Decision at First Sight: An Attention Network for Direct Maintenance Optimization from Sensor Data

Mohammad Badfar[1], Iman Kazemian[1], Ratna Babu Chinnam[1], and Murat Yildirim[1]

[1] *Department of Industrial and Systems Engineering, Wayne State University, Detroit, Michigan, 48201, United States*
mohammadbadfar@wayne.edu
iman.kazemian@wayne.edu
ratna.chinnam@wayne.edu
murat@wayne.edu

ABSTRACT

Maintenance planning is a crucial strategy in industrial systems, where maintenance costs can consume up to 40% of total production expenses, and downtime costs can reach hundreds of thousands of dollars per hour. Despite its importance, the implementation of advanced maintenance approaches remains limited due to challenges such as insufficient resources, lack of expertise, inadequate funding, and difficulty converting vast operational data into actionable decisions. This paper introduces a novel attention-based deep learning model for maintenance scheduling that bypasses traditional degradation modeling and optimization techniques. The proposed model operates directly on sensor data, leveraging a multi-head attention mechanism within an encoder-decoder architecture to generate maintenance schedules. The cost function of the model is flexible and can be customized to accommodate different maintenance scenarios, making it adaptable to various operational requirements. The model's performance is validated through comparisons with the state-of-the-art predict-then-optimize benchmark, demonstrating its ability to generate cost-effective maintenance schedules. For commercial lithium-ion battery fleets, ATOM achieves a 22–35% reduction in maintenance expenses relative to predict-then-optimize approach. This approach provides a scalable, data-driven solution for dynamic and complex maintenance environments, eliminating the need for explicit remaining useful life (RUL) estimates or predefined degradation models.

1. INTRODUCTION

Maintenance costs can consume up to 40% of total production costs in heavy industries like oil and gas, manufacturing, and utilities (W. Wang, 2012; Deloitte, 2017), underscoring the critical importance of effective maintenance strategies. Additionally, the average cost of downtime in manufacturing is approximately $260,000 per hour (Research, 2016), highlighting the significant financial impact of equipment failures and unplanned downtime. Condition-Based Maintenance (CBM) has been widely recognized as a crucial strategy for managing the health and performance of large-scale industrial systems such as power generation units, manufacturing facilities, and renewable energy infrastructures. Despite decades of efforts, however, many organizations struggle to effectively deploy CBM. Only 11% of companies have implemented predictive maintenance solutions at scale (PwC & Mainnovation, 2017). This indicates that a vast majority of assets are not benefiting from CBM, revealing a significant opportunity for improvement. One significant reason is that companies have accumulated vast amounts of operational data, including sensor data and maintenance records, over the years but often lack the means to convert this data directly into actionable maintenance decisions (Carvalho et al., 2019).

According to a recent survey, key challenges to improving maintenance practices include lack of resources or staff, insufficient understanding of new maintenance options and technologies, lack of available funding, and inadequate training (Engineering, 2018). Despite the abundance of operational data, organizations face a significant shortage of skilled professionals capable of extracting actionable insights and integrating them into practical maintenance strategies, making effective monitoring, degradation modeling, and decision optimization challenging (Manyika, 2011). The asset management industry has also experienced a talent shortage due to an aging workforce and a lack of new skilled entrants (Deloitte & Institute, 2018), further exacerbating the reliance on experts to analyze complex data. Companies previously depended on professionals proficient in asset management and data science to work with this data, but the talent gap has made this increasingly difficult. Traditionally, the focus has been on developing predictive models to forecast equipment failures. However, without effectively bridging these predictions to decision-making processes, the full potential of CBM remains unrealized. The real opportunity lies in turning data into decisions-operationalizing this wealth of information to optimize maintenance actions. In this context, our approach aims to directly integrate data-driven insights with decision optimization, enabling more effective maintenance scheduling and resource allocation. By addressing the gap between data accumulation and actionable decision-making, we seek to unlock the untapped potential of CBM for a broader range of assets.

CBM policies utilize real-time insights into the system's health condition to make informed maintenance decisions. These policies focus on continuously monitoring the health status of machines, using data from sensors and other diagnostic tools to predict the remaining useful life (RUL) of each asset. CBM effectively manages operations and maintenance by leveraging detailed machine health information to optimize intervention timing (Ali & Abdelhadi, 2022). By relying on sensor-driven asset life prediction, CBM policies enable maintenance actions to be mapped directly to the machine's condition, allowing for timely interventions and reducing unexpected failures (Li, Peng, Li, & Jiang, 2020). Unlike traditional time-based maintenance, which follows fixed schedules, CBM is considered a more beneficial and realistic approach, ensuring that maintenance is performed only when needed, thereby minimizing unnecessary costs and downtime (Ahmad & Kamaruddin, 2012). Condition monitoring plays a crucial role in this process, reducing the number of interventions compared to time-based maintenance by continuously assessing system performance and identifying potential failures in advance (Quatrini, Costantino, Di Gravio, & Patriarca, 2020). This approach supports proactive, data-driven decision-making, leading to improved asset reliability and optimized operational efficiency (Sharma, Mittal, & Soni, 2024). Additionally, CBM integrates monitoring techniques that facilitate operational decisions in areas such as production planning, spare parts management, and reliability improvement, making it a key strategy for modern industrial asset management (Li et al., 2020).

