2 Multinominal Random Forest (MRF)

MRF is a supervised machine learning algorithm that is used for classification problems where the target variable has multiple classes or categories. It is an extension of the RF algorithm, which combines multiple DT to produce a more accurate prediction. In the case of MRF, the algorithm builds a collection of DT, where each tree is trained on a subset of the data and uses a random selection of features to make predictions. During training, the algorithm calculates the importance of each feature in making accurate predictions and uses this information to weight the importance of each tree. MRF is particularly useful for problems with a large number of classes or categories since it can efficiently handle the high-dimensional feature space that arises from this type of problem. Additionally, MRF is less prone to overfitting than other classification algorithms, which can be a problem when dealing with large datasets. Applications of MRF include text classification, image recognition, and bioinformatics. It is also used in areas where multi-class problems arise, such as customer segmentation, medical diagnosis, and fault detection [100, 101].

3.3.3 Support Vector Machines (SVM)

SVM is a supervised machine learning algorithm used for classification and regression analysis. The goal of SVM is to find a hyperplane that separates data points of different classes in a high-dimensional space. In other words, SVM attempts to find the optimal decision boundary that maximizes the margin, or the distance between the decision boundary and the nearest data points of each class. To accomplish this, SVM maps the data into a higher-dimensional feature space where it becomes easier to find a hyperplane that can separate the data. SVM uses a kernel function to compute the dot product between the data points, which allows the algorithm to work in the high-dimensional space efficiently without actually computing the coordinates of the data points in that space. SVM is particularly useful when the number of features is greater than the number of samples and the data is not linearly separable. SVM can also handle non-linear decision boundaries by using kernel functions such as polynomial, radial basis function, or sigmoid function. SVM has many applications, including image classification, text classification, bioinformatics, and financial forecasting. SVM has become a popular algorithm for classification and regression analysis due to its effectiveness and efficiency in solving complex problems. SVM has also been shown to generalize well on new, unseen data, making it a valuable tool for many applications [102, 103].

3.3.4 Stochastic Gradient Descent (SGD)

SGD is an optimization algorithm commonly used in machine learning to find the parameters or coefficients of a model that minimize a given cost or loss function. It is a variant of the Gradient Descent algorithm, which updates the model parameters iteratively to minimize the cost function. However, while Gradient Descent calculates the average gradient over the entire training set, SGD calculates the gradient for each training example individually and updates the model parameters based on that gradient. The main advantage of using SGD over Gradient Descent is that it can converge faster as it considers only one training example at a time, making it more computationally efficient. Furthermore, SGD can help to prevent the model from getting stuck in local optima by introducing randomness into the optimization process [104].

3.3.5 Multinomial Logistic Regression (MLR)

MLR is a statistical based machine learning model used for predicting outcomes with more than two categories. It is a variant of the logistic regression algorithm, which is used for binary classification problems. In MLR, the dependent variable or outcome variable has three or more categories, and the goal is to model the probability of each category as a function of one or more independent variables or predictors. In MLR, the algorithm estimates the probability of each category using a set of coefficients, which are calculated using maximum likelihood estimation. The coefficients represent the impact of each predictor variable on the probability of each category, with positive coefficients indicating that the variable increases the probability of the category and negative coefficients indicating that the variable decreases the probability of the category. MLR is commonly used in many fields, including social sciences, economics, and medical research, where there are multiple outcomes of interest. MLR is often used to model customer behavior, market segmentation, and disease classification. Additionally, MLR can be used to identify the most important predictors for each category, which can be useful in understanding the factors that influence different outcomes [105, 106].

3.3.6 K-Nearest Neighbors (KNN)

KNN is a supervised machine learning algorithm used for classification and regression analysis. It is a non-parametric algorithm, which means that it does not make any assumptions about the underlying distribution of the data. In KNN, the algorithm classifies new data points based on the class or category of the K nearest neighbors in the training set. The value of K is a hyperparameter that is set by the user and determines the number of neighbors used to classify a new data point. To find the K nearest neighbors, the algorithm calculates the distance between the new data point and each point in the training set. The most common distance metric used is the Euclidean distance, although other metrics such as Manhattan distance and Minkowski distance can also be used. Once the K nearest neighbors have been identified, the algorithm assigns the new data point to the class that is most common among the K neighbors. For regression analysis, KNN calculates the average value of the K nearest neighbors and assigns that value to the new data point. KNN is simple to understand and easy to implement, making it a popular choice in many applications. It is also non-parametric, which means it can handle data that does not follow a specific distribution. However, KNN can be computationally expensive, especially for large datasets, and it does not work well with high-dimensional data [107].

3.3.7 Multinomial Naïve Bayes (MNB)

MNB is a probabilistic machine learning algorithm used for classification problems with discrete features. It is a variant of the Naive Bayes algorithm, which is based on Bayes' theorem of conditional probability. In MNB, the algorithm models the probability of each class based on the frequency of the feature occurrences in the training set. It assumes that the features are conditionally independent of each other, which means that the presence or absence of one feature does not affect the presence or absence of any other feature. To make a prediction, the algorithm calculates the probability of each class given the occurrence of each feature in the new data point. The class with the highest probability is then assigned as the prediction. MNB is particularly useful for text classification problems, such as spam filtering and sentiment analysis, where the features are typically the frequency of occurrence of words in a document. MNB has also been shown to perform well in many other applications, including image recognition and bioinformatics. MNB is computationally efficient and can handle large datasets with many features. However, it can be sensitive to irrelevant features and may perform poorly when the features are not conditionally independent [108].

3.3.8 Extreme Gradient Boosting (XGBoost)

XGBoost is a popular supervised machine learning algorithm used for regression, classification, and ranking tasks. It is a variant of the Gradient Boosting algorithm that is designed to improve performance and reduce computation time. In XGBoost, the algorithm builds an ensemble of decision trees, where each tree is trained to correct the errors of the previous tree. The algorithm iteratively adds trees to the ensemble until the model's performance on the validation set no longer improves. XGBoost uses a gradient descent optimization technique to minimize a specific loss function, which is typically the mean squared error or log loss, depending on the problem. XGBoost incorporates several features that improve the performance and efficiency of the algorithm, including, regularization to prevent overfitting by penalizing the complexity of the model, tree pruning to remove redundant or unnecessary branches in the decision trees, and parallel processing to speed up computation by distributing the workload across multiple processors. XGBoost has been shown to be highly effective in many machine learning applications, including image recognition, text classification, and natural language processing. It has won numerous machine learning competitions and is widely used in industry and academia due to its accuracy and efficiency [109],[110],[111].

3.3.9 Adaptive Boost (AdaBoost)

AdaBoost is a machine learning algorithm used for regression and classification problems. It is an ensemble method that combines multiple weak learners to form a strong learner. In AdaBoost, the algorithm assigns a weight to each training example based on its classification error in the previous iteration. The weights of misclassified examples are increased, and the weights of correctly classified examples are decreased. The algorithm then trains a new weak learner on the updated weighted data and repeats the process. The weak learners are typically decision trees with a maximum depth of one, which are also called decision stumps. Once the weak learners have been trained, AdaBoost combines them to form a strong learner. The final classification is based on a weighted majority vote of the weak learners, where each weak learner's weight is proportional to its accuracy on the training set. AdaBoost is particularly effective in handling noisy data and can achieve high accuracy with relatively small datasets. However, it is sensitive to outliers and can overfit to noisy data if the number of weak learners is too high. AdaBoost has been applied in various domains, including face detection, natural language processing, and medical diagnosis [112, 113].

3.3.10 Light Gradient Boosted Machine (LightGBM)

LightGBM i is an open-source gradient boosting framework used for supervised machine learning tasks such as regression and classification. It is a variant of the gradient boosting algorithm that is designed to be fast and efficient, with lower memory usage and faster training speed. LightGBM uses a tree-based learning algorithm and gradient-based optimization to minimize the loss function. It builds a tree-based model by adding decision trees in a greedy fashion, where each new tree is trained to correct the errors of the previous trees. LightGBM employs a leaf-wise approach to tree building, which grows the tree leaf by leaf and reduces the number of splits needed to reach the same depth, resulting in a shallower and more balanced tree. LightGBM also includes several features that improve its efficiency and accuracy, such as, gradient-based one-side sampling, which reduces the computation required by only using a subset of the data with large gradients, exclusive feature bundling, which bundles features with similar values into a single feature, reducing the number of features and improving accuracy, and histogram-based algorithms, which replace continuous features with discrete bins to speed up training and reduce memory usage. LightGBM has been shown to perform well on large-scale datasets with high-dimensional features and is widely used in industry and academia. It has won several machine learning competitions and is particularly useful for applications such as search ranking, recommendation systems, and image classification [114].

3.3.11 Linear Discriminant Analysis (LDA)

LDA is a supervised machine learning algorithm used for classification problems with two or more classes. It is a dimensionality reduction technique that projects the data onto a lower-dimensional space while preserving the separation between the classes. In LDA, the algorithm finds a linear combination of features that best separates the classes. It does this by maximizing the between-class scatter and minimizing the within-class scatter. The between-class scatter measures the distance between the means of the classes, while the within-class scatter measures the variation within each class. Once the optimal linear combination has been found, LDA projects the data onto this new feature space. The new features are ordered by their discriminatory power, with the most important features being those that best separate the classes. LDA is particularly useful for datasets with many features and few samples. By reducing the dimensionality of the data, LDA can improve the accuracy and generalization performance of the classification model. LDA has many applications, including image recognition, speech recognition, and bioinformatics. One limitation of LDA is that it assumes that the data follows a normal distribution and that the covariance matrices are equal for all classes. If these assumptions are not met, the performance of the algorithm may be compromised [115].

3.3.12 Quadratic Discriminant Analysis (QDA)

QDA is a supervised machine learning algorithm used for classification problems with two or more classes. It is a variant of Linear Discriminant Analysis (LDA) that relaxes the assumption of equal covariance matrices for all classes. In QDA, the algorithm finds a quadratic boundary that separates the classes by modeling the distribution of the data for each class. Unlike LDA, which assumes that the covariance matrices are equal for all classes, QDA allows for different covariance matrices for each class. This allows QDA to capture more complex decision boundaries and can lead to better classification accuracy in certain situations. Once the covariance matrices have been estimated, QDA calculates the discriminant function for each class, which is used to classify new data points. The class with the highest discriminant function value is assigned as the predicted class. QDA is particularly useful when the covariance matrices differ significantly between classes or when the distribution of the data is non-normal. However, QDA can be sensitive to overfitting when the number of features is large compared to the number of samples. In these cases, regularized versions of QDA, such as Shrinkage QDA, can be used to improve performance. QDA has many applications, including speech recognition, image classification, and bioinformatics [116].

3.4 Deep Learning (DL) Models

For all models, dense layers were used with Softmax activation function, and hidden layers were used with Rectified Linear Activation (ReLU). LSTM and ALSTM were made up of 8 units. All DL and DHL models were trained for more than 14 epochs (See Fig. 6).

Figure 10 Using multiple epochs to achieve minimum loss and highest performance measurements values for DL models

3.4.1 Multilayer Perceptron (MLP)

MLP is is a type of ANN that is commonly used in supervised machine learning applications for both classification and regression tasks. MLP is a feedforward neural network consisting of multiple layers of neurons that transform the input data to a desired output. In MLP, the input data is first passed through one or more hidden layers, each of which consists of several neurons that perform a linear transformation followed by a non-linear activation function. The output of each hidden layer is then passed to the next layer until the final layer is reached, which produces the network's output. The weights of the MLP are adjusted during training to minimize a loss function, such as the mean squared error or cross-entropy loss, using backpropagation. Backpropagation is a gradient-based optimization technique that adjusts the weights of the network in the direction of the steepest descent of the loss function. MLP is a versatile algorithm that can handle both numerical and categorical data, and can be used for various types of machine learning tasks, including image classification, natural language processing, and time-series forecasting. However, MLP is prone to overfitting when the number of parameters is too large, and it can be difficult to interpret the results of the model due to its complex structure [117–119]. See Fig. 7 for a demonstration of the proposed MLP model.

3.4.2 Long Short Term Memory (LSTM)

LSTM is a type of Recurrent Neural Network (RNN) that is designed to handle the problem of vanishing gradients in traditional RNNs. LSTMs are particularly well-suited for processing sequential data, such as text, speech, and time-series data. In LSTM, the network consists of repeating modules of memory cells, input gates, output gates, and forget gates. The memory cells store information about the sequence, while the gates regulate the flow of information into and out of the cells. The input gate controls the amount of new information that is added to the memory cell, while the forget gate determines how much of the previous memory should be retained. The output gate determines the amount of information that is outputted from the memory cell. LSTM allows for the learning of long-term dependencies by allowing information to be retained in the memory cells over many time steps. This is achieved by the use of the gates, which selectively pass information and prevent the problem of vanishing gradients. LSTM has been applied in various domains, including speech recognition, text analysis, and time-series forecasting. It has been shown to outperform traditional RNNs on many tasks and has become a popular choice in many machine learning applications [120].

3.4.3 Attention-based Long Short Term Memory (ALSTM)

ALTSM is a type of neural network architecture that combines the strengths of LSTM and attention mechanisms to improve the performance of sequence-to-sequence tasks, such as machine translation and speech recognition. In ALSTM, the input sequence is first processed by an LSTM network, which learns to encode the input sequence into a fixed-length vector representation. The attention mechanism is then applied to the encoded representation, which allows the network to focus on the most relevant parts of the sequence during decoding. During decoding, the attention mechanism computes a weight for each element in the encoded sequence based on its relevance to the current decoding step. The weighted sum of the encoded sequence is then used as the input to the next decoding step. ALSTM has several advantages over traditional LSTM networks. It can handle long sequences with variable-length inputs and outputs, and it allows the network to focus on the most relevant parts of the sequence during decoding. This results in improved performance on sequence-to-sequence tasks. ALSTM has been successfully applied in various domains, including machine translation, speech recognition, and image captioning. It has become a popular choice in many machine learning applications and has contributed to significant advances in natural language processing and computer vision [121], [122].

3.4.4 Convolutional Neural Network (CNN)

CNN is a type of neural network that is commonly used in deep learning for image and video analysis. It is particularly well-suited for tasks that require the network to learn hierarchical representations of the input data, such as object detection and recognition. CNN is composed of several layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply a set of filters to the input image, which detect features such as edges, corners, and shapes. The pooling layers downsample the feature maps, reducing the dimensionality of the data while retaining important information. The fully connected layers perform the classification, using the features learned by the convolutional and pooling layers to predict the output class. CNNs are designed to be translation invariant, meaning that they can detect the same feature regardless of its position in the input image. This is achieved by the use of the convolutional layers, which scan the input image with a small filter and apply it to all locations in the image. CNNs have many applications in computer vision, including image classification, object detection, face recognition, and semantic segmentation. They have achieved state-of-the-art performance on many benchmarks and have been widely adopted in industry and academia [123], [124]. Figure 8, Fig. 9, and Fig. 10 show the architecture of the proposed Two layers CNN-ALSTM model, the CNN-LSTM model, and the Two layers CNN model, respectively, that were built.

