



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Eduardo Cunha Cabral

**PROGFORMER: UM MODELO TRANSFORMER QUE EXPLORA
ESTRATÉGIAS DE TRANSIÇÃO E BALANCEAMENTO DINÂMICO DE PERDAS
PARA PREDIÇÃO DE VIDA ÚTIL RESTANTE EM MOTORES TURBOFAN**

Florianópolis
2026

Eduardo Cunha Cabral

**PROGFORMER: UM MODELO TRANSFORMER QUE EXPLORA
ESTRATÉGIAS DE TRANSIÇÃO E BALANCEAMENTO DINÂMICO DE PERDAS
PARA PREDIÇÃO DE VIDA ÚTIL RESTANTE EM MOTORES TURBOFAN**

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Ciência da Computação.

Orientador: Prof. Eduardo Camilo Inacio, Dr.

Florianópolis

2026

Catálogo na fonte pela Biblioteca Universitária da Universidade Federal de Santa Catarina.
Arquivo compilado às 17:05h do dia 11 de julho de 2026.

Eduardo Cunha Cabral

PROGFORMER: UM MODELO TRANSFORMER QUE EXPLORA ESTRATÉGIAS DE TRANSIÇÃO E BALANCEAMENTO DINÂMICO DE PERDAS PARA PREDIÇÃO DE VIDA ÚTIL RESTANTE EM MOTORES TURBOFAN / Eduardo Cunha Cabral; Orientador, Prof. Eduardo Camilo Inacio, Dr. - Florianópolis, 15 de junho de 2026.

79 p.

Trabalho de Conclusão de Curso - Universidade Federal de Santa Catarina, Departamento de Informática e Estatística, Curso de Graduação em Ciência da Computação.

Inclui referências

1. Manutenção Preditiva, 2. Vida Útil Remanescente, 3. Deep Learning, 4. Transformer, 5. Função de Perda Assimétrica, I. Prof. Eduardo Camilo Inacio, Dr. II. Curso de Graduação em Ciência da Computação III. PROGFORMER: UM MODELO TRANSFORMER QUE EXPLORA ESTRATÉGIAS DE TRANSIÇÃO E BALANCEAMENTO DINÂMICO DE PERDAS PARA PREDIÇÃO DE VIDA ÚTIL RESTANTE EM MOTORES TURBOFAN

CDU 02:141:005.7

Eduardo Cunha Cabral

**PROGFORMER: UM MODELO TRANSFORMER QUE EXPLORA
ESTRATÉGIAS DE TRANSIÇÃO E BALANCEAMENTO DINÂMICO DE PERDAS
PARA PREDIÇÃO DE VIDA ÚTIL RESTANTE EM MOTORES TURBOFAN**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação, e foi aprovado em sua forma final pelo Curso de Graduação em Ciência da Computação do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina.

Florianópolis, 15 de junho de 2026.

Profa. Lúcia Helena Martins Pacheco, Dra.
Coordenador(a) do Curso de Graduação em
Ciência da Computação

Banca Examinadora:

Prof. Eduardo Camilo Inacio, Dr.
Orientador
Universidade Federal de Santa Catarina – UFSC

Profa. Patrícia Della Mea Plentz, Dra.
Universidade Federal de Santa Catarina, UFSC

Profa. Analucia Schiaffino Morales, Dra.
Universidade Federal de Santa Catarina, UFSC

RESUMO

A Manutenção Preditiva (PdM) desempenha um papel crítico na indústria aeronáutica, onde a estimativa precisa da Vida Útil Remanescente (RUL) de componentes, como motores turbofan, é essencial para garantir a segurança operacional e a eficiência logística. Embora abordagens baseadas em *Deep Learning*, como os *Transformers*, tenham demonstrado alta capacidade de modelagem sequencial, as estratégias de treinamento convencionais baseadas na minimização do Erro Quadrático Médio (MSE) falham em capturar a assimetria de risco inerente ao problema, onde previsões tardias podem levar a falhas catastróficas. Além disso, a otimização direta de funções de perda assimétricas baseadas em risco frequentemente causa instabilidade numérica e colapso do modelo em regimes operacionais complexos devido à explosão de gradientes. Para mitigar esse problema, este trabalho propõe o modelo *ProgFormer* (*Prognostics Transformer*) para a previsão de RUL, introduzindo estratégias avançadas de dinâmica de treinamento: um *framework* de Transição de Objetivo em Fases (*Phased Objective Transition* - POT) e um mecanismo de Balanceamento Dinâmico de Perdas (*Dynamic Loss Balancing* - DLB). Os experimentos realizados no conjunto de dados C-MAPSS demonstraram que a introdução da perda assimétrica reduziu de forma consistente a pontuação de risco operacional nos regimes estáveis. Nos regimes de múltipla condição, as estratégias POT e DLB do *ProgFormer* foram determinantes para estabilizar a dinâmica de aprendizado e evitar o colapso do modelo. A abordagem desenvolvida alcançou resultados robustos mesmo no cenário mais crítico, apresentando-se como uma alternativa viável em relação às arquiteturas consolidadas para a implementação de sistemas prognósticos industriais focados em segurança.

Palavras-chave: Manutenção Preditiva; Vida Útil Remanescente; Deep Learning; Transformer; Função de Perda Assimétrica.

ABSTRACT

Predictive Maintenance (PdM) plays a critical role in the aeronautical industry, where accurately estimating the Remaining Useful Life (RUL) of components, such as turbofan engines, is essential for ensuring operational safety and logistic efficiency. Although Deep Learning-based approaches, such as Transformers, have demonstrated high sequential modeling capacity, conventional training strategies based on minimizing Mean Squared Error (MSE) fail to capture the inherent asymmetric risk of the problem, where late predictions can lead to catastrophic failures. Furthermore, the direct optimization of risk-based asymmetric loss functions often causes numerical instability and model collapse in complex operational regimes due to exploding gradients. To mitigate this problem, this work proposes the ProgFormer (Prognostics Transformer) model for RUL prediction, introducing advanced training dynamics strategies: a Phased Objective Transition (POT) framework and a Dynamic Loss Balancing (DLB) mechanism. Experiments conducted on the C-MAPSS dataset demonstrated that the introduction of the asymmetric loss consistently reduced the operational risk score in stable regimes. In multiple operating condition regimes, the ProgFormer's POT and DLB strategies were decisive in stabilizing the learning dynamics and preventing model collapse. The developed approach achieved robust results even in the most critical scenario, presenting itself as a viable alternative compared to the consolidated architectures for implementing safety-focused industrial prognostic systems.

Keywords: Predictive Maintenance; Remaining Useful Life; Deep Learning; Transformer; Asymmetric Loss Function.

LISTA DE FIGURAS

Figura 1	–	Arquitetura do modelo Transformer	22
Figura 2	–	Configuração esquemática do motor turbofan simulado no C-MAPSS . . .	24
Figura 3	–	Comparação da Função Score com o RMSE	26
Figura 4	–	Visão geral do <i>ProgFormer</i> , evidenciando a arquitetura base e a seleção das estratégias de otimização.	31
Figura 5	–	Dispersão das predições de RUL utilizando MSE Padrão para os quatro conjuntos de dados C-MAPSS.	44
Figura 6	–	Dispersão das predições de RUL utilizando otimização Assimétrica Direta (Perda C-MAPSS).	46
Figura 7	–	Dispersão das predições de RUL utilizando a Função Híbrida Estática. . .	48
Figura 8	–	Dispersão das predições de RUL estabilizadas pela estratégia <i>Binary</i> POT. . .	50
Figura 9	–	Dispersão das predições de RUL estabilizadas pela estratégia <i>Hybrid</i> POT. . .	52
Figura 10	–	Dispersão das predições de RUL utilizando a estratégia DLB.	54

LISTA DE TABELAS

Tabela 1	–	Comparação de Abordagens para Predição de RUL	29
Tabela 2	–	Comparação de Abordagens para Predição de RUL incluindo o modelo proposto	35
Tabela 3	–	Desempenho preditivo do modelo <i>baseline</i> treinado com Erro Quadrático Médio (MSE).	43
Tabela 4	–	Desempenho preditivo do modelo otimizado de forma direta pela Função de Perda C-MAPSS.	45
Tabela 5	–	Desempenho preditivo do modelo otimizado pela Função Híbrida Estática (MSE + C-MAPSS).	47
Tabela 6	–	Desempenho preditivo do modelo otimizado por <i>Binary</i> POT nos regimes multivariados.	49
Tabela 7	–	Desempenho preditivo do modelo otimizado por <i>Hybrid</i> POT.	51
Tabela 8	–	Desempenho preditivo do modelo otimizado por DLB.	53
Tabela 9	–	Comparação de desempenho do <i>ProgFormer</i> com as abordagens de referência.	55

LISTA DE ABREVIATURAS E SIGLAS

PdM	Manutenção Preditiva
PM	Manutenção Preventiva
RUL	Vida Útil Remanescente
RNN	Rede Neural Recorrente
LSTM	Memória de Curto Prazo Longo
MSE	Erro Quadrático Médio
ProgFormer	Transformer para Prognóstico
C-MAPSS	Simulação Comercial Modular de Sistema Aero-Propulsivo
POT	Transição de Objetivo em Fases
DLB	Balanceamento Dinâmico de Perdas
MLP	Perceptron de Múltiplas Camadas
ReLU	Unidade Linear Retificada
Adam	Estimação de Momento Adaptativo
CNN	Rede Neural Convolucional
EMA	Média Móvel Exponencial
RMSE	Raiz do Erro Quadrático Médio
BiLSTM	Memória de Curto Prazo Longo Bidirecional
DCNN	Rede Neural Convolucional Profunda
ARR	Técnica de Reponderação Adaptativa
FFG	Portão de Fusão de Características
PVA	Ângulo Vetorial de Predição
GAT	Rede de Atenção em Grafos
PLO	Optimizador de Luz Polar
FFN	Rede Feed-Forward
AutoDiff	Diferenciação Automática

GAP	Agregação Global Média
IAG	Inteligência Artificial Generativa

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	15
1.1.1	Objetivos Específicos	15
1.2	ORGANIZAÇÃO DO DOCUMENTO	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	APRENDIZADO DE MÁQUINA E REDES NEURAIS	17
2.1.1	<i>Deep Learning</i>	17
2.1.2	O Neurônio Artificial	18
2.1.3	<i>Multilayer Perceptron</i>	18
2.1.4	Otimização de Modelos	19
2.1.5	Redes Neurais Convolucionais	19
2.1.6	Redes Neurais Recorrentes	20
2.1.7	<i>Arquitetura Transformer</i>	21
2.2	ESTRATÉGIAS AVANÇADAS DE TREINAMENTO	22
2.2.1	Aprendizado Curricular (<i>Curriculum Learning</i>)	22
2.2.2	Ponderação Adaptativa e Média Móvel Exponencial	23
2.3	C-MAPSS E RUL	23
2.3.1	Manutenção Preditiva	23
2.3.2	A Simulação C-MAPSS	24
2.3.3	Métricas de Avaliação	24
3	TRABALHOS RELACIONADOS	27
4	PROGFORMER	30
4.1	VISÃO GERAL DO <i>PROGFORMER</i>	30
4.2	ARQUITETURA BASE <i>TRANSFORMER ENCODER</i>	31
4.2.1	<i>Input Embedding e Positional Encoding</i>	31
4.2.2	<i>Encoder Block</i>	32
4.2.3	<i>Regression Head</i>	32
4.3	DINÂMICAS DE TREINAMENTO	33
4.3.1	<i>Phased Objective Transition</i>	33
4.3.2	<i>Dynamic Loss Balancing</i>	34
5	METODOLOGIA	36
5.1	PREPARAÇÃO DOS DADOS E PRÉ-PROCESSAMENTO	36
5.1.1	Organização do Conjunto de Dados C-MAPSS	36
5.1.2	Seleção de Atributos	36

5.1.3	Normalização e Escalonamento dos Dados	37
5.1.4	Definição da Variável Alvo	37
5.1.5	Geração de Sequências	37
5.2	TREINAMENTO E OTIMIZAÇÃO	38
5.2.1	Funções Objetivo	38
5.2.2	Modelo Híbrido Estático	39
5.2.3	Transição de Objetivo em Fases	39
5.2.4	Balanceamento Dinâmico de Perdas	40
5.3	PROTOCOLO DE AVALIAÇÃO	40
5.3.1	Métricas de Avaliação	40
5.3.2	Configuração dos Experimentos	41
5.3.3	Uso de Inteligência Artificial Generativa	41
6	RESULTADOS	43
6.0.1	Desempenho do Modelo Base MSE	43
6.0.2	Perda C-MAPSS	45
6.0.3	Função Híbrida Estática	47
6.0.4	Transição de Objetivo em Fases	49
6.0.5	Balanceamento Dinâmico de Perdas	52
6.0.6	Análise Comparativa com os Trabalhos Relacionados	55
7	CONCLUSÕES	57
	REFERÊNCIAS	59
	APÊNDICE A – CÓDIGO FONTE	63
	APÊNDICE B – ARTIGO	64

1 INTRODUÇÃO

O cenário industrial contemporâneo vivencia uma transformação profunda, frequentemente caracterizada como Indústria 4.0, onde a conexão entre máquinas inteligentes, automação digital e análise avançada de dados redefine a eficiência operacional. No centro desta revolução encontra-se uma mudança de paradigma na gestão do ciclo de vida de ativos críticos: a transição de estratégias de manutenção tradicionais para a Manutenção Preditiva (*Predictive Maintenance* - PdM). Esta mudança é particularmente vital em setores onde a segurança é inegociável, como na indústria aeronáutica. Neste cenário, os motores aeronáuticos, especificamente os *turbofans*, destacam-se como sistemas termodinâmicos altamente complexos, projetados para operar sob extremas condições de pressão, temperatura e intenso estresse mecânico. Diferentemente de maquinários industriais comuns, onde uma falha resulta primordialmente em perdas financeiras por tempo de inatividade, a degradação não detectada em componentes críticos de um *turbofan* em voo pode culminar em falhas catastróficas e fatalidades (DUAN; ZHANG; SHI, 2024). Devido a essa severidade operacional, atuar no domínio de prognósticos aeronáuticos exige ir além da simples precisão estatística, tornando imperativa a adoção de estratégias preditivas que priorizem ativamente a mitigação de riscos (SAXENA *et al.*, 2008).

Historicamente, as filosofias de manutenção evoluíram através de estágios distintos. A forma mais primitiva, a Manutenção Corretiva, opera de forma reativa, permitindo que o ativo funcione até falhar. Embora maximize a utilização teórica do componente, essa abordagem é operacionalmente inaceitável na aviação devido aos riscos severos à segurança. As limitações desse modelo levaram à adoção da Manutenção Preventiva (*Preventive Maintenance* - PM), baseada em intervenções agendadas estatisticamente. Contudo, pesquisas indicam que a PM é inerentemente ineficiente, resultando frequentemente na substituição prematura de componentes que ainda possuem vida útil significativa (SAXENA *et al.*, 2008). A PdM representa o ápice desta evolução, impulsionada pelo Monitoramento Baseado em Condição para prever o estado futuro do equipamento.

A métrica central neste domínio é a Vida Útil Remanescente (*Remaining Useful Life* - RUL). A RUL é definida como o tempo, tipicamente medido em ciclos operacionais ou horas, desde o momento atual até que o ativo atinja um limiar de falha onde não pode mais desempenhar sua função com segurança (SAXENA *et al.*, 2008). A estimativa precisa da RUL é uma tarefa de prognóstico que permite aos operadores agendar manutenções dinamicamente, otimizando a logística e a disponibilidade da frota. Apesar da transparência oferecida pelos modelos determinísticos, que descrevem a propagação de danos via equações diferenciais, a complexidade computacional inerente a esses métodos restringe sua utilização em sistemas que apresentam múltiplas interações dinâmicas.

Para a realização da predição da RUL, as metodologias estabelecidas na literatura dividem-se fundamentalmente em três categorias: modelos baseados na física, modelos estatísticos e matemáticos, e modelos orientados a dados (DUAN; ZHANG; SHI, 2024). Os modelos baseados

na física utilizam princípios de mecânica de fratura, fadiga e termodinâmica para descrever exatamente a propagação do dano ao longo do tempo. Apesar da transparência causal oferecida por essas equações diferenciais, a imensa dificuldade em modelar todas as interações dinâmicas e o alto custo computacional restringem fortemente sua aplicação em sistemas aeronáuticos multivariados. Como alternativa, os modelos estatísticos, a exemplo dos Filtros de Kalman e dos processos de Markov, estimam a degradação de forma probabilística. No entanto, essas abordagens frequentemente assumem premissas de linearidade ou distribuições de ruído ideais que falham em refletir as severas flutuações causadas pelas inconstantes condições de voo reais (DUAN; ZHANG; SHI, 2024).

Consequentemente, o campo migrou para abordagens orientadas a dados. Facilitados pela disponibilidade de grandes bases de dados históricas, esses métodos aprendem o mapeamento funcional entre as entradas dos sensores e os estados de degradação diretamente a partir dos dados. Neste contexto, modelos de *Deep Learning*, como as Redes Neurais Recorrentes (*Recurrent Neural Networks* - RNNs) e, mais especificamente, as *Long Short-Term Memory* (LSTMs), estabeleceram-se historicamente como um padrão de referência devido à sua capacidade de capturar dependências temporais em dados sequenciais (TIPPAREDDY, 2024). Para superar as dificuldades inerentes às RNNs no processamento de sequências longas e complexas (GOOD-FELLOW; BENGIO; COURVILLE, 2016), arquiteturas mais recentes, como os *Transformers*, têm se destacado como alternativas promissoras.

Embora a aplicação de técnicas de *Deep Learning* na prognose de falhas tenha demonstrado resultados promissores em termos de precisão geral, a implementação prática desses modelos em sistemas críticos de segurança enfrenta desafios substanciais que não são adequadamente resolvidos pelas abordagens convencionais. A literatura atual frequentemente prioriza a minimização do Erro Quadrático Médio (*Mean Squared Error* - MSE), uma métrica simétrica que penaliza erros de predição positivos (atrasados) e negativos (adiantados) com a mesma severidade. No entanto, no contexto da manutenção aeronáutica, esses erros possuem consequências drasticamente distintas. Uma predição adiantada resulta apenas em manutenção conservadora, enquanto uma predição atrasada pode levar a falhas catastróficas em serviço (SAXENA *et al.*, 2008). Portanto, a ausência de mecanismos de otimização que priorizem explicitamente a segurança operacional constitui uma lacuna crítica na pesquisa vigente.

Outro obstáculo significativo reside na disparidade de desempenho entre ambientes controlados e cenários operacionais reais. Enquanto modelos padrão apresentam excelente acurácia em condições estáveis, seus desempenhos degradam quando submetidos a regimes de voo variáveis. Estudos indicam que arquiteturas convencionais encontram dificuldades em modelar as complexas dependências espaço-temporais inerentes a esses regimes dinâmicos, resultando em uma capacidade de generalização limitada que compromete a viabilidade prática do sistema em operações de campo (ZENG; WU; YIWEI, 2023).

Para avaliar empiricamente estes desafios e validar as propostas arquiteturais sem cometer riscos físicos ou esbarrar na confidencialidade de dados industriais, a comunidade científica recorre a simulações de alta fidelidade. Neste cenário, o conjunto de dados C-MAPSS (*Com-*

mercial Modular Aero-Propulsion System Simulation), desenvolvido pela NASA (FREDERICK; DECASTRO; LITT, 2007), estabeleceu-se como a principal referência. Ao fornecer trajetórias de degradação de motores *turbofan* até a falha completa (*run-to-failure*) sob múltiplas condições de voo e modos de desgaste, este simulador cria o ambiente multivariado ideal para testar a estabilidade e a eficácia das estratégias avançadas de treinamento propostas neste trabalho.

1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver um modelo baseado em *Transformers* para a predição da RUL em sistemas aeronáuticos complexos, incorporando estratégias de treinamento que priorizem a segurança operacional e garantam a estabilidade da convergência em regimes multivariados.

1.1.1 Objetivos Específicos

Para alcançar o objetivo geral proposto, foram definidos os seguintes objetivos específicos:

- Construir uma função de perda assimétrica orientada à segurança, visando mitigar o risco de predições tardias perigosas e quantificar o ganho de confiabilidade em relação à otimização simétrica padrão.
- Desenvolver estratégias de gerenciamento da dinâmica de treinamento para equilibrar as funções de erro e mitigar a instabilidade numérica em superfícies complexas de aprendizado.
- Propor uma arquitetura baseada em *Transformers* capaz de processar séries temporais multivariadas e integrar as restrições de segurança na estimativa da RUL.
- Avaliar o desempenho do modelo proposto em diferentes níveis de complexidade operacional, utilizando os *datasets* padronizados do C-MAPSS.
- Analisar o desempenho da abordagem proposta em comparação com arquiteturas preditivas consolidadas na literatura, como os modelos baseados em redes recorrentes (RNNs).

1.2 ORGANIZAÇÃO DO DOCUMENTO

O restante deste trabalho está estruturado em seis capítulos, organizados para conduzir o leitor desde os conceitos fundamentais até a análise dos experimentos realizados. O Capítulo 2 apresenta a Fundamentação Teórica, estabelecendo as bases sobre PM e o *dataset* C-MAPSS, além de revisar os fundamentos de redes neurais e a arquitetura *Transformer*. O Capítulo 3 discute os trabalhos relacionados que motivaram esta pesquisa, mapeando o panorama de modelos preditivos na literatura. O Capítulo 4 detalha a proposta do modelo, definindo a sua concepção estrutural e os princípios teóricos de sua dinâmica de treinamento orientada a risco.

Em seguida, o Capítulo 5 descreve a Metodologia, detalhando o *pipeline* de pré-processamento, a parametrização da rede e a formulação matemática das estratégias POT e DLB. O Capítulo 6 apresenta e discute os Resultados, detalhando a avaliação experimental nos diferentes regimes de complexidade. Por fim, o Capítulo 7 apresenta as conclusões deste trabalho de pesquisa, sintetizando as contribuições e delineando caminhos para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo estabelece as bases conceituais necessárias para a compreensão dos modelos preditivos desenvolvidos. Inicialmente, são abordados os fundamentos das redes neurais artificiais e seus mecanismos de otimização. Em seguida, detalham-se as arquiteturas de processamento de sequências, com ênfase nas RNNs e *Transformers*. Posteriormente, apresentam-se estratégias avançadas de treinamento para otimização sob múltiplos objetivos. Por fim, discute-se o domínio da manutenção preditiva e as especificidades do conjunto de dados C-MAPSS.