Sensor-driven asset life prediction has gained significant attention in recent years. Traditional methods such as Gamma process and Markov decision process (MDP) are often limited by their reliance on simplified assumptions, linearity, and stationarity (Rezaeianjouybari & Shang, 2020; Saeed et al., 2025; Xiang & Foo, 2021; Rupprecht & Wang, 2022). In recent years, machine learning-based, data-driven methods have gained significant traction for predicting asset life and failure probabilities, offering more adaptive and accurate estimations (Y. Wang, Zhao, & Addepalli, 2020). These predictions are subsequently employed to represent the machine's health status, serving as a foundation for various maintenance planning strategies (Lee & Mitici, 2023; Wesendrup & Hellingrath, 2023; Zhang et al., 2023; Elbasheer et al., 2022). Instead of relying solely on estimated failure time, recent state-of-the-art CBM approaches leverage sensor-driven asset life prediction techniques to estimate the Remaining Life Distribution (RLD) of assets and compute failure probabilities at different future time steps (Yildirim, Sun, & Gebraeel, 2016; Yildirim, Gebraeel, & Sun, 2017; Fallahi, Bakir, Yildirim, & Ye, 2022). These failure probabilities, combined with the predicted costs of preventive maintenance and failure replacement, are integrated into a dynamic maintenance cost function to optimize maintenance decisions.

The Predict-then-Optimize approach is a widely used framework for maintenance optimization, where sensor-driven asset life predictions are employed to optimize the maintenance schedule while considering operational constraints and cost trade-offs. In this approach, asset life predictions, such as RUL or RLD, serve as the primary decision inputs, enabling maintenance planners to schedule interventions proactively based on predicted failure risks. Mathematical optimization-based methods then use this dynamic maintenance cost function to determine optimal maintenance actions that balance preventive maintenance costs, failure risks, and operational limitations. However, without considering how predictions will be used, high predictive accuracy does not necessarily lead to optimal decisions. Bastani et al. (Bastani, Zhang, & Zhang, 2022) highlight that traditional Predict-then-Optimize models focus solely on minimizing prediction error, neglecting the impact of these predictions on downstream optimization, which can result in misaligned objectives and suboptimal real-world outcomes. Similarly, Elmachtoub and Grigas (Elmachtoub & Grigas, 2022) emphasize that standard machine learning models are not designed to incorporate the structure of the optimization problem, limiting their effectiveness in decision-making. Moreover, abstraction in deep learning-based diagnostics often leads to the loss of critical time-series information. Lv et al. (Lv, Guo, Zhou, Qian, & Liu, 2023) note that predictive maintenance models frequently simplify complex sensor signals into high-level fault indicators, overlooking important degradation patterns that influence maintenance decisions. Likewise, Sadana et al. (Sadana et al., 2024) discuss how contextual optimization models may discard valuable variations when compressing high-dimensional sensor data into simplified life predictions, reducing their ability to support informed decision-making.

In this paper, we propose a deep attention-based model for maintenance planning that bypasses the need for both asset life prediction and traditional mathematical optimization techniques. Instead, the model operates directly on raw sensor signals, learning to generate optimal maintenance schedules without relying on predefined degradation states or complex mathematical formulations. The attention mechanism enables the model to focus on the most relevant features of the sensor data, allowing it to effectively learn patterns and correlations that inform maintenance decisions, without losing information due to avoiding abstraction in the prediction phase. By eliminating the need for sensor-driven prediction, this approach reduces reliance on costly and complex engineering processes. Additionally, the proposed model inherently offers adaptability, meaning it can adjust to varying operational conditions, account for uncertainties in asset degradation, and generalize across different maintenance scenarios without requiring extensive retraining. Ultimately, our model provides a novel, data-driven alternative for maintenance scheduling, leveraging the power of deep learning to produce cost-effective maintenance plans based solely on real-time sensor input. The contributions of this work are as follows:

- **Direct Sensor-to-Maintenance Framework:** A deep attention-based model that generates maintenance schedules directly from raw sensor data, eliminating degradation modeling and optimization steps.
- **Attention-Driven Decision Making:** An attention mechanism that automatically identifies informative signal patterns to support maintenance scheduling without requiring degradation-state estimation.
- **Scalable CBM Alternative:** A cost-effective and scalable maintenance planning approach that reduces dependence on complex models and domain expertise.

The remainder of this paper is organized as follows. Section 2 formulates the maintenance planning problem and outlines the associated objectives and constraints. Section 3 presents the proposed attention-based model for maintenance planning. Section 4 describes the experimental setup and discusses the results. Finally, Section 5 concludes the paper and highlights directions for future research.

2. MAINTENANCE PLANNING PROBLEM

Maintenance planning involves determining optimal schedules for performing maintenance activities on assets to minimize costs and ensure operational efficiency. In this framework, we consider a set of assets, each associated with a RUL at the start of the planning horizon. Each asset can receive one preventive maintenance action, and after this maintenance, it is assumed that the asset will not fail for the remainder of the planning horizon.

We define a binary decision variable $x_{i,t}$ for each asset i at time t, where $x_{i,t} = 1$ if preventive maintenance is performed on asset i at time t, and $x_{i,t} = 0$ otherwise. Additionally, we introduce binary variables $y_{i,t}$ to indicate whether corrective maintenance is required for asset i at time t; that is, $y_{i,t} = 1$ if asset i fails at time t, and $y_{i,t} = 0$ otherwise. Finally, we define a binary variable z_t to indicate whether a maintenance visit occurs at time t, where $z_t = 1$ if at least one maintenance action is scheduled at time t, and $z_t = 0$ otherwise.

The objective is to minimize the total maintenance cost over the planning horizon, encompassing preventive maintenance costs, corrective maintenance costs, and visit costs. This total cost can be expressed as:

$$\min_{\{x_{i,t}, y_{i,t}\}} \sum_{i=1}^{I} \left(\sum_{t=1}^{T} C_{\text{PM},i} x_{i,t} + \sum_{t=1}^{T} C_{\text{CM},i} y_{i,t} \right) + \sum_{t=1}^{T} V_t z_t, \tag{1}$$

where:

- $C_{\text{PM},i}$ is the cost of performing preventive maintenance on asset i,
- $C_{\text{CM},i}$ is the cost of corrective maintenance for asset i,
- V_t is the visit cost at time t, and
- z_t is a binary variable indicating whether a maintenance visit occurs at time t.

