Deep learning representations for prognostics and health management

Alaaeddine Chaoub

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Deep learning representations for prognostics and health management

THÈSE

présentée et soutenue publiquement le 10 Juillet 2024

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Alaaeddine Chaoub