2.1 APRENDIZADO DE MÁQUINA E REDES NEURAIAS

O Aprendizado de Máquina é um subcampo da inteligência artificial que se dedica ao desenvolvimento de algoritmos capazes de aprender padrões a partir de dados, sem serem explicitamente programados para regras específicas (GOODFELLOW; BENGIO; COURVILLE, 2016). No contexto da manutenção preditiva, a tarefa é frequentemente modelada como um problema de regressão supervisionada, onde o objetivo é mapear um conjunto de entradas sensoriais x (como temperatura e pressão) para uma saída contínua y (a Vida Útil Remanescente) (LEI *et al.*, 2022).

2.1.1 *Deep Learning*

O *Deep Learning* é uma subárea do aprendizado de máquina que busca resolver problemas centrais da inteligência artificial através da construção de modelos complexos a partir de conceitos mais simples. A premissa fundamental desta abordagem é permitir que o computador aprenda pela experiência e entenda o mundo em termos de uma hierarquia de conceitos, onde cada conceito é definido através de sua relação com conceitos mais simples. Essa hierarquia, quando visualizada, possui muitas camadas de profundidade, o que origina o termo “aprendizado profundo” (LECUN; BENGIO; HINTON, 2015).

A principal distinção entre o *Deep Learning* e os algoritmos de aprendizado de máquina tradicionais reside na capacidade de realizar o Aprendizado de Representações. Em abordagens clássicas, o desempenho do modelo depende fortemente da extração manual de atributos, onde o conhecimento de domínio humano é necessário para transformar dados brutos em uma representação adequada para o classificador ou regressor. No entanto, para muitas tarefas complexas, como a análise de séries temporais multivariadas de sensores de aeronaves, é difícil saber quais características devem ser extraídas previamente (GOODFELLOW; BENGIO; COURVILLE, 2016).

O *Deep Learning* soluciona esse problema aprendendo as representações diretamente a partir dos dados brutos. Uma rede neural profunda é composta por múltiplas camadas de processamento, onde cada camada aplica uma transformação não-linear à saída da camada anterior. Dessa forma, as camadas iniciais podem aprender características de baixo nível (como

flutuações simples em um sensor), enquanto as camadas subsequentes agregam essas informações para formar representações abstratas de alto nível, como o estado de saúde global do motor. Essa capacidade de descobrir automaticamente a estrutura latente em dados de alta dimensionalidade é o que torna o *Deep Learning* particularmente eficaz para a prognose de falhas em sistemas complexos (LEI *et al.*, 2022).

2.1.2 O Neurônio Artificial

A unidade elementar de processamento nas redes neurais é o neurônio artificial. Inspirado na neurobiologia, este modelo matemático simplificado opera como um processador de informação que recebe múltiplos sinais de entrada e produz um único sinal de saída. Segundo Bishop (2006), a operação fundamental de um neurônio consiste em calcular uma combinação linear das variáveis de entrada $x = \{x_1, \dots, x_D\}$, ponderada por um vetor de parâmetros adaptativos $w = \{w_1, \dots, w_D\}$, conhecidos como pesos sinápticos.

Matematicamente, o neurônio agrega as informações de entrada somando os produtos de cada entrada pelo seu respectivo peso e adicionando um parâmetro de viés, denotado por w_0 ou b . O resultado dessa operação linear é expresso na Equação 2.1:

$$a = \sum_{i=1}^D w_i x_i + w_0 \quad (2.1)$$

onde a representa o valor da pré-ativação. Para que o neurônio desempenhe uma função útil em tarefas de classificação ou regressão não-linear, essa soma ponderada a é processada por uma função de ativação, resultando na saída final $y = f(a)$. A introdução de uma não-linearidade nesta etapa é fundamental: sem ela, uma rede neural profunda, independentemente do número de camadas, iria se comportar matematicamente como uma única transformação linear, limitando drasticamente sua capacidade de representação.

Dentre as abordagens não-lineares, destaca-se a função de Unidade Linear Retificada (*Rectified Linear Unit* - ReLU), definida como $f(a) = \max(0, a)$. A ReLU consolidou-se como o padrão para o processamento de camadas ocultas por oferecer propriedades computacionais vantajosas: é eficiente de calcular, promove a esparsidade nas ativações e mitiga o problema do desaparecimento de gradiente, permitindo o treinamento eficaz de arquiteturas profundas (GLOROT; BORDES; BENGIO, 2011). Por outro lado, para a camada de saída em tarefas de regressão, como a estimativa da RUL, utiliza-se tipicamente uma função de ativação Linear. Isso permite que a rede preveja valores contínuos, mapeando os padrões aprendidos diretamente para o número de ciclos restantes. O processo de aprendizado da rede consiste, portanto, no ajuste iterativo dos parâmetros w e w_0 para que essa saída y se aproxime do valor alvo desejado (BISHOP, 2006).

2.1.3 Multilayer Perceptron

O *Multilayer Perceptron* (MLP) constitui a classe mais proeminente de redes neurais *feed-forward*. Esses modelos podem ser entendidos como uma extensão das funções lineares,

onde as próprias funções de base (camadas ocultas) são adaptativas e aprendidas a partir dos dados, em vez de serem fixas previamente (HAYKIN, 2009). A arquitetura organiza-se em camadas de neurônios conectadas sequencialmente, permitindo que a informação flua da entrada, através de uma ou mais camadas ocultas, até a saída, sem a formação de ciclos.

A capacidade de modelagem do MLP decorre diretamente da introdução das não-linearidades (como a ReLU) entre essas camadas. Enquanto uma rede de camada única é limitada a resolver problemas linearmente separáveis, a adição de camadas ocultas com funções de ativação não-lineares confere à rede a capacidade de extrair características progressivamente mais complexas. Hornik, Stinchcombe e White (1989) destacam que redes de duas camadas (com uma camada oculta), utilizando funções de ativação adequadas, satisfazem a propriedade de aproximação universal, sendo capazes de aproximar qualquer função contínua em um domínio compacto com precisão arbitrária, desde que o número de unidades ocultas seja suficiente. Em modelos voltados para a estimativa escalar da RUL, as redes MLP desempenham frequentemente o papel crucial de regressor final, mapeando as representações temporais de alta dimensão extraídas por arquiteturas profundas (como os *Transformers*) para a predição definitiva.

2.1.4 Otimização de Modelos

O processo de aprendizado consiste em minimizar uma Função de Perda (*Loss Function*), denotada por $J(\theta)$, que quantifica a discrepância entre a predição do modelo $\hat{y}$ e o valor real y . Em problemas de regressão, a métrica padrão é o MSE, dado pela Equação 2.2:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.2)$$

A minimização dessa função é realizada através do algoritmo de Retropropagação, que aplica a regra da cadeia do cálculo diferencial para computar o gradiente da função de perda em relação a cada peso da rede. Esses gradientes indicam a direção na qual os pesos devem ser ajustados para reduzir o erro (RUMELHART; HINTON; WILLIAMS, 1986).

Para a atualização dos pesos em arquiteturas profundas, emprega-se frequentemente o otimizador Adam (*Adaptive Moment Estimation*) (KINGMA; BA, 2014). O Adam é um algoritmo de gradiente descendente estocástico que calcula taxas de aprendizado adaptativas para cada parâmetro. Ele armazena uma média móvel exponencial dos gradientes passados e dos gradientes ao quadrado, permitindo uma convergência mais estável e rápida em superfícies de erro complexas e ruidosas, como as encontradas em dados de sensores industriais.

2.1.5 Redes Neurais Convolucionais

Embora as Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNN) tenham sido popularizadas pelo sucesso no processamento de imagens, seus princípios matemáticos são diretamente aplicáveis a dados sequenciais. Ao invés de conectar todos os neurônios à entrada

completa, a operação de convolução aplica filtros que percorrem a janela de dados, isolando e extraindo padrões morfológicos locais da série temporal (LECUN; BENGIO; HINTON, 2015).

No contexto de séries temporais unidimensionais, a convolução é aplicada deslizando-se um filtro sobre a sequência de entrada ao longo do eixo temporal. Matematicamente, para uma entrada x e um filtro w , a operação de convolução discreta $s(t)$ em um passo de tempo t é definida conforme a Equação 2.3:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (2.3)$$

Onde o símbolo $*$ denota o operador matemático de convolução e a representa o índice temporal auxiliar sobre o qual a soma é computada. Essa operação permite que a rede extraia características locais invariantes à translação, identificando padrões como picos ou tendências de curto prazo, independentemente de onde ocorram na janela temporal. A eficácia das CNNs reside na sua capacidade de aprender hierarquias de características através do empilhamento de camadas, reduzindo a dimensionalidade dos dados enquanto preserva as informações salientes para a tarefa de prognóstico (FERREIRA; GONÇALVES, 2022).

2.1.6 Redes Neurais Recorrentes

As RNNs são arquiteturas projetadas nativamente para o processamento de sequências, por meio de conexões de retroalimentação que mantêm um estado latente contínuo, atuando como uma memória do contexto histórico. A cada passo de tempo t , a rede recebe uma entrada x_t e o estado oculto do passo anterior h_{t-1} , atualizando seu estado interno. No entanto, as RNNs tradicionais sofrem com o problema do desaparecimento de gradiente, o que dificulta o aprendizado de dependências temporais longas (GOODFELLOW; BENGIO; COURVILLE, 2016).

Para solucionar essa limitação, Hochreiter e Schmidhuber (1997) propuseram a arquitetura *Long Short-Term Memory* (LSTM). A versão moderna dessa arquitetura, consolidada na literatura (GOODFELLOW; BENGIO; COURVILLE, 2016), introduz uma célula de memória cujo fluxo é regulado por três unidades principais:

- **Portão de Esquecimento (*Forget Gate*):** Decide qual informação descartar do estado da célula.
- **Portão de Entrada (*Input Gate*):** Decide qual nova informação armazenar na célula.
- **Portão de Saída (*Output Gate*):** Controla qual informação será passada para o próximo estado oculto.

Essa estrutura permite que a rede mantenha o gradiente estável durante o treinamento por retropropagação através do tempo, tornando as LSTMs eficazes na captura de padrões de degradação que evoluem lentamente ao longo de centenas de ciclos (HOCHREITER; SCHMIDHUBER, 1997).

2.1.7 Arquitetura *Transformer*

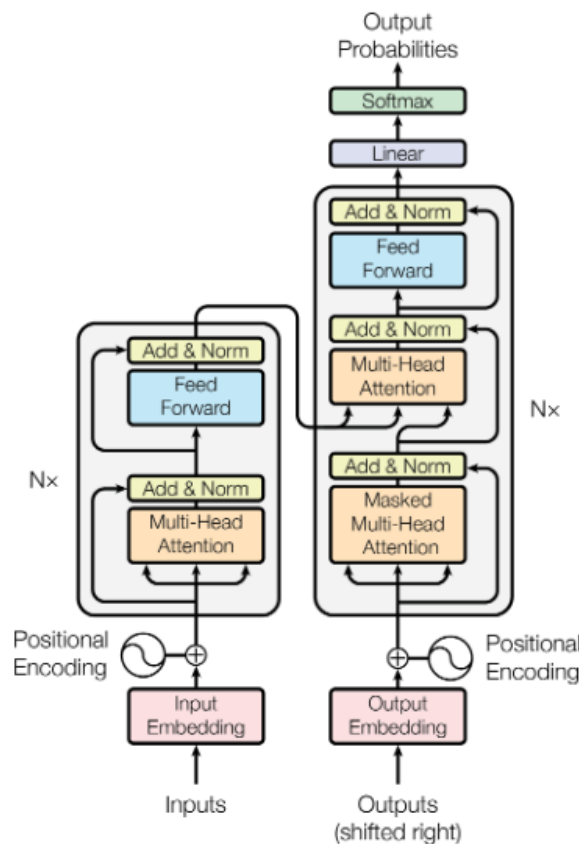
O *Transformer* representa uma mudança de paradigma na modelagem de sequências ao descartar totalmente a recorrência em favor do mecanismo de Auto-Atenção (VASWANI *et al.*, 2017). Ao contrário das RNNs, que processam os dados sequencialmente passo a passo, o *Transformer* processa toda a sequência de entrada simultaneamente, permitindo uma paralelização massiva durante o treinamento.

O núcleo da arquitetura é a função de atenção, que mapeia uma consulta e um conjunto de pares chave-valor para uma saída. A atenção calcula uma soma ponderada dos valores, onde o peso atribuído a cada valor é determinado pela compatibilidade entre a consulta e a chave correspondente. A equação da "Atenção por Produto Escalar Escalado" (*Scaled Dot-Product Attention*) é definida na Equação 2.4:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.4)$$

onde d_k é a dimensão das chaves, utilizada como fator de escala para evitar gradientes extremamente pequenos. Esse mecanismo permite que o modelo capture dependências globais entre quaisquer dois pontos da sequência, independentemente da distância temporal entre eles. Como o modelo não possui recorrência ou convolução, ele é invariante à ordem dos dados. Para incorporar a noção de sequência, Vaswani *et al.* (2017) introduziram a Codificação Posicional que adiciona vetores fixos ou aprendíveis aos *embeddings* de entrada, fornecendo ao modelo informações sobre a posição relativa ou absoluta de cada elemento na série temporal.

Figura 1 – Arquitetura do modelo Transformer



Fonte: (VASWANI *et al.*, 2017)

Para refinar a capacidade do modelo de focar em diferentes aspectos da informação simultaneamente, a arquitetura utiliza o mecanismo de Atenção Múltipla (*Multi-Head Attention*), conforme ilustrado na Figura 1. Em vez de realizar uma única função de atenção, este mecanismo projeta linearmente as consultas, chaves e valores h vezes com diferentes projeções aprendidas. Estas projeções resultam em h “cabeças” de atenção paralelas, permitindo que o modelo atente conjuntamente a informações oriundas de diferentes subespaços de representação em posições distintas. As saídas dessas cabeças são concatenadas e projetadas novamente, resultando em uma representação final mais rica do que seria possível com uma única cabeça de atenção.

2.2 ESTRATÉGIAS AVANÇADAS DE TREINAMENTO

Para lidar com funções de perda complexas e instabilidades durante o processo de otimização, abordagens que modulam a dinâmica de treinamento tornam-se essenciais em arquiteturas modernas.

2.2.1 Aprendizado Curricular (*Curriculum Learning*)

Treinar redes neurais profundas sob objetivos altamente não lineares ou superfícies de erro complexas pode ser desafiador, especialmente quando ocorrem gradientes abruptos desde o

início da otimização. O Aprendizado Curricular (*Curriculum Learning*) soluciona esse problema estruturando o treinamento como uma sequência de tarefas progressivamente mais difíceis. Esta abordagem permite que o modelo desenvolva representações estáveis e generalizáveis em cenários mais simples antes de focar em objetivos especializados e rigorosos (BENGIO *et al.*, 2009). No contexto da manutenção preditiva, a transição metodológica de uma função de perda simétrica (como o MSE) para uma função assimétrica de risco pode ser interpretada como um currículo de otimização.

2.2.2 Ponderação Adaptativa e Média Móvel Exponencial

Em arquiteturas de aprendizado multi-objetivo, os modelos frequentemente combinam múltiplas funções de perda. A abordagem padrão de utilizar pesos escalares estáticos pressupõe que a importância de cada componente de perda permanece constante. Na prática, diferentes objetivos evoluem em taxas distintas e possuem magnitudes de gradiente que podem desestabilizar o aprendizado (CHEN *et al.*, 2017). Para mitigar este efeito, métodos de ponderação adaptativa ajustam dinamicamente a contribuição de cada objetivo com base no estado do treinamento.

Para estimar o comportamento dessas perdas de forma estável, emprega-se amplamente o conceito de Média Móvel Exponencial (*Exponential Moving Average* - EMA). O EMA produz uma representação suavizada de sinais temporais contínuos, reduzindo drasticamente a sensibilidade a ruídos pontuais e flutuações de curto prazo durante a otimização (KINGMA; BA, 2014).

2.3 C-MAPSS E RUL

A validação de algoritmos de prognóstico requer dados que representem fielmente a degradação de sistemas complexos. Devido à escassez de dados públicos de falhas reais em aeronaves, a comunidade científica adota simulações de alta fidelidade como padrão.

2.3.1 Manutenção Preditiva

A gestão eficiente de ativos industriais é um pilar fundamental na era da Indústria 4.0. Historicamente, as estratégias de manutenção dividiam-se em duas categorias principais: a Manutenção Corretiva, onde o reparo ocorre apenas após a falha funcional, e a Manutenção Preventiva, que se baseia em intervenções agendadas periodicamente, independentemente do estado real do equipamento. Embora a abordagem preventiva reduza o risco de falhas catastróficas comparada à corretiva, ela frequentemente resulta no desperdício de vida útil remanescente de componentes saudáveis e em custos operacionais desnecessários (FERREIRA; GONÇALVES, 2022).

Neste contexto, a PdM emerge como uma evolução estratégica. Fundamentada no Monitoramento Baseado em Condição, a PdM utiliza dados de sensores em tempo real (como vibração, temperatura e pressão) para inferir o estado de saúde atual do sistema e prognosticar

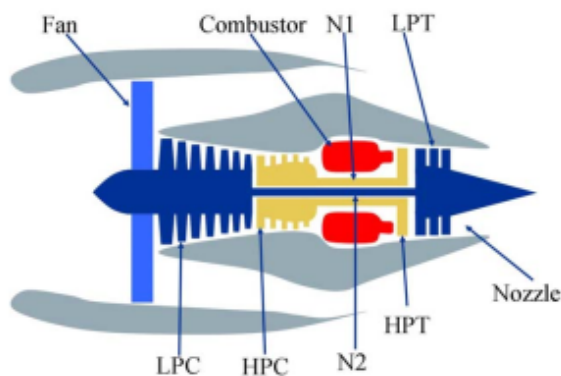
sua degradação futura. O objetivo central é realizar a intervenção no momento ótimo, antes que a falha ocorra, mas somente quando estritamente necessário, maximizando assim a disponibilidade e a confiabilidade do ativo (LEI *et al.*, 2022).

A implementação eficaz da PdM depende da capacidade de prever com precisão a RUL. A RUL é a estimativa do tempo de operação restante até que o equipamento atinja um limiar de falha funcional. Técnicas modernas baseadas em dados, particularmente aquelas que empregam *Deep Learning*, têm se destacado por sua habilidade em modelar as relações complexas e não-lineares entre os dados brutos dos sensores e a degradação do sistema, superando as limitações de escalabilidade dos modelos físicos tradicionais (FERREIRA; GONÇALVES, 2022).

2.3.2 A Simulação C-MAPSS

O conjunto de dados utilizado neste trabalho foi gerado pelo software C-MAPSS (FREDERICK; DECASTRO; LITT, 2007), desenvolvido pela NASA. O software simula um motor turbofan de classe comercial (90.000 lb de empuxo), modelando a termodinâmica de componentes rotativos essenciais, incluindo Fan, Compressor de Baixa Pressão (LPC), Compressor de Alta Pressão (HPC), Turbina de Alta Pressão (HPT) e Turbina de Baixa Pressão (LPT) (SAXENA *et al.*, 2008). A disposição desses módulos no motor simulado é ilustrada na Figura 2.

Figura 2 – Configuração esquemática do motor turbofan simulado no C-MAPSS



Fonte: (FREDERICK; DECASTRO; LITT, 2007)

2.3.3 Métricas de Avaliação

A avaliação de modelos de prognóstico em sistemas críticos exige métricas que quantifiquem não apenas a precisão média, mas também a segurança da predição. Este trabalho utiliza as duas métricas oficiais estabelecidas na competição *PHM08 Data Challenge*¹, que refletem prioridades distintas na operação aeronáutica.

A primeira métrica é a Raiz do Erro Quadrático Médio (*Root Mean Squared Error - RMSE*), amplamente utilizada em problemas de regressão. Ela penaliza erros de predição de

¹ Os dados e as métricas do desafio encontram-se disponíveis no repositório oficial da NASA: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>

forma quadrática e simétrica, fornecendo uma medida da precisão global do modelo, dada na Equação 2.5:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (2.5)$$

onde N é o número de amostras do conjunto de dados, y_i é o valor real e $\hat{y}_i$ é o valor previsto pelo modelo. No entanto, no contexto da manutenção aeronáutica, os custos associados aos erros de predição são inerentemente assimétricos. Uma predição tardia (estimar que o motor ainda possui vida útil quando, na verdade, está prestes a falhar) acarreta riscos de segurança crítica e falhas catastróficas em operação. Por outro lado, uma predição adiantada (estimar que a falha ocorrerá antes do real) resulta em manutenção preventiva precoce, o que, embora seguro, gera custos operacionais por desperdício de vida útil do componente.

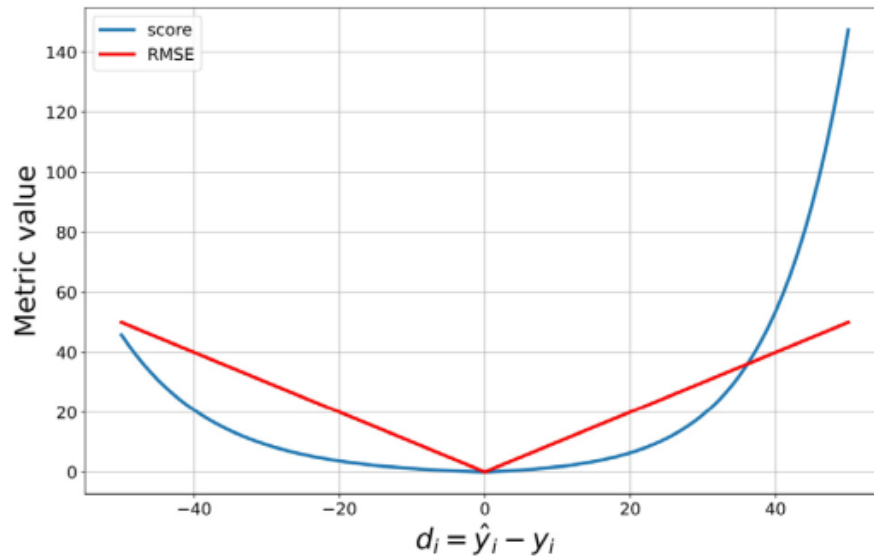
Para modelar essa assimetria de risco, utiliza-se uma função de pontuação assimétrica, definida na Equação 2.6:

$$S = \sum_{i=1}^N s_i, \quad \text{onde } s_i = \begin{cases} e^{-\frac{d_i}{13}} - 1 & \text{para } d_i < 0 \\ e^{\frac{d_i}{10}} - 1 & \text{para } d_i \geq 0 \end{cases} \quad (2.6)$$

Onde S é o *Score* total do modelo, obtido pela soma das penalidades de todas as amostras, N é o número total de unidades avaliadas, s_i é a penalidade associada à i -ésima amostra e d_i é o erro da previsão de RUL dessa amostra, definido como a diferença entre o RUL estimado e o RUL verdadeiro.