Based on the RUL of each asset, we can determine potential failure times if preventive maintenance is not performed. The failure indicator $y_{i,t}$ is defined based on the maintenance decision $x_{i,t}$ and the expected failure time RUL_i for asset i:

$$y_{i,t} = \begin{cases} 1, & \text{if } x_{i,\tau} = 0 \;\forall\; \tau = 1, \ldots, RUL_i \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

This means that if preventive maintenance is not performed on asset i before its failure time, the asset will fail at its expected failure time RUL_i, necessitating corrective maintenance. To account for uncertainties in the RUL predictions, a buffer parameter Δ is introduced. This buffer ensures that preventive maintenance for asset i is scheduled Δ periods earlier than the predicted failure time RUL_i, reducing the risk of unexpected failures.

Operational constraints can be incorporated to reflect real-world limitations. For example, the total number of preventive maintenance actions cannot exceed a maximum allowed number:

$$\sum_{i=1}^{I} \sum_{t=1}^{T} x_{i,t} \leq N^{\max}, \tag{3}$$

where $N^{\max}$ is the maximum number of preventive maintenance actions permitted during the planning horizon.

Similarly, if there are limitations on maintenance visits per time period, we can define constraints such as:

$$\sum_{i \in \mathcal{V}_t} x_{i,t} \leq V_t^{\max}, \quad \forall t \in \{1, 2, \ldots, T\}, \tag{4}$$

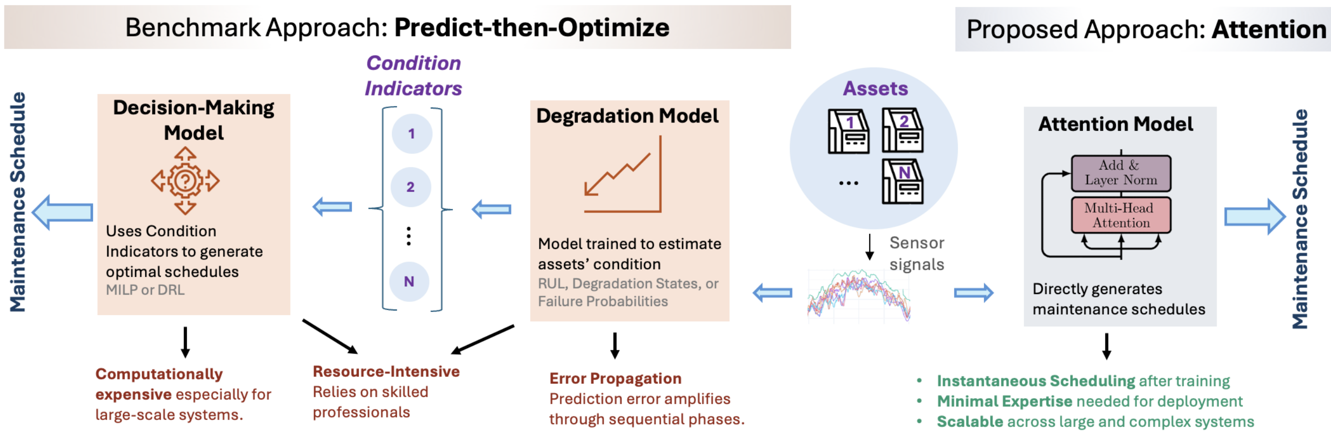

Figure 1. **Comparison of Predict-then-Optimize approaches and the proposed attention-based maintenance planning method.** The benchmark approach involves estimating asset conditions followed by optimization, amplifying errors through sequential phases, and requiring costly engineering expertise. In contrast, the proposed method bypasses prediction entirely, directly generating maintenance schedules from raw sensor data, reducing complexity, cost, and implementation challenges.

where:

- $\mathcal{V}_t$ is the set of assets available for maintenance at time t,
- $V_t^{\max}$ is the maximum number of maintenance visits allowed at time t.

The RUL of the assets is inherently stochastic and must be estimated using data-driven approaches or other predictive methods. By obtaining estimated RUL values, we can determine the expected failure times RUL_i for each asset. With these estimates, the maintenance planning problem becomes an integer programming problem. This problem can be solved using conventional integer optimization methods such as Branch and Bound, Cutting Planes, or heuristic algorithms like Genetic Algorithms and Tabu Search.

3. Attention-Based Model for Maintenance Planning

In this section, we introduce our proposed attention-based deep learning model for maintenance scheduling. This model is designed to learn and generate optimal maintenance schedules for a set of assets by solely utilizing the assets' sensor signals. By leveraging the rich and continuous data provided by these sensors, the model can identify patterns and indicators of asset health and predict maintenance needs without relying on explicit RUL estimates.

3.1. Model Architecture

The proposed maintenance scheduling model is based on an attention mechanism that operates on a set of assets with specific features and generates a maintenance schedule for each asset. Fig. 2 provides an overview of the proposed model, illustrating its key components and the interactions between them. The model follows an encoder-decoder architecture, where attention is used to focus on relevant information at each step of the scheduling process. The input to the model consists of a set of assets, each represented by a feature vector $\mathbf{a}_i \in R^d$, where $i = 1, \ldots, N$ denotes the asset index and d is the number of features. The input for all assets is represented as a matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N]^\top \in R^{N\times d}$.

The encoder processes the input features of each asset and outputs a set of context-aware embeddings, which are used by the decoder to generate a maintenance schedule. The encoder consists of multiple layers of multi-head attention (MHA) and feed-forward neural networks (FFN).