3.4.5 Fully Convolutional Neural Network (FCN)

FCN is a type of neural network that is commonly used for semantic segmentation tasks in computer vision, such as object detection and image segmentation. It is an extension of CNN that allows for end-to-end segmentation of images without the need for additional post-processing steps. Unlike traditional CNNs, which typically have a fully connected layer at the end of the network for classification, FCNs replace the fully connected layer with convolutional layers to produce a dense output map of class probabilities [125]. This allows the network to classify each pixel in the input image instead of the entire image as a whole. FCNs use several techniques to improve segmentation performance, including skip connections, upsampling, and multi-scale inputs. Skip connections allow the network to reuse features from earlier layers in the network, improving accuracy and reducing artifacts. Upsampling techniques such as transposed convolution or nearest-neighbor interpolation are used to increase the resolution of the output map. Multi-scale inputs, which involve processing the image at different scales, help the network to better capture objects of varying sizes. FCNs have achieved state-of-the-art performance on many benchmark datasets for semantic segmentation, including Pascal VOC, Cityscapes, and COCO. They have been widely adopted in industry and academia for various computer vision tasks [124, 126].

3.4.6 Learning Vector Quantization (LVQ)

LVQ is a type of supervised learning ANN algorithm that learns to classify input data by mapping it to a set of predefined classes. In LVQ, the network consists of a set of prototype vectors, each of which represents a class in the output space. During training, the prototype vectors are adjusted to minimize a distance metric between the input data and the prototypes. The distance metric can be based on various measures, such as Euclidean distance or cosine similarity. LVQ can be divided into several variants, including LVQ1, LVQ2, and LVQ3. LVQ1 assigns the input to the closest prototype vector, while LVQ2 uses a winner-takes-all approach to select the closest prototype vector and adjust it and its neighbors. LVQ3 uses a more flexible approach by using a range of vectors to adapt to the input. LVQ has several advantages over other classification algorithms, such as its ability to handle non-linearly separable data and its interpretability. LVQ is also robust to noise and outliers and can learn from small datasets. LVQ has been applied in various domains, including image classification, text classification, and speech recognition. It has been shown to perform well on many benchmark datasets and has been widely adopted in industry and academia [127–129].

3.5 Deep Hybrid Learning (DHL) Models

Using DL algorithms, features were extracted from the training and testing datasets, and ML algorithms performed detection. Four distinct models were applied to the dataset. Model one incorporated an Attention-based Long Short-Term Memory Fully Convolutional Neural Network with Extreme Gradient Boosting (ALSTM-FCN with XGBoost), while model two utilized an Attention-based Long Short-Term Memory Fully Convolutional Neural Network with Adaptive Boost (ALSTM-FCN with AdaBoost). Each model featured two 1D convolutional hidden layers, operating on a 1D sequence and containing 32 kernels (filters) with a size of 2. These kernels stored values learned throughout training. Every hidden convolutional layer was paired with batch normalization to normalize input via a transformation that kept the mean output close to 0 and the output standard deviation close to 1. These layers served as feature extractors. Rectified Linear Activation (ReLU) enabled the model to learn more complex functions, enhancing training results. A GlobalAveragePooling1D layer followed the ReLU to preserve information about "less important" outputs. Dropout layers with probability values of 0.2 and 0.3 were added to minimize overfitting risk. One hundred LSTM cells were deemed optimal. The GlorotUniform, or Xavier Uniform, was the default weight initializer for both models. An attention mechanism aided the algorithm's learning process. Both models employed the Adam Optimization Algorithm with a constant learning rate of 0.03. After the dropout layers, a concatenation layer combined all previous outputs along a specified dimension. These new outputs fed into the ML algorithm (XGBoost or AdaBoost), using a RandomizedSearchCV for random parameter combination selection, enhancing model generalizability. Illustrations of the ALSTM-FCN with XGBoost and ALSTM-FCN with AdaBoost models can be found in

Figs. 11 and 12, respectively. The third deep learning model, CNN with XGBoost, was similar to the previous two but featured dense neuron layers and lacked the ALSTM mechanism [130].

The last model is DF, an ML algorithm that combines RF with DL techniques to improve the accuracy of prediction models. It is a type of ensemble learning method that combines multiple decision trees to form a single prediction model. In DF, the algorithm first trains a random forest on the input data to generate a set of features that are used to train a deep learning model. The deep learning model is typically a multi-layer neural network that can learn complex representations of the data. DF uses two structures of DL models MLP and CNN. The MLP is used for tabular data, while the CNN is used for image data. DF has several advantages over traditional deep learning algorithms, such as faster training times, better generalization performance, and improved interpretability. It can handle large datasets with high-dimensional features and can achieve high accuracy with relatively small datasets. DF is also less prone to overfitting than traditional deep learning algorithms. DF has been applied in various domains, including computer vision, natural language processing, and speech recognition. It has won several machine learning competitions and is widely used in industry and academia [131–133].

Results and Discussion

A confusion matrix was obtained for every DL and ML model. However, we only included one as an example here. Figure 13 shows the confusion matrix for the lightGBM model. A confusion matrix is a table used to evaluate the performance of a classification model by displaying the number of correct and incorrect predictions. It is particularly useful for assessing the performance of a classifier in a multi-class problem. The matrix consists of rows and columns that represent the actual and predicted classes, respectively. The confusion matrix has four main components, True Positives (TP): The number of instances where the model correctly predicted the positive class. True Negatives (TN): The number of instances where the model correctly predicted the negative class. False Positives (FP): The number of instances where the model incorrectly predicted the positive class (also known as Type I error). False Negatives (FN): The number of instances where the model incorrectly predicted the negative class (also known as Type II error). The TP, TN, FP, and FN values obtained from the confusion matrix is used to calculate performance measurement values such as accuracy, precision, recall, and F-measure. Keep in mind that the misclassification cost of a false negative is much higher than the cost of a false positive, especially in a mass production environment [62].

Accuracy is a metric used to evaluate the performance of classification models. It is the proportion of correct predictions made by the model out of the total number of predictions. In the context of RxM, accuracy plays a significant role in the effectiveness of the maintenance strategy. RxM relies on predictive models to forecast equipment failures and provide recommendations for optimal maintenance actions. A higher accuracy of these predictive models means that the maintenance team can make better-informed decisions, which can lead to improved equipment reliability, reduced downtime, and cost savings. However, it's important to note that accuracy alone might not provide a complete picture of a model's performance, particularly in cases of imbalanced datasets. In such situations, other metrics like precision, recall, and F1-score can be more insightful in evaluating a model's effectiveness. Combining these metrics with accuracy can offer a more comprehensive understanding of the model's performance and its impact on RxM.

Precision is a performance metric used to evaluate classification models, specifically focusing on the correctness of positive predictions. It is the ratio of true positive predictions to the total number of positive predictions made by the model. In the context of RxM, precision plays a crucial role in minimizing false alarms or unnecessary maintenance actions. A higher precision indicates that the predictive model is accurately identifying equipment failures or anomalies, which helps maintenance teams focus their efforts on the right issues. When implementing a RxM strategy, it is essential to balance precision with other metrics such as recall, which measures the model's ability to identify all actual positive cases. A model with high precision but low recall might miss important equipment failures, leading to unplanned downtime and increased maintenance costs. In summary, precision helps to ensure that RxM actions are targeted and effective by reducing false alarms. However, it should be considered alongside other performance metrics, such as recall and accuracy, to gain a comprehensive understanding of the model's overall impact on the maintenance strategy.

The F-measure is a performance metric that combines both precision and recall to provide a single, balanced measure of a classification model's performance. It is the harmonic mean of precision and recall. In the context of RxM, the F-measure is useful for evaluating the effectiveness of a maintenance model in identifying equipment failures or potential issues while minimizing false alarms. A higher F-measure indicates that the model performs well in terms of both precision (correct positive predictions) and recall (capturing all actual positive cases). Using the F-measure to assess a prescriptive maintenance model helps ensure that the model strikes a balance between identifying potential failures and reducing unnecessary maintenance actions. This balance is crucial for optimizing maintenance resources, minimizing costs, and reducing the risk of unexpected breakdowns or catastrophic failures. In summary, the F-measure serves as a valuable metric for evaluating the overall performance of a prescriptive maintenance model, as it accounts for both precision and recall. By striving to maximize the F-measure, maintenance teams can work towards improving the effectiveness of their RxM strategy, ultimately leading to better resource allocation and cost savings. Table 7 shows a summary of the performance measurements of the ML models.

Table 7
Summary of performance measurements for ML models

ML Model	Accuracy	Precision	Recall	F-measure
LightGBM	0.93	0.94	0.93	0.93
XGBoost	0.92	0.90	0.92	0.90
AdaBoost	0.82	0.84	0.82	0.82
MRF	0.81	0.83	0.81	0.81
DT	0.82	0.82	0.82	0.82
SVM	0.64	0.75	0.64	0.63
SGD	0.63	0.75	0.63	0.64
MLR	0.88	0.86	0.88	0.87
KNN	0.68	0.73	0.68	0.69
MNB	0.63	0.71	0.63	0.61
QDA	0.86	0.89	0.86	0.87
LDA	0.81	0.83	0.81	0.81

Based on the results in Table 7, we conclude that LightGBM had the highest performance, followed by XGBoost (above 90% accuracy and recall for the three of them). The reasons for LightGBM having the highest performance than other gradient boosting algorithms and having a better performance than the rest of ML models lies in the fact that LightGBM is designed to handle large datasets efficiently, making it a suitable choice for scenarios with a large number of data points or high-dimensional data. In addition, LightGBM uses an innovative sampling technique called Gradient-based One-Side Sampling. GOSS retains instances with large gradients while randomly sampling instances with small gradients. This strategy speeds up the training process while maintaining the quality of the model. LightGBM uses an Exclusive Feature Bundling (EFB) technique, which bundles mutually exclusive features together to reduce the number of features the algorithm has to handle. This technique further reduces memory consumption and training time. It is also noticed that SGD, SVM, KNN, and MNB had the lowest performance (less than 70% accuracy values for the four of them). On the other hand, the rest of the ML models (LDA, QDA, MLR, MRF, and DT) have achieved a good performance measurement with accuracies and recall values between 80% and 89%. It's essential to note that the performance of any algorithm depends on the specific problem and dataset. It is always a good practice to try multiple algorithms and compare their performance to choose the best one for your specific use case. Table 8 shows a summary of the performance measurements of the DL models.

Table 8
Summary of performance measurements for DL models

Models	Accuracy	Precision	Recall	F-measure
MLP	0.61	0.73	0.61	0.60
CNN-LSTM	0.60	0.72	0.60	0.60
2CNN	0.68	0.66	0.68	0.64
LVQ	0.63	0.76	0.63	0.63
2CNN-ALSTM	0.61	0.71	0.61	0.60

It can be seen from Table 8 that DL models have performed poorly compared to ML models. Thus, putting DL models at a disadvantage when it comes to failure detection compared to ML models. DL models typically require large amounts of data to learn the underlying patterns effectively. With smaller datasets, they may overfit or underfit the data, leading to suboptimal performance compared to ML models that can work well with smaller datasets. In cases where the problem or the underlying patterns in the data are simple, a traditional ML model might be sufficient to capture these patterns and perform well, without the need for a more complex DL model. It is crucial to recognize that the performance of DL and ML models is highly dependent on the specific problem, dataset, and model architecture. In some cases, DL models may significantly outperform ML models, while in others, the reverse may be true. It is generally recommended to explore multiple models and approaches to find the best solution for a given problem. All proposed DL models showed performance measurement values between 60% and 73%. Table 9 shows a summary of the performance measurements of the DHL models.

Table 9
Summary of performance measurements for DHL models

Models	Accuracy	Precision	Recall	F-measure
ALSTM-FCN with XGBoost	0.81	0.83	0.81	0.81
ALSTM-FCN with AdaBoost	0.75	0.76	0.75	0.76
CNN with XGBoost	0.78	0.80	0.78	0.78
DF	0.90	0.88	0.90	0.89

Table 9 shows DHL models have a performance level between ML and DL models. However, it did perform better than the DL model but not as well as ML models. All proposed DHL models showed performance measurement values between 75% and 90%. DL and ML models have different strengths and weaknesses, and combining them in a DHL model can be a two edged sowrd. However, in some cases, it can capitalize on their complementary strengths. For example in the case of DF, the DL part can be good at learning complex patterns and representations, while ML part can be more interpretable and require less computation. The performance of DHL models is highly dependent on the specific problem, dataset, and model architecture. In some cases, DHL models may outperform both standalone DL and ML models, while in others, they may perform somewhere in between. The key is to carefully design and evaluate the DHL model based on the specific requirements of the task at hand. Figure 14 shows a summary of average accuracies for failure detection for all models, Fig. 15 shows a comparison summary of performance measurements of all models, and Fig. 16 shows a comparison of detection accuracy values for each of the five failure types for all models.

Based on the values obtained from Fig. 16, it can be seen that some models have performed severely inadequate (accuracy less than 50%) in detecting certain types of failures. In comparison, other models have outperformed the rest in detecting certain types of failures. For example, RNF had less than 50% detection accuracy through all proposed models except for SGD and LVQ, where both scored a detection accuracy north of 99%. XGBoost and LightGBM scored a detection accuracy similar to that of SGD and LVQ but when detecting TWF. At the same time, XGBoost and MLR achieved similar high accuracy in detecting PWF. On the other hand, MLR, MNB, and QDA have resulted in a near perfect accuracy of 99% in detecting OSF. Finally, QDA achieved a similar accuracy in detecting HDF. Overall, RNF failure had the lowest detection accuracy since it was the only failure where a few of the models (for example, DF, GB, SVM, MLR, QDA, LDA, and MLP) caompletly failed in detecting it yielding an accuracy of 0%. Table 10 shows a recommendation for which model is best to use in failure detection with respect to the five different type of failures that were presented in the dataset.

Table 10
Recommended models to use to predict certain failure types

Failure Type	Recommended Model
HDF	QDA
OSF	QDA, MNB, and MLR
PWF	MLR and XGBoost
TWF	XGBoost and LightGBM
RNF	LVQ and SGD

The output feature (also known as the target variable or label) or the failure type in our case can affect the performance of a model. The choice of output feature, its distribution, and its relationship with input features all play a role in determining the quality of the predictions made by the model. Since the output feature has more than two categories (five failure types), you are dealing with a multiclass classification problem. Some algorithms can handle multiclass classification better than other algorithms. Understanding the nature of the categorical output feature and its relationship with input features is essential to build an appropriate model and select suitable techniques to maximize performance.