A assimetria é controlada pelos denominadores nos exponenciais: 10 para erros tardios e 13 para erros adiantados. Essa modelagem força o algoritmo de aprendizado a adotar um comportamento conservador.

Figura 3 – Comparação da Função Score com o RMSE



Fonte: (KIM; CHOI; LIM, 2024)

A Figura 3 indica visualmente que, para uma mesma magnitude de erro, a penalidade imposta pela função de pontuação assimétrica diverge significativamente daquela imposta pelo RMSE. Observa-se que, na região de predições tardias ($d_i \geq 0$), a curva da pontuação exibe um crescimento exponencial abrupto, atingindo valores de custo muito superiores aos da métrica RMSE. Esse comportamento gráfico confirma a imposição de uma penalidade severa para a subestimação do risco de falha, enquanto a região de predições adiantadas ($d_i < 0$) apresenta uma curva de crescimento mais suave, incentivando o modelo a priorizar a segurança operacional.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma revisão de pesquisas selecionadas que fundamentam a análise comparativa do modelo proposto. Ressalta-se que as obras aqui discutidas não foram elencadas por meio de uma busca exaustiva, mas escolhidas estrategicamente como uma amostragem representativa para isolar e avaliar o impacto real do gerenciamento da função de perda em contraposição à complexidade arquitetural.

Para compor esse cenário de referência, os trabalhos de Wang *et al.* (2018), Boujamza e Lissane Elhaq (2022) e Li, Ding e Sun (2018) foram selecionados para demarcar os limites de desempenho de arquiteturas clássicas estritamente recorrentes. Em paralelo, a fim de estabelecer um teto comparativo associado ao empilhamento massivo de parâmetros, incluíram-se as pesquisas híbridas de Zeng, Wu e Yiwei (2023) e Zhang *et al.* (2025). Por fim, os estudos de Kim, Choi e Lim (2024) e Zhou *et al.* (2024) foram eleitos por concorrerem diretamente com os objetivos desta pesquisa no que tange à mitigação de riscos operacionais e ao tratamento do desbalanceamento de dados.

No domínio das redes recorrentes, Wang *et al.* (2018) estabelecem um modelo robusto com a proposta de uma BiLSTM Profunda (*Deep Bidirectional LSTM*). A arquitetura utiliza camadas bidirecionais para processar as sequências de sensores tanto na ordem cronológica quanto na reversa, permitindo que o modelo capture dependências temporais passadas e futuras dentro da janela deslizante. Este trabalho demonstra que o processamento bidirecional supera as LSTMs unidirecionais padrão, tornando-se uma referência essencial para avaliar o desempenho de modelos sequenciais puros.

Avançando na complexidade das RNNs, Boujamza e Lissane Elhaq (2022) introduzem a *Attention-LSTM*, uma arquitetura híbrida que incorpora mecanismos de atenção à estrutura recorrente. Diferentemente da BiLSTM, este modelo utiliza a atenção para ponderar a importância relativa de cada passo de tempo na sequência antes do processamento final, permitindo que a rede foque em momentos críticos da degradação. Os resultados obtidos posicionam este modelo como um representante avançado das abordagens baseadas em recorrência.

Representando as abordagens não-recorrentes baseadas em convolução, Li, Ding e Sun (2018) propõem o uso de Redes Neurais Convolucionais Profundas (*Deep Convolutional Neural Networks* - DCNN). O método aplica filtros convolucionais unidimensionais ao longo do eixo temporal para extrair automaticamente atributos locais de curto prazo, eliminando a necessidade de engenharia de atributos manual. Este trabalho serve como um comparativo fundamental para demonstrar a eficácia da extração de padrões locais versus a modelagem de dependências globais realizada pelos *Transformers*.

No contexto de *Transformers*, Kim, Choi e Lim (2024) abordam o desafio do desequilíbrio de dados inerente aos conjuntos *run-to-failure*, onde amostras saudáveis superam largamente as amostras de falha. Os autores propõem uma Técnica de Reponderação Adaptativa (*Adaptive Reweighting Technique* - ARR) acoplada a um *Transformer Encoder* puro, que ajusta dinamicamente

mente a importância das amostras durante o treinamento. Esta abordagem é o competidor direto para o modelo *Transformer* proposto neste trabalho, focando na otimização via manipulação da distribuição de dados em vez da função de perda.

Focando na segurança operacional, Zhou *et al.* (2024) apresentam o modelo FFG-Transformer, uma arquitetura híbrida que combina CNNs e *Transformers* através de um portão de fusão de atributos (*Feature Fusion Gate*). A principal contribuição deste trabalho é a introdução da função de perda PVA (*Prediction Vector Angle*), desenhada especificamente para penalizar previsões tardias através da minimização do ângulo vetorial entre a previsão e o alvo. Este estudo serve como referência primária para avaliar a eficácia de funções de perda assimétricas.

Explorando a modelagem de dependências complexas, Zeng, Wu e Yiwei (2023) desenvolvem o modelo AGATT. A arquitetura inova ao utilizar canais paralelos com Redes de Atenção em Grafos (*Graph Attention Networks* - GATs) para capturar explicitamente as correlações espaciais (entre sensores) e temporais (entre passos de tempo) antes de fundi-las em um *Transformer*. Este modelo representa o extremo da complexidade arquitetural, servindo para verificar se a adição de contexto operacional simples pode competir com modelagens estruturais complexas.

Por fim, abordando a otimização de hiperparâmetros em arquiteturas híbridas, Zhang *et al.* (2025) propõem o modelo PLO-Transformer-LSTM. Esta abordagem substitui o decodificador padrão do *Transformer* por uma camada LSTM para melhor capturar a sequencialidade temporal e emprega um algoritmo de otimização meta-heurístico, denominado PLO (*Polar Lights Optimizer*), para o ajuste automático de hiperparâmetros. O trabalho destaca-se pela busca de robustez através da otimização algorítmica, oferecendo um contraponto às abordagens focadas em funções de custo ou engenharia de características.

Tabela 1 – Comparação de Abordagens para Predição de RUL

Trabalho	Arquitetura Base	Principal Contribuição / Foco	Função de Perda
Wang <i>et al.</i> (2018)	RNN (Bi-LSTM)	Processamento temporal bidirecional para captura de contexto passado-futuro.	MSE (Simétrica)
Li, Ding e Sun (2018)	CNN (DCNN)	Uso de convoluções profundas para extração automática de características locais.	MSE (Simétrica)
Boujamza e Lissane Elhaq (2022)	RNN (Attn-LSTM)	Introdução de mecanismo de atenção temporal em redes recorrentes.	MSE (Simétrica)
Zeng, Wu e Yiwei (2023)	Híbrida (GAT + <i>Transformer</i>)	Modelagem explícita de dependências espaciais e temporais via Grafos.	MSE (Simétrica)
Kim, Choi e Lim (2024)	<i>Transformer</i>	Técnica de Reponderação Adaptativa (ARR) para lidar com desequilíbrio de dados.	MSE (Ponderado)
Zhou <i>et al.</i> (2024)	Híbrida (CNN + <i>Transformer</i>)	Portão de Fusão de Características (FFG) e minimização do ângulo vetorial.	PVA (Assimétrica)
Zhang <i>et al.</i> (2025)	Híbrida (<i>Transformer</i> + LSTM)	Otimização automática de hiperparâmetros via algoritmo Polar Lights (PLO).	MSE (Simétrica)

Fonte: Elaboração Própria.

A Tabela 1 apresenta um resumo estruturado das abordagens discutidas, destacando a arquitetura base, a principal contribuição de cada estudo e a função de perda utilizada para o treinamento. Isto permite visualizar as lacunas que motivam este trabalho: observa-se que a grande maioria das abordagens foca no aumento da complexidade arquitetural (como a hibridização de modelos) ou em técnicas de tratamento de dados (como reponderação de classes), mas mantém a otimização baseada na perda MSE padrão. Evidencia-se, portanto, a ausência de metodologias que combinem o poder de representação de modelos puramente baseados em atenção com funções de perda orientadas à segurança, apontando para a necessidade de investigar dinâmicas de treinamento avançadas que viabilizem essa otimização de forma estável.

4 PROGFORMER

Conforme evidenciado na revisão da literatura, a grande maioria das pesquisas voltadas para a predição da RUL foca no incremento da complexidade arquitetural, combinando redes convolucionais, recorrentes e mecanismos de atenção, enquanto relega o processo de otimização à tradicional minimização do MSE. Este trabalho argumenta que, em domínios críticos para a segurança como a aviação, o poder preditivo da arquitetura não pode ser desvinculado do perfil de risco operacional.

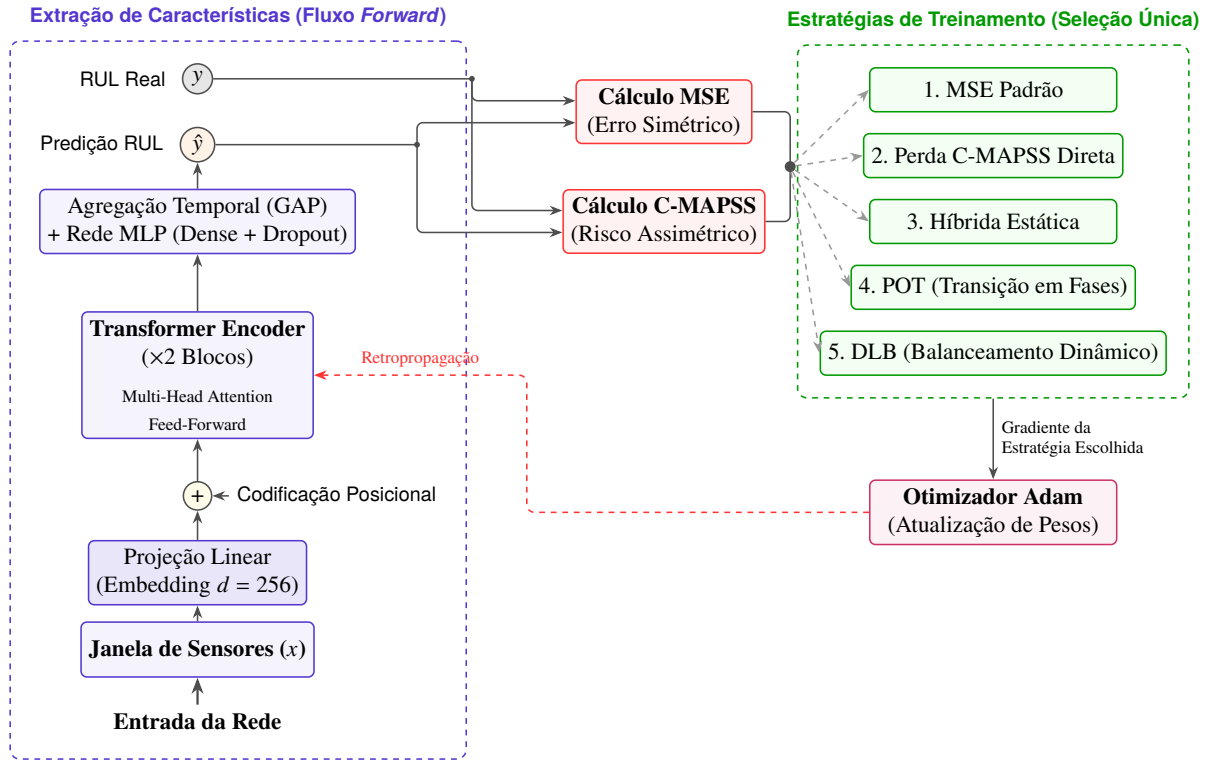
Neste capítulo, apresenta-se o *ProgFormer* (*Prognostics Transformer*), uma abordagem preditiva fundamentada em codificadores baseados em atenção e estratégias de otimização de risco. A concepção desta arquitetura foi motivada pela necessidade de formular uma estrutura que integre restrições de segurança diretamente no processo de aprendizado, superando as limitações de confiabilidade e instabilidade numérica encontradas nos trabalhos relacionados. A presente seção introduz a visão geral do modelo, detalha a escolha da arquitetura base e aprofunda-se nas dinâmicas avançadas de treinamento que constituem o principal diferencial deste trabalho.

4.1 VISÃO GERAL DO *PROGFORMER*

O modelo *ProgFormer* fundamenta-se na premissa de que a estabilidade numérica e a mitigação de riscos operacionais não devem ser tratadas como etapas secundárias, mas sim integradas de forma nativa ao núcleo de aprendizado do modelo. A Figura 4 ilustra a estrutura conceitual do *ProgFormer*, evidenciando sua divisão em dois pilares complementares: a arquitetura de extração de características e a dinâmica de treinamento.

Em vez de propor uma hibridização exaustiva de modelos que aumenta indiscriminadamente o custo computacional e a quantidade de parâmetros, o *ProgFormer* mantém uma arquitetura intencionalmente enxuta. A inovação desloca-se da estrutura da rede para a forma como o modelo é penalizado e atualizado. Essa mudança de paradigma transforma a função de perda de um mero avaliador estático em um componente dinâmico que modula o processo de otimização.

Figura 4 – Visão geral do ProgFormer, evidenciando a arquitetura base e a seleção das estratégias de otimização.



Fonte: Elaboração Própria.

4.2 ARQUITETURA BASE TRANSFORMER ENCODER

O modelo *ProgFormer* é fundamentado na arquitetura *Transformer* desenvolvida por Vaswani *et al.* (2017), adaptada especificamente para a tarefa de regressão em séries temporais. Diferentemente de sua aplicação original em Processamento de Linguagem Natural (que utiliza uma estrutura Codificador-Decodificador), o *ProgFormer* adota exclusivamente a pilha isolada de Codificadores (*Encoder*). Essa escolha justifica-se pois o objetivo do modelo é extrair uma representação latente robusta da sequência de entrada para prever um valor escalar contínuo (a RUL), e não gerar uma nova sequência de dados.

4.2.1 Input Embedding e Positional Encoding

A entrada do modelo consiste em um tensor de dimensão (B, W, F) , onde B é o tamanho do lote (*batch size*), $W = 30$ é o comprimento da janela temporal e F é o número de características (14 ou 24, dependendo do conjunto de dados).

Como o mecanismo de auto-atenção é invariante à ordem, a informação sequencial deve ser injetada explicitamente. Inicialmente, os vetores de características de cada passo de tempo são projetados linearmente para uma dimensão superior $d_{\text{model}} = 256$ através de uma camada densa. Em seguida, adiciona-se a esta projeção uma Codificação Posicional (*Positional Encoding*)

baseada em funções senoidais de diferentes frequências, conforme definido por Vaswani *et al.* (2017) e expresso nas Equações 4.1 e 4.2:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \quad (4.1)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \quad (4.2)$$

Esta abordagem permite que o modelo distinga a ordem temporal dos eventos de degradação sem introduzir parâmetros aprendíveis adicionais nesta etapa.

4.2.2 Encoder Block

O núcleo da arquitetura consiste em um empilhamento de dois blocos codificadores idênticos. Essa quantidade foi selecionada para equilibrar a capacidade de extração de características hierárquicas com a eficiência computacional, mitigando o risco de *overfitting* frequentemente observado em arquiteturas excessivamente profundas aplicadas a conjuntos de dados de porte moderado. Cada bloco é composto por duas subcamadas principais:

- **Mecanismo de Auto-Atenção Múltipla (*Multi-Head Self-Attention*):** Configurado com quatro cabeças de atenção (*heads*). Neste trabalho, definiu-se a dimensão da chave (d_k) igual à dimensão do modelo (256), permitindo que cada cabeça opere em um subespaço de alta dimensionalidade para capturar dependências temporais complexas nos sinais dos sensores.
- **Rede Feed-Forward (FFN):** Uma rede neural totalmente conectada aplicada a cada posição separadamente e de forma idêntica. Ela consiste em duas transformações lineares com uma ativação não-linear (ReLU) entre elas. A dimensão da camada interna da FFN foi definida como 256.

Em torno de cada uma dessas duas subcamadas, aplica-se uma conexão residual seguida por uma Normalização de Camada (*Layer Normalization*). Além disso, utiliza-se regularização *Dropout* com taxa de 0,1 após a atenção e a FFN para prevenção de *overfitting* (VASWANI *et al.*, 2017).

4.2.3 Regression Head

A saída final da pilha de codificadores é uma sequência de vetores codificados de dimensão ($W, 256$). Para realizar a predição da RUL, a saída não é submetida a um achatamento vetorial direto, mas sim a uma operação de Agregação Temporal (*Global Average Pooling* - GAP), que condensa a dimensão temporal e gera um único vetor de características de tamanho 256, resumindo todo o contexto da janela deslizante.

O vetor resultante alimenta um MLP (*Multilayer Perceptron*), composto por:

- Uma camada densa com 128 neurônios e função de ativação ReLU;
- Uma camada de *Dropout* com taxa de 0,1;
- Um neurônio de saída linear, responsável por produzir a estimativa final dos ciclos restantes antes da falha crítica.

4.3 DINÂMICAS DE TREINAMENTO

A verdadeira vantagem do *ProgFormer* reside em sua abordagem em relação à função de custo. Como discutido anteriormente, o MSE simétrico penaliza previsões otimistas e pessimistas com pesos idênticos, uma premissa fatal na indústria aeronáutica, onde a superestimação da vida útil custa muito mais caro que sua subestimação (SAXENA *et al.*, 2008).

Em contrapartida, a otimização direta de funções de pontuação exponenciais assimétricas provoca a explosão de gradientes e o colapso irreversível da rede, um fenômeno de instabilidade numérica documentado na literatura de aprendizado profundo (GOODFELLOW; BENGIO; COURVILLE, 2016), manifestando-se drasticamente em cenários com múltiplos regimes operacionais devido à severidade desproporcional das penalidades iniciais geradas pela função de risco do C-MAPSS.

Para quebrar esse impasse metodológico, o *ProgFormer* disponibiliza um conjunto de cinco estratégias distintas de otimização. Durante o processo de treinamento, realiza-se a seleção única de apenas uma dessas abordagens, buscando identificar a configuração que melhor garanta a estabilidade da convergência sem sacrificar a sensibilidade ao risco.

Para fins de clareza metodológica, é fundamental destacar que a arquitetura do *ProgFormer* não opera como uma *pipeline* sequencial por onde os dados transitam obrigatoriamente através de todas as funções de perda ou estratégias de otimização durante um mesmo ciclo de aprendizado. Pelo contrário, o *design* da solução foi concebido em torno de uma arquitetura base comum que é submetida a configurações de treinamento independentes (aqui denominadas variantes). Cada variante é treinada de forma isolada do início ao fim: seja utilizando exclusivamente a perda simétrica padrão (MSE), empregando apenas a função assimétrica orientada a risco (Perda C-MAPSS) ou integrando as estratégias avançadas de transição (POT) e balanceamento (DLB). Esta abordagem de avaliação em configurações separadas foi estruturada para isolar o impacto de cada mecanismo, permitindo uma comparação empírica clara e justa sobre qual estratégia confere, de fato, a maior segurança operacional ao modelo.

4.3.1 *Phased Objective Transition*

Inspirado pelos fundamentos do Aprendizado Curricular (BENGIO *et al.*, 2009), o POT estrutura um "currículo de otimização" no qual a função objetivo da rede evolui ao longo do tempo de treinamento.

A premissa desta estratégia é não expor o modelo à complexidade e aos gradientes abruptos da função de risco assimétrica de forma prematura. Durante o estágio inicial de treinamento,

o *ProgFormer* é otimizado exclusivamente pela perda simétrica (MSE), assegurando que os gradientes fluam de forma controlada e que a rede mapeie a variedade geral de degradação. Uma vez que a arquitetura atinge uma estabilidade preditiva satisfatória, o modelo inicia uma transição sistemática para a perda assimétrica. Essa transição permite que a rede ajuste finamente seus parâmetros para priorizar a margem de segurança operacional, mitigando a instabilidade numérica associada às grandes predições errôneas (*outliers*) do início do treinamento.

4.3.2 *Dynamic Loss Balancing*

Embora o agendamento por fases do POT seja uma solução metodológica rigorosa, ele depende de transições previamente agendadas que podem não ser ideais para todos os limiares de degradação. Para contornar esta restrição, o *ProgFormer* oferece um segundo mecanismo de otimização: o DLB autônomo.

Na estratégia DLB, em vez de transitar estaticamente entre as métricas, o modelo otimiza continuamente uma função de perda híbrida, a soma ponderada do MSE e da perda assimétrica. A inovação, baseada na literatura de aprendizado multi-objetivo (CHEN *et al.*, 2017), reside na capacidade do algoritmo de atualizar seus próprios pesos dinamicamente ao final de cada época.

Através do monitoramento contínuo das taxas de evolução de ambas as funções usando EMA (KINGMA; BA, 2014), o *ProgFormer* identifica qual componente da perda está apresentando maior dificuldade de convergência e realoca o peso do gradiente para favorecê-la. Caso a perda assimétrica cause picos agudos de gradiente, o DLB reduz autonomamente a sua influência e amplia o peso da estabilidade do MSE. Através desta estratégia, o *ProgFormer* atua como um sistema que se auto-regula, assegurando a confiabilidade preditiva exigida pelo ambiente industrial sob regimes de voo complexos.

Tabela 2 – Comparação de Abordagens para Predição de RUL incluindo o modelo proposto

Trabalho	Arquitetura Base	Principal Contribuição / Foco	Função de Perda
Wang <i>et al.</i> (2018)	RNN (Bi-LSTM)	Processamento temporal bidirecional para captura de contexto passado-futuro.	MSE (Simétrica)
Li, Ding e Sun (2018)	CNN (DCNN)	Uso de convoluções profundas para extração automática de características locais.	MSE (Simétrica)
Boujamza e Lissane Elhaq (2022)	RNN (Attn-LSTM)	Introdução de mecanismo de atenção temporal em redes recorrentes.	MSE (Simétrica)
Zeng, Wu e Yiwei (2023)	Híbrida (GAT + <i>Transformer</i>)	Modelagem explícita de dependências espaciais e temporais via Grafos.	MSE (Simétrica)
Kim, Choi e Lim (2024)	<i>Transformer</i>	Técnica de Reponderação Adaptativa (ARR) para lidar com desequilíbrio de dados.	MSE (Ponderado)
Zhou <i>et al.</i> (2024)	Híbrida (CNN + <i>Transformer</i>)	Portão de Fusão de Características (FFG) e minimização do ângulo vetorial.	PVA (Assimétrica)
Zhang <i>et al.</i> (2025)	Híbrida (<i>Transformer</i> + LSTM)	Otimização automática de hiperparâmetros via algoritmo Polar Lights (PLO).	MSE (Simétrica)
ProgFormer (Este trabalho)	<i>Transformer Encoder</i>	Gerenciamento dinâmico da superfície de perda e mitigação ativa do risco operacional.	Adaptativa / Evolutiva (POT / DLB)

Fonte: Elaboração Própria.