The multi-head attention mechanism allows the model to attend to different parts of the input simultaneously. For a single attention head, the output is calculated as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{5}$$

where Q, K, and V are the query, key, and value matrices, respectively, all derived from linear transformations of the input feature matrix $\mathbf{A}$. d_k is the dimension of the query/key vectors. Multi-head attention is applied by repeating this mechanism across multiple heads:

$$\text{MHA}(Q, K, V) = [\text{head}_1, \ldots, \text{head}_h]W^O \tag{6}$$

where h is the number of heads, and W^O is a learned output projection matrix.

After the attention layer, the output is passed through a feed-forward network applied independently to each asset's representation:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{7}$$

where W_1, W_2, b_1, and b_2 are learned parameters. This

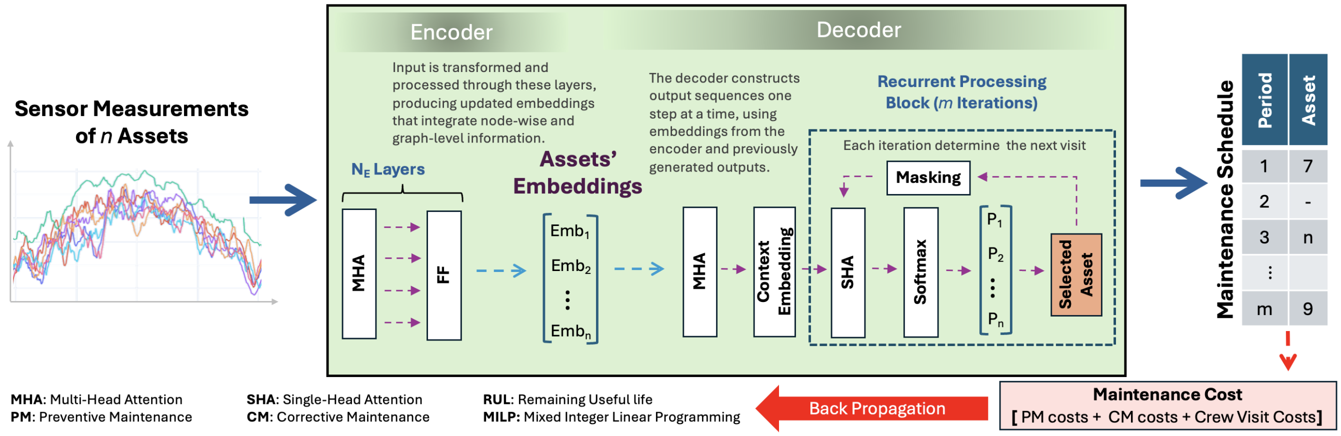

Figure 2. The proposed attention-based Sensor-to-Decision maintenance planning model.

layer introduces non-linearity into the model, improving its representational power. Both the multi-head attention and feed-forward layers use residual connections and layer normalization to stabilize training:

The decoder sequentially generates the maintenance schedule by selecting one asset at a time for maintenance. It uses masked multi-head attention to ensure that the model only attends to previously selected assets, preventing information leakage from future decisions. The attention mechanism in the decoder is the same as in the encoder but applied to the previously selected assets.

After self-attention, the decoder computes attention over the encoder's output to incorporate the context provided by the entire set of assets. This allows the model to dynamically decide which asset should be selected for maintenance based on the current step:

$$\text{Cross-Attention}(Q, K, V) = \text{MHA}(Q_{\text{decoder}}, K_{\text{encoder}}, V_{\text{encoder}}) \tag{8}$$

where Q_{decoder} is the query matrix from the decoder and $K_{\text{encoder}}, V_{\text{encoder}}$ are the key and value matrices from the encoder. The decoder then passes the output through a feed-forward network similar to the encoder, followed by residual connections and layer normalization.

At each time step, the decoder produces a probability distribution over all assets, indicating which asset should be selected for maintenance:

$$p_i = \text{softmax}(W_{\text{out}}\mathbf{z}_i) \tag{9}$$

where W_{out} is a learned weight matrix, and p_i represents the probability of selecting asset i for maintenance at the current step.

To introduce flexibility and enhance decision-making in the maintenance scheduling model, a dummy asset is incorporated into the decoder's selection process. The dummy asset is fed to the model as input, similar to other real assets, but its sensor values are all set to zero. During each step of the decoder's operation, the dummy asset is included in the set of selectable assets, allowing the model to represent a time period where no PM is performed. By enabling the selection of the dummy asset, the model gains the ability to make PM actions optional, effectively skipping unnecessary PMs when maintenance on real assets is not economically justified or operationally critical. This mechanism optimizes the overall maintenance schedule, balancing cost savings with reliability by avoiding redundant or premature PMs.

The model is trained using a loss function that reflects the total maintenance cost, which consists of both preventive and corrective maintenance costs. Let $\hat{S}$ be the predicted schedule, and $C(\hat{S})$ the corresponding total cost. The objective is to minimize the expected maintenance cost:

$$\mathcal{L} = E_{\hat{S}\sim p(\hat{S}|\mathbf{A})}[C(\hat{S})] \tag{10}$$

where $p(\hat{S}|\mathbf{A})$ is the probability of the predicted schedule given the input features $\mathbf{A}$. The cost function $C(\hat{S})$ accounts for both preventive maintenance costs and penalties associated with asset failures. During training, the RUL of each asset is used to compute these maintenance costs, ensuring that the scheduling decisions take into account the likelihood of asset failure. The model is trained end-to-end with backpropagation used to update the parameters of the encoder, decoder, and attention layers. The objective during training is to learn to generate maintenance schedules that minimize the total cost while respecting operational constraints.