Conclusion

This paper evaluated the fidelity and efficiency of an array of ML, DL, and DHL models in early failure detection task on a synthetic dataset. All models have shown inherent strengths and weaknesses in detecting failures. For example, some models showed very high accuracies (of up to 99%, while others showed heavily poor performance (0% – 50%) in detecting failures. Future work can be done to confirm the reliability of such a RxM approach through a design of experimetns approach. In addition, future work can focus on applying the same models to different datasets and validating their efficiency. In summary future work can address the following areas:

- Advanced algorithms: Continuously research and explore new ML, DL, and DHL algorithms, architectures, and techniques that can lead to improved performance and adaptability in RxM applications.
- Feature engineering: Investigate more sophisticated feature engineering techniques to better represent the input data and capture underlying patterns, which can lead to improved model performance.
- Transfer learning: Utilize transfer learning and pre-trained models to leverage knowledge from similar domains, reducing training time and potentially improving model performance in RxM applications.
- Ensemble learning: Combine multiple ML, DL, and DHL models using ensemble learning techniques like bagging, boosting, or stacking to improve overall model performance and reduce overfitting.
- Model interpretability: Develop models with improved interpretability to better understand the factors influencing RxM predictions and facilitate more effective decision-making.
- Incorporate domain knowledge: Work closely with domain experts to incorporate valuable domain knowledge into the modeling process, leading to more accurate and useful RxM models.
- Real-time learning: Implement real-time learning and online learning algorithms that can adapt to changes in the underlying data and system dynamics, allowing for more accurate and timely RxM predictions.
- Edge computing: Utilize edge computing to perform ML, DL, and DHL model training and inference on edge devices, reducing latency and enabling real-time RxM applications.
- Scalability: Develop models and techniques that can scale efficiently to handle large-scale RxM applications with vast amounts of data and numerous equipment assets.
- Integration with other technologies: Combine RxM models with other tools and technologies, such as IoT, digital twins, resource allocation, optimiation, and augmented reality, to yield more results and to create more comprehensive and effective RxM solutions.

By continuously researching and integrating these advancements into RxM applications, practitioners can further improve the effectiveness and efficiency of maintenance processes, ultimately leading to reduced downtime, cost savings, and increased asset lifespan.

Declarations

Funding

The reported research work received partial financial support from the Office of Naval Research MEEP Program (Award Number: N00014-19-1-2728) as well as from the Lutcher Brown Distinguished Chair Professorship fund of the University of Texas at San Antonio.

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Contributions of the Authors

Mohammad Shahin took care of conceptualization, methodology, investigation, original draft, and final revisions. Ali Hosseinzadeh took care of the dataset and the review. Neda Zand helped in methodology. F. Frank Chen contributed in resources. Finally, all authors read and approved the final manuscript.

References

1. Carrasqueira M, Cruz-Machado V (2008) Strategic logistics: Re-designing companies in accordance with Lean Principles. International Journal of Management Science and Engineering Management 3:. https://doi.org/10.1080/17509653.2008.10671056
2. Sanders A, Elangeswaran C, Wulfsberg J (2016) Industry 4.0 implies lean manufacturing: research activities in Industry 4.0 function as enablers for lean manufacturing,. Journal of Industrial Engineering and Management 9:811–833
3. Shahin M, Chen FF, Hosseinzadeh A (2023) Machine-based identification system via optical character recognition. Flexible Services and Manufacturing Journal. https://doi.org/10.1007/s10696-023-09497-8
4. Dombrowski U, Malorny C (2014) Lean after sales service – An opportunity for OEMs to ensure profits. In B. Grabot, B. Vallespir, S. Gomes, A. Bouras, & D. Kiritsis (Eds.),. In: Advances in production management systems innovative and knowledge-based production management in a global-local world. Springer, Berlin, pp 618–625
5. McConnell K, Lindrooth R, Wholey D, Maddox T (2013) Management practices and the quality of care in cardiac units. JAMA Internal Medicine 173:684–692. https://doi.org/10.1001/jamainternmed.2013.3577
6. Womack J, Jones D (2003) Lean thinking: Banish waste and create wealth in your corporation, 2nd ed. Free Press, Simon & Schuster., New York
7. Zuehlke D (2010) SmartFactory-Towards a factory-of-things. Annual Reviews in Control 34:129–138
8. Hopp W, Spearman M (2004) Commissioned paper to pull or not to pull: What is the question? Manufacturing & Service Operations Management 6:133–148

9. De Treville S, Antonakis J (2006) Could lean production job design be intrinsically motivating? Contextual, configurational, and levels-of-analysis issues. Journal of Operations Management 24:99–123. https://doi.org/10.1016/j.jom.2005.04.001
10. : Invisible waste. In: EWWR. https://ewwr.eu/thematic_focus/2020-invisible-waste/. Accessed 6 Jul 2022
11. Graban M (2012) The Invisible Waste in Healthcare Leads to Visible Harm to Patients. In: Lean Blog. https://www.leanblog.org/2012/10/the-invisible-waste-in-healthcare-with-visible-harm/. Accessed 6 Jul 2022
12. Abhishek D, Vikas D, Alakshendra P (2015) Lean Manufacturing: An Approach for Waste Elimination. International Journal of Engineering Research & Technology (IJERT) 4:1–5. https://doi.org/10.17577/IJERTV4IS040817
13. Shahin M, Chen FF, Hosseinzadeh A, Maghanaki M (2023) Waste Reduction via Computer Vision-based Inspection: Towards Lean Systems in Metal Production
14. Salma A, Anas C, Mohammed EH (2021) Bibliographic Study on the Difficulties Encountered by SMEs During the Implementation of Lean Manufacturing. Journal of Advanced Manufacturing Systems 20:163–190. https://doi.org/10.1142/S0219686721500098
15. He Y, Changchao G, Zhaoxiang C (2017) Integrated predictive maintenance strategy for manufacturing systems by combining quality control and mission reliability analysis. International Journal of Production Research 55:5841–5862. https://doi.org/10.1080/00207543.2017.1346843
16. Froger A, Gendreau J, Mendoza E (2016) Maintenance Scheduling in the Electricity Industry: A Literature Review. European Journal of Operational Research 251:695–706
17. Ding S, Kamaruddin S (2014) Maintenance Policy Optimization – Literature Review and Directions. The International Journal of Advanced Manufacturing Technology 76:1263–1283
18. Mechefske C, Wang Z (2001) Using Fuzzy Linguistics to Select Optimum Maintenance and Condition Monitoring Strategies. Mechanical Systems & Signal Processing 15:1129–1140
19. Prajapati A(1), Bechtel J(2), Ganesan S(3) (2012) Condition based maintenance: A survey. Journal of Quality in Maintenance Engineering 18:384–400. https://doi.org/10.1108/13552511211281552
20. Peng Y, Dong M, Zuo J (2010) Current Status of Machine Prognostics in Condition-based Maintenance: A Review. International Journal of Advanced Manufacturing Technology 50:297–313
21. Kothamasu R, Huang SH, VerDuin WH (2006) System health monitoring and prognostics — a review of current paradigms and practices. The International Journal of Advanced Manufacturing Technology 28:1012–1024. https://doi.org/10.1007/s00170-004-2131-6
22. Ferreiro S, Konde E, Prado A (2016) Industry 4.0: Predictive Intelligent Maintenance for Production Equipment
23. Javed K, Gouriveau R, Zerhouni N (2017) State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology

readiness levels. Mechanical Systems and Signal Processing 94:214–236. https://doi.org/10.1016/j.ymssp.2017.01.050

24. Sakib N, Wuest T (2018) Challenges and Opportunities of Condition-based Predictive Maintenance: A Review. Procedia CIRP 78:267–272. https://doi.org/10.1016/j.procir.2018.08.318
25. Alaswad S, Xiang Y (2017) A review on condition-based maintenance optimization models for stochastically deteriorating system. Reliability Engineering and System Safety 157:54–63. https://doi.org/10.1016/j.ress.2016.08.009
26. Gouriveau R, Medjaher K, Zerhouni N (2016) From prognostics and health systems management to predictive maintenance 1 : monitoring and prognostics. Wiley
27. Wang X, Li L, Xie M (2019) Optimal preventive maintenance strategy for leased equipment under successive usage-based contracts. International Journal of Production Research 57:5705–5724. https://doi.org/10.1080/00207543.2018.1542181
28. Albrice D, Branch M (2015) A Deterioration Model for Establishing an Optimal Mix of Time-Based Maintenance (TbM) and Condition-Based Maintenance (CbM) for the Enclosure System
29. Montero Jimenez JJ, Schwartz S, Vingerhoeds R, et al (2020) Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. Journal of Manufacturing Systems 56:539–557. https://doi.org/10.1016/j.jmsy.2020.07.008
30. Momber AW, Möller T, Langenkämper D, et al (2022) A Digital Twin concept for the prescriptive maintenance of protective coating systems on wind turbine structures. Wind Engineering 46:949–971. https://doi.org/10.1177/0309524X211060550
31. Cho A d., Carrasco R a., Ruz G a. (2022) Improving Prescriptive Maintenance by Incorporating Post-Prognostic Information Through Chance Constraints. IEEE Access, Access, IEEE 10:55924–55932. https://doi.org/10.1109/ACCESS.2022.3177537
32. Choubey S, Benton R, Johnsten T (2019) Prescriptive Equipment Maintenance: A Framework. 2019 IEEE International Conference on Big Data (Big Data), Big Data (Big Data), 2019 IEEE International Conference on 4366–4374. https://doi.org/10.1109/BigData47090.2019.9006213
33. Consilvio A, Sanetti P, Anguita D, et al (2019) Prescriptive Maintenance of Railway Infrastructure: From Data Analytics to Decision Support. 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2019 6th International Conference on 1–10. https://doi.org/10.1109/MTITS.2019.8883331
34. Livera A, Theristis M, Charalambous A, et al (2021) Decision support system for corrective maintenance in large-scale photovoltaic systems. 2021 IEEE 48th Photovoltaic Specialists Conference (PVSC), Photovoltaic Specialists Conference (PVSC), 2021 IEEE 48th 0306–0311. https://doi.org/10.1109/PVSC43889.2021.9518796
35. Meissner R, Rahn A, Wicke K (2021) Developing prescriptive maintenance strategies in the aviation industry based on a discrete-event simulation framework for post-prognostics decision making. Reliability Engineering and System Safety 214:. https://doi.org/10.1016/j.ress.2021.107812

36. Kosky P, Robert B, Wise G (2020) Exploring Engineering - An Introduction to Engineering and Design, Fifth. Elsevier Inc.
37. Theissler A, Elger G, Kettelgerdes M (2021) Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. Reliability Engineering & System Safety 215:. https://doi.org/10.1016/j.ress.2021.107864
38. Atay H, Arslankayaa S (2015) Maintenance management and lean manufacturing practices in a firm which produces dairy products. Procedia - Social and Behavioral Sciences, pp 214–225
39. Xia L, Zheng P, Li X, et al (2022) Toward cognitive predictive maintenance: A survey of graph-based approaches. Journal of Manufacturing Systems 64:107–120. https://doi.org/10.1016/j.jmsy.2022.06.002
40. Pedersen E, Huniche, M (2011) Determinants of lean success and failure in the Danish public sector. International Journal of Public Sector Management, 24:403–420. https://doi.org/10.1108/09513551111147141
41. Li Z, Wang K, He Y (2016) Industry 4.0 - Potentials for Predictive Maintenance. 6th International Workshop of Advanced Manufacturing and Automation. https://doi.org/10.2991/iwama-16.2016.8
42. Powell D, Skjelstad L (2012) RFID for the extended lean enterprise. International Journal of Lean Six Sigma 3:172–186. https://doi.org/10.1108/20401461211282691
43. Mostafa S, Lee S, Dumrak J, et al (2015) Lean thinking for a maintenance process. Production & Manufacturing Research 3:236–272. https://doi.org/10.1080/21693277.2015.1074124
44. Shahin M, Chen FF, Bouzary H, Krishnaiyer K (2020) Integration of Lean practices and Industry 4.0 technologies: smart manufacturing for next-generation enterprises. Int J Adv Manuf Technol 107:2927–2936. https://doi.org/10.1007/s00170-020-05124-0
45. Lee C, Cao Y, Ng K (2017) Supply Chain Management in the Big Data Era. In: Big Data Analytics for Predictive Maintenance Strategies. IGI Global
46. Müller J, Kiel D, Voigt K (2018) What Drives the Implementation of Industry 4.0? The Role of Opportunities and Challenges in the Context of Sustainability. Sustainability 10:. https://doi.org/10.3390/su10010247
47. Abidi M, Mohammed M, Alkhalefah H (2022) Predictive Maintenance Planning for Industry 4.0 Using Machine Learning for Sustainable Manufacturing. Sustainability 14:1–27
48. Ferreira C, Gonçalves G (2022) Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. Journal of Manufacturing Systems 63:550–562. https://doi.org/10.1016/j.jmsy.2022.05.010
49. Sahli A, Evans R, Manohar A (2021) Predictive Maintenance in Industry 4.0: Current Themes. Procedia, pp 1948–1953
50. Mentzas G, Bousdekis A (2017) Condition-Based Predictive Maintenance in the Frame of Industry 4.0. IFIP International Conference on Advances in Production Management Systems 513:

51. Fernandes M, Canito A, Canedo V, Marreiors G (2019) Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry. International Journal of Information Management 46:252–262
52. Zhai S, Gehring B, Reinhart G (2021) Enabling predictive maintenance integrated production scheduling by operation-specific health prognostics with generative deep learning. Journal of Manufacturing Systems 61:830–855. https://doi.org/10.1016/j.jmsy.2021.02.006
53. Leukel J, González J, Riekert M (2021) Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. Journal of Manufacturing Systems 61:87–96. https://doi.org/10.1016/j.jmsy.2021.08.012
54. Hashemian H, Bean W (2011) State-of-the-art predictive maintenance techniques. IEEE Transactions on Instrumentation and measurement 60:3480–3492
55. Wang X, Zhang Y, Wang L, et al (2019) Maintenance grouping optimization with system multi-level information based on BN lifetime prediction model. Journal of Manufacturing Systems 50:201–211. https://doi.org/10.1016/j.jmsy.2019.01.002
56. Selcuk S (2017) Predictive maintenance, its implementation and latest trends. Journal of Engineering Manufacture 231:1670–1679
57. Wang K (2013) Toward Zero-defect manufacuturing - a data mining approach. Advances in Manufacturing 1:62–74
58. Alfeo AL, Cimino MGCA, Vaglini G (2022) Degradation stage classification via interpretable feature learning. Journal of Manufacturing Systems 62:972–983. https://doi.org/10.1016/j.jmsy.2021.05.003
59. Wu J-Y, Wu M, Chen Z, et al (2021) A joint classification-regression method for multi-stage remaining useful life prediction. Journal of Manufacturing Systems 58:109–119. https://doi.org/10.1016/j.jmsy.2020.11.016
60. Tiddens W, Braaksma J, Tinga T (2020) Exploring predictive maintenance applications in industry. Journal of Quality in Maintenance Engineering 26:
61. Bekar E, Skoogh A, Nyqvist P (2020) An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study. Advances in Mechanical Engineering 12:1–14
62. Explainable Artificial Intelligence for Predictive...: Results. https://resolver-ebscohost-com.libproxy.txstate.edu/openurl?sid=EBSCO%3aedseee&genre=book&issn=edseee.IEEEConferenc&ISBN=9781728187013&volume=&issue=&date=&spage=69&pages=69-74&title=2020+Third+International+Conference+on+Artificial+Intelligence+for+Industries+(AI4I)%2c+Artificial+Intelligence+for+Industries+(AI4I)%2c+2020+Third+International+Conference+on%2c+AI4I&atitle=Explainable+Artificial+Intelligence+for+Predictive+Maintenance+Applications&aulast=Matzka%2c+Stephan&id=DOI%3a10.1109%2fAI4I49448.2020.00023&site=ftf-live. Accessed 6 Jul 2022