A Tabela 2 sintetiza o posicionamento teórico do *ProgFormer* frente aos modelos avaliados. Fica evidente que, enquanto a maioria das pesquisas contemporâneas busca o aprimoramento preditivo por meio do aumento da complexidade estrutural (hibridização de arquiteturas) e mantém a otimização atrelada a perdas estáticas (predominantemente o MSE), este trabalho consolida um caminho metodológico inverso. Ao adotar uma arquitetura extratora intencionalmente enxuta, baseada puramente no *Transformer Encoder*, a inovação desloca-se inteiramente para a governança da função de custo. Essa quebra de paradigma estabelece que a mitigação ativa do risco operacional, gerida pelas estratégias adaptativas POT e DLB, atue como o núcleo do aprendizado da rede, fundamentando as validações experimentais que serão detalhadas no capítulo subsequente.

5 METODOLOGIA

Este capítulo descreve a abordagem experimental desenvolvida para a validação do modelo *ProgFormer* na predição da Vida Útil Remanescente em motores turbofan. A metodologia está estruturada em cinco etapas principais: preparação e pré-processamento dos dados, definição da arquitetura de extração, formalização das estratégias avançadas de dinâmica de treinamento (POT e DLB), configuração experimental e apresentação dos resultados obtidos nos diferentes regimes operacionais do conjunto C-MAPSS.

Para isolar o impacto das estratégias de treinamento propostas e estabelecer um comparativo rigoroso, o modelo *Transformer* otimizado puramente com a função de MSE é formalmente estabelecido como o *baseline* desta pesquisa. A escolha do MSE reflete o padrão metodológico predominante na literatura para a avaliação de modelos de prognóstico baseados em *Deep Learning* (WANG *et al.*, 2018; LI; DING; SUN, 2018). Desta forma, o desenho experimental garante que a comparação não foca na busca por arquiteturas mais simples ou mais complexas, mas sim em comprovar que a alteração da função de perda (de uma métrica simétrica para uma métrica orientada a risco) é o fator determinante para a segurança operacional.

5.1 PREPARAÇÃO DOS DADOS E PRÉ-PROCESSAMENTO

A qualidade dos dados de entrada é determinante para o desempenho de modelos de *Deep Learning*. O *pipeline* de pré-processamento foi projetado para tratar as especificidades das séries temporais do C-MAPSS, abordando a variabilidade de escalas, a presença de ruído e a definição do alvo de predição.

5.1.1 Organização do Conjunto de Dados C-MAPSS

O conjunto de dados C-MAPSS é fornecido em sua origem já particionado em conjuntos de treinamento e teste. Os arquivos de treinamento contêm trajetórias completas de degradação, permitindo que o modelo aprenda o comportamento do motor desde o estado saudável até a falha. Os arquivos de teste, por outro lado, contêm trajetórias censuradas que terminam em um ponto arbitrário antes da falha. Esse conjunto de dados é estruturado em quatro subconjuntos distintos, a saber, FD001, FD002, FD003 e FD004, com complexidades operacionais distintas. Cada amostra do conjunto é representada por uma linha contendo 26 colunas: o identificador da unidade, o índice temporal em ciclos, 3 valores de configurações operacionais e 21 leituras de sensores físicos. O objetivo do modelo é prever a RUL especificamente para o último ciclo disponível em cada sequência de teste.

5.1.2 Seleção de Atributos

Este trabalho adota estratégias de seleção de atributos distintas, baseadas na complexidade do regime operacional de cada subconjunto, visando evitar ruídos indesejados e focar na

informação útil.

Para os conjuntos de dados FD001 e FD003, que operam sob uma única condição (Nível do Mar), as três colunas de configurações operacionais e sete sensores específicos ($s_1, s_5, s_6, s_{10}, s_{16}, s_{18}, s_{19}$) apresentam variância nula ou desprezível. Conforme prática consolidada na literatura, a inclusão dessas variáveis constantes não agrega informação preditiva e pode introduzir viés numérico na rede (ZHANG *et al.*, 2025). Portanto, para estes casos, utilizou-se uma seleção fixa das 14 características com desvio padrão não-nulo.

Para os conjuntos de dados FD002 e FD004, que operam sob seis condições de voo distintas (variando altitude, número de Mach e posição do acelerador), a abordagem foi alterada para uma seleção dinâmica. Nestes cenários, os sensores que antes eram constantes passam a variar em resposta às mudanças de voo. Além disso, as configurações operacionais tornam-se vitais para contextualizar as leituras dos sensores. Portanto, a etapa de pré-processamento foi configurada para reter automaticamente todas as colunas que apresentassem desvio padrão superior a zero ($\sigma > 0$), resultando em um vetor de entrada com 24 características (21 sensores e 3 configurações operacionais).

5.1.3 Normalização e Escalonamento dos Dados

Dada a heterogeneidade das grandezas físicas medidas pelos sensores, a normalização é mandatória para garantir a convergência do otimizador. Utilizou-se o método Min-Max Scaling para projetar todas as características no intervalo $[0, 1]$.

Para evitar o vazamento de dados, os parâmetros do escalonador (valores mínimos e máximos) foram calculados exclusivamente com base no conjunto de treinamento. Esses mesmos parâmetros foram então aplicados para transformar o conjunto de teste, simulando um cenário realista onde o modelo não tem acesso a estatísticas futuras.

5.1.4 Definição da Variável Alvo

A degradação de motores aeronáuticos não é linear durante toda a sua vida útil, ela tende a ser negligenciável no início e acelerada nos estágios finais. Treinar um modelo para prever uma RUL linearmente decrescente durante a fase saudável, onde os sensores permanecem estáveis, pode induzir o modelo ao erro e ao *overfitting*.

Para mitigar isso, adotou-se a abordagem de RUL Linear por Partes, amplamente aceita na literatura desde o trabalho seminal de Heimes (2008). Definiu-se um limiar de saturação de 125 ciclos. Dessa forma, qualquer RUL real superior a 125 foi truncada para 125, instruindo o modelo a tratar qualquer motor com mais de 125 ciclos restantes como saudável.

5.1.5 Geração de Sequências

Como o núcleo da arquitetura do *ProgFormer* é dependente de informações contextuais (através do mecanismo de auto-atenção) para processar séries temporais multivariadas, os dados contínuos foram reestruturados em matrizes utilizando janelas deslizantes (*sliding windows*).

Definiu-se um comprimento de janela de $W = 30$ ciclos. Esta escolha alinha-se às melhores práticas reportadas para o *benchmark* C-MAPSS, oferecendo um balanço ótimo: é larga o suficiente para incorporar a trajetória histórica da falha sem impor custo computacional excessivo ou ultrapassar o tamanho mínimo das séries de teste mais curtas fornecidas (KIM; CHOI; LIM, 2024).

- **No Treinamento:** a janela desliza com passo de 1 ciclo sobre toda a vida útil dos motores, gerando o máximo de amostras de treinamento possível.
- **No Teste:** gera-se apenas uma sequência por motor, correspondente aos últimos W ciclos registrados, para realizar a predição pontual exigida pelo *benchmark*. Se uma trajetória de teste possuir comprimento menor que W , aplica-se preenchimento (*padding*) com zeros no início da sequência para garantir que todas as entradas possuam a mesma dimensionalidade, uniformizando a janela temporal e a quantidade de características extraídas para cada motor avaliado.

5.2 TREINAMENTO E OTIMIZAÇÃO

A eficácia de uma rede neural profunda para predição de RUL depende não apenas de sua capacidade de representação arquitetural, mas, fundamentalmente, da definição da função objetivo e da estratégia de otimização empregada. Este trabalho investiga uma progressão metodológica de funções de custo, partindo de funções objetivo até os mecanismos dinâmicos propostos para o *ProgFormer*.

5.2.1 Funções Objetivo

Para o baseline e para a fase de estabilização inicial, utilizou-se o MSE, cuja definição formal foi apresentada anteriormente na Equação 2.2. Sendo uma métrica simétrica e convexa, o MSE favorece uma convergência estável e rápida, penalizando grandes erros de forma quadrática, independentemente da direção (adiantado ou atrasado).

Para a abordagem proposta focada em segurança, implementou-se uma Função de Perda C-MAPSS. A função de pontuação original do desafio PHM08 não é diretamente adequada para o treinamento via gradiente descendente devido à sua formulação original. Para torná-la uma função de perda viável, adaptou-se a fórmula para operar como uma média contínua sobre o lote, garantindo a diferenciabilidade em todos os pontos. A perda L_{Safety} é definida na Equação 5.1:

$$L_{Safety} = \frac{1}{N} \sum_{i=1}^N \left(\exp \left(\frac{|d_i|}{k_i} \right) - 1 \right) \quad (5.1)$$

Onde $d_i = \hat{y}_i - y_i$ é o erro de predição, e o divisor k_i é dinâmico: $k_i = 13$ se $d_i < 0$ (erro adelantado) e $k_i = 10$ se $d_i \geq 0$ (erro tardio). A implementação computacional desta função utilizou operações primitivas da biblioteca TensorFlow (`tf.where`, `tf.abs`, `tf.exp`). Isso permitiu a

utilização do mecanismo de Diferenciação Automática (*Automatic Differentiation* - AutoDiff), que constrói o grafo computacional das operações e calcula os gradientes exatos ($\nabla_{\theta} L_{Safety}$) através da regra da cadeia durante a retropropagação. O uso do AutoDiff elimina a necessidade de derivação analítica manual e assegura que os gradientes da penalidade exponencial sejam corretamente propagados para a atualização dos pesos da rede (GOODFELLOW; BENGIO; COURVILLE, 2016).

5.2.2 Modelo Híbrido Estático

Como primeira tentativa de mitigar o colapso numérico causado pela otimização direta da função exponencial, define-se um objetivo híbrido que busca aliar a estabilidade matemática do MSE à consciência de risco do C-MAPSS. A função híbrida estática é expressa como uma soma ponderada na Equação 5.2:

$$L_{Hybrid} = \lambda_{MSE} \cdot L_{MSE} + \lambda_{Safety} \cdot L_{Safety} \quad (5.2)$$

Onde λ_{MSE} e λ_{Safety} são coeficientes escalares fixos. Na formulação estática, ambos os coeficientes são inicializados como $\lambda_{MSE} = \lambda_{Safety} = 1$. Embora esta abordagem forneça uma melhoria em relação à otimização assimétrica pura, a diferença de escala e a divergência de estabilidade entre os dois componentes de perda revelam as limitações de pesos estáticos em cenários de alta variância, motivando o desenvolvimento dos mecanismos POT e DLB pelo *ProgFormer*.

5.2.3 Transição de Objetivo em Fases

O POT estrutura um currículo de treinamento, evitando expor a rede a gradientes exponenciais severos antes que os parâmetros da arquitetura convirjam para uma representação estável do motor. O estudo propõe duas abordagens para esta transição:

1. **Binary POT:** Consiste em uma mudança de fase discreta. Durante as primeiras 50 épocas (Fase de Estabilização), o modelo é treinado exclusivamente com a perda simétrica (L_{MSE}). A partir da época 51 (Fase de Refinamento), o objetivo é alterado abrupta e integralmente para a perda assimétrica (L_{Safety}). Esta abordagem previne a explosão inicial de gradientes, estabilizando o aprendizado em cenários complexos de forma binária.
2. **Hybrid POT:** Em vez de um corte abrupto, esta abordagem utiliza um agendamento estruturado dos coeficientes λ_{MSE} e λ_{Safety} da função híbrida (Equação 5.2) ao longo de 100 épocas. A cada 20 épocas, o peso do MSE λ_{MSE} decresce linearmente em passos de 0,25 ($1.0 \rightarrow 0.75 \rightarrow 0.50 \rightarrow 0.25 \rightarrow 0.0$), enquanto o peso do C-MAPSS (λ_{Safety}) cresce proporcionalmente ($0.0 \rightarrow 0.25 \rightarrow 0.50 \rightarrow 0.75 \rightarrow 1.0$). Este currículo linear suave desloca de forma progressiva o foco do *ProgFormer* de uma otimização puramente estatística (MSE) para uma otimização rigorosamente voltada ao risco operacional.

5.2.4 Balanceamento Dinâmico de Perdas

Embora o POT seja eficaz, este baseia-se num agendamento rígido predefinido. Para dotar o modelo de autonomia na gestão do equilíbrio entre estabilidade e risco, implementou-se o DLB. Em vez de coeficientes estáticos ou agendados, o DLB atualiza dinamicamente os pesos de ponderação, λ_{MSE} e λ_{Safety} , em tempo real, baseando-se na dificuldade de convergência de cada função (CHEN *et al.*, 2017).

Ao final de cada época t , os valores atuais de ambas as funções de perda são extraídos. Para evitar reações instáveis a flutuações pontuais dos lotes, o DLB rastreia o histórico dessas perdas utilizando uma EMA, com fator de suavização $\alpha = 0.7$, conforme a Equação 5.3:

$$\text{EMA}_t = \alpha \cdot \text{EMA}_{t-1} + (1 - \alpha) \cdot L_t \quad (5.3)$$

Para tratar a severa diferença de magnitude entre as métricas, cada EMA_t é normalizada em relação ao seu valor inicial da primeira época, produzindo as escalas $\tilde{L}_t^{\text{mse}}$ e $\tilde{L}_t^{\text{safety}}$. Com essas estimativas normalizadas, a importância relativa (pesos alvo λ^*) de cada objetivo é calculada de forma inversamente proporcional ao seu progresso, conforme demonstrado na Equação 5.4:

$$\lambda_{\text{MSE}}^* = \frac{\tilde{L}_t^{\text{safety}}}{\tilde{L}_t^{\text{mse}} + \tilde{L}_t^{\text{safety}}}, \quad \lambda_{\text{Safety}}^* = \frac{\tilde{L}_t^{\text{mse}}}{\tilde{L}_t^{\text{mse}} + \tilde{L}_t^{\text{safety}}} \quad (5.4)$$

Esta formulação força o sistema a atribuir mais peso à função de perda que apresentar a menor taxa relativa de melhoria naquele momento. Para garantir a suavidade da transição e evitar colapsos por oscilação de pesos, os novos pesos não são aplicados diretamente. Em vez disso, utiliza-se uma regra de atualização por *momentum* (fator $\beta = 0.3$), descrita na Equação 5.5:

$$\lambda_t = (1 - \beta) \cdot \lambda_{t-1} + \beta \cdot \lambda_t^* \quad (5.5)$$

Adicionalmente, os pesos finais são restritos ao intervalo $[0.2, 0.8]$ para assegurar que nenhum dos objetivos seja completamente ignorado durante a otimização.

5.3 PROTOCOLO DE AVALIAÇÃO

A validação das estratégias de treinamento do *ProgFormer* foi conduzida sob um protocolo experimental rigoroso, desenhado para quantificar tanto a precisão das estimativas quanto a segurança operacional das predições em cenários de falha crítica.

5.3.1 Métricas de Avaliação

Para avaliar a qualidade das predições de RUL, foram utilizadas as duas métricas padrão estabelecidas na literatura do conjunto de dados C-MAPSS. A primeira métrica é o RMSE, que avalia a precisão global do modelo penalizando grandes desvios entre o valor predito ($\hat{y}$) e o valor real (y). O RMSE foi definido formalmente na Equação 2.5. Embora o RMSE seja uma medida estatística robusta, ele é simétrico, tratando erros adiantados (conservadores) e atrasados (perigosos) com a mesma severidade. Para endereçar a natureza crítica da manutenção

aeronáutica, utiliza-se a Pontuação C-MAPSS (*Score*), uma métrica assimétrica que penaliza predições tardias exponencialmente mais do que predições adiantadas. O *Score* foi definido formalmente na Equação 2.6. O objetivo da otimização é minimizar ambas as métricas, com ênfase na redução do *Score* para garantir a segurança.

5.3.2 Configuração dos Experimentos

Todos os experimentos numéricos e avaliações arquiteturais foram implementados em linguagem Python (versão 3.12.3), utilizando o *framework* de aprendizado profundo TensorFlow/Keras (versão 2.13.0). O ambiente de *hardware* foi suportado por um processador AMD Ryzen 5 3600 (6 núcleos), alocado com 16GB de memória RAM, e aceleração computacional garantida por uma unidade gráfica NVIDIA GeForce GTX 1650, equipada com 4GB de memória dedicada.

O otimizador global eleito para todas as variantes do *ProgFormer* foi o Adam, mantendo suas taxas de momento em parâmetros padronizados ($\beta_1 = 0.9$ e $\beta_2 = 0.999$). Visando gerenciar a volatilidade inerente aos componentes exponenciais de perda, a taxa de aprendizado (*learning rate*) foi calibrada em $\alpha = 10^{-4}$ durante as fases otimizadas por MSE, sofrendo uma redução para $\alpha = 10^{-5}$ nas fases submetidas majoritariamente aos objetivos assimétricos. O tamanho do lote de treinamento foi estipulado em 64 amostras para cenários de condição operacional única (FD001/FD003) e expandido para 256 amostras nas bases de múltipla condição operacional (FD002/FD004).

Por fim, todos os modelos foram submetidos a um limite máximo estipulado de 100 épocas de treinamento. O monitoramento do aprendizado foi realizado através de uma subamostra de validação, correspondente a 20% dos dados de treino. Para as abordagens baseadas em otimização estática, adotou-se o protocolo de Parada Antecipada (*Early Stopping*) para evitar *overfitting*. Nestes casos, se a função de perda não manifestasse declínio após 10 épocas ininterruptas, a simulação era suspensa e os tensores de peso restaurados para sua configuração de desempenho ótimo registrada. Em contrapartida, para as estratégias em que a função de perda é alterada dinamicamente ao decorrer do treinamento (*Hybrid POT* e DLB), o mecanismo de *Early Stopping* não foi aplicado. Essa restrição metodológica justifica-se pelo fato de que a transição e a realocação contínua dos pesos da função de custo geram oscilações naturais e esperadas na curva de perda de validação, o que poderia acionar o gatilho de parada de forma incorreta e prematura.

5.3.3 Uso de Inteligência Artificial Generativa

Recentemente, diretrizes regulatórias foram estabelecidas no Brasil para orientar a aplicação de ferramentas de Inteligência Artificial Generativa (IAG) no ambiente acadêmico e científico. De acordo com a Política de Integridade na Atividade Científica, instituída pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (2026), a transparência quanto ao emprego de IAG tornou-se um requisito obrigatório em todas as fases da pesquisa. Essa exigência

é corroborada internamente pelas normativas da Universidade Federal de Santa Catarina (2025). Para atender a essas determinações éticas e institucionais, esta subseção detalha o escopo de utilização do modelo *Gemini 3.1 Pro* (GOOGLE, 2026) e *Claude Opus 4.8* (ANTHROPIC, 2026) no desenvolvimento desta monografia.

No que tange à elaboração do texto, a plataforma foi empregada como um assistente de revisão gramatical e aprimoramento de coesão textual. Seu uso incluiu sugestões de reescrita para otimizar a clareza de trechos específicos e na adequação de formatações do código em $\text{\LaTeX}$. Ressalta-se que a concepção original da pesquisa, o desenvolvimento das estratégias de otimização, a construção dos argumentos e a interpretação crítica dos resultados são de autoria exclusiva do autor.

Em relação ao desenvolvimento computacional, a arquitetura algorítmica do *ProgFormer* e a lógica de programação subjacente foram concebidas e estruturadas integralmente pelo autor. A inteligência artificial atuou estritamente como uma ferramenta de suporte para a depuração de erros de compilação (*debugging*) e para o esclarecimento de detalhes e comandos específicos de bibliotecas de *Deep Learning*, como *TensorFlow/Keras*, *NumPy* e *Pandas*.

É importante destacar que a IAG não foi responsável por processar os dados brutos da base C-MAPSS, gerar as previsões de RUL ou a análise comparativa dos modelos. Em estrito cumprimento à Conselho Nacional de Desenvolvimento Científico e Tecnológico (2026), o autor atesta que revisou todo o material apresentado, assumindo responsabilidade sobre a integridade do documento final.

6 RESULTADOS

Este capítulo apresenta a avaliação experimental das estratégias de treinamento desenvolvidas para o modelo *ProgFormer*. Para responder à complexidade do problema de forma clara, a discussão dos resultados é conduzida de maneira progressiva.

A análise inicia-se pelo estabelecimento de um *baseline* estatístico através do desempenho do modelo base MSE. Em seguida, investigam-se as instabilidades numéricas causadas pela Perda C-MAPSS direta e os resultados iniciais da Função Híbrida Estática. Posteriormente, o estudo detalha a eficácia das estratégias adaptativas propostas para recuperar a convergência e garantir a segurança preditiva, avaliando a POT e o DLB. Por fim, a avaliação culmina em uma análise comparativa global de todas as variantes desenvolvidas frente aos trabalhos de referência da literatura.

6.0.1 Desempenho do Modelo Base MSE

Para estabelecer um referencial de comparação, a arquitetura *Transformer Encoder* foi inicialmente submetida ao treinamento clássico da literatura, utilizando exclusivamente a minimização do MSE. Por ser uma função matemática convexa e simétrica, o MSE garante uma superfície de erro com gradientes comportados, facilitando a convergência do otimizador algorítmico.

A Tabela 3 sintetiza as métricas de avaliação obtidas por esta abordagem fundamental nos quatro subconjuntos de dados.

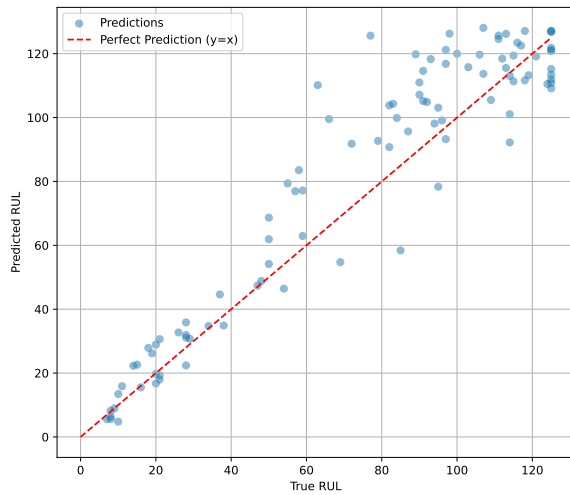
Tabela 3 – Desempenho preditivo do modelo *baseline* treinado com Erro Quadrático Médio (MSE).