Since the cost function is not differentiable with respect to the model parameters, we use the REINFORCE algorithm (Williams, 1992) to estimate gradients. REINFORCE improves the decision policy by learning from sampled maintenance schedules generated by the decoder. The probability of a schedule is determined by the sequence of action probabilities at each step. To reduce variance during training, we use a greedy rollout baseline that evaluates the cost of a deterministic policy selecting the most likely feasible action at each step. The baseline is periodically updated when the learned policy statistically outperforms it on a large evaluation set. This self-critical training strategy stabilizes learning and encourages continuous improvement of the policy.

3.2. Cost Function

The cost function used in our attention-based maintenance scheduling model is equivalent to the objective function of the mathematical maintenance planning model introduced in the previous section. This equivalence ensures that our deep learning model directly optimizes the same cost criteria, bridging traditional optimization methods with advanced machine learning techniques for maintenance planning.

Our model is designed with a flexible cost function framework capable of accommodating various maintenance cost considerations. This flexibility allows for adaptation to different maintenance strategies, such as balancing preventive and corrective maintenance costs, applying penalties for early or late maintenance, and accounting for operational expenses like crew visit costs. By handling diverse cost functions, the model becomes a versatile tool for optimizing maintenance schedules across different operational contexts.

To demonstrate the model's capabilities, we present a commonly used maintenance cost function that the model can optimize. Let

$$\pi = (\pi_1, \pi_2, \ldots, \pi_N) \tag{11}$$

represent the sequence of assets scheduled for maintenance, where π_j denotes the index of the asset scheduled at the j-th time period. Each asset i has a Remaining Useful Life, denoted by RUL_i, which serves as the ground truth for calculating maintenance costs during training.

The total maintenance cost $C(\pi)$ consists of three main components: preventive maintenance costs, corrective maintenance costs, and crew visit costs. The cost function is defined as:

$$C(\pi) = \sum_{t=1}^{T} \sum_{i=1}^{N} \left(C_i^{\text{PM}}\, x_{i,t} + C_i^{\text{CM}}\, y_{i,t} \right) + \sum_{t=1}^{T} V_t\, z_t \tag{12}$$

where:

- $x_{i,t}$ is a binary variable equal to 1 if asset i undergoes preventive maintenance at time t (i.e., before failure, $t < RUL_i$), and 0 otherwise.
- $y_{i,t}$ is a binary variable equal to 1 if asset i undergoes corrective maintenance at time t (i.e., after failure, $t \geq RUL_i$), and 0 otherwise.
- C_i^{PM} and C_i^{CM} are the costs associated with preventive and corrective maintenance for asset i, respectively.
- V_t represents the crew visit cost at time t, and z_t is a binary variable equal to 1 if at least one asset is maintained at time t, accounting for the operational expenses of dispatching maintenance teams.

This cost function effectively balances the trade-off between the generally lower costs of preventive maintenance and the higher costs associated with corrective maintenance after asset failure. By including crew visit costs, the model also considers the logistical aspects of scheduling, aiming to minimize the number of maintenance trips required.

During training, the model leverages both the assets' features and the ground-truth RUL values to learn optimal maintenance decisions that minimize the total cost $C(\pi)$. The attention mechanism enables the model to focus on relevant features that influence maintenance needs, such as operational conditions, asset health indicators, and failure probabilities. By learning from the actual RUL data, the model understands when an asset is likely to fail and can schedule maintenance accordingly to prevent costly corrective actions.

After training, the model can generate maintenance schedules solely based on the assets' features, without requiring the ground-truth RUL values. This capability is particularly valuable in real-world applications where the exact RUL may not be known in advance. By relying on the learned relationships between features and maintenance needs, the model can predict optimal maintenance actions that minimize costs while maintaining asset reliability.

The flexibility of the cost function allows practitioners to tailor the model to specific operational needs and cost structures. Whether the goal is to prioritize asset reliability, reduce maintenance expenses, or optimize resource allocation, the attention-based model can be customized accordingly. This adaptability makes it a powerful tool for strategic maintenance planning across various industries and operational scenarios.

3.3. Constraint Handling

Our attention-based maintenance scheduling model is designed to accommodate a variety of operational constraints, enhancing its practicality and effectiveness in real-world applications. These constraints include limitations on the number of preventive maintenance actions per time period, the total number of PMs, the availability of maintenance crews, the number of allowable visits, and other resource-based or logistical restrictions. By effectively incorporating these constraints, the model ensures that the generated maintenance schedules are not only cost-efficient but also feasible within the operational context.

Incorporating these constraints into the model is achieved through the masking process within the decoder. The masking process is a crucial feature of the attention mechanism, which enables the model to selectively consider or ignore certain actions during the decoding phase. By dynamically adjusting the mask, the model can prevent the selection of assets for maintenance if scheduling them would violate specified constraints. For example, to enforce a constraint on the maximum number of PMs per time period, the mask can be updated at each decoding step to exclude assets that would exceed this limit. This is accomplished by setting the attention weights to zero for those assets, effectively removing them from consideration at that time step. Similarly, constraints on the total number of PMs can be enforced by keeping a cumulative count of scheduled PMs and updating the mask accordingly.

4. EXPERIMENTS AND RESULTS

In this section, we present a series of experiments designed to evaluate the performance of our proposed attention-based deep learning model for maintenance scheduling. The primary goal is to demonstrate the model's ability to generate optimal maintenance schedules using only the assets' sensor signals, without relying on explicit Remaining Useful Life estimates or additional predictive models. We aim to showcase how the model handles different maintenance strategies and operational constraints across various scenarios.

4.1. Experimental Setup

The dataset utilized in this study is sourced from Zhu et al. (Zhu et al., 2022) and comprises cycling data from 130 commercial lithium-ion batteries. These batteries were subjected to a variety of operational and environmental conditions, such as differing temperatures and discharge rates, and they differ in electrode compositions. The dataset captures the degradation behavior of the batteries over time, making it ideal for tasks like estimating battery health and predicting future performance—both critical elements in maintenance planning.