63. Dua D, Graff C (2019) AI4I 2020 Predictive Maintenance Dataset. In: UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science
64. Pastorino J, Biswas AK (2020) Hey ML, what can you do for me? 2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Artificial Intelligence and Knowledge Engineering (AIKE), 2020 IEEE Third International Conference on, AIKE 116–119. https://doi.org/10.1109/AIKE48582.2020.00023
65. Waldrop MM (2019) What are the limits of deep learning? Proceedings of the National Academy of Sciences of the United States of America 116:1074–1077
66. Majlesi A, Koodiani HK, de Rincon OT, et al (2023) Artificial neural network model to estimate the long-term carbonation depth of concrete exposed to natural environments. Journal of Building Engineering 106545
67. Shinde K, Thakare A (2021) Deep Hybrid Learning Method for Classification of Fetal Brain Abnormalities. 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), Artificial Intelligence and Machine Vision (AIMV), 2021 International Conference on 1–6. https://doi.org/10.1109/AIMV53313.2021.9670994
68. Bhattacharya A (2021) Deep Hybrid Learning — a fusion of conventional ML with state of the art DL. In: Medium. https://towardsdatascience.com/deep-hybrid-learning-a-fusion-of-conventional-ml-with-state-of-the-art-dl-cb43887fe14. Accessed 25 Feb 2022
69. Adib QAR, Mehedi MdHK, Sakib MdS, et al (2021) A Deep Hybrid Learning Approach to Detect Bangla Fake News. In: 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). pp 442–447
70. Zonta T, da Costa CA, Zeiser FA, et al (2022) A predictive maintenance model for optimizing production schedule using deep neural networks. Journal of Manufacturing Systems 62:450–462. https://doi.org/10.1016/j.jmsy.2021.12.013
71. Han T, Pang J, Tan ACC (2021) Remaining useful life prediction of bearing based on stacked autoencoder and recurrent neural network. Journal of Manufacturing Systems 61:576–591. https://doi.org/10.1016/j.jmsy.2021.10.011
72. Reinforcement Learning for Predictive Maintenance of Industrial Plants. Information Technologies and Control 11:21–28. https://doi.org/10.2478/itc-2013-0004
73. Zhao R(1), Wang D(1), Mao K(1), et al (2017) Machine health monitoring using local feature-based gated recurrent unit networks. IEEE Transactions on Industrial Electronics 65:1539–1548. https://doi.org/10.1109/TIE.2017.2733438
74. Dong D, Li X-Y, Sun F-Q (2017) Life prediction of jet engines based on LSTM-recurrent neural networks. 2017 Prognostics and System Health Management Conference (PHM-Harbin), Prognostics and System Health Management Conference (PHM-Harbin), 2017 1–6. https://doi.org/10.1109/PHM.2017.8079264
75. Hinchi AZ, Tkiouat M (2018) Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network. Procedia Computer Science 127:123–132.

https://doi.org/10.1016/j.procs.2018.01.106

76. Zhang J, Wang P, Yan R, Gao RX (2018) Long short-term memory for machine remaining life prediction. Journal of Manufacturing Systems 48:78–86. https://doi.org/10.1016/j.jmsy.2018.05.011
77. Ren L, Sun Y, Cui J, Zhang L (2018) Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. Journal of Manufacturing Systems 48:71–77. https://doi.org/10.1016/j.jmsy.2018.04.008
78. Huang W, Cheng J, Yang Y, Guo G (2019) An improved deep convolutional neural network with multi-scale information for bearing fault diagnosis. Neurocomputing 359:77–92. https://doi.org/10.1016/j.neucom.2019.05.052
79. Li X, Ding Q, Sun J-Q (2018) Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering and System Safety 172:1–11. https://doi.org/10.1016/j.ress.2017.11.021
80. Lu F, Wu J, Huang J, Qiu X (2019) Aircraft engine degradation prognostics based on logistic regression and novel OS-ELM algorithm. Aerospace Science and Technology 84:661–671. https://doi.org/10.1016/j.ast.2018.09.044
81. Li S, Fang H, Shi B (2018) Multi-Step-Ahead Prediction with Long Short Term Memory Networks and Support Vector Regression. In: Chinese Control Conference, CCC. IEEE Computer Society, pp 8104–8109
82. Singh K, Malik H, Sharma R (2017) Condition monitoring of wind turbine gearbox using electrical signatures. 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Microelectronic Devices, Circuits and Systems (ICMDCS), 2017 International conference on 1–6. https://doi.org/10.1109/ICMDCS.2017.8211718
83. Gajewski J, Vališ D (2017) The determination of combustion engine condition and reliability using oil analysis by MLP and RBF neural networks. Tribology International 115:557–572. https://doi.org/10.1016/j.triboint.2017.06.032
84. Ayo-Imoru R m., Cilliers A c. (2018) Continuous machine learning for abnormality identification to aid condition-based maintenance in nuclear power plant. Annals of Nuclear Energy 118:61–70. https://doi.org/10.1016/j.anucene.2018.04.002
85. Luwei K c. (1), Yunusa-Kaltungo A(1), Sha'aban Y a. (2) (2018) Integrated fault detection framework for classifying rotating machine faults using frequency domain data fusion and Artificial Neural Networks. Machines 6:. https://doi.org/10.3390/MACHINES6040059
86. Onel M, Kieslich CA, Guzman YA, Pistikopoulos EN (2018) Simultaneous Fault Detection and Identification in Continuous Processes via nonlinear Support Vector Machine based Feature Selection. Computer Aided Chemical Engineering 44:2077–2082. https://doi.org/10.1016/B978-0-444-64241-7.50341-4
87. Jin W(1), Shi Z(1), Siegel D(1), et al (2015) Development and evaluation of health monitoring techniques for railway point machines. In: 2015 IEEE Conference on Prognostics and Health

Management: Enhancing Safety, Efficiency, Availability, and Effectiveness of Systems Through PHAf Technology and Application, PHM 2015. Institute of Electrical and Electronics Engineers Inc.

88. Duan C, Deng C, Li N (2019) Reliability assessment for CNC equipment based on degradation data. The International Journal of Advanced Manufacturing Technology 100:421–434. https://doi.org/10.1007/s00170-018-2548-y
89. von Birgelen A, Buratti D, Mager J, Niggemann O (2018) Self-Organizing Maps for Anomaly Localization and Predictive Maintenance in Cyber-Physical Production Systems. Procedia CIRP 72:480–485. https://doi.org/10.1016/j.procir.2018.03.150
90. Zhao Y, Wang S, Xiao F (2013) Pattern recognition-based chillers fault detection method using Support Vector Data Description (SVDD). APPLIED ENERGY 112:1041–1048. https://doi.org/10.1016/j.apenergy.2012.12.043
91. Laib dit Leksir Y, Mansour M, Moussaoui A (2018) Localization of thermal anomalies in electrical equipment using Infrared Thermography and support vector machine. Infrared Physics and Technology 89:120–128. https://doi.org/10.1016/j.infrared.2017.12.015
92. Xiao Y, Wang H, Xu W, Zhou J (2016) Robust one-class SVM for fault detection. Chemometrics and Intelligent Laboratory Systems 151:15–25. https://doi.org/10.1016/j.chemolab.2015.11.010
93. Zheng A, Casari A (2018) Feature engineering for machine learning : principles and techniques for data scientists, First edition. O'Reilly Media
94. Witten IH, Frank E, Hall MA, Pal CJ (2017) Data mining : practical machine learning tools and techniques, Fourth edition. Morgan Kaufmann
95. Ethem Alpaydin (2014) Introduction to Machine Learning. The MIT Press, Cambridge, MA
96. Kuhn M, Johnson K (2013) Applied predictive modeling. Springer
97. Loh W-Y, Shih Y-S Loh, W.-Y.and Shih, Y.-S. (1997). Split selection methods for classification trees. Vol.7, No.4. http://www3.stat.sinica.edu.tw/statistica/j7n4/j7n41/j7n41.htm. Accessed 17 Jun 2020
98. Kass GV (1980) An Exploratory Technique for Investigating Large Quantities of Categorical Data. Journal of the Royal Statistical Society Series C (Applied Statistics) 29:119–127. https://doi.org/10.2307/2986296
99. Salzberg SL (1994) C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Mach Learn 16:235–240. https://doi.org/10.1007/BF00993309
100. Breiman L (2001) Random Forests. Machine Learning 45:5–32. https://doi.org/10.1023/A:1010933404324
101. Bai J, Li Y, Li J, et al (2022) Multinomial random forest. Pattern Recognition 122:108331. https://doi.org/10.1016/j.patcog.2021.108331
102. Erigha ED, Ayo FE, Dada OO, Folorunso O (2017) Intrusion Detection System Based on Support Vector Machines and the Two-Phase Bat Algorithm. Journal of Information System Security 13:135–149
103. Wang L (2005) Support vector machines : theory and applications. Springer

104. Saha S, Singh K (2018) Detection of Intrusion by Stochastic Gradient Descent and K-means Clustering
105. Zhang P, Wang R, Xiu N (2022) Multinomial logistic regression classifier via lq,0-proximal Newton algorithm. Neurocomputing 468:148–164. https://doi.org/10.1016/j.neucom.2021.10.005
106. Chan HC, Chang CC, Chen PA, Lee JT (2019) Using multinomial logistic regression for prediction of soil depth in an area of complex topography in Taiwan. CATENA 176:419–429. https://doi.org/10.1016/j.catena.2019.01.030
107. Samui P, Sekhar S, Balas VE (2017) Handbook of Neural Computation. Elsevier
108. Koc L, Mazzuchi TA, Sarkani S (2012) A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. Expert Systems with Applications 39:13492–13500. https://doi.org/10.1016/j.eswa.2012.07.009
109. Friedman J, Hastie T, Tibshirani R (2000) Additive Logistic Regression: A Statistical View of Boosting. The Annals of Statistics 28:337–407. https://doi.org/10.1214/aos/1016218223
110. Friedman JH (2001) Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29:1189–1232. https://doi.org/10.1214/aos/1013203451
111. Chen T, Guestrin C (2016) XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 785–794. https://doi.org/10.1145/2939672.2939785
112. Freund Y, Schapire RE (1999) A Short Introduction to Boosting. In: In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann, pp 1401–1406
113. Freund Y, Schapire RE (1997) A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55:119–139. https://doi.org/10.1006/jcss.1997.1504
114. Ke G, Meng Q, Finley T, et al (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.
115. Altman EI (1968) Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. The Journal of Finance 23:589–609. https://doi.org/10.2307/2978933
116. Ghosh A, SahaRay R, Chakrabarty S, Bhadra S (2021) Robust generalised quadratic discriminant analysis. Pattern Recognition 117:107981. https://doi.org/10.1016/j.patcog.2021.107981
117. Svozil D, Kvasnicka V, Pospíchal J (1997) Introduction to multi-layer feed-forward neural networks. Chemometrics and Intelligent Laboratory Systems 39:43–62. https://doi.org/10.1016/S0169-7439(97)00061-0
118. Rumelhart DE, McClelland JL (1986) Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations. MIT Press, Cambridge, MA, USA
119. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations. MIT Press, Cambridge, MA, USA, pp 318–362

120. Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. Neural Computation 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
121. Bahdanau D(1), Cho K(2), Bengio Y(2) (2015) Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. International Conference on Learning Representations, ICLR
122. Yang S, Tan M, Xia S, Liu F (2020) A method of intrusion detection based on Attention-LSTM neural network. In: Proceedings of the 2020 5th International Conference on Machine Learning Technologies. Association for Computing Machinery, New York, NY, USA, pp 46–50
123. Ciaburro G (2017) Neural Networks with R. Packt Publishing
124. Shahin M, Chen FF, Bouzary H, et al (2022) A novel fully convolutional neural network approach for detection and classification of attacks on industrial IoT devices in smart manufacturing systems. The International Journal of Advanced Manufacturing Technology 1–13. https://doi.org/10.1007/s00170-022-10259-3
125. Shahin M, Chen Ff, Bouzary H, et al (2023) Implementation of a Novel Fully Convolutional Network Approach to Detect and Classify Cyber-Attacks on IoT Devices in Smart Manufacturing Systems. In: Kim K-Y, Monplaisir L, Rickli J (eds) Flexible Automation and Intelligent Manufacturing: The Human-Data-Technology Nexus. Springer International Publishing, Cham, pp 107–114
126. Fazle Karim, Somshubra Majumdar, Houshang Darabi (2019) Insights Into LSTM Fully Convolutional Networks for Time Series Classification. IEEE Access 7:67718–67725. https://doi.org/10.1109/ACCESS.2019.2916828
127. Pourghasemi HR, Gayen A, Lasaponara R, Tiefenbacher JP (2020) Application of learning vector quantization and different machine learning techniques to assessing forest fire influence factors and spatial modelling. Environmental Research 184:109321. https://doi.org/10.1016/j.envres.2020.109321
128. Akarslan E (2022) Learning Vector Quantization based predictor model selection for hourly load demand forecasting. Applied Soft Computing 108421. https://doi.org/10.1016/j.asoc.2022.108421
129. Esfandiarpour-Boroujeni I, Shamsabadi MS, Shirani H, et al (2020) Comparison of error and uncertainty of decision tree and learning vector quantization models for predicting soil classes in areas with low altitude variations. CATENA 191:104581. https://doi.org/10.1016/j.catena.2020.104581
130. Shahin M, Chen FF, Hosseinzadeh A, et al (2022) A deep hybrid learning model for detection of cyber attacks in industrial IoT devices. Int J Adv Manuf Technol 123:1973–1983. https://doi.org/10.1007/s00170-022-10329-6
131. Yu H, Chen C, Lu N (2021) Improving the Accuracy of Maintenance Decision-Making via Deep Forest-Based Failure Prognostics. 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST), Science and Technology Innovation (IAECST), 2021 3rd International Academic Exchange Conference on 525–529. https://doi.org/10.1109/IAECST54258.2021.9695668