Dataset	RMSE	C-MAPSS Score
FD001	14,55	517,26
FD002	18,05	2.243,00
FD003	13,90	1.256,09
FD004	23,35	5.726,00

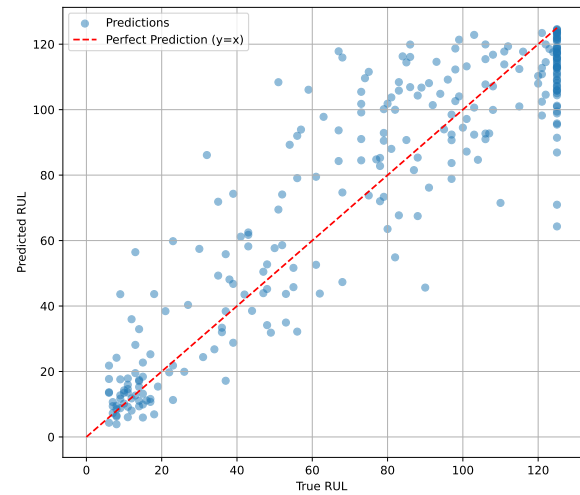
Fonte: Elaboração Própria.

Conforme evidenciado pelos dados, o modelo demonstrou alta capacidade de extração de características temporais, atingindo valores de RMSE excepcionalmente competitivos em todos os níveis de complexidade. Notavelmente, no conjunto multivariado FD002, o modelo atingiu um erro de apenas 18,05, indicando que o mecanismo de atenção múltipla foi capaz de mapear a degradação subjacente mesmo em meio ao ruído de seis condições de voo distintas. No entanto, a análise do risco operacional quantificado pelo *Score* revela a falha inerente a este paradigma de treinamento estatístico.

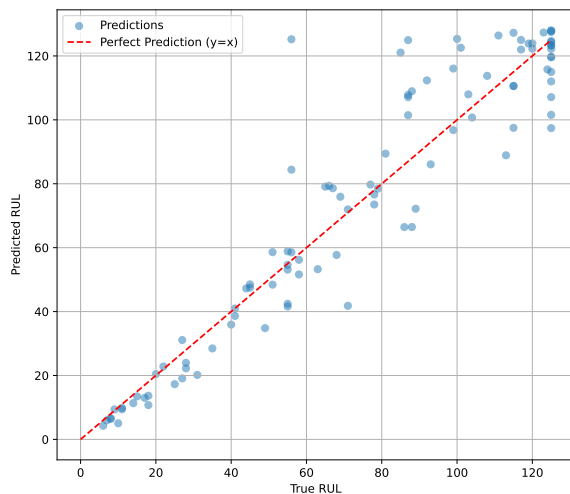
Figura 5 – Dispersão das previsões de RUL utilizando MSE Padrão para os quatro conjuntos de dados C-MAPSS.



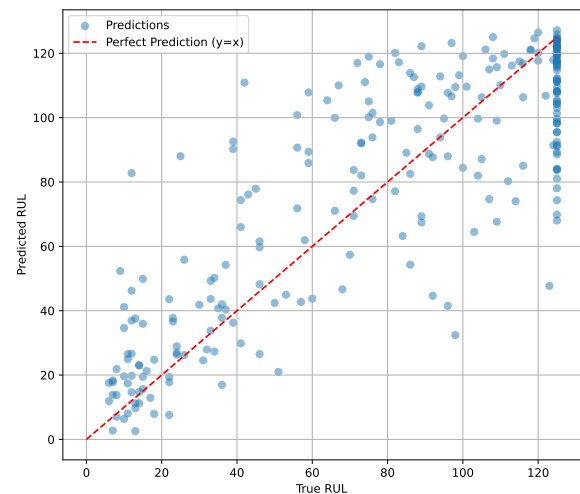
(a) FD001



(b) FD002



(c) FD003



(d) FD004

Fonte: Elaboração Própria.

Em regimes de condição única (FD001 e FD003), a ausência de penalidade direcional permitiu que a rede gerasse distribuições estatisticamente simétricas em torno da diagonal ideal ($y = x$), conforme ilustrado nos quadrantes (a) e (c) da Figura 5. Essa simetria resulta em uma quantidade severa de pontos localizados acima da linha de referência (superestimação). No contexto aeronáutico, prever que o motor suportará mais ciclos do que a realidade permite é classificado como um erro tardio crítico. Este comportamento resultou em *Scores* punitivos de 517,26 para o conjunto FD001 e 1.256,09 para o conjunto FD003.

O cenário de risco exacerba-se nos regimes de condição múltipla (FD002 e FD004). Devido à alta variância induzida pelas trocas de altitude e velocidade, o modelo treinado com MSE tende a suavizar a incerteza prevendo valores médios. No quadrante (d) (FD004, com dois modos de

falha sob seis condições de voo), observa-se que essa tendência à média exacerbou as predições otimistas na faixa de fim de vida ($RUL < 50$), resultando em um *Score* alarmante de 5.726,00.

Esses resultados corroboram de forma empírica a premissa que motiva esta pesquisa: uma alta precisão estatística global (baixo RMSE) alcançada via otimização simétrica não se traduz em confiabilidade operacional. A arquitetura aprende a evolução da degradação de forma geral, mas seu foco no erro absoluto a torna negligente ao risco de falhas catastróficas.

6.0.2 Perda C-MAPSS

Constatada a ineficiência da função simétrica em garantir a confiabilidade operacional, a segunda etapa da avaliação experimental investigou o impacto do treinamento guiado exclusivamente pela Função de Perda C-MAPSS. O objetivo primário desta abordagem foi forçar a rede a internalizar o risco, penalizando exponencialmente os erros de superestimação da vida útil desde a primeira época de treinamento. A Tabela 4 expõe os resultados quantitativos obtidos por essa estratégia.

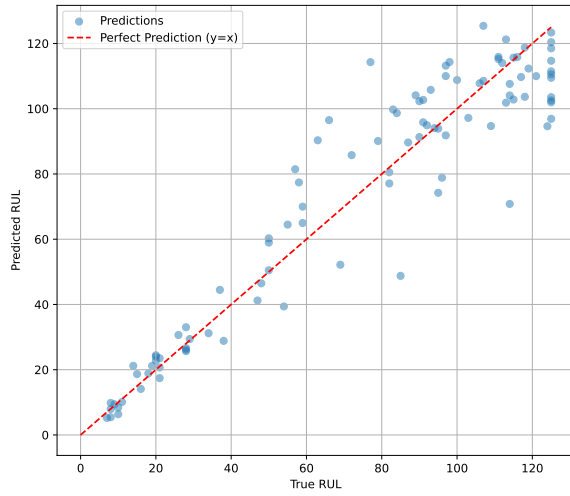
Tabela 4 – Desempenho preditivo do modelo otimizado de forma direta pela Função de Perda C-MAPSS.

Dataset	RMSE	C-MAPSS Score
FD001	13,09	259,87
FD002	87,75	1.409.215,00
FD003	12,46	293,25
FD004	44,74	18.434,00

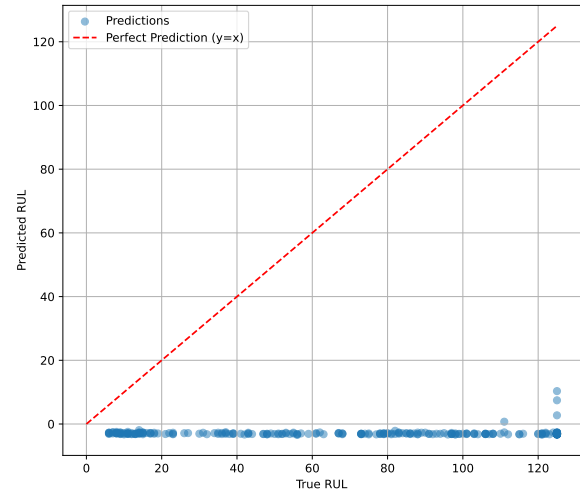
Fonte: Elaboração Própria.

A análise revela uma dicotomia extrema no comportamento do modelo, evidenciando um severo limite de aplicabilidade para funções exponenciais puras. O impacto da técnica variou drasticamente em função da complexidade operacional do conjunto de dados.

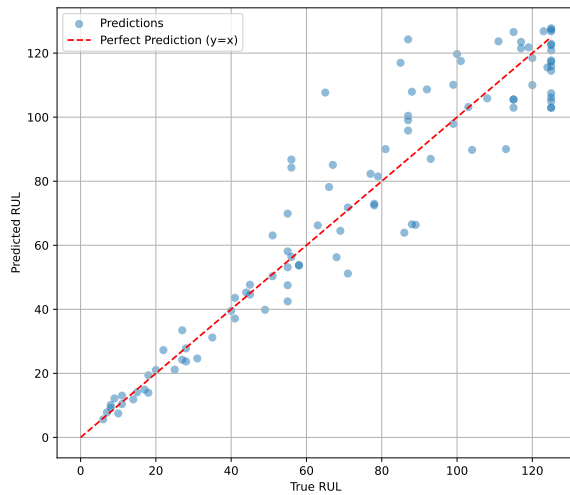
Figura 6 – Dispersão das previsões de RUL utilizando otimização Assimétrica Direta (Perda C-MAPSS).



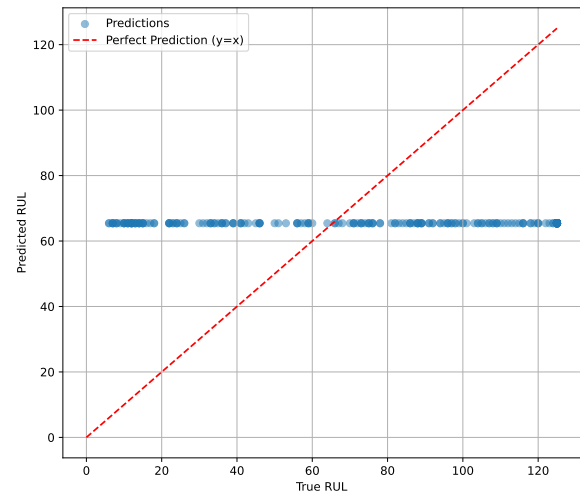
(a) FD001



(b) FD002



(c) FD003



(d) FD004

Fonte: Elaboração Própria.

Nos cenários que operam sob uma única condição de voo e apresentam comportamento sensorial altamente estável (FD001 e FD003), a otimização direta pelo risco obteve resultados satisfatórios. Os quadrantes (a) e (c) da Figura 6 ilustram que o gradiente da perda exponencial forçou a maioria das previsões a se alocarem próximo da linha da diagonal perfeita, expurgando as falhas tardias encontradas no modelo MSE. Este deslocamento conservador reduziu o *Score* de forma maciça: uma melhoria de 49,7% no FD001 e 76,6% no FD003 em relação ao *baseline*. Além de mitigar o risco, as métricas de precisão global (RMSE) também registraram queda (13,09 e 12,46, respectivamente), demonstrando que a penalização severa operou adicionalmente como um fator de regularização e redução de variância na arquitetura da rede.

Contudo, ao submeter o modelo aos regimes complexos e altamente ruidosos (FD002 e

FD004), a rede neural sofreu um grave colapso de aprendizado. A severidade dos gradientes exponenciais gerada nos primeiros ciclos de treinamento, momento em que o modelo opera com pesos aleatórios e ainda não mapeou as seis condições diferentes de voo, desestabilizou o otimizador.

O quadrante (b) exibe o colapso numérico sob o conjunto FD002. Para evitar a penalidade de superestimar a vida útil de motores imersos em ruído, a rede convergiu para um mínimo local, prevendo valores próximos a zero ($RUL \approx 0$) para praticamente todas as unidades. Esse comportamento hiperdefensivo tornou o modelo inutilizável (RMSE de 87,75) e explodiu o *Score* preditivo para níveis acima de 1,4 milhão devido aos severos erros acumulados.

O colapso manifestou-se de forma distinta no conjunto FD004 (quadrante d). Incapaz de processar simultaneamente múltiplos modos de falha e variações de regime sob uma punição assimétrica severa, a rede entrou em estagnação completa, passando a emitir uma predição média constante (RUL em torno de 65 ciclos) que formou uma linha horizontal no gráfico de dispersão, atingindo um *Score* catastrófico de 18.434,00.

Estes eventos demonstram que, embora os objetivos focados em segurança sejam insubstituíveis na indústria, a sua implementação crua e não mediada em arquiteturas de *Deep Learning* é insustentável perante a volatilidade de um cenário real de operação.

6.0.3 Função Híbrida Estática

Como primeira técnica para tentar mitigar o colapso numérico causado pela otimização assimétrica pura, investigou-se o treinamento da rede utilizando a Função Híbrida Estática definida na Equação 5.2. O intuito era que o componente MSE ancorasse as predições e proovesse estabilidade inicial, enquanto o componente C-MAPSS puxasse os resultados finais para a zona de segurança. A Tabela 5 apresenta o desempenho quantitativo do modelo sob esta abordagem.

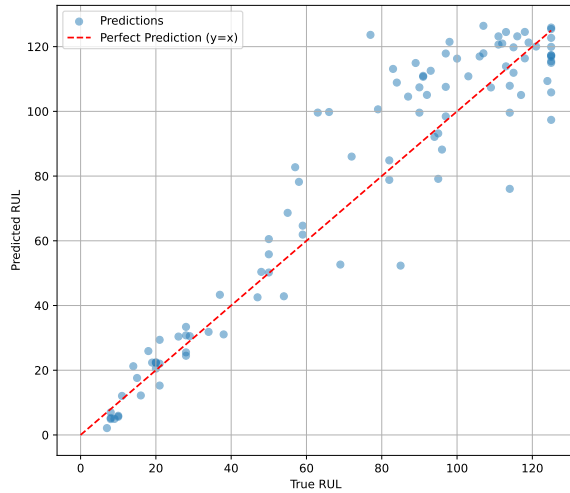
Tabela 5 – Desempenho preditivo do modelo otimizado pela Função Híbrida Estática (MSE + C-MAPSS).

Dataset	RMSE	C-MAPSS Score
FD001	13,99	399,85
FD002	80,84	841.619,00
FD003	12,42	298,12
FD004	80,39	679.277,00

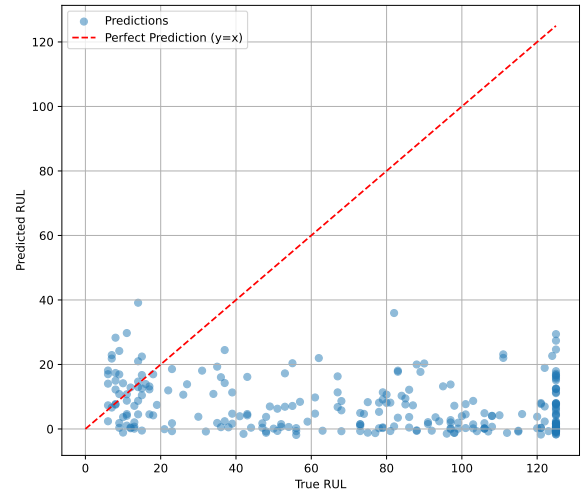
Fonte: Elaboração Própria.

A análise inicial das métricas sugere uma falha nos regimes complexos. Contudo, a observação do comportamento gráfico, evidencia que a hibridização é um caminho de otimização altamente viável.

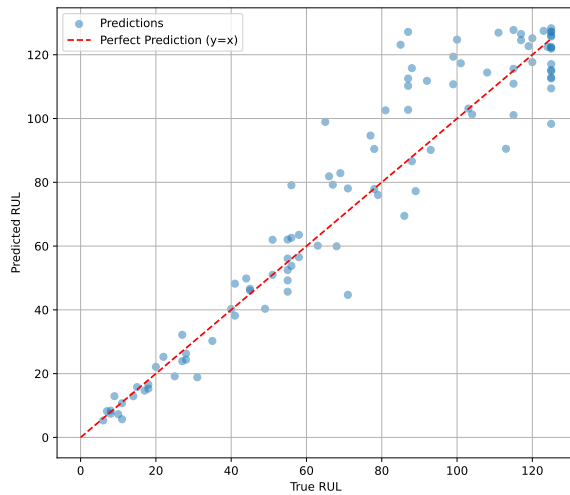
Figura 7 – Dispersão das predições de RUL utilizando a Função Híbrida Estática.



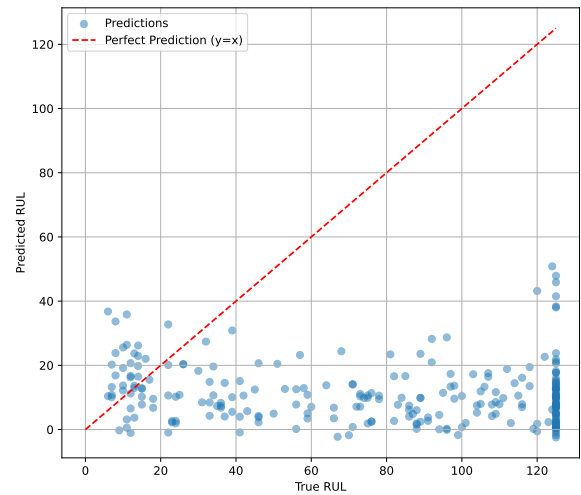
(a) FD001



(b) FD002



(c) FD003



(d) FD004

Fonte: Elaboração Própria.

Nos cenários de baixa variância, a estratégia híbrida demonstrou resultados mistos. No conjunto FD001 (quadrante a), a imposição do MSE limitou a capacidade da perda C-MAPSS de puxar os resultados para baixo, resultando em um *Score* de 399,85, um desempenho inferior ao modelo puramente assimétrico e caracterizando um meio-termo matemático. Por outro lado, no conjunto FD003 (quadrante c), ao ancorar as predições com o erro quadrático, o modelo lidou melhor com a ambiguidade dos múltiplos modos de falha, alcançando o menor RMSE de toda a análise para este subconjunto (12,42) sem sacrificar substancialmente o seu *Score*.

A principal descoberta, contudo, reside na análise visual dos regimes de condição múltipla (FD002 e FD004). Embora os valores de RMSE (80,84 e 80,39) e *Score* (841.619 e 679.277) permaneçam inaceitáveis, os quadrantes (b) e (d) revelam uma dinâmica promissora quando

contrastados com os gráficos do modelo puramente assimétrico (Figura 6).

Enquanto a otimização assimétrica direta sofreu um colapso completo (formando linhas horizontais planas em $RUL \approx 0$ no FD002 e $RUL \approx 65$ no FD004), o modelo Híbrido Estático tentou ativamente recuperar a correlação linear. Observa-se nos gráficos do modelo híbrido que as previsões voltaram a se espalhar ao longo do eixo X, com o componente MSE exercendo uma "força gravitacional" para mapear a tendência real de degradação, rompendo a estagnação.

Essa dispersão visual comprova que a hibridização de objetivos é um caminho de otimização estruturalmente viável e correto. A falha na convergência não decorre da combinação das perdas, mas exclusivamente da natureza estática dos pesos (1 : 1). Com coeficientes fixos, os gradientes de magnitude exponencial do C-MAPSS eventualmente sobrepujam os gradientes do MSE durante as flutuações de ruído nas épocas iniciais. Esse visual solidifica a premissa de que, para que o modelo aprenda de forma segura e precisa, a transição entre os objetivos deve ser gerida de forma gradual.

6.0.4 Transição de Objetivo em Fases

Para solucionar definitivamente o problema da instabilidade nos estágios iniciais de treinamento, substituiu-se a ponderação estática pelo *framework* POT. A premissa central desta abordagem foi blindar a arquitetura contra os gradientes exponenciais severos enquanto seus parâmetros ainda fossem aleatórios ou imaturos. O modelo foi forçado a consolidar sua representação de extração de características via MSE antes de ser submetido à rigorosa penalidade de risco assimétrico. Foram avaliadas duas dinâmicas de transição: *Binary* POT e *Hybrid* POT.

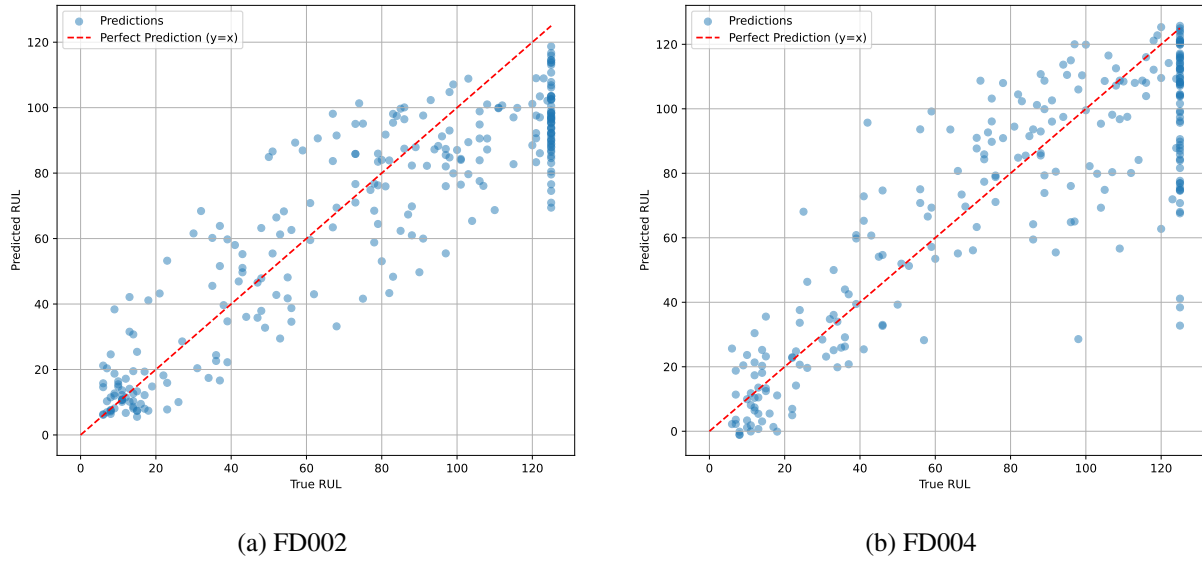
A estratégia *Binary* POT consiste em uma mudança de fase abrupta. Durante as primeiras 50 épocas (Fase de Estabilização), o modelo é treinado exclusivamente com a perda simétrica (L_{MSE}). A partir da época 51 (Fase de Refinamento), o objetivo é alterado integralmente para a perda assimétrica (L_{Safety}). Esta abordagem foi projetada de forma direcionada aos cenários de condição múltipla (FD002 e FD004), visando prevenir a explosão de gradientes que corrompeu o modelo base. A Tabela 6 apresenta o desempenho desta abordagem.

Tabela 6 – Desempenho preditivo do modelo otimizado por *Binary* POT nos regimes multivariados.