The RUL of each battery is defined as the number of remaining cycles before it reaches 80% of its nominal capacity. The model receives only the summary data from a single cycle, including high-level statistics such as the remaining capacity, as input features. It is tasked with learning and generating the optimal maintenance schedule based solely on the summary data of a randomly selected cycle. This setup challenges the model to predict long-term maintenance needs using limited, cycle-specific information, thereby enabling schedule optimization with minimal data input.

Although the current implementation relies on a sampling strategy based on key battery health indicators, the ATOM architecture employs a scalable encoder–decoder framework built on Multi-Head Attention (MHA). Due to its ability to capture long-range temporal dependencies, the MHA mechanism enables the framework to be extended to directly process raw, high-frequency sensor data. This capability allows forecasting and scheduling tasks to be integrated into a single end-to-end policy network.

The attention-based maintenance scheduling model was trained and tested using subsets of the dataset. Specifically, 100 batteries were allocated for training, while the remaining 30 batteries were reserved for testing. Given the limited number of batteries and the utilization of only a single cycle's summary data as input per sample, a sampling strategy was employed to generate the training data. In each training sample, n batteries (where n represents the number of maintenance assets) were randomly selected from the training battery pool. For each selected battery, a random cycle was chosen, and the summary data from that cycle were used as input features. The RUL, defined as the number of cycles remaining before the battery reaches 80% capacity, served as the ground truth for training. This sampling process was repeated for each training epoch, resulting in a unique dataset of 25,600 samples per epoch. Each sample comprised n batteries with 7 features per battery, yielding a dataset shape of (25,600, n, 7).

Following preliminary experiments, hyperparameter tuning was conducted. The final model architecture consisted of one multi-head attention (MHA) layer with two attention heads. The initial embedding size for each battery was set to 8. Training was performed over 450 epochs with a batch size of 16. To ensure exposure to a diverse range of battery conditions, a new dataset was generated for each epoch, allowing the model to encounter various battery states during training.

Evaluation was conducted on the test set of 30 batteries using the same sampling strategy as in training. Specifically, 128 samples were generated from the test batteries, each comprising n randomly selected batteries. For each battery in a sample, a random cycle was selected, and the RUL was calculated based on the remaining cycles until the battery reached 80% capacity.

The model's performance was assessed using several metrics:

- **Number of Preventive Maintenance Actions (No. of PMs):** The total count of times preventive maintenance was scheduled before asset failure.
- **Number of Failures (No. of Failures):** The total count of assets that failed without receiving preventive maintenance.
- **Maintenance Cost:** The total cost incurred, encompass-

ing preventive and corrective maintenance expenses, as well as any applicable visit costs.

- **Number of Visits (if applicable):** The total number of distinct instances when maintenance crews were dispatched.

Performance results were averaged across the 128 test samples, yielding an overall assessment of the model's ability to generate effective and cost-efficient maintenance schedules. This evaluation demonstrated the model's capacity to optimize maintenance planning using minimal input data, solely based on the assets' sensor signals.

4.2. Benchmarks

To evaluate the effectiveness of our proposed model, we compare it against two benchmarks. The first benchmark, referred to as the *Oracle*, assumes perfect knowledge of the RUL for each asset. Using these known RULs, we solve a Linear Programming problem using the PuLP v3.3.0 library to obtain the optimal maintenance schedule, which minimizes costs by performing maintenance at the ideal times for each asset. This benchmark provides an idealized baseline, representing the best possible outcome with complete RUL information—something that is not achievable in real-world applications due to the unpredictable nature of asset degradation.

The second benchmark follows a *Predict-then-Optimize (PtO)* approach. Here, a deep neural network model is trained on historical data to predict RULs in real time. These predicted RULs are then used as inputs for the linear programming model using the PuLP v3.3.0 library, similar to the Oracle benchmark, to produce a maintenance schedule. Unlike the Oracle solution, this benchmark reflects a practical scenario where RULs must be estimated, introducing potential inaccuracies. This benchmark serves as a competitive standard for our model, as our approach aims to outperform this predict-then-optimize method by directly integrating predictive and decision-making capabilities.

We compare the results of our attention-based model (referred to as *Attention*) with these two benchmarks across all conducted experiments. To ensure a fair comparison, we use consistent parameters, such as preventive maintenance cost rate, corrective maintenance cost rate, and other relevant factors that influence maintenance decisions. Additionally, we apply the same operational constraints across all approaches, including limits on the number of preventive maintenance actions allowed per time period, the availability of maintenance crews, and the number of maintenance locations. This uniformity ensures that differences in performance arise solely from each method's ability to handle uncertainty and optimize scheduling, rather than from discrepancies in parameter settings or constraints. The maintenance optimization model is also run with various values of buffer Δ, and the results for the best-performing Δ are reported in the results section. By analyzing our model's performance relative to these benchmarks, we gain insights into its potential advantages in real-world scenarios, where predictive accuracy and effective decision-making under constraints are critical for optimal maintenance scheduling.

4.3. Case Study I: Single Maintenance Crew

In the single maintenance scenario, only one preventive maintenance action is allowed per time period. Each of the 20 assets can receive a single preventive maintenance intervention over a planning horizon of 30 time periods, ensuring maintenance actions are distributed across time.

Table 1. Benchmark Analysis for a 20-Asset Problem in Single Maintenance Crew Case Study

Performance metric	PtO	Attention	Oracle
No. of PMs	15.03	16.52	16.34
No. of Failures	4.01	3.48	1.78
Maintenance Cost ($1000)	55.11	42.89	34.16

Table 1 presents the results for maintenance cost and the number of preventive maintenance (PM) actions across three methods: Attention, Oracle, and PtO. The Oracle benchmark, which uses actual RUL values, serves as an ideal benchmark since it represents the best possible outcome that would be difficult to achieve in real-world settings without precise RUL data.