132. Zhou Z-H, Feng J (2017) Deep Forest: Towards An Alternative to Deep Neural Networks. 3553–3559
133. Xia H, Tang J (2021) An Improved Deep Forest Regression. 2021 3rd International Conference on Industrial Artificial Intelligence (IAI), Industrial Artificial Intelligence (IAI), 2021 3rd International Conference on 1–6. https://doi.org/10.1109/IAI53119.2021.9619276

Figures

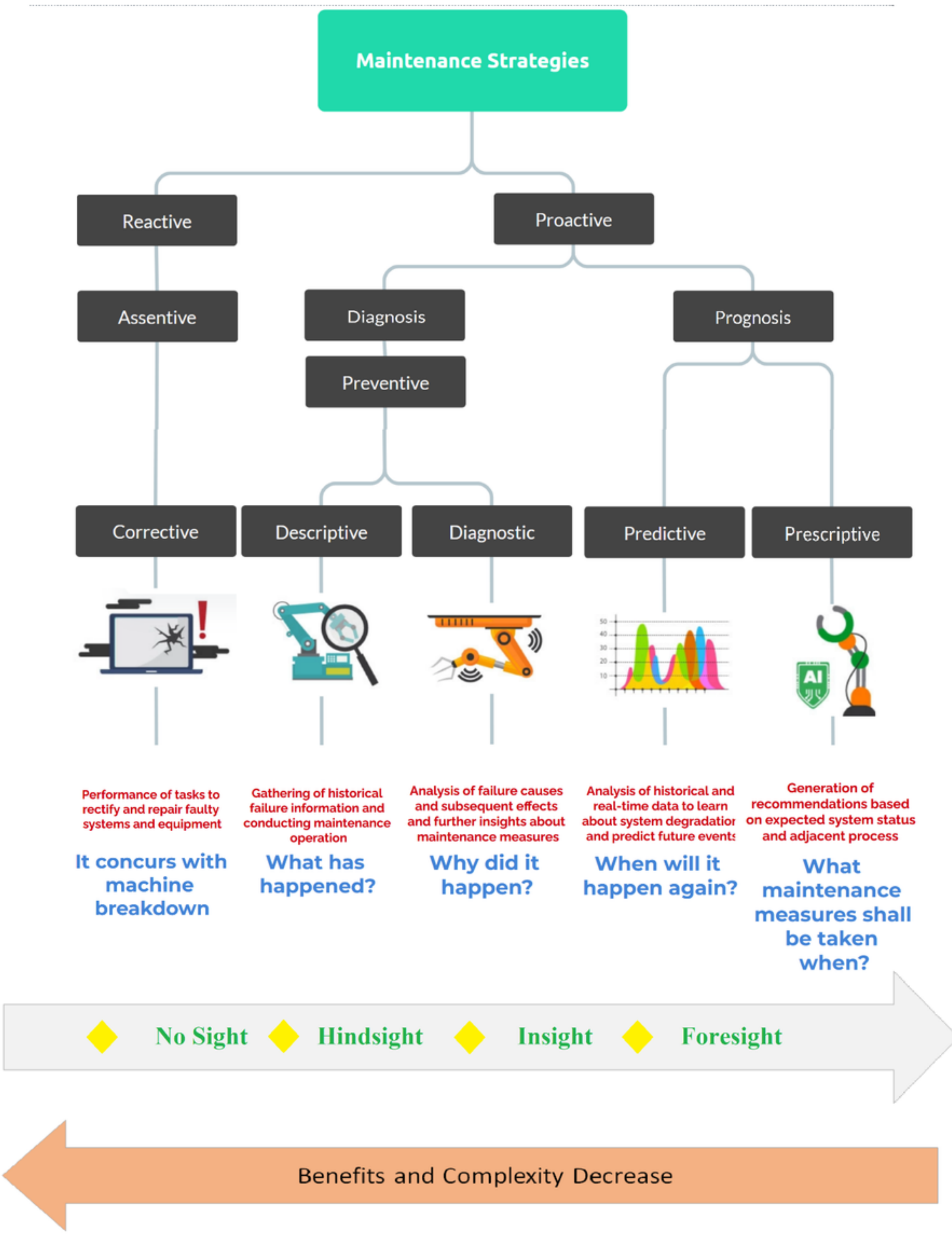

Figure 1

A summarized relationship between the different types of maintenance strategies and their development process