Dataset	RMSE	C-MAPSS Score
FD002	21,08	1.578,00
FD004	23,05	4.737,00

Fonte: Elaboração Própria.

A introdução do agendamento discreto de fases provou ser o mecanismo definitivo para viabilizar a otimização de segurança em ambientes de alta variância, resolvendo o colapso numérico que afetava as abordagens anteriores.

Figura 8 – Dispersão das previsões de RUL estabilizadas pela estratégia *Binary* POT.

Fonte: Elaboração Própria.

O colapso degenerativo observado na otimização assimétrica pura foi completamente erradicado. No conjunto FD002 (quadrante a), o *Binary* POT obteve sucesso. Ao ancorar a extração de características no MSE durante as primeiras 50 épocas, a rede aprendeu a filtrar o ruído inerente às mudanças de altitude e velocidade. Quando o treinamento alternou bruscamente para a Função C-MAPSS, o modelo não explodiu; ele apenas refinou a representação estável recém-adquirida, deslocando a nuvem de pontos para a zona de segurança (abaixo da diagonal).

Esse movimento coordenado resultou no melhor *Score* de toda a pesquisa para o FD002 (1.578,00). Aceitou-se um ligeiro aumento no erro médio global (RMSE 21,08, em comparação aos 18,05 do MSE padrão) como o *trade-off* matemático estritamente necessário para empurrar as previsões de um ambiente tão ruidoso para longe da zona de perigo.

O cenário multivariado crítico de FD004 (quadrante b) consolidou a eficácia desta abordagem. Diferente da linha horizontal estagnada gerada pela perda direta, a rede recuperou integralmente a capacidade preditiva temporal. O modelo alcançou a marca mais segura do estudo para este subconjunto extremo (RMSE 23,05 e *Score* de 4.737,00). A análise visual comprova uma aderência muito mais forte à diagonal do que a observada no modelo treinado apenas com MSE, eliminando severamente as superestimções críticas nas regiões de falha iminente ($RUL < 40$).

Em contraste com a mudança abrupta, a estratégia *Hybrid* POT implementou um "currículo linear" de aprendizado. Ao longo das 100 épocas de treinamento, o peso da otimização estatística (MSE) foi reduzido gradualmente, enquanto o peso da otimização de risco (C-MAPSS) foi proporcionalmente aumentado. Esta variante foi desenhada para promover uma adaptação mais fluida, minimizando o choque numérico da transição. A Tabela 7 consolida os resultados desta abordagem em todos os quatro conjuntos de dados.

Tabela 7 – Desempenho preditivo do modelo otimizado por *Hybrid* POT.

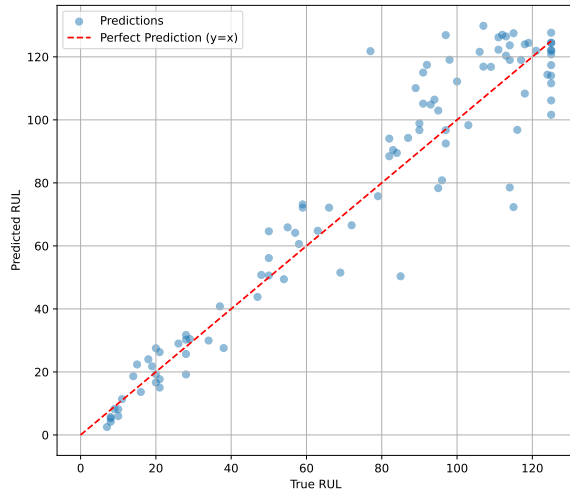
Dataset	RMSE	C-MAPSS Score
FD001	12,95	307,05
FD002	18,52	1.650,00
FD003	13,77	405,55
FD004	25,98	6.745,50

Fonte: Elaboração Própria.

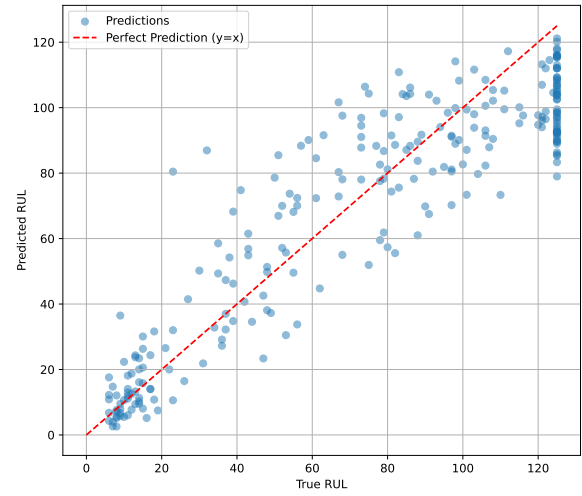
A estratégia de transição suave apresentou um desempenho altamente satisfatório ao estabelecer um equilíbrio notável entre a precisão temporal e a segurança preditiva. Particularmente nos cenários de condição operacional única, a exposição gradual do *Hybrid* POT permitiu que o modelo desenvolvesse mapeamentos latentes de alta fidelidade. No conjunto FD001 (quadrante **a**), a estratégia alcançou o melhor RMSE de toda a pesquisa para este subconjunto (12,95), mantendo um *Score* robusto (307,05). No FD003 (quadrante **c**), a transição evitou o colapso perante os múltiplos modos de falha, embora o tempo de exposição prolongado ao MSE tenha resultado em uma aversão ao risco levemente inferior à obtida pela perda direta (com o *Score* de 405,55).

Ao avaliar os cenários de condição múltipla, o *Hybrid* POT demonstrou notável resiliência no FD002 (quadrante **b**). A transição evitou a explosão de gradientes, guiando o modelo a um excelente RMSE (18,52) e um *Score* competitivo (1.650,00).

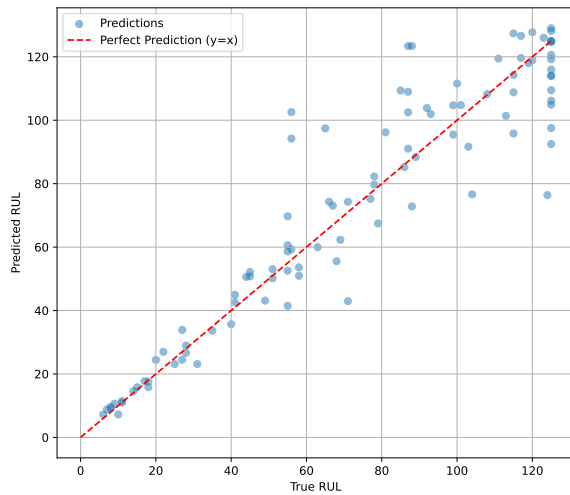
A limitação metodológica do currículo linear, no entanto, manifestou-se no extremo cenário do FD004 (quadrante **d**). A combinação de ruído severo e múltiplos modos de falha exigia uma imposição rigorosa de risco. A redução branda e progressiva do componente MSE demorou a aplicar a "gravidade" necessária para penalizar os *outliers* tardios no final da vida útil da turbina. Como reflexo, embora o modelo tenha evitado o colapso, ele registrou uma quantidade substancial de superestimações na faixa de $RUL < 40$, culminando em um *Score* inseguro de 6.745,50 e um RMSE de 25,98.

Figura 9 – Dispersão das previsões de RUL estabilizadas pela estratégia *Hybrid* POT.

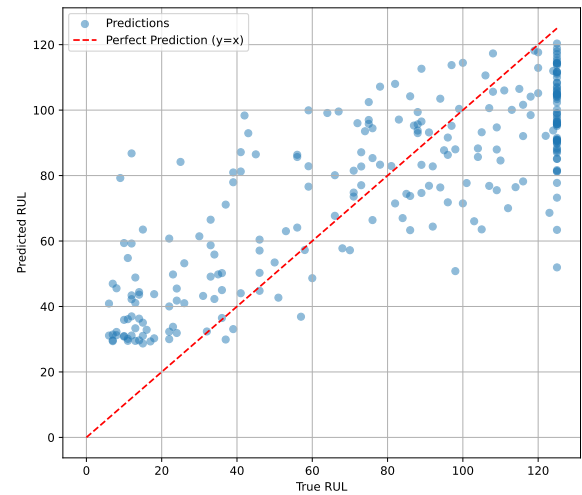
(a) FD001



(b) FD002



(c) FD003



(d) FD004

Fonte: Elaboração Própria.

Isso estabelece que, para ambientes críticos caracterizados por extrema incerteza cruzada, a imposição das restrições de segurança (*Binary* POT) é preferível ao amaciamento dos gradientes (*Hybrid* POT). Contudo, ambas as variantes baseiam-se em épocas pré-agendadas, motivando a investigação final de uma alternativa que gerencie este processo de forma autônoma: o Balanceamento Dinâmico de Perdas.

6.0.5 Balanceamento Dinâmico de Perdas

Embora o *framework* POT tenha solucionado o colapso numérico, ambas as suas variantes (*Binary* e *Hybrid*) sofrem de uma limitação fundamental: elas dependem de hiperparâmetros

engessados (épocas de corte ou taxas de decaimento lineares) definidos previamente. Visando dotar o *ProgFormer* de autonomia para gerenciar o próprio risco, implementou-se o DLB.

Nesta abordagem, a rede monitora o histórico de convergência de ambas as funções de perda através de EMA e atualiza dinamicamente os pesos (λ_{MSE} e λ_{Safety}) a cada época, favorecendo a função que apresentar maior dificuldade de otimização. A Tabela 8 consolida as métricas obtidas pelo DLB.

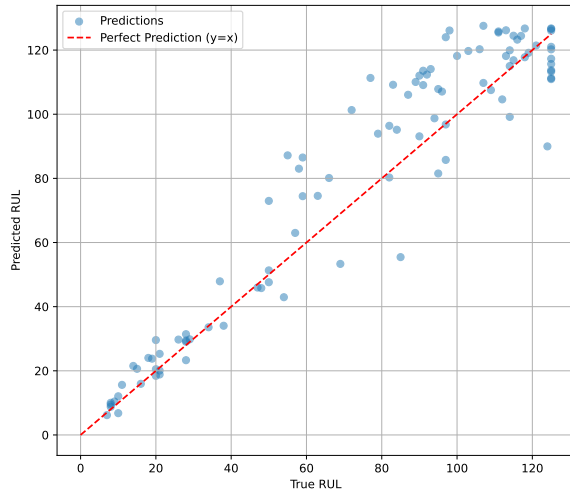
Tabela 8 – Desempenho preditivo do modelo otimizado por DLB.

Dataset	RMSE	C-MAPSS Score
FD001	13,48	311,02
FD002	20,19	3.718,23
FD003	15,85	1.114,52
FD004	24,99	7.936,91

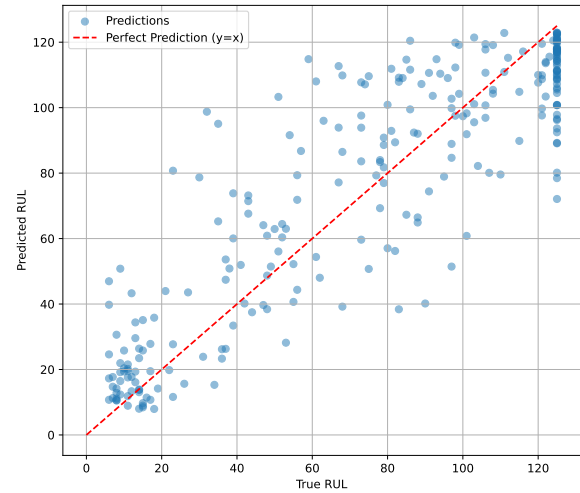
Fonte: Elaboração Própria.

A análise dos resultados revela que o algoritmo autônomo apresenta um comportamento distinto do agendamento manual. O foco primário do DLB é a sobrevivência e a estabilidade da convergência, operando como um mecanismo de auto-regulação conservador.

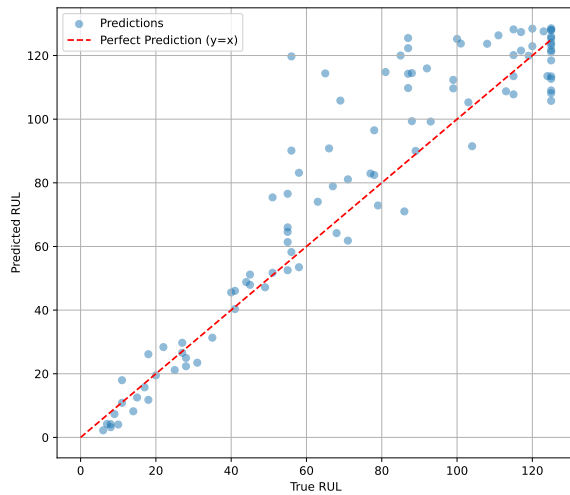
Figura 10 – Dispersão das previsões de RUL utilizando a estratégia DLB.



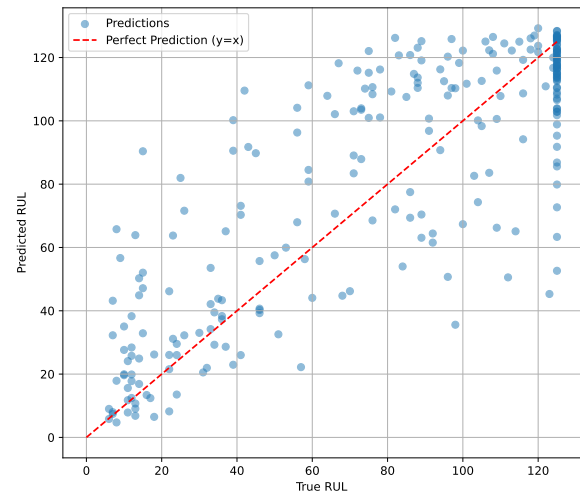
(a) FD001



(b) FD002



(c) FD003



(d) FD004

Fonte: Elaboração Própria.

Nos cenários de condição única, o DLB obteve resultados altamente satisfatórios no FD001 (quadrante a), mantendo um *Score* seguro de 311,02 e RMSE de 13,48. No entanto, o desafio apresentou-se no conjunto FD003 (quadrante c). Devido à incerteza conflitante gerada pelos dois modos de falha, os gradientes da perda C-MAPSS oscilaram fortemente. Para evitar a desestabilização (uma vez que os pesos são dinâmicos), o DLB identificou a volatilidade e reagiu reduzindo a influência do componente exponencial, ancorando-se majoritariamente no MSE. Isso evitou o colapso, mas resultou em um sacrifício na segurança final (*Score* de 1.114,52).

A verdadeira prova de fogo ocorreu nos cenários de condição múltipla (FD002 e FD004). Conforme observado nos quadrantes (b) e (d), o DLB atingiu seu propósito central: a rede navegou pelos gradientes explosivos de forma completamente autônoma, sem sofrer o colapso

degenerativo da perda direta ou do híbrido estático. Os preditores mantiveram a correlação linear com a vida útil sem prever a média constante.

Embora os *Scores* obtidos (3.718,23 para FD002 e 7.936,91 para FD004) não superem a marca excelente do *Binary* POT, a estratégia DLB solidifica o *ProgFormer* como um *framework* robusto, capaz de ingerir ambientes ruidosos e encontrar caminhos matemáticos seguros de forma orgânica.

6.0.6 Análise Comparativa com os Trabalhos Relacionados

Com as dinâmicas internas de otimização validadas, a última etapa deste estudo consiste em posicionar o *ProgFormer* frente às arquiteturas que compõem os modelos de referência na predição de RUL do conjunto C-MAPSS. Para assegurar uma comparação abrangente e transparente, todas as variantes de treinamento propostas para o modelo foram dispostas lado a lado com os sete trabalhos de referência discutidos na Seção 3. Os resultados consolidados são apresentados na Tabela 9, onde os melhores desempenhos globais de cada coluna encontram-se destacados em negrito.

Tabela 9 – Comparação de desempenho do *ProgFormer* com as abordagens de referência.

Modelo (Autor, Ano)	RMSE				C-MAPSS Score			
	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
Bi-LSTM (WANG <i>et al.</i> , 2018)	13,65	23,18	13,74	24,86	295,00	4.130,00	317,00	5.430,00
DCNN (LI; DING; SUN, 2018)	12,61	22,36	12,64	23,31	272,70	10.412,00	284,00	12.466,00
Attention-LSTM (BOUJAMZA; LISSANE ELHAQ, 2022)	13,95	17,65	12,72	20,21	320,00	2.102,00	223,00	3.100,00
Transformer + ARR (KIM; CHOI; LIM, 2024)	11,36	18,97	11,28	20,69	192,22	2.433,15	133,41	2.842,44
FFG-Transformer (ZHOU <i>et al.</i> , 2024)	11,36	13,24	11,80	15,16	173,89	714,98	215,32	1.049,53
AGATT (ZENG; WU; YIWEI, 2023)	11,45	13,87	10,88	16,43	204,00	323,00	205,00	632,00
PLO-Transformer (ZHANG <i>et al.</i> , 2025)	10,26	12,97	11,61	14,36	149,91	728,06	278,61	1.160,60
ProgFormer (MSE Padrão)	14,55	18,05	13,90	23,35	517,26	2.243,00	1.256,09	5.726,00
ProgFormer (Perda C-MAPSS)	13,09	87,75	12,46	44,74	259,87	1.409.215,00	293,25	18.434,00
ProgFormer (Híbrido Estático)	13,99	80,84	12,42	80,39	399,85	841.619,00	298,12	679.277,00
ProgFormer (Binary POT)	–	21,08	–	23,05	–	1.578,00	–	4.737,00
ProgFormer (Hybrid POT)	12,95	18,52	13,77	25,98	307,05	1.650,00	405,55	6.745,50
ProgFormer (DLB)	13,48	20,19	15,85	24,99	311,02	3.718,23	1.114,52	7.936,91

Fonte: Elaboração Própria.

Observa-se que, na minimização do erro global absoluto, os modelos híbridos mais recentes, como o AGATT de Zeng, Wu e Yiwei (2023) e o PLO-Transformer de Zhang *et al.* (2025), retêm os melhores resultados dentre os estudos analisados. Estes modelos alcançam níveis excepcionais de precisão e segurança ao fundir múltiplos módulos extratores, combinando o mecanismo de atenção espacial de Grafos (GATs), redes convolucionais profundas e otimizadores meta-heurísticos, o que lhes confere uma imensa capacidade de representação latente da física complexa da turbina, refletida nos menores valores globais destacados na Tabela 9.

Contudo, a competitividade do *ProgFormer* reside em sua eficiência orientada ao risco contrastada com sua extrema simplicidade arquitetural. Utilizando apenas blocos *Transformer Encoder* puros, sem hibridização de camadas extratoras de características, o modelo proposto

foi capaz de superar ou equiparar-se a arquiteturas clássicas consolidadas, como redes recorrentes bidirecionais (Bi-LSTM) e convolucionais (DCNN), especialmente no quesito de risco operacional (*Score*).

Destaca-se que, no ruidoso cenário do FD002, o *Score* de 1.578,00 obtido pela estratégia *Binary* POT é inferior às pontuações de modelos recorrentes com atenção (Attention-LSTM, com *Score* de 2.102,00). Além disso, supera abordagens recentes em *Transformers* que utilizam técnicas complexas de reponderação adaptativa de amostras para lidar com desbalanceamento, como o Transformer + ARR de Kim, Choi e Lim (2024) (*Score* de 2.433,15). Isso comprova a premissa fundamental da pesquisa: alterar e agendar dinamicamente a superfície de perda tem um impacto de mitigação de risco tão ou mais robusto do que manipular exclusivamente a arquitetura extratora ou as distribuições dos dados de entrada.

Em suma, os resultados estabelecem o *ProgFormer* como uma alternativa competitiva válida. Enquanto a literatura atual concentra seus esforços no aumento massivo da complexidade estrutural e no empilhamento de redes para melhorar as predições, este estudo demonstra que a exploração rigorosa da dinâmica de otimização é um vetor de aprimoramento preditivo eficaz.

7 CONCLUSÕES

Este trabalho investigou o desafio crítico da predição da RUL em motores aeronáuticos, abordando uma lacuna fundamental na literatura de prognósticos: a negligência do risco operacional durante a otimização de modelos de aprendizado profundo. Para solucionar esta problemática, a pesquisa cumpriu integralmente o seu objetivo principal ao propor e validar o *ProgFormer*, um *framework* preditivo baseado em uma arquitetura *Transformer Encoder* pura, cujo diferencial reside no desenvolvimento de estratégias avançadas de dinâmica de treinamento estritamente orientadas à segurança.

Os objetivos específicos delineados na introdução foram sistematicamente alcançados através da avaliação do modelo nos quatro subconjuntos de dados padronizados do C-MAPSS. Primeiramente, demonstrou-se de forma empírica que o treinamento tradicional guiado pelo MSE produz previsões simétricas que frequentemente superestimam a vida útil da turbina. Ao implementar a função de perda assimétrica orientada ao risco (Perda C-MAPSS), o modelo obteve sucesso expressivo em regimes operacionais estáveis (FD001 e FD003), forçando a rede a adotar um comportamento conservador e mitigando ativamente o risco de previsões tardias perigosas.

No entanto, conforme antecipado pelo escopo do trabalho, a imposição direta da penalidade exponencial desencadeou instabilidades numéricas e colapsos de aprendizado nos cenários ruidosos de condição múltipla (FD002 e FD004). Para superar este desafio e garantir a convergência em superfícies de erro multivariadas, as estratégias de gerenciamento da dinâmica de treinamento foram desenvolvidas e aplicadas com êxito. A estratégia POT solucionou o colapso ao utilizar o MSE como um escudo estabilizador inicial. A variante *Binary* POT consolidou-se como a abordagem metodológica mais robusta para ambientes altamente variáveis, alcançando as melhores marcas de mitigação de risco do estudo. Paralelamente, a implementação do DLB atendeu ao objetivo de criar um mecanismo autônomo, provando que o modelo é capaz de rastrear seu próprio histórico de convergência e ajustar a ponderação dos gradientes em tempo real para evitar a degeneração preditiva.

A etapa final do estudo apresentou também uma análise comparativa frente aos trabalhos relacionados. Ao manter uma arquitetura intencionalmente enxuta, o *ProgFormer* superou algumas das abordagens consolidadas baseadas em RNNs e CNNs no quesito de segurança operacional (*Score*). Esses resultados consolidam a tese central defendida nesta pesquisa: a gestão inteligente da superfície de perda e da dinâmica de otimização exerce um impacto de confiabilidade preditiva tão profundo quanto o empilhamento de camadas neurais.

As contribuições deste trabalho deram origem a um artigo aceito para publicação na *Brazilian Conference on Intelligent Systems* (BRACIS 2026) (SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2026), intitulado *Exploring Transformers for Predicting the Remaining Useful Life of Turbofan Engines*. A versão *pre-print* do artigo encontra-se no Apêndice B.