In terms of maintenance cost, the Oracle benchmark achieves the lowest cost at 34.16, providing an ideal benchmark that reflects perfect knowledge of RUL. This serves as a baseline for optimal performance, achievable only with accurate and complete information about the assets' degradation states. The Attention model closely follows with a maintenance cost of 42.89, which is approximately 26% higher than Oracle. This demonstrates the model's ability to generate cost-effective schedules even without explicit RUL data, by leveraging sensor-driven insights to make informed decisions. The slight increase in cost compared to Oracle can be attributed to Attention's cautious approach, favoring preventive maintenance to avoid failures.

On the other hand, PtO incurs a significantly higher maintenance cost of 55.11, about 61% above the Oracle benchmark. This substantial gap underscores the challenges posed by prediction errors in the PtO pipeline. Even with the addition of a buffer (delta = 3) that forces the optimization model to perform preventive maintenance actions earlier than the predicted RUL, PtO still fails to execute enough preventive interventions. Specifically, PtO schedules only 15.03 preventive maintenance actions, compared to 16.34 and 16.52 for Oracle and Attention, respectively. This insufficient number of PMs suggests that the buffer, while helpful in partially mitigating the effects of RUL prediction errors, is not enough to overcome the limitations of the predictive model. As a result, PtO ends up relying more heavily on corrective maintenance, which is costlier and contributes to its overall inefficiency. These findings highlight the importance of accurate RUL predictions for the PtO approach and demonstrate how Attention offers a robust alternative by bypassing the prediction stage altogether.

4.4. Case Study II: Opportunistic Maintenance

In the opportunistic maintenance scenario, visit costs are incorporated into the cost function. This scenario considers 20 assets over a planning horizon of 30 time periods, where maintenance actions can be coordinated to minimize additional costs associated with each maintenance visit.

Table 2 presents the results for maintenance cost, number of preventive maintenance (PM) actions, and visit count across the three methods: Attention, Oracle, and PtO. Here, the Oracle benchmark, which uses actual RUL values, serves as the ideal benchmark, representing the most efficient outcomes achievable under perfect information.

Table 2. Benchmark Analysis for a 20-Asset Problem in Opportunistic Maintenance Case Study

Performance metric	PtO	Attention	Oracle
No. of PMs	15.11	16.89	13.11
No. of Failures	3.89	3.07	3.11
Maintenance Cost ($1000)	54.00	47.63	44.22
Visit Count	10.33	15.59	10.00

For maintenance cost, the Oracle benchmark achieves the lowest value at 44.22, establishing the ideal baseline for this scenario. By utilizing precise RUL information, Oracle is able to optimize maintenance schedules perfectly, ensuring that interventions are timed exactly when needed to minimize both preventive and corrective maintenance costs. The Attention model follows closely with a maintenance cost of 47.63, which is only 8% higher than the Oracle benchmark. This small difference highlights Attention's ability to generate cost-efficient schedules without relying on explicit RUL data. By directly processing sensor signals and using an attention mechanism to identify relevant patterns, Attention demonstrates strong performance in cost reduction while maintaining practical applicability across different operational settings.

In comparison, PtO incurs a significantly higher maintenance cost of 54.00, approximately 22% above the Oracle benchmark. This reflects the inefficiency of PtO's predict-then-optimize framework, where errors in RUL prediction propagate through the optimization stage, leading to suboptimal schedules. To address this issue, a buffer (delta = 3) was added to the PtO optimization model, forcing it to schedule preventive maintenance actions three periods before the predicted RUL. While this adjustment helps reduce the number of failures, it does not sufficiently improve PtO's performance in this scenario. Specifically, PtO schedules only 15.11 PMs, which is fewer than Attention's 16.89 but still higher than Oracle's 13.11. This outcome indicates that even with the buffer, PtO struggles to achieve an optimal balance between preventive and corrective maintenance, as its reliance on inaccurate RUL predictions limits its ability to generate effective schedules. The Oracle benchmark, by contrast, performs the fewest PMs and achieves the most efficient cost balance, underscoring the value of accurate RUL data. Meanwhile, the Attention model, although performing more PMs than both PtO and Oracle, provides a viable and cost-efficient alternative by circumventing the need for explicit RUL predictions.

5. CONCLUSION

This paper presents a novel attention-based deep learning model for maintenance scheduling, which eliminates the need for traditional degradation modeling and optimization methods. By operating directly on sensor data, the proposed model leverages a multi-head attention mechanism within an encoder-decoder architecture to generate cost-effective and operationally feasible maintenance schedules. The flexibility of the model's cost function makes it adaptable to various maintenance scenarios, and its performance is validated against state-of-the-art predict-then-optimize benchmarks, demonstrating its effectiveness in reducing maintenance costs without relying on explicit remaining useful life (RUL) estimates.

While the model offers significant advancements in maintenance planning, future work will focus on enhancing its adaptability and generalizability. On one hand, the model will be extended to handle data distribution shifts caused by asset aging or changes in operational conditions, ensuring robust performance in dynamic environments. On the other hand, efforts will be made to enable the model to generalize across different numbers of assets, making it scalable and applicable to diverse industrial systems. These enhancements will further establish the proposed approach as a robust and reliable tool for addressing complex maintenance challenges in real-world applications.

REFERENCES

Ahmad, R., & Kamaruddin, S. (2012). A review of condition-based maintenance decision-making. *European journal of industrial engineering*, *6*(5), 519–541.