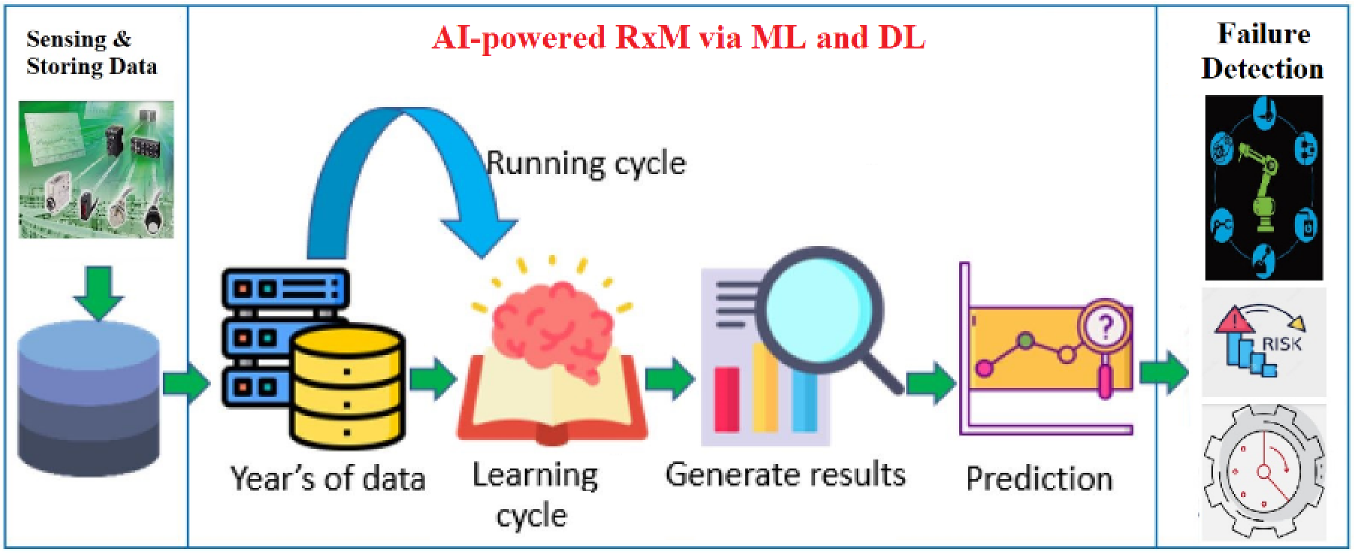

Figure 2

The framework of an AI-powered RxM via ML and DL

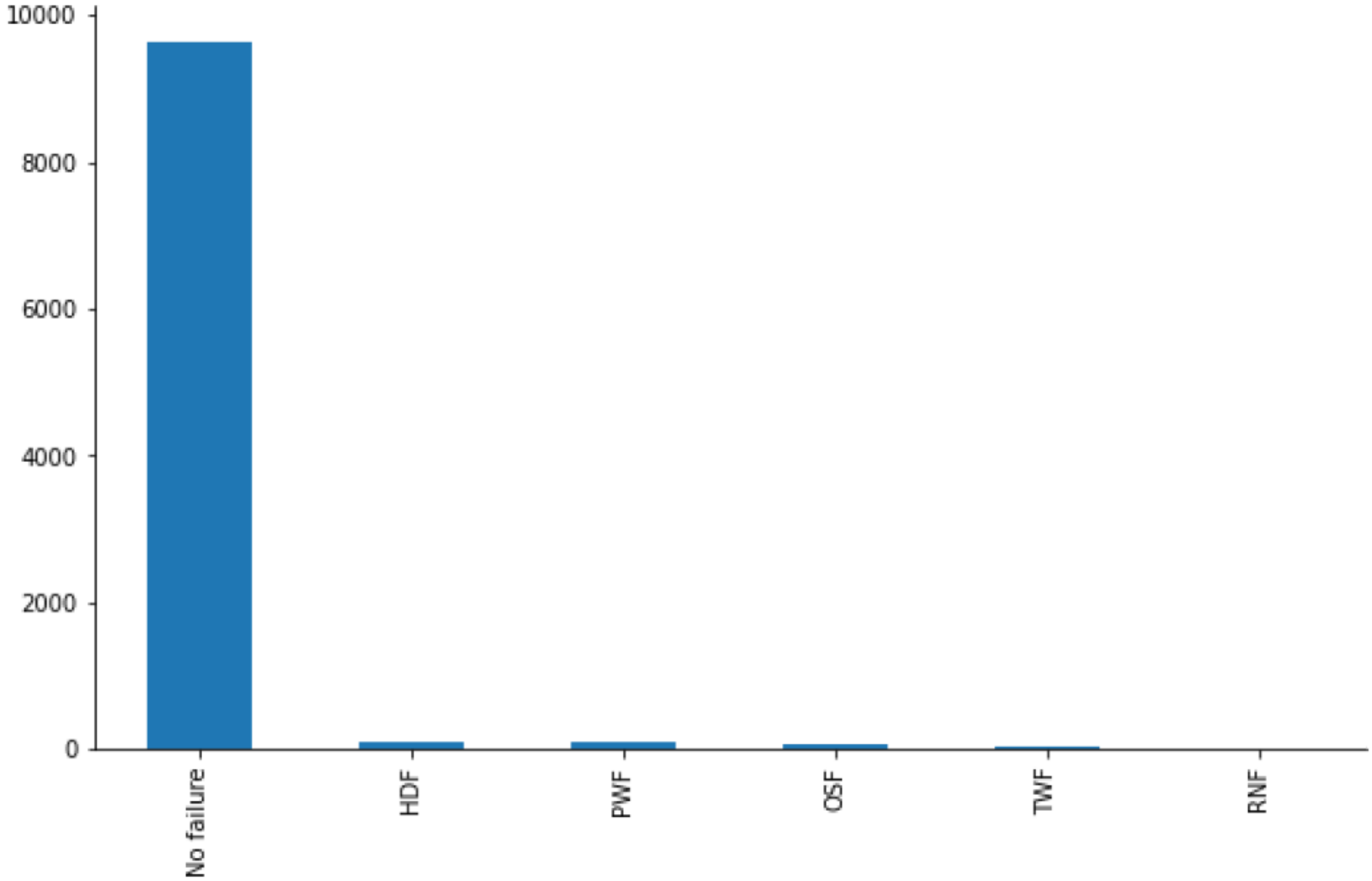

Figure 3

Statistical description of the failure features

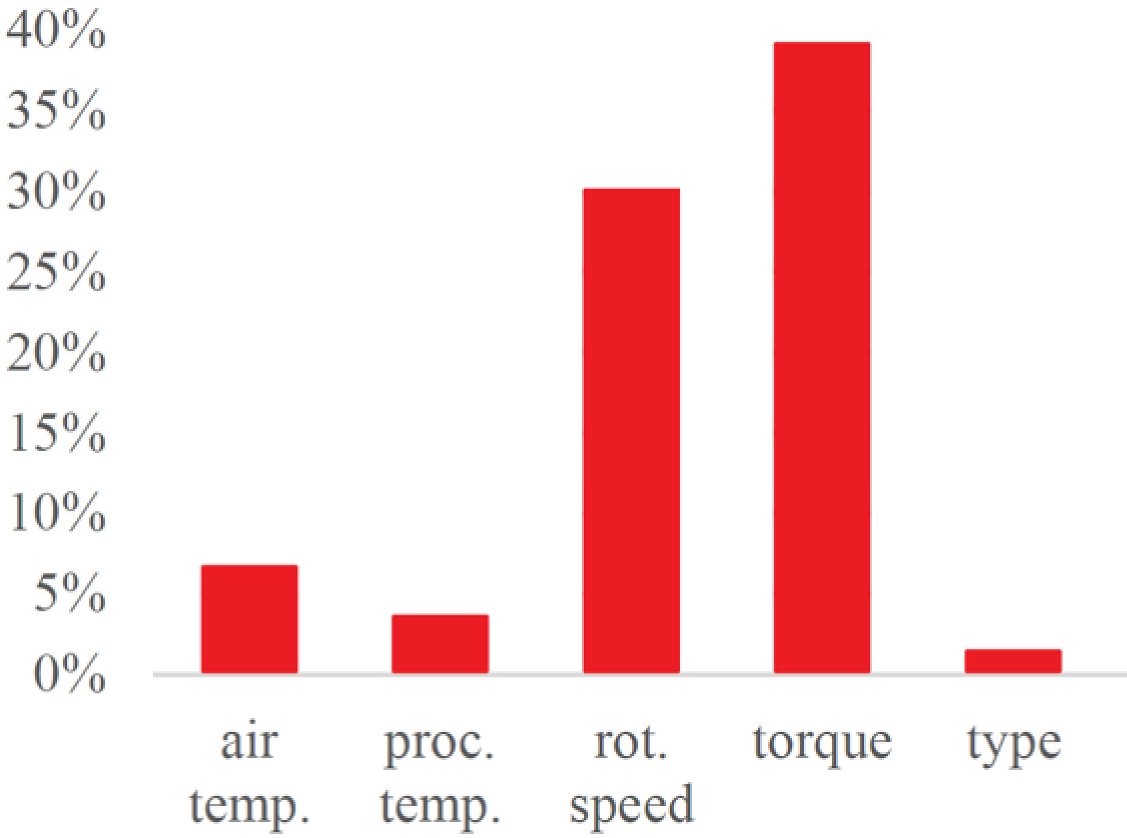

Figure 4

The six most crucial input features for ML models

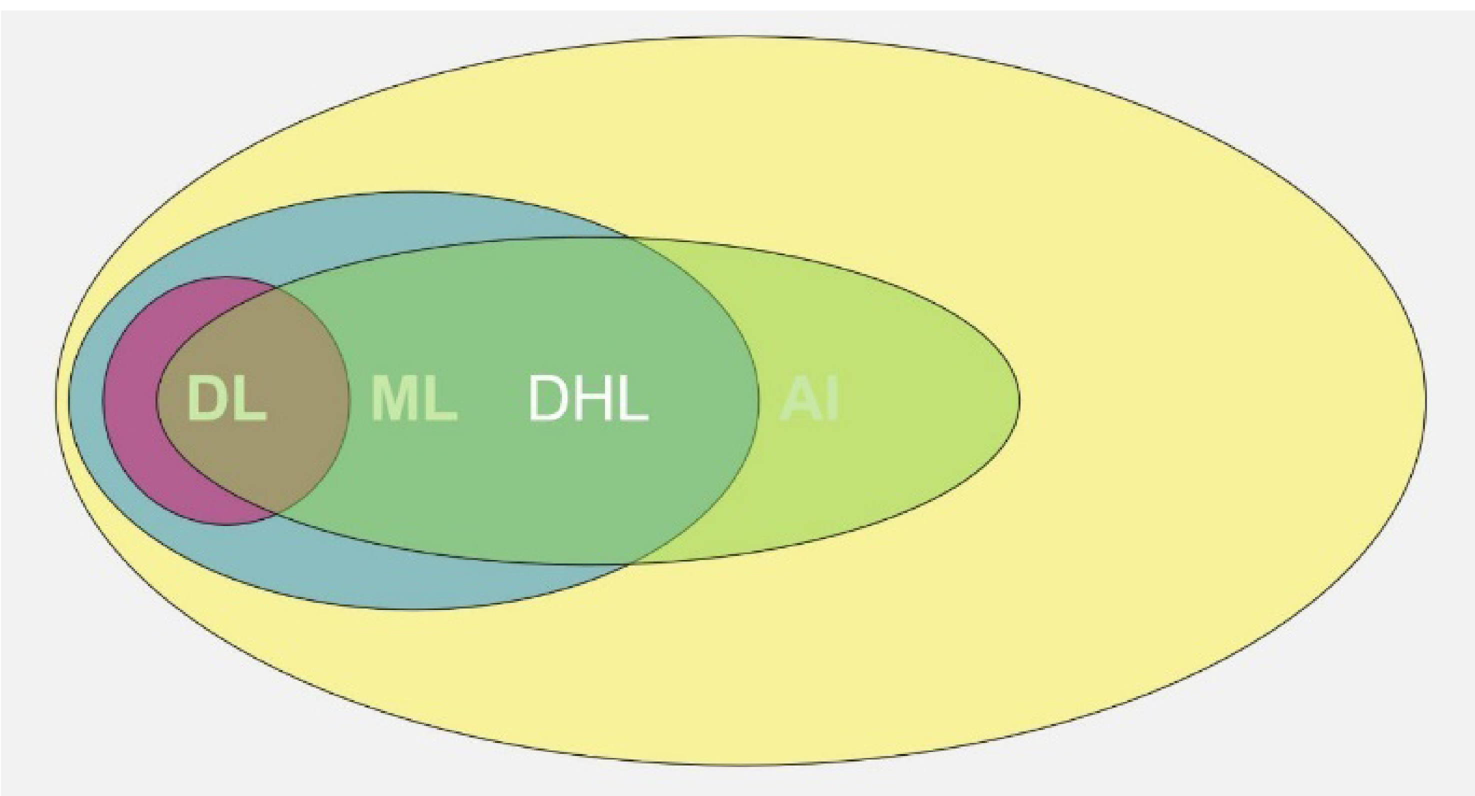

Figure 5

Relationship between DL, ML, DHL, and AI

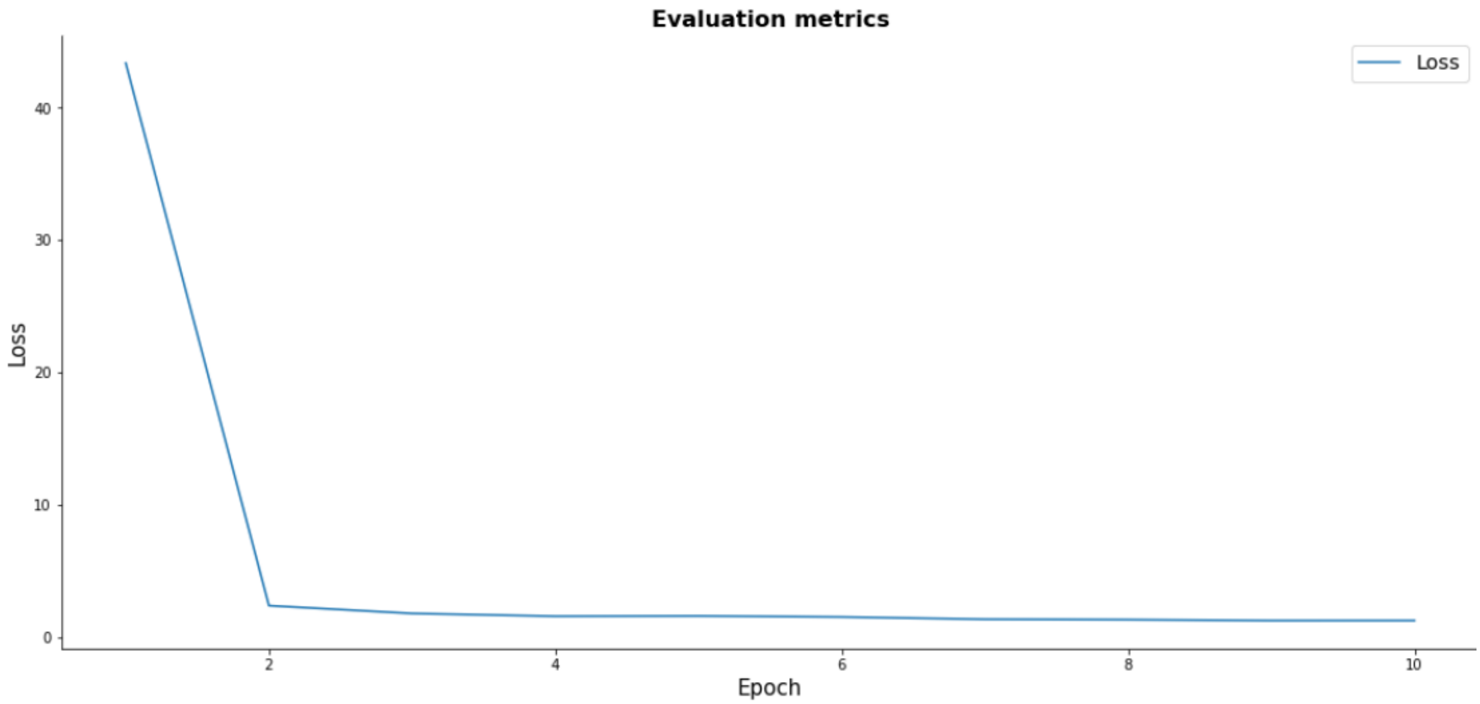

Figure 6

Fig. 10 Using multiple epochs to achieve minimum loss and highest performance measurements values for DL models