Como perspectiva para trabalhos futuros, sugere-se a extrapolação destas metodologias

de treinamento dinâmico para os modelos que atualmente retêm os recordes de precisão global na literatura. A integração dos mecanismos POT e DLB às arquiteturas híbridas complexas, como aquelas que fundem Redes de Atenção em Grafos e Redes Convolucionais Profundas, desponta como um horizonte científico promissor. Acredita-se que a união harmônica entre a imensa capacidade de representação latente destas arquiteturas massivas e a otimização intrinsecamente segura e auto-regulada concebida neste estudo resultará em uma abordagem preditiva de excelência para sistemas de manutenção preditiva em setores críticos.

REFERÊNCIAS

ANTHROPIC. **Claude Opus 4.8**. [S.l.: s.n.], 2026. Acesso em: 8 jul. 2026. Disponível em: <<https://www.anthropic.com/claude/opus>>. Citado na p. 42.

BENGIO, Yoshua *et al.* Curriculum learning. *In: PROCEEDINGS of the 26th Annual International Conference on Machine Learning*. Montreal, Quebec, Canada: Association for Computing Machinery, 2009. (ICML '09), p. 41–48. DOI: 10.1145/1553374.1553380. Disponível em: <<https://doi.org/10.1145/1553374.1553380>>. Citado nas pp. 23, 33.

BISHOP, Christopher M. **Pattern Recognition and Machine Learning**. New York: Springer, 2006. ISBN 978-0-387-31073-2. Citado na p. 18.

BOUJAMZA, Abdeltif; LISSANE ELHAQ, Saad. Attention-based LSTM for Remaining Useful Life Estimation of Aircraft Engines. **IFAC-PapersOnLine**, v. 55, n. 12, p. 450–455, 2022. 14th IFAC Workshop on Adaptive and Learning Control Systems ALCOS 2022. ISSN 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2022.07.353>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405896322007546>>. Citado nas pp. 27, 29, 35, 55.

CHEN, Zhao *et al.* GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks, nov. 2017. DOI: 10.48550/arXiv.1711.02257. Citado nas pp. 23, 34, 40.

CONSELHO NACIONAL DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO. **Portaria nº 2.664, de 6 de março de 2026: institui a Política de Integridade na Atividade Científica**. [S.l.]: [s.n.], 2026. Disponível em: <<http://memoria2.cnpq.br>>. Citado nas pp. 41, 42.

DUAN, Yuntao; ZHANG, Tao; SHI, Dunhuang. Anomaly Detection and Remaining Useful Life Prediction for Turbofan Engines with a Key Point-Based Approach to Secure Health Management. **Sensors**, v. 24, p. 8022, dez. 2024. DOI: 10.3390/s24248022. Citado nas pp. 13, 14.

FERREIRA, Carlos; GONÇALVES, Gil. Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. **Journal of Manufacturing Systems**, Elsevier, v. 63, p. 550–562, 2022. DOI: 10.1016/j.jmsy.2022.05.010. Citado nas pp. 20, 23, 24.

FREDERICK, Dean K.; DECASTRO, Jonathan A.; LITT, Jonathan S. **User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)**. [S.l.], 2007. Citado nas pp. 15, 24.

GLOROT, Xavier; BORDES, Antoine; BENGIO, Yoshua. Deep sparse rectifier neural networks. In: PROCEEDINGS of the fourteenth international conference on artificial intelligence and statistics. [S.l.: s.n.], 2011. P. 315–323. Citado na p. 18.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado nas pp. 14, 17, 20, 33, 39.

GOOGLE. **Gemini**. [S.l.: s.n.], 2026. Acesso em: 8 jul. 2026. Disponível em: <<https://gemini.google.com/>>. Citado na p. 42.

HAYKIN, Simon. **Neural Networks and Learning Machines**. 3. ed. [S.l.]: Pearson, 2009. Citado na p. 19.

HEIMES, Felix. Recurrent neural networks for remaining useful life estimation. In: 2008 International Conference on Prognostics and Health Management. [S.l.: s.n.], nov. 2008. P. 1–6. DOI: 10.1109/PHM.2008.4711422. Citado na p. 37.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado na p. 20.

HORNIK, Kurt; STINCHCOMBE, Maxwell; WHITE, Halbert. Multilayer feedforward networks are universal approximators. **Neural networks**, Elsevier, v. 1, n. 5, p. 359–366, 1989. Citado na p. 19.

KIM, Gyeongho; CHOI, Jae Gyeong; LIM, Sunghoon. Using transformer and a reweighting technique to develop a remaining useful life estimation method for turbofan engines. **Engineering Applications of Artificial Intelligence**, v. 133, p. 108475, 2024. ISSN 0952-1976. DOI: <https://doi.org/10.1016/j.engappai.2024.108475>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S095219762400633X>>. Citado nas pp. 26, 27, 29, 35, 38, 55, 56.

KINGMA, Diederik; BA, Jimmy. Adam: A Method for Stochastic Optimization. **International Conference on Learning Representations**, dez. 2014. Citado nas pp. 19, 23, 34.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Citado nas pp. 17, 20.

LEI, Jianxin *et al.* A Review: Prediction Method for the Remaining Useful Life of the Mechanical System. **Journal of Failure Analysis and Prevention**, v. 22, nov. 2022. DOI: 10.1007/s11668-022-01532-4. Citado nas pp. 17, 18, 24.

LI, Xiang; DING, Qian; SUN, Jian-Qiao. Remaining useful life estimation in prognostics using deep convolution neural networks. **Reliability Engineering & System Safety**, Elsevier, v. 172, p. 1–11, 2018. Citado nas pp. 27, 29, 35, 36, 55.

RUMELHART, David E; HINTON, Geoffrey E; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **Nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986. Citado na p. 19.

SAXENA, Abhinav *et al.* Damage propagation modeling for aircraft engine run-to-failure simulation. **International Conference on Prognostics and Health Management**, out. 2008. DOI: 10.1109/PHM.2008.4711414. Citado nas pp. 13, 14, 24, 33.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. **36th Brazilian Conference on Intelligent Systems (BRACIS 2026)**. [S.l.: s.n.], 2026. Disponível em: <<https://bracis.sbc.org.br/2026/>>. Citado na p. 57.

TIPPAREDDY, Samaikya. **Predicting Remaining Useful Life of Turbofan Engines on CMAPSS and N-CMAPSS using Deep Recurrent Neural Networks**. 2024. Master's Project – San José State University. DOI: 10.31979/etd.cyfe-z4sj. Disponível em: <https://scholarworks.sjsu.edu/etd_projects/1360>. Citado na p. 14.

UNIVERSIDADE FEDERAL DE SANTA CATARINA. **Resolução Normativa nº 217/2025/CUn**. [S.l.]: [s.n.], 2025. Citado na p. 42.

VASWANI, Ashish *et al.* Attention is All you Need. In: GUYON, I. *et al.* (Ed.). **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2017. Citado nas pp. 21, 22, 31, 32.

WANG, Jiujuan *et al.* Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network. **2018 Prognostics and System Health Management Conference (PHM-Chongqing)**, p. 1037–1042, 2018. Disponível em: <<https://api.semanticscholar.org/CorpusID:57753485>>. Citado nas pp. 27, 29, 35, 36, 55.

ZENG, Ming; WU, Feng; YIWEI, Cheng. Remaining Useful Life Prediction via Spatio-Temporal Channels and Transformer. **IEEE Sensors Journal**, PP, p. 1–1, dez. 2023. DOI: 10.1109/JSEN.2023.3324330. Citado nas pp. 14, 27–29, 35, 55.

ZHANG, Enlai *et al.* PLO-Transformer-LSTM: a remaining useful life prediction method for aircraft engines. **Engineering Research Express**, v. 7, ago. 2025. DOI: 10.1088/2631-8695/adece8. Citado nas pp. 27–29, 35, 37, 55.

ZHOU, Zhihao *et al.* An aircraft engine remaining useful life prediction method based on predictive vector angle minimization and feature fusion gate improved transformer model. **Journal of Manufacturing Systems**, v. 76, p. 567–584, out. 2024. DOI: 10.1016/j.jmsy.2024.08.025. Citado nas pp. 27–29, 35, 55.

Apêndices

APÊNDICE A – CÓDIGO FONTE

Todo o código fonte desenvolvido para este trabalho, incluindo as variantes do modelo *ProgFormer* (perdas MSE, C-MAPSS, híbrida estática, em fases e dinâmica), os pesos treinados e os *scripts* utilizados para gerar os gráficos apresentados ao longo deste documento, encontra-se disponível publicamente no seguinte repositório: <https://codigos.ufsc.br/tcc2/tcc-progformer>.

APÊNDICE B – ARTIGO

Este apêndice apresenta o artigo produzido a partir deste trabalho, intitulado *Exploring Transformers for Predicting the Remaining Useful Life of Turbofan Engines*, que foi aceito na BRACIS 2026, sendo reproduzido na íntegra nas páginas seguintes.

Exploring Transformers for Predicting the Remaining Useful Life of Turbofan Engines

Eduardo Cunha Cabral¹[0009–0000–3495–3676] and Eduardo Camilo Inacio¹[0000–0003–3424–8269]

Universidade Federal de Santa Catarina (UFSC), Florianópolis, Brazil
eduardo.cabral@grad.ufsc.br

Abstract. Precise Remaining Useful Life (RUL) estimation in turbofan engines is vital for operational safety and cost-efficiency in the aviation industry. Although Transformer-based architectures have set new benchmarks in sequence modeling, conventional training objectives centered on the Mean Squared Error (MSE) fail to account for the asymmetric risk profile of predictive maintenance. This paper evaluates the viability of Transformers for RUL prediction using the CMAPSS dataset, focusing on the numerical instabilities triggered by asymmetric exponential loss functions in complex operational regimes. We demonstrate that direct optimization of safety-biased losses often leads to model collapse due to exploding gradients. To mitigate these effects, our methodology utilizes a Phased Objective Transition (POT) framework, comprising a Binary POT strategy for discrete stabilization and a Hybrid POT for curriculum-based refinement. Furthermore, an independent Dynamic Loss Balancing (DLB) mechanism was implemented to provide autonomous stabilization based on training dynamics. Experimental results indicate that both the POT framework and the DLB model provide viable alternatives for implementing risk-aware deep learning in industrial prognostic systems.

Keywords: Remaining Useful Life · Transformer · Asymmetric Loss · Phased Objective Transition · Dynamic Loss Balancing.

1 Introduction

Accurate estimation of Remaining Useful Life (RUL) is a fundamental task in Prognostics and Health Management (PHM) [2], particularly in safety-critical domains such as the aviation industry. Reliable RUL predictions enable condition-based maintenance, reducing operational costs while preventing unexpected failures. Among the benchmark datasets for this problem, the CMAPSS turbofan engine simulation has become a standard for evaluating data-driven approaches due to its realistic degradation patterns and varying operational conditions [8].

Traditional RUL prediction methods have relied on statistical models and conventional machine learning techniques. However, recent advances in deep learning have significantly improved performance by capturing complex temporal dependencies in multivariate sensor data. In particular, Transformer-based

architectures [9] have demonstrated remarkable success in sequence modeling tasks due to their ability to learn long-range dependencies through self-attention mechanisms. This capability makes them especially attractive for RUL estimation, where degradation patterns may span long temporal horizons.

Despite these advances, most existing approaches optimize models using symmetric loss functions such as the Mean Squared Error (MSE). While MSE provides stable gradients and facilitates convergence, it fails to capture the asymmetric risk profile inherent to predictive maintenance. In real-world scenarios, overestimating the RUL (late prediction) can lead to catastrophic system failures, whereas underestimation (early prediction) is comparatively less critical. To address this limitation, the PHM 2008 challenge introduced a scoring function that penalizes late predictions more severely than early ones [8]. However, directly optimizing such asymmetric exponential losses poses significant challenges, as their steep gradients can lead to numerical instability, gradient explosion, and convergence to trivial solutions.

To address this challenge, this work investigates a set of training objectives and strategies designed to balance optimization stability and risk-aware learning. As baselines, three primary loss formulations are considered: a standard MSE model for stable convergence, a model trained directly with the asymmetric CMAPSS loss, and a hybrid model combining both objectives through a weighted formulation. While these approaches provide different trade-offs between accuracy and safety, they also expose the limitations of static objective design, particularly in complex regimes.

The main contributions of this work are summarized as follows:

- A systematic analysis of the instability issues arising from the direct optimization of asymmetric exponential loss functions for RUL prediction, particularly in complex multi-regime scenarios of the CMAPSS dataset.
- The proposal of Phased Objective Transition (POT) strategies, including Binary POT and Hybrid POT, to stabilize training by introducing structured transitions between symmetric and asymmetric objectives.
- The introduction of a Dynamic Loss Balancing (DLB) mechanism based on loss dynamics, which adaptively adjusts the contribution of multiple objectives during training without requiring gradient-level modifications.
- A comprehensive empirical evaluation across all four CMAPSS subsets (FD001–FD004), comparing MSE, CMAPSS, hybrid, POT, and DLB approaches, highlighting the trade-offs between predictive accuracy, stability, and safety-oriented performance.

The remainder of this paper is organized as follows: Section 2 details the theoretical foundation of asymmetric loss and objective scheduling. Section 3 reviews related works. Section 4 presents the proposed methodology, including the Phased Object Transition (POT) framework and Dynamic Loss Balancing (DLB) mechanism. Section 5 discusses the experimental results, and Section 6 concludes with our final remarks and directions for future works.

2 Background

This section outlines the theoretical concepts underlying the proposed training strategies. It first contrasts symmetric loss functions with the asymmetric CMAPSS scoring function required for risk-aware predictive maintenance. Finally, it introduces curriculum learning and adaptive loss weighting, which form the conceptual basis for the objective scheduling and dynamic balancing mechanisms used in this study.

2.1 Symmetric vs Asymmetric Loss Functions in RUL

Conventional optimization objectives in data-driven prognostics are typically based on symmetric loss functions, such as the MSE, which assign equal penalties to overestimation and underestimation of the target variable. This property results in balanced gradient updates with respect to prediction errors, making such losses attractive from an optimization standpoint due to their stability and smooth convergence behavior.

However, in the context of RUL estimation, the direction of the prediction error carries distinct operational implications. Asymmetric loss functions address this limitation by assigning different penalties depending on whether the model overestimates or underestimates the true RUL, thereby enabling the incorporation of domain-specific risk preferences into the learning objective [8]. This formulation allows models to prioritize safety-critical errors without altering the underlying predictive architecture.

From an optimization perspective, introducing asymmetry modifies the structure of the loss landscape. In contrast to symmetric objectives, where gradient magnitudes depend solely on the error magnitude, asymmetric losses can amplify gradients selectively based on the direction of the error. This behavior may lead to biased updates, increased sensitivity to certain regions of the error space, and potential instability during training, particularly in deep neural networks [4]. These characteristics highlight the need for training strategies that can effectively manage the trade-off between stability and risk-aware optimization.

2.2 CMAPSS Scoring Function

To properly evaluate predictive models in safety-critical domains, the optimization objective must reflect the physical risks of the system. Unlike symmetric metrics such as MSE, the CMAPSS scoring function applies an asymmetric, exponential penalty to prediction errors.

Let $t_{\text{RUL},i}$ represent the true RUL (measured in operational cycles) of the i -th engine unit, and let $\hat{t}_{\text{RUL},i}$ denote the model's predicted RUL for that same unit. The prediction error, d_i , is defined as the difference between the prediction and the ground truth, as observed in Equation 1:

$$d_i = \hat{t}_{\text{RUL},i} - t_{\text{RUL},i} \quad (1)$$

An overestimation, predicting the engine will last longer than it actually does, results in $d_i > 0$, while an underestimation (predicting an earlier failure) yields $d_i < 0$. The total safety score S across n units is mathematically defined in Equation 2.

$$S = \begin{cases} \sum_{i=1}^n \left[\exp\left(-\frac{d_i}{a_1}\right) - 1 \right], & \text{for } d_i < 0 \\ \sum_{i=1}^n \left[\exp\left(\frac{d_i}{a_2}\right) - 1 \right], & \text{for } d_i \geq 0 \end{cases} \quad (2)$$

In this formulation, $a_1 = 13$ and $a_2 = 10$ act as scaling constants that dictate the steepness of the penalty. Because a_2 is smaller than a_1 , the penalty grows significantly faster for late predictions ($d_i > 0$). This ensures that the model is heavily penalized for catastrophic overestimations, while premature maintenance predictions ($d_i < 0$) are tolerated with a much gentler penalty curve.

From an optimization perspective, the exponential growth of the scoring function introduces significant challenges. During early training stages, when prediction errors are typically large, the resulting gradients can become excessively high, increasing the risk of numerical instability and divergence. This issue is particularly relevant in deep learning models, where large gradient magnitudes can disrupt parameter updates and hinder convergence [4]. Consequently, while the CMAPSS score provides a realistic evaluation criteria, directly optimizing it requires careful handling to ensure stable training.

2.3 Curriculum Learning and Objective Scheduling

Training deep neural networks under complex or highly non-linear objectives can be challenging, particularly when the loss function exhibits steep gradients or uneven optimization landscapes. Curriculum learning addresses this issue by structuring the training process as a sequence of progressively more difficult tasks, allowing the model to first learn stable and generalizable representations before focusing on more specialized objectives [1].

In the context of RUL estimation, the transition from symmetric to asymmetric loss functions can be interpreted as a curriculum over the optimization objective. Symmetric losses such as MSE provide a smooth and stable training signal, allowing the model to capture the underlying degradation dynamics without being affected by large gradient magnitudes. In contrast, asymmetric exponential losses introduce sharper penalties that emphasize safety-critical errors but may destabilize training if applied from the outset. Viewing this transition as a curriculum provides a principled framework for managing this trade-off.

Objective scheduling strategies extend this idea by explicitly controlling how the training objective evolves over time. Instead of optimizing a fixed loss function, the model is exposed to a sequence of objectives that gradually shift the optimization focus. This paradigm allows the learning process to adapt to different stages of training, prioritizing stability in early epochs and progressively incorporating risk-aware constraints. Such strategies are closely related to broader principles in deep learning optimization, where controlling gradient magnitudes and learning dynamics is critical for stable convergence [4].

This perspective directly motivates the use of Phased Objective Transition (POT) strategies, where the learning process is structured to control the interaction between stability-oriented and safety-oriented objectives. By framing the transition between loss functions as a curriculum, POT provides a systematic mechanism to mitigate instability while preserving the benefits of asymmetric error penalization.

2.4 Adaptive Loss Weighting

In multi-objective learning settings, models are often required to optimize multiple loss functions simultaneously, each capturing different aspects of the task. A common approach is to combine these objectives using fixed scalar weights. However, static weighting schemes assume that the relative importance and scale of each loss remain constant throughout training. In practice, this assumption rarely holds, as different objectives may evolve at different rates and exhibit distinct optimization dynamics [3].

Adaptive loss weighting methods address this limitation by dynamically adjusting the contribution of each objective based on the state of the training process. Instead of relying on predefined weights, these approaches seek to balance the optimization by considering factors such as gradient magnitudes, convergence speed, or relative loss improvement. To obtain stable estimates of these quantities, Exponential Moving Averages (EMA) are commonly employed, as they provide a smoothed representation of temporal signals while reducing sensitivity to short-term fluctuations [7].

In the context of RUL estimation, combining symmetric and asymmetric objectives introduces an inherent imbalance. The MSE component typically provides stable and well-scaled gradients, while the asymmetric exponential component may produce large updates when prediction errors are significant. This discrepancy can lead to unstable training or suboptimal convergence if fixed weights are used, particularly in early training stages where errors are large.

This motivates the use of Dynamic Loss Balancing (DLB) strategies, where the contribution of each loss component is continuously adjusted according to its relative training dynamics. By leveraging EMA to track the evolution of each objective, DLB enables a smoother and more robust estimation of their relative progress, allowing the weighting mechanism to react adaptively without introducing instability. This dynamic adjustment provides a practical mechanism to maintain training stability while progressively incorporating asymmetric error penalization.

3 Related Works

Recent approaches to RUL prediction on the CMAPSS dataset span recurrent, convolutional, and attention-based architectures, as summarized in Table 1. Early deep learning methods focused on recurrent models, with Wang *et al.* [10] proposing a deep BiLSTM to capture bidirectional temporal dependencies, and

Boujamza *et al.* [2] extending this approach with attention mechanisms to emphasize critical time steps.

Table 1. Comparison of RUL Prediction Approaches for Turbofan Engines

Work	Architecture	Main Contribution	Loss
Wang et al. (2018)	BiLSTM	Bidirectional temporal modeling	MSE
Boujamza et al. (2022)	Attn-LSTM	Temporal attention mechanism	MSE
Zeng et al. (2023)	GAT + Transformer	Spatial-temporal dependency modeling	MSE
Kim et al. (2024)	Transformer	Adaptive reweighting (ARR)	Weighted MSE
Zhou et al. (2024)	CNN + Transformer	Feature fusion and PVA loss	PVA (Asymmetric)
Zhang et al. (2025)	Transformer + LSTM	PLO-based hyperparameter optimization	MSE
This Work	Transformer Encoder	Objective scheduling and DLB	MSE + CMAPSS

More recent work has shifted toward Transformer-based and hybrid architectures. Kim *et al.* [6] combined a Transformer encoder with adaptive reweighting to address data imbalance, while Zeng *et al.* [11] incorporated graph attention mechanisms to model spatial and temporal dependencies. Zhang *et al.* [12] further explored hybrid designs by integrating Transformer and LSTM components with meta-heuristic hyperparameter optimization.

Despite these architectural advances, most methods rely on symmetric loss functions such as MSE. Only a limited number of studies explicitly consider asymmetric risk, such as the PVA loss proposed by Zhou *et al.* [13]. This suggests that, while model architectures have evolved significantly, the role of training objectives and optimization strategies remains comparatively underexplored.

In contrast, this work focuses on the training process itself, investigating how objective scheduling and adaptive loss weighting can enable stable optimization under asymmetric loss functions.

4 Methodology

This section details the experimental framework developed to evaluate risk-aware RUL prediction. It first outlines the data preparation pipeline for the CMAPSS dataset and describes the core Transformer Encoder architecture. Subsequently,

it defines the proposed training strategies, introducing the baseline, the POT framework and the DLB mechanism. Finally, it details the experimental setup.

4.1 Data Pipeline and Preprocessing

Data preparation follows a pipeline designed for the CMAPSS benchmark to ensure consistency with the existing literature. For datasets operating under constant conditions, namely FD001 and FD003, a fixed selection of 14 sensors with non-null variance is retained. In contrast, for the multi-regime datasets FD002 and FD004, a dynamic feature selection is implemented to retain all columns with a standard deviation greater than zero, resulting in an input vector comprised of 21 sensors and 3 operational settings. To ensure numerical stability for the optimizer, all input features undergo Min-Max scaling to a range between zero and one, with scaling parameters derived solely from the training set to prevent data leakage. Regarding the target variable, a piecewise linear RUL labeling strategy is adopted with a maximum threshold of 125 cycles [5]. This approach prevents the model from trying to learn negligible degradation patterns during the early healthy stages where the sensor signals do not show perceptible change. Finally, the continuous multivariate time-series are segmented into overlapping sliding windows with a length of 30 cycles.

4.2 Architecture

The core of the predictive model is a Transformer Encoder architecture adapted for regression tasks. The input window is first projected into a 256-dimensional latent space via a linear dense layer. Sinusoidal positional encodings are then added to the embeddings to incorporate temporal order [9]. The model backbone consists of two identical encoder blocks, each featuring a four-head Multi-Head Self-Attention mechanism followed by a position-wise feed-forward network with 256 hidden units and ReLU activation. Residual connections and layer normalization are applied around each sub-layer, alongside a dropout rate of 0.1. Finally, a Global Average Pooling layer condenses the temporal features into a single vector, which is processed by a regression head comprising a 128-unit dense layer and a final linear neuron to produce the RUL estimate.

4.3 Proposed Training Strategies

This study investigates protocols to address the trade-off between absolute error minimization and the asymmetric risk requirements of the aeronautical industry.

Mean Squared Error (MSE) The fundamental baseline employs the standard MSE loss function. This objective treats all errors equally by penalizing the square of the difference between the predicted and true RUL. While this provides a smooth loss surface for initial convergence, it is fundamentally flawed for safety-critical systems as it does not differentiate between a conservative early prediction and a dangerous late prediction.

CMAPSS Score To incorporate asymmetric risk into the training process, the CMAPSS scoring function introduced in the PHM 2008 challenge is employed [8]. As detailed in Section 2.2, this function assigns exponentially larger penalties to late predictions than to early ones, reflecting the higher operational risk associated with overestimating RUL.

While the CMAPSS score provides a realistic objective for safety-critical applications, its exponential structure introduces optimization challenges. In particular, large prediction errors during early training stages can lead to steep gradients, increasing the risk of instability and poor convergence.

In this work, the CMAPSS score is evaluated both as a standalone objective and as a component within composite training strategies, enabling a systematic analysis of its impact on model behavior and stability.

Hybrid Model To combine the smooth optimization properties of the MSE loss with the safety-aware characteristics of the CMAPSS scoring function, we define a hybrid objective as a weighted sum of both components, as described in Equation 3,

$$L_{\text{hybrid}} = \lambda_1 \cdot L_{\text{mse}} + \lambda_2 \cdot L_{\text{cmapss}} \quad (3)$$

where λ_1 and λ_2 are scalar coefficients controlling the contribution of each loss component. In this study, both coefficients are set to $\lambda_1 = \lambda_2 = 1$, providing a balanced formulation in terms of coefficient scaling.

The MSE component corresponds to the mean squared error between predicted and true RUL values, while the CMAPSS component follows the asymmetric exponential formulation described in Section 2.2, computed as the mean penalty over the batch to enable gradient-based optimization.

Due to differences in scale between the MSE and CMAPSS components, equal weighting does not necessarily imply equal influence during optimization. Nevertheless, this formulation provides a simple and effective baseline that jointly accounts for prediction accuracy and asymmetric risk.

This observation motivates the exploration of adaptive weighting strategies, as addressed by the Phased Objective Transition (POT) and Dynamic Loss Balancing (DLB) methods described in the following sections.

Phased Objective Transition (POT) In complex regimes such as FD002 and FD004, the high variance of sensor data combined with the steep gradients of an exponential loss function often leads to numerical instability or model collapse. To mitigate this, this study adopts two distinct Phased Objective Transition (POT) strategies designed to stabilize the training process by gradually introducing the asymmetric safety constraints. This approach is conceptually related to curriculum learning, where simpler objectives are used in early stages to facilitate stable optimization [1].

The first approach, referred to as Binary POT, divides the training into two stages. The model undergoes an initial 50-epoch stabilization phase using only the symmetric MSE loss to learn the general degradation manifold without the

risk of gradient explosion. This is followed by a 50-epoch refinement phase where the objective is switched entirely to the CMAPSS loss component. This staged transition reduces the impact of large gradients during early training and helps prevent convergence to degenerate solutions, such as constant predictions.

The second approach, referred to as Hybrid POT, utilizes a structured weight curriculum over 100 epochs. Unlike the Binary POT, this strategy builds upon the hybrid loss function defined in Equation 3, where the loss is expressed as a combination of MSE and CMAPSS components. While the baseline hybrid model employs fixed weights ($\lambda_1 = \lambda_2 = 1$), the Hybrid POT introduces a time-dependent weighting schedule. The coefficients λ_1 and λ_2 are adjusted every 20 epochs in a stepwise manner: the MSE weight decreases ($1.0 \rightarrow 0.75 \rightarrow 0.5 \rightarrow 0.25 \rightarrow 0.0$), while the CMAPSS weight increases proportionally ($0.0 \rightarrow 0.25 \rightarrow 0.5 \rightarrow 0.75 \rightarrow 1.0$). This progressive transition allows the optimizer to gradually shift from accuracy-driven learning to risk-aware optimization, reducing instability while preserving the structural guidance provided by the MSE objective during early training.

4.4 Dynamic Loss Balancing (DLB)

To address the limitations of static weighting strategies, we propose a Dynamic Loss Balancing mechanism that adaptively adjusts the contribution of the MSE and CMAPSS loss components throughout training. In contrast to the static (λ_1, λ_2) coefficients used in the hybrid and POT strategies, DLB introduces dynamically updated weights ($w_{\text{mse}}, w_{\text{cmapss}}$) that are adjusted at each epoch based on the observed training dynamics. The method is implemented as a lightweight callback within the training loop, operating without requiring modifications to the optimization process.

At the end of each epoch, the current values of both components are retrieved from the training logs. To reduce sensitivity to short-term fluctuations, the method maintains an EMA of each loss. Given a smoothing factor $\alpha = 0.7$, the EMA is updated as described in Equation 4.

$$\text{EMA}_t = \alpha \cdot \text{EMA}_{t-1} + (1 - \alpha) \cdot L_t \quad (4)$$

To account for scale differences, each EMA is normalized against its initial value at the first epoch, denoted as $\tilde{L}_t^{\text{mse}}$ and $\tilde{L}_t^{\text{cmapss}}$. The relative importance of each objective is then determined inversely proportional to its normalized magnitude, updating the weights as shown in Equation 5:

$$w_{\text{mse}}^* = \frac{\tilde{L}_t^{\text{cmapss}}}{\tilde{L}_t^{\text{mse}} + \tilde{L}_t^{\text{cmapss}}}, \quad w_{\text{cmapss}}^* = \frac{\tilde{L}_t^{\text{mse}}}{\tilde{L}_t^{\text{mse}} + \tilde{L}_t^{\text{cmapss}}} \quad (5)$$

This formulation increases the weight of the loss component that exhibits slower relative improvement, promoting a more balanced optimization between objectives.

To avoid instability and ensure that neither objective is neglected, the weights are constrained to the interval $[0.2, 0.8]$. Furthermore, instead of applying abrupt

changes, the weights are updated smoothly using a momentum-like rule controlled by a factor $\beta = 0.3$, as defined in Equation 6.

$$w_t = (1 - \beta) \cdot w_{t-1} + \beta \cdot w_t^* \quad (6)$$

Finally, the complementary constraint $w_{\text{cmapss}} = 1 - w_{\text{mse}}$ is enforced to maintain a combination of the two objectives.

The DLB mechanism continuously adapts the loss weights during training, enabling a dynamic trade-off between accuracy and safety-oriented objectives while maintaining training stability.

4.5 Experimental Setup

Experiments are conducted on the CMAPSS benchmark dataset [8], which comprises four subsets (FD001–FD004) with increasing levels of operational complexity. FD001 and FD003 correspond to single operating conditions, while FD002 and FD004 include multiple regimes and fault modes, resulting in higher variability.

All models are trained for 100 epochs using the Adam optimizer. Hyperparameters such as learning rate and batch size are selected empirically for each configuration to ensure stable training. A validation split of 20% is used. For POT and DLB strategies, weight updates are applied at the epoch level following the procedures described in Section 4.

Performance is evaluated using Root Mean Squared Error (RMSE) for predictive accuracy and the CMAPSS score for safety-oriented assessment. All experiments share the same architecture and preprocessing pipeline, differing only in the loss formulation and weighting strategy.

5 Results

This section presents the evaluation of the proposed training strategies across the CMAPSS dataset. It begins by establishing a baseline using standard MSE optimization. Subsequently, it analyzes the instability introduced by direct asymmetric loss optimization, evaluates the stabilizing effects of the POT and DLB frameworks, and concludes with a comparative analysis against state-of-the-art approaches.

5.1 Baseline Performance

To establish a reference point, the Transformer model was first trained using the standard Mean Squared Error (MSE) loss. Table 2 summarizes the performance obtained using this configuration. As expected, the model achieves consistent and stable convergence across all datasets, with lower error values observed in the simpler single-regime scenarios (FD001 and FD003) and higher errors in the more complex multi-regime datasets (FD002 and FD004).

Table 2. Study of proposed loss functions and weighting strategies across CMAPSS datasets.

Training Strategy	RMSE				CMAPSS Score			
	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
Standard MSE (Baseline)	14.55	18.05	13.90	23.35	517.26	2,243.00	1,256.09	5,726.00
CMAPSS Loss	13.09	87.75	12.46	44.74	259.87	1,409,215.00	293.25	18,434.00
Hybrid Loss	13.99	80.84	12.42	80.39	399.85	841,619.00	298.12	679,277.00
Binary POT	–	21.08	–	23.05	–	1,578.00	–	4,737.00
Hybrid POT	12.95	18.52	13.77	25.98	307.05	1,650.00	405.55	6,745.50
Dynamic Loss Balancing	13.48	20.19	15.85	24.99	311.02	3,718.23	1,114.52	7,936.91

Despite its stable convergence and competitive RMSE values, the MSE-based model does not explicitly account for the asymmetric risk associated with prediction errors. Late predictions are penalized equally to early predictions, which is misaligned with safety-critical requirements. This limitation motivates the exploration of alternative formulations.

5.2 Effect of Loss Functions

To evaluate the impact of incorporating asymmetric risk, the model was trained using the CMAPSS loss in isolation, as well as the static hybrid loss defined in Equation 3. The results, presented in Table 2, highlight a clear trade-off between predictive stability and risk-aware optimization.

While the CMAPSS loss improves safety-oriented performance in simpler datasets, it introduces severe instability in complex regimes. In FD002, the model collapses during training, failing to learn meaningful degradation patterns (Fig. 1). FD004 follows a similar pattern. The hybrid loss partially mitigates this issue by incorporating the stabilizing effect of MSE. However, the model still exhibits unstable behavior in complex datasets, often converging to systematically underestimated predictions (Fig. 2).

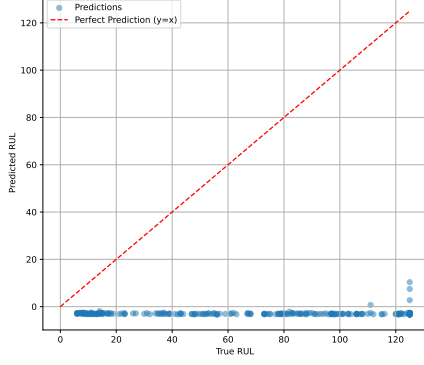
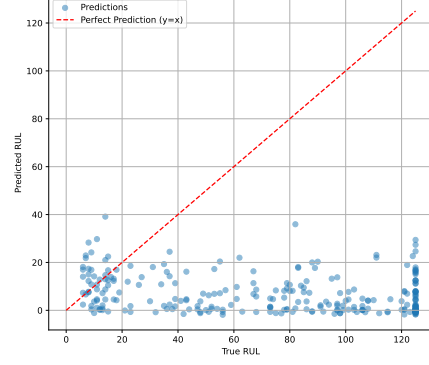
These findings confirm that directly optimizing asymmetric exponential losses is challenging in high-variance settings, motivating the structured training strategies explored next.

5.3 Phased Objective Transition (POT)

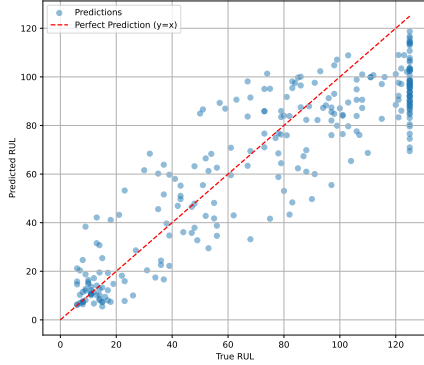
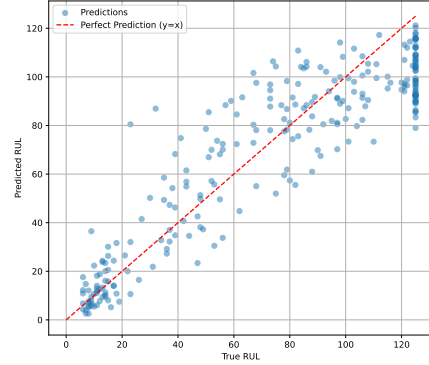
To mitigate the instability observed when optimizing asymmetric loss functions, the Binary POT and Hybrid POT strategies were evaluated (Table 2).

The Binary POT strategy, designed specifically to address instability in complex datasets, successfully prevents the collapse observed in direct CMAPSS optimization (Fig. 3). In both FD002 and FD004, the model converges to stable solutions. However, this stabilization comes at the cost of limited improvement in the CMAPSS score.

The Hybrid POT strategy extends this idea by introducing a gradual transition between objectives (Fig. 4). This approach yields competitive performance

**Fig. 1.** FD002 CMAPSS Loss**Fig. 2.** FD002 Hybrid Loss

in simpler datasets and achieves stable training in complex regimes, avoiding the collapse observed in previous static approaches.

**Fig. 3.** FD002 Binary POT**Fig. 4.** FD002 Hybrid POT

Overall, Phased Objective Transition is effective as a stabilization mechanism, particularly in high-variance scenarios. However, the predefined nature of the transition schedule limits its ability to adapt to dataset-specific dynamics.

5.4 Dynamic Loss Balancing (DLB)

To address the limitations of static schedules, the Dynamic Loss Balancing (DLB) mechanism was evaluated. As shown in Table 2, DLB adaptively adjusts the contribution of the MSE and CMAPSS components based on their relative evolution, successfully preventing severe collapse.

In the challenging FD002 scenario, the model produces stable predictions without collapsing (Fig. 5), confirming the effectiveness of the adaptive weighting mechanism. However, despite improved stability, the results indicate that DLB does not consistently lead to superior performance in terms of CMAPSS scores compared to simpler strategies. This suggests that balancing loss magnitudes alone may not fully align the optimization process with the asymmetric risk objective.

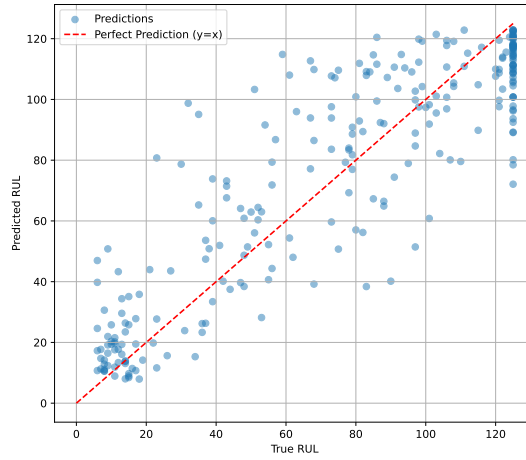


Fig. 5. Predictions for FD002 using Dynamic Loss Balancing.

5.5 Comparative Analysis and Discussion

The experimental results highlight a fundamental trade-off between optimization stability and the incorporation of asymmetric risk. Directly optimizing the CMAPSS loss introduces severe instability, and fixed-weight static hybrid combinations remain insufficient in high-variance settings.

Phased Objective Transition (POT) effectively prevents collapse, confirming that staged or progressive exposure to asymmetric objectives improves stability. Dynamic Loss Balancing (DLB) further ensures stable convergence across all datasets via autonomous adaptive weighting.

Table 3 provides a comprehensive comparison between the proposed training strategies and representative state-of-the-art methods. While recent approaches achieve superior performance through increased architectural complexity, our findings emphasize that beyond architectural innovation, effective objective scheduling and training strategies are essential for enabling robust and risk-aware RUL prediction.

Table 3. Comparison with state-of-the-art methods on the CMAPSS dataset.

Model (Author, Year)	Arch.	RMSE				CMAPSS Score			
		FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
Bi-LSTM (Wang et al., 2018)	RNN	13.65	23.18	13.74	24.86	295.00	4,130.00	317.00	5,430.00
Attention-LSTM (Boujanza et al., 2022)	RNN+Attn	13.95	17.65	12.72	20.21	320.00	2,102.00	223.00	3,100.00
Transformer + ARR (Kim et al., 2024)	Transformer	11.36	18.97	11.28	20.69	192.22	2,433.15	133.41	2,842.44
FFG-Transformer (Zhou et al., 2024)	Hybrid	11.36	13.24	11.80	15.16	173.89	714.98	215.32	1,049.53
AGATT (Zeng et al., 2023)	Graph+Transf.	11.45	13.87	10.88	16.43	204.00	323.00	205.00	632.00
PLO-Transformer (Zhang et al., 2025)	Hybrid	10.26	12.97	11.61	14.36	149.91	728.06	278.61	1,160.60
Proposed (MSE)	Transformer	14.55	18.05	13.90	23.35	517.26	2,243.00	1,256.09	5,726.00
Proposed (CMAPSS)	Transformer	13.09	87.75	12.46	44.74	259.87	1,409,215.00	293.25	18,434.00
Proposed (Hybrid)	Transformer	13.99	80.84	12.42	80.39	399.85	841,619.00	298.12	679,277.00
Proposed (Binary POT)	Transformer	–	21.08	–	23.05	–	1,578.00	–	4,737.00
Proposed (Hybrid POT)	Transformer	12.95	18.52	13.77	25.98	307.05	1,650.00	405.55	6,745.50
Proposed (DLB)	Transformer	13.48	20.19	15.85	24.99	311.02	3,718.23	1,114.52	7,936.91

6 Conclusion

This work investigated the challenges of incorporating asymmetric, safety-oriented objectives into Transformer-based models for Remaining Useful Life (RUL) prediction on the CMAPSS benchmark. The results demonstrate that while the CMAPSS loss better reflects operational risk, its direct optimization leads to severe instability in complex scenarios, often resulting in model collapse or degenerate solutions. Through a systematic evaluation, it was shown that hybrid loss formulations, Phased Objective Transition (POT), and Dynamic Loss Balancing (DLB) can effectively mitigate these issues by improving training stability. However, none of the proposed strategies consistently outperformed the baseline across all datasets, highlighting a persistent trade-off between predictive accuracy and risk-aware optimization. These findings emphasize that training strategy plays a critical role in enabling the practical use of asymmetric objectives in industrial prognostic systems.

Future work will focus on developing more principled approaches for integrating asymmetric risk into the optimization process. This includes exploring gradient-based balancing techniques, alternative loss formulations with improved numerical properties, and adaptive scheduling methods that account for dataset-specific dynamics. Additionally, extending the evaluation to real-world industrial datasets and incorporating uncertainty quantification may provide further insights into the deployment of risk-aware models in safety-critical applications.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. p. 41–48. ICML ’09, Association for Computing Machinery, New York, NY, USA (2009). <https://doi.org/10.1145/1553374.1553380>, <https://doi.org/10.1145/1553374.1553380>

2. Boujamza, A., Lissane Elhaq, S.: Attention-based lstm for remaining useful life estimation of aircraft engines. *IFAC-PapersOnLine* **55**(12), 450–455 (2022). <https://doi.org/https://doi.org/10.1016/j.ifacol.2022.07.353>, <https://www.sciencedirect.com/science/article/pii/S2405896322007546>, 14th IFAC Workshop on Adaptive and Learning Control Systems ALCOS 2022
3. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks (11 2017). <https://doi.org/10.48550/arXiv.1711.02257>
4. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press (2016), <http://www.deeplearningbook.org>
5. Heimes, F.: Recurrent neural networks for remaining useful life estimation. In: 2008 International Conference on Prognostics and Health Management. pp. 1 – 6 (11 2008). <https://doi.org/10.1109/PHM.2008.4711422>
6. Kim, G., Choi, J.G., Lim, S.: Using transformer and a reweighting technique to develop a remaining useful life estimation method for turbofan engines. *Engineering Applications of Artificial Intelligence* **133**, 108475 (2024). <https://doi.org/https://doi.org/10.1016/j.engappai.2024.108475>, <https://www.sciencedirect.com/science/article/pii/S095219762400633X>
7. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. *International Conference on Learning Representations* (12 2014)
8. Saxena, A., Goebel, K., Simon, D., Eklund, N.: Damage propagation modeling for aircraft engine run-to-failure simulation. *International Conference on Prognostics and Health Management* (10 2008). <https://doi.org/10.1109/PHM.2008.4711414>
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 30. Curran Associates, Inc. (2017)
10. Wang, J., Wen, G., Yang, S., qiang Liu, Y.: Remaining useful life estimation in prognostics using deep bidirectional lstm neural network. 2018 Prognostics and System Health Management Conference (PHM-Chongqing) pp. 1037–1042 (2018), <https://api.semanticscholar.org/CorpusID:57753485>
11. Zeng, M., Wu, F., Yiwei, C.: Remaining useful life prediction via spatio-temporal channels and transformer. *IEEE Sensors Journal* **PP**, 1–1 (12 2023). <https://doi.org/10.1109/JSEN.2023.3324330>
12. Zhang, E., Wu, L., Ren, Z., Gao, H., Liu, D., Wang, Q.: Plo-transformer-lstm: a remaining useful life prediction method for aircraft engines. *Engineering Research Express* **7** (08 2025). <https://doi.org/10.1088/2631-8695/adece8>
13. Zhou, Z., Long, Z., Wang, R., Bai, M., Liu, J., Yu, D.: An aircraft engine remaining useful life prediction method based on predictive vector angle minimization and feature fusion gate improved transformer model. *Journal of Manufacturing Systems* **76**, 567–584 (10 2024). <https://doi.org/10.1016/j.jmsy.2024.08.025>