Ali, A., & Abdelhadi, A. (2022). Condition-based monitoring and maintenance: state of the art review. *Applied Sciences*, *12*(2), 688.

Bastani, H., Zhang, D. J., & Zhang, H. (2022). Applied machine learning in operations management. *Innovative Technology at the Interface of Finance and*

Operations: Volume I, 189–222.

Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R. d. P., Basto, J. P., & Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024.

Deloitte. (2017). *Predictive maintenance and the smart factory.* Deloitte Insights. (`https://www2.deloitte.com/us/en/insights/focus/industry-4-0/predictive-maintenance-smart-factory.html`)

Deloitte, & Institute, T. M. (2018). *2018 skills gap in manufacturing study.* Deloitte Insights. (`https://www2.deloitte.com/us/en/pages/manufacturing/articles/future-of-manufacturing-skills-gap-study.html`)

Elbasheer, M., Longo, F., Mirabelli, G., Padovano, A., Solina, V., & Talarico, S. (2022). Integrated prescriptive maintenance and production planning: a machine learning approach for the development of an autonomous decision support agent. *IFAC-PapersOnLine*, *55*(10), 2605–2610.

Elmachtoub, A. N., & Grigas, P. (2022). Smart "predict, then optimize". *Management Science*, *68*(1), 9–26.

Engineering, P. (2018). *2018 maintenance study.* Plant Engineering Magazine. (`https://www.plantengineering.com/articles/2018-maintenance-report/`)

Fallahi, F., Bakir, I., Yildirim, M., & Ye, Z. (2022). A chance-constrained optimization framework for wind farms to manage fleet-level availability in condition based maintenance and operations. *Renewable and Sustainable Energy Reviews*, *168*, 112789.

Lee, J., & Mitici, M. (2023). Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics. *Reliability Engineering & System Safety*, *230*, 108908.

Li, Y., Peng, S., Li, Y., & Jiang, W. (2020). A review of condition-based maintenance: Its prognostic and operational aspects. *Frontiers of Engineering Management*, *7*(3), 323–334.

Lv, Y., Guo, X., Zhou, Q., Qian, L., & Liu, J. (2023). Predictive maintenance decision-making for variable faults with non-equivalent costs of fault severities. *Advanced Engineering Informatics*, *56*, 102011.

Manyika, J. (2011). Big data: The next frontier for innovation, competition, and productivity. *McKinsey Global Institute*, *1*.

PwC, & Mainnovation. (2017). *Predictive maintenance 4.0: Beyond the hype.* (`https://www.pwc.nl/en/publicaties/predictive-maintenance-4-0.html`)

Quatrini, E., Costantino, F., Di Gravio, G., & Patriarca, R. (2020). Condition-based maintenance—an extensive literature review. *Machines*, *8*(2), 31.

Research, A. (2016). *The critical cost of downtime.* (`https://www.aberdeen.com/opspro-essentials/critical-cost-downtime/`)

Rezaeianjouybari, B., & Shang, Y. (2020). Deep learning for prognostics and health management: State of the art, challenges, and opportunities. *Measurement*, *163*, 107929.

Rupprecht, T., & Wang, Y. (2022). A survey for deep reinforcement learning in markovian cyber–physical systems: Common problems and solutions. *Neural Networks*, *153*, 13–36.

Sadana, U., Chenreddy, A., Delage, E., Forel, A., Frejinger, E., & Vidal, T. (2024). A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*.

Saeed, A., A. Khan, M., Akram, U., J. Obidallah, W., Jawed, S., & Ahmad, A. (2025). Deep learning based approaches for intelligent industrial machinery health management and fault diagnosis in resource-constrained environments. *Scientific Reports*, *15*(1), 1114.

Sharma, J., Mittal, M. L., & Soni, G. (2024). Condition-based maintenance using machine learning and role of interpretability: a review. *International Journal of System Assurance Engineering and Management*, *15*(4), 1345–1360.

Wang, W. (2012). An overview of the recent advances in delay-time-based maintenance modelling. *Reliability Engineering & System Safety*, *106*, 165–178.

Wang, Y., Zhao, Y., & Addepalli, S. (2020). Remaining useful life prediction using deep learning approaches: A review. *Procedia manufacturing*, *49*, 81–88.

Wesendrup, K., & Hellingrath, B. (2023). Post-prognostics demand management, production, spare parts and maintenance planning for a single-machine system using reinforcement learning. *Computers & Industrial Engineering*, *179*, 109216.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*, 229–256.

Xiang, X., & Foo, S. (2021). Recent advances in deep reinforcement learning applications for solving partially observable markov decision processes (pomdp) problems: Part 1—fundamentals and applications in games, robotics and natural language processing. *Machine Learning and Knowledge Extraction*, *3*(3), 554–581.

Yildirim, M., Gebraeel, N. Z., & Sun, X. A. (2017). Integrated predictive analytics and optimization for opportunistic maintenance and operations in wind farms. *IEEE Transactions on power systems*, *32*(6), 4319–4328.

Yildirim, M., Sun, X. A., & Gebraeel, N. Z. (2016). Sensor-driven condition-based generator maintenance scheduling—part i: Maintenance problem. *IEEE Transactions on Power Systems*, *31*(6), 4253–4262.

Zhang, Y., Zhang, D., Zhang, X., Qiu, L., Chan, F. T., Wang, Z., & Zhang, S. (2023). Guided probabilistic reinforcement learning for sampling-efficient maintenance scheduling of multi-component system. *Applied Mathematical Modelling*, *119*, 677–697.

Zhu, J., Wang, Y., Huang, Y., Bhushan Gopaluni, R., Cao, Y., Heere, M., … others (2022). Data-driven capacity estimation of commercial lithium-ion batteries from voltage relaxation. *Nature communications*, *13*(1), 2261.