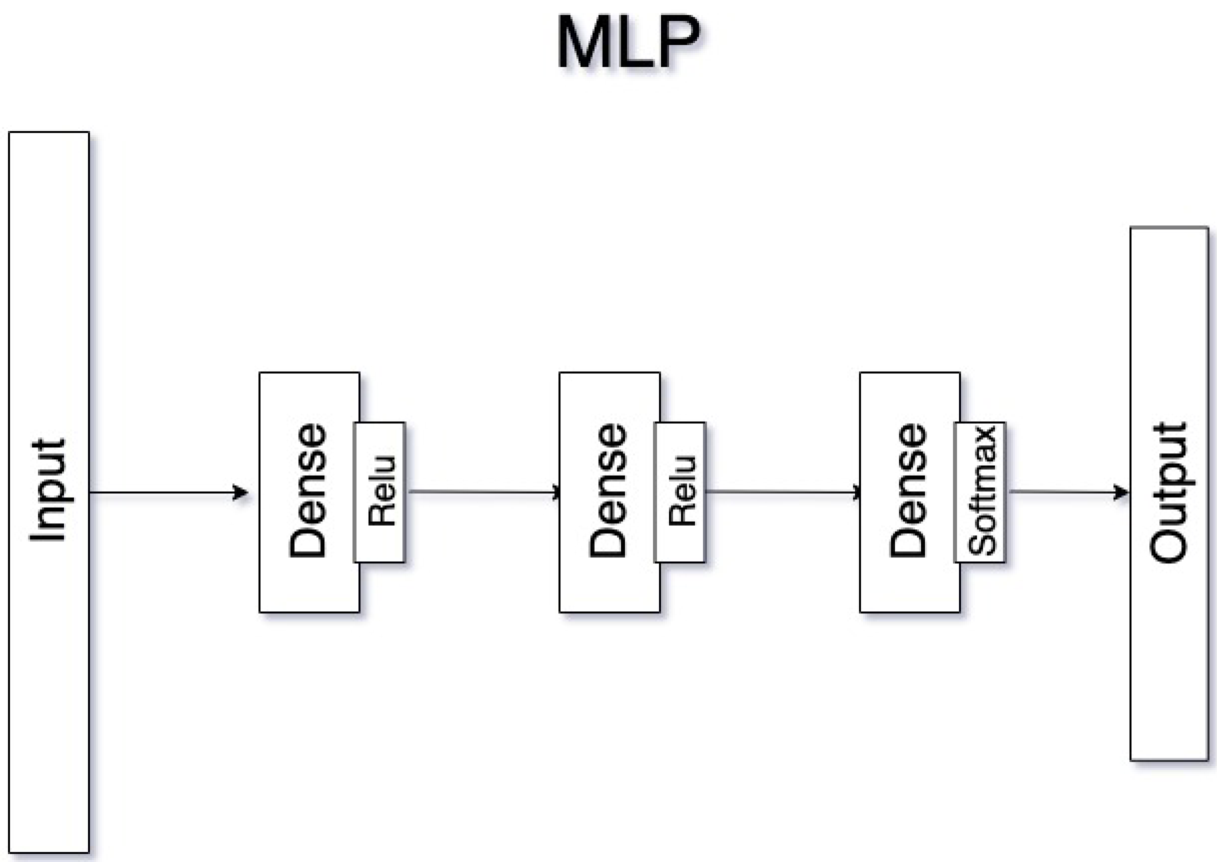

Figure 7

The proposed architecture of the MLP model

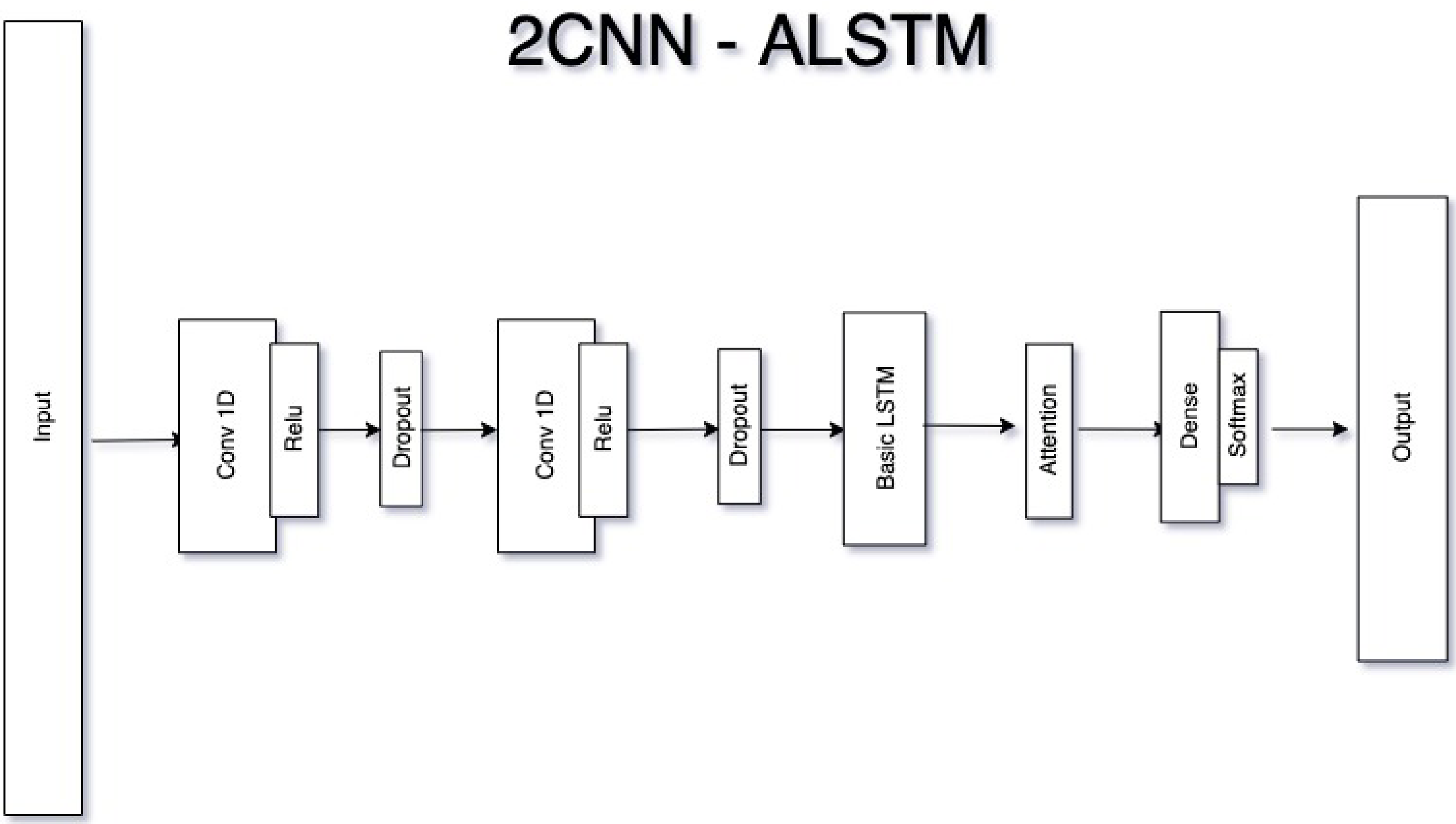

Figure 8

Architecture of the Two layers CNN-ALSTM model

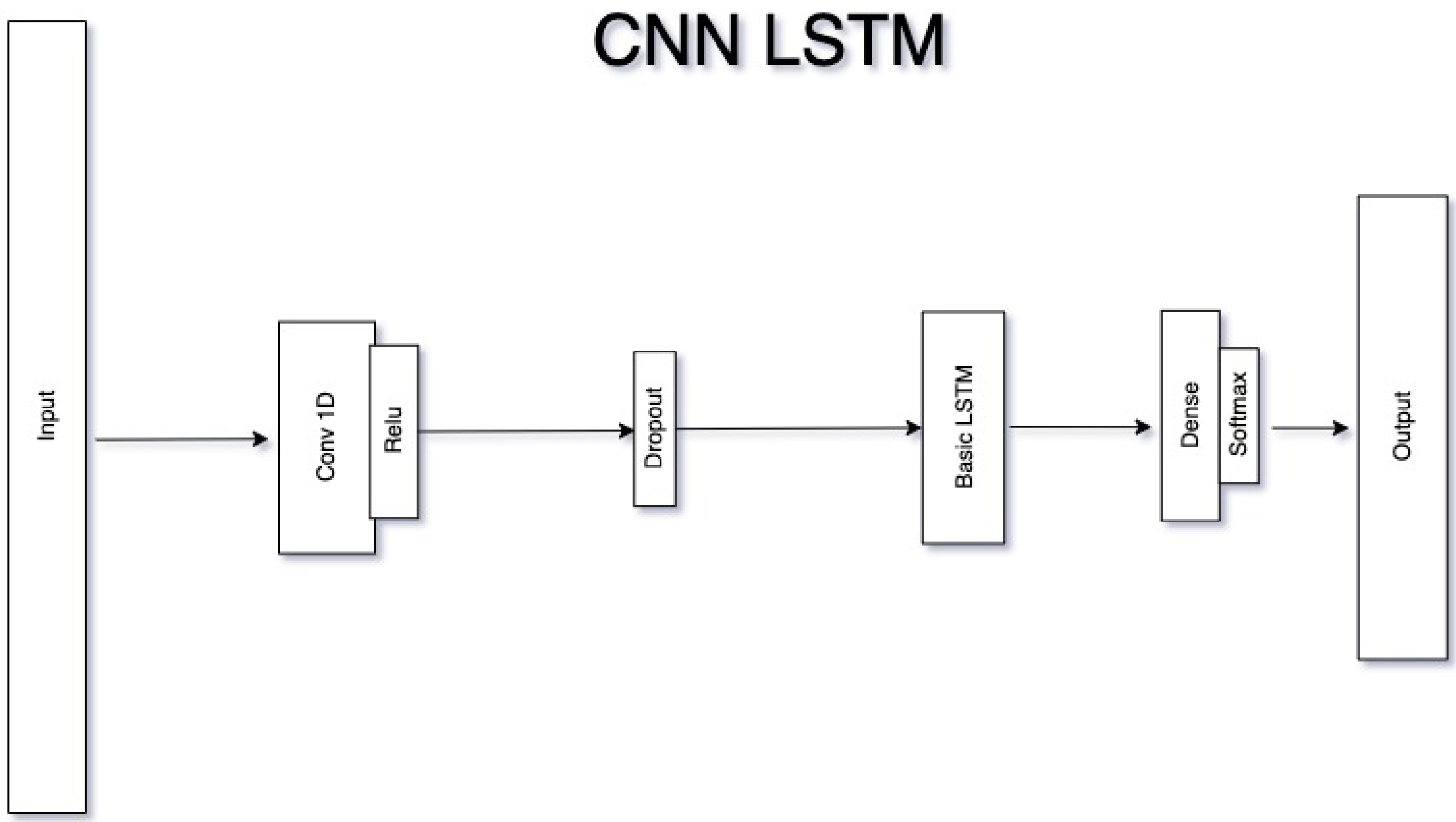

Figure 9

Architecture of the CNN-LSTM model

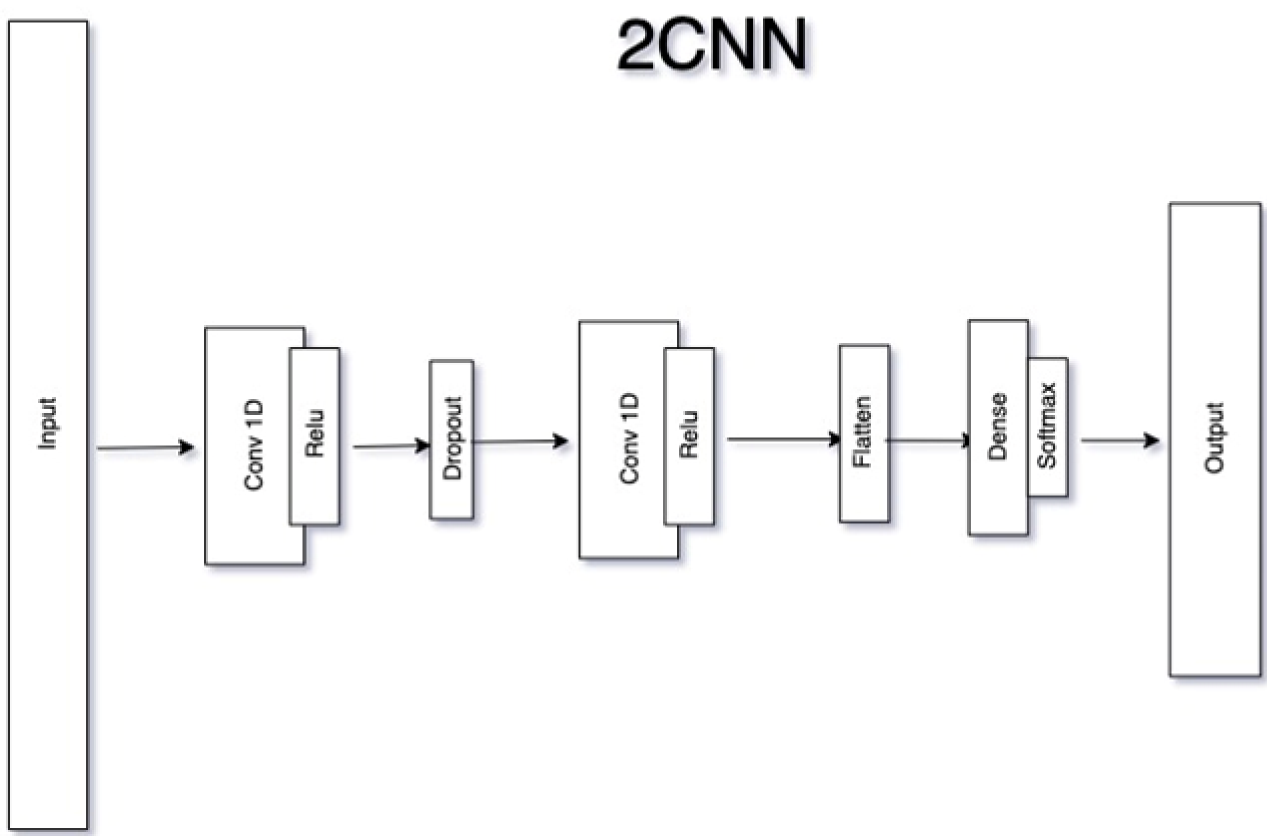

Figure 10

Architecture of the Two layers CNN model

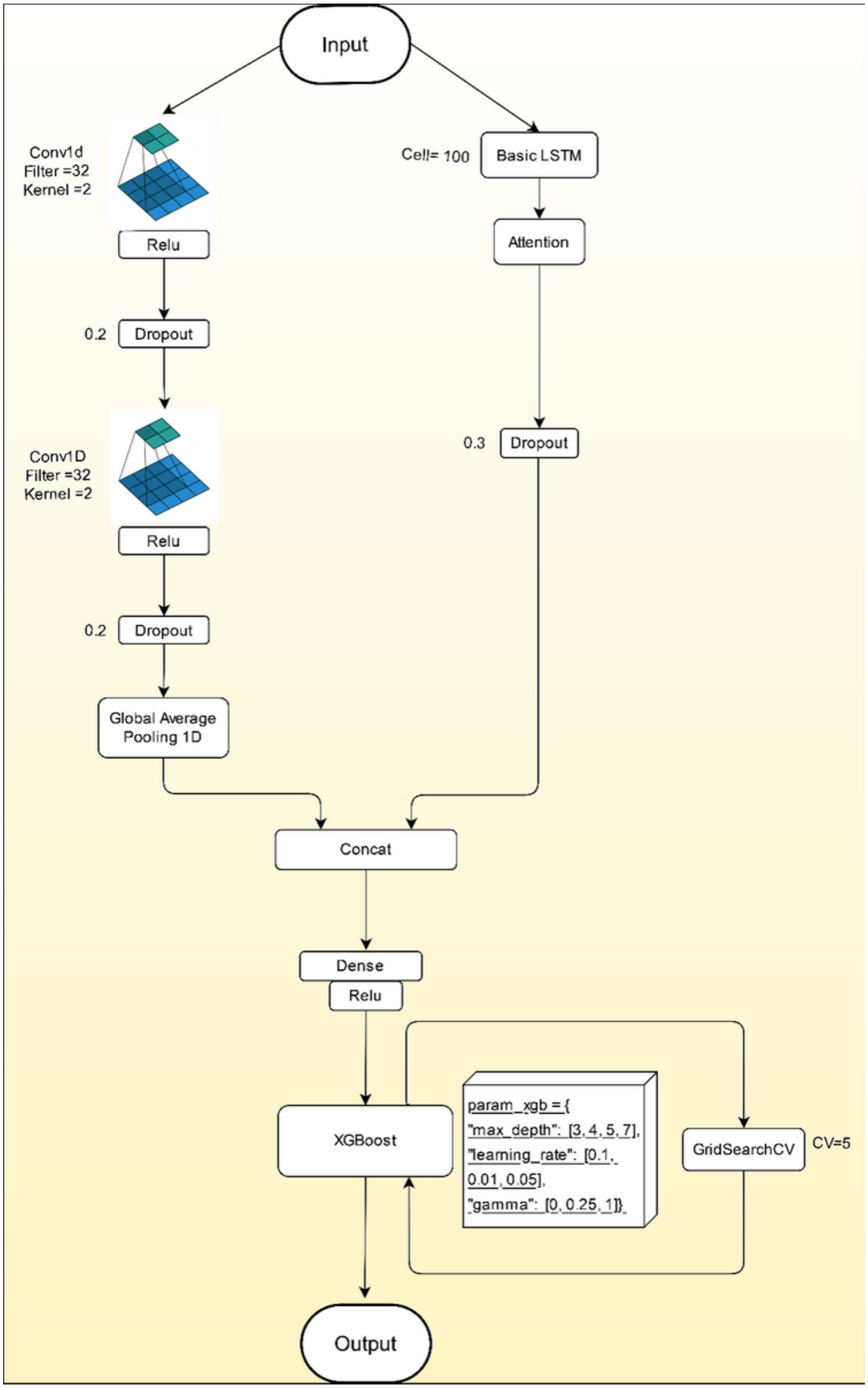

Figure 11

ALSTM-FCN with XGBoost model

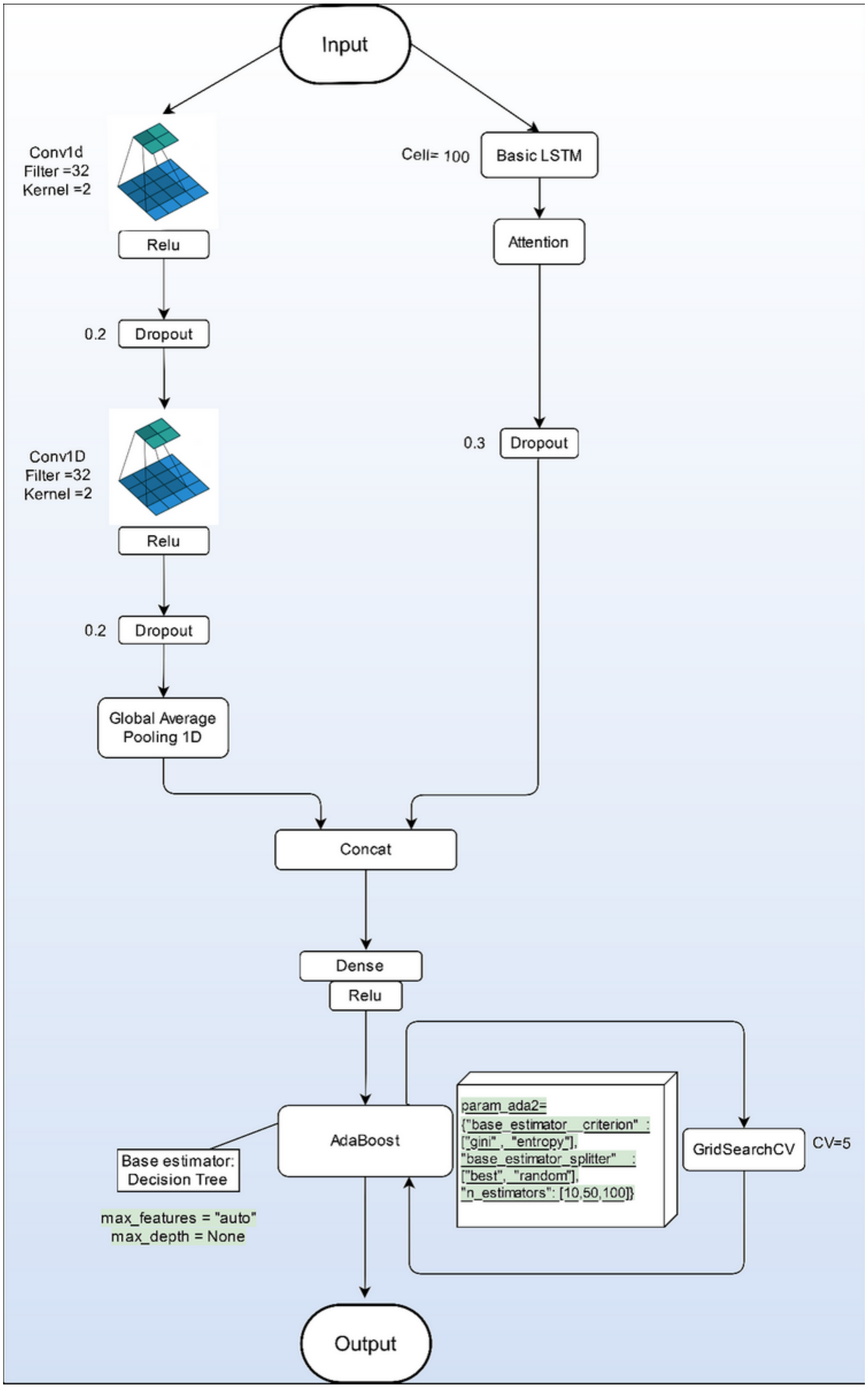

Figure 12

ALSTM-FCN with AdaBoost model

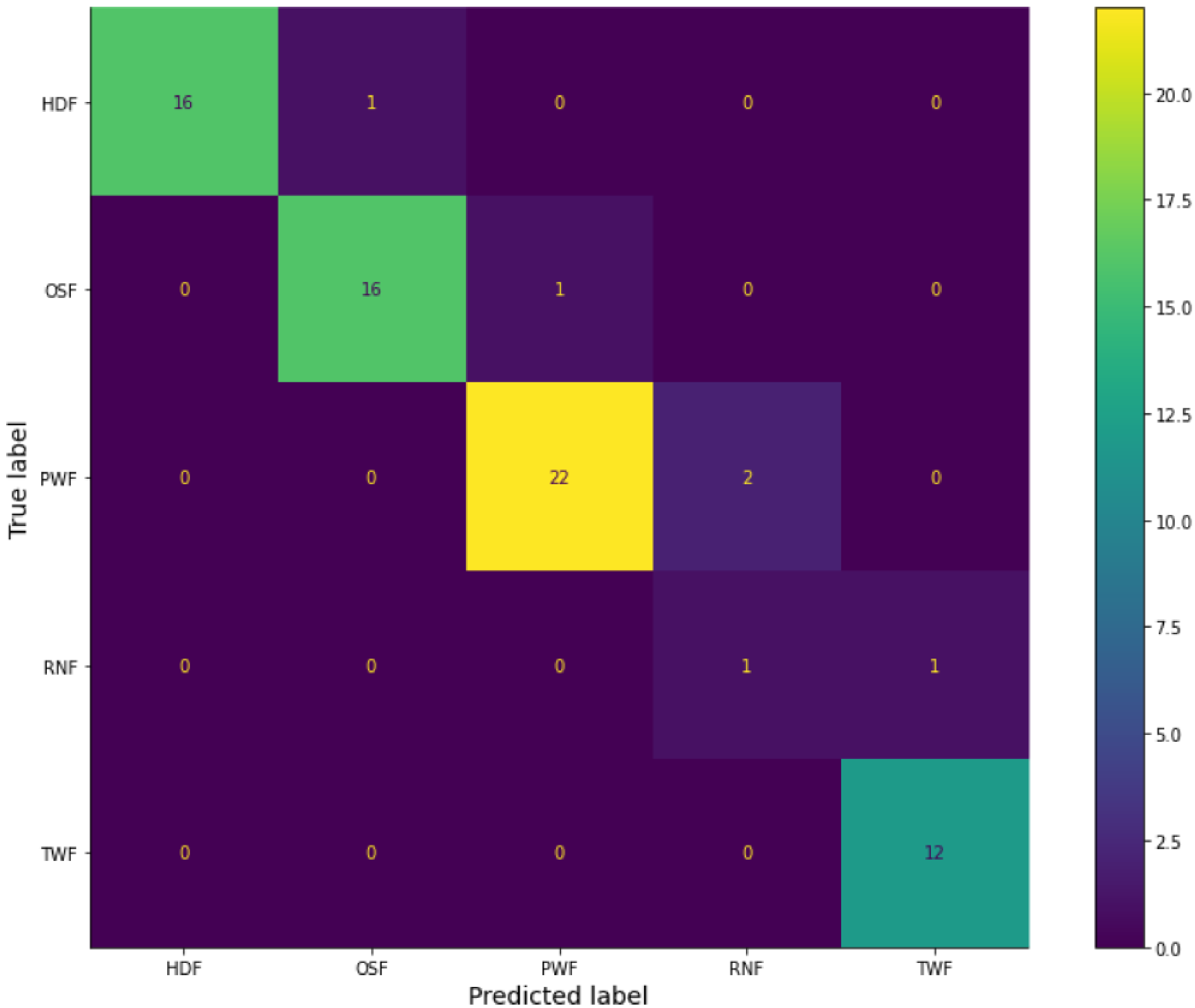

Figure 13

Confusion matrix for the LightGBM model

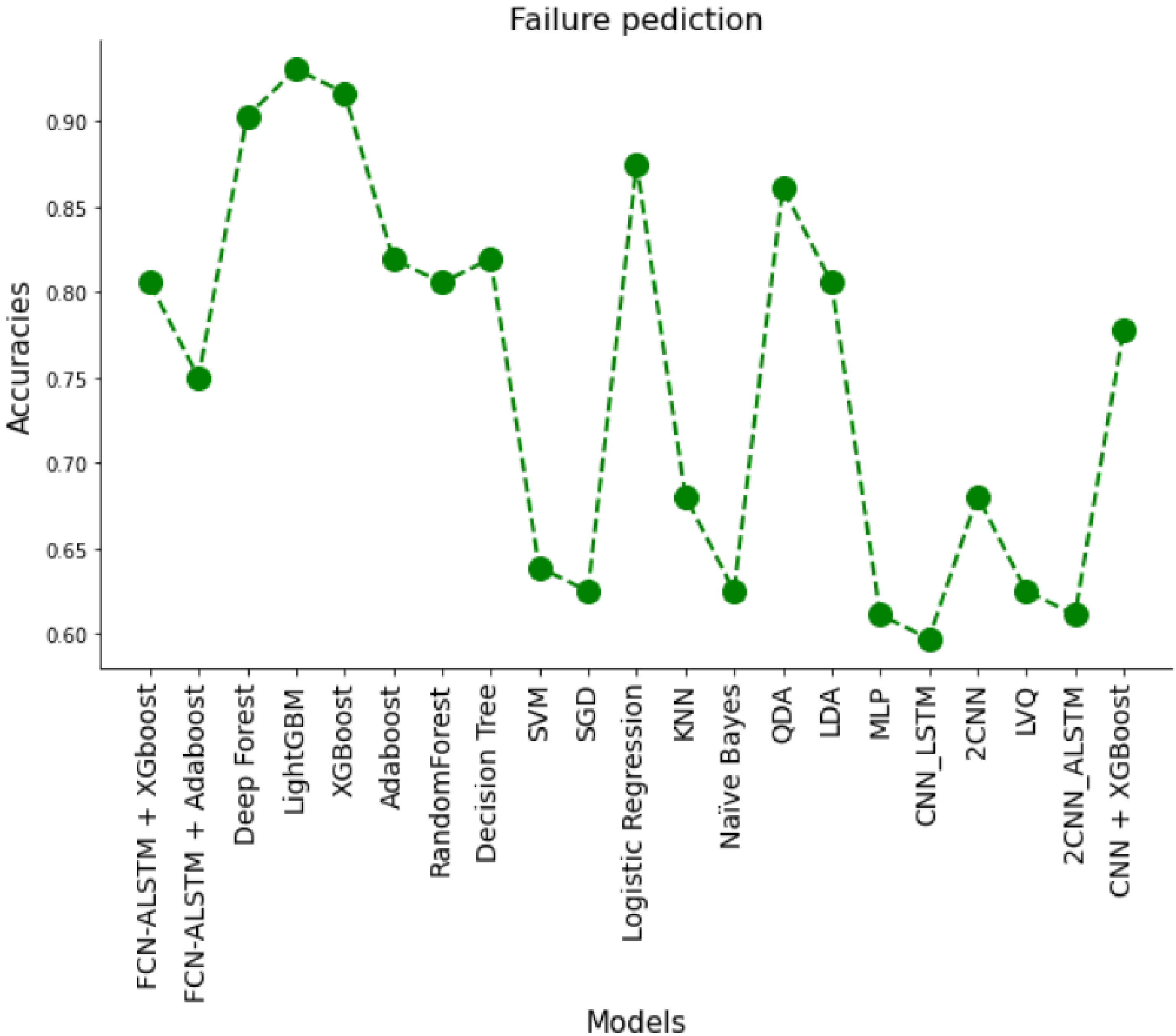

Figure 14

A a comparison of average accuracies for failure detection for all models

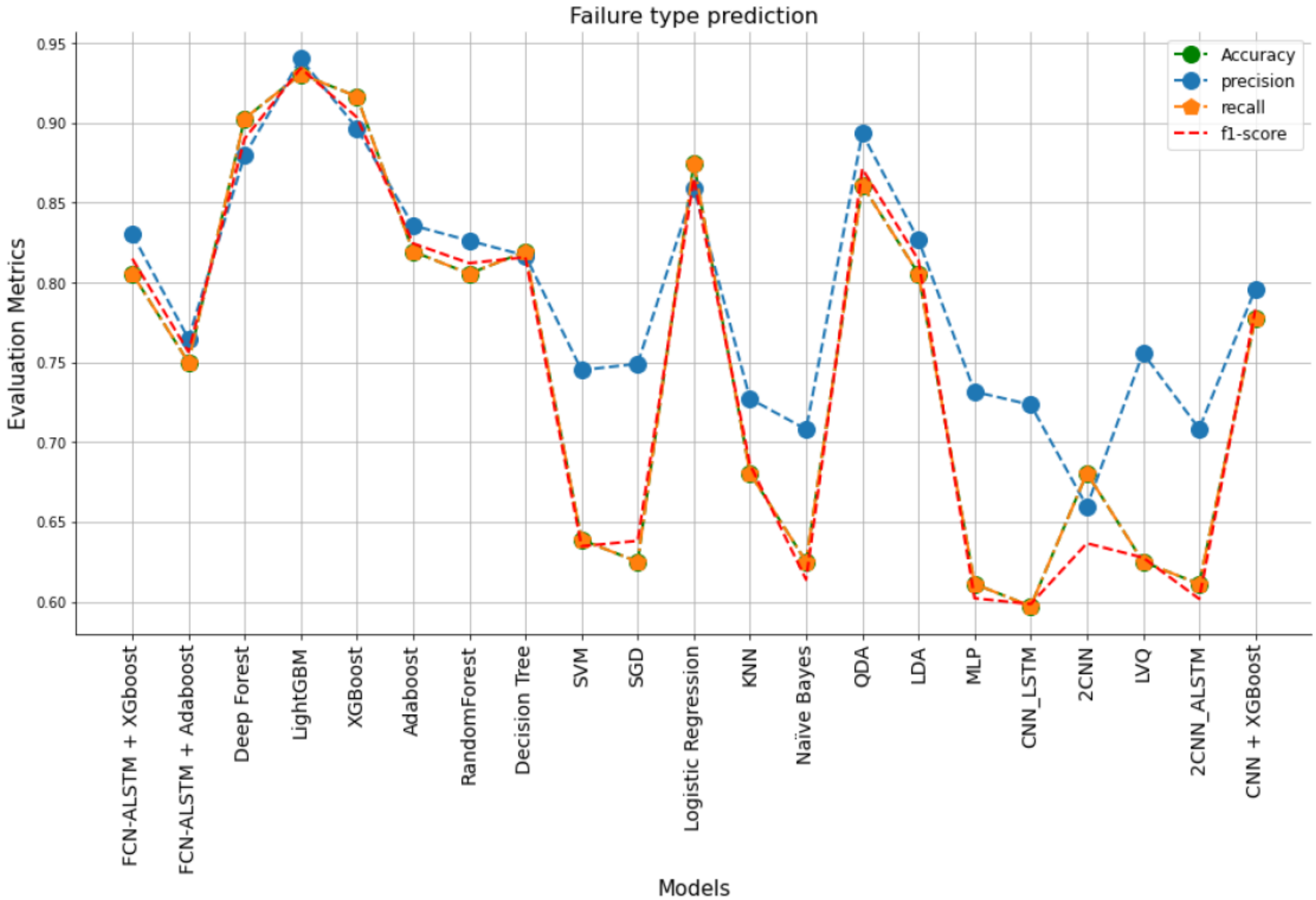

Figure 15

A comparison of performance measurements of failure detection for all models

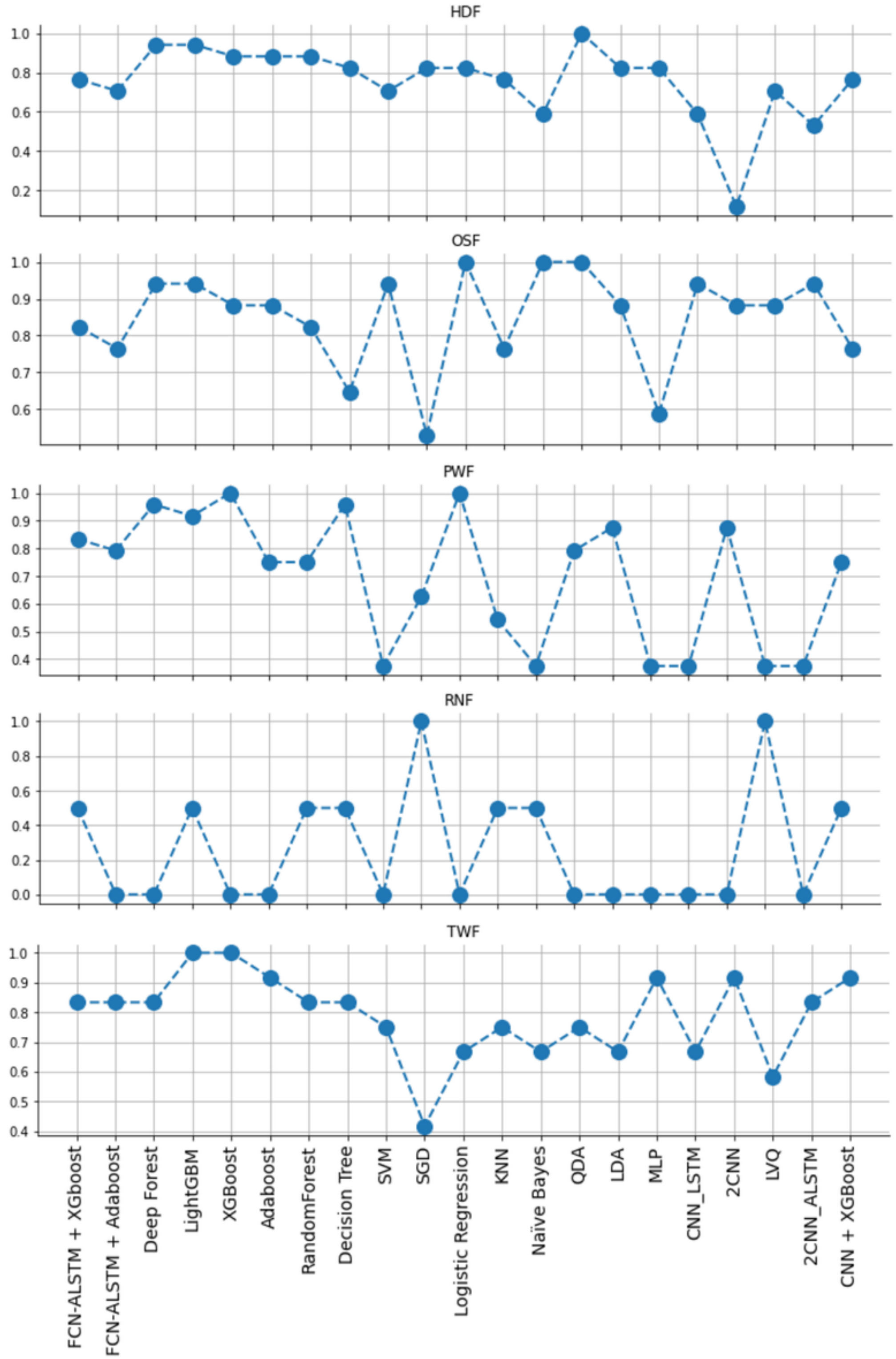

Figure 16

A comparison of detection accuracy values for each of the five failure types for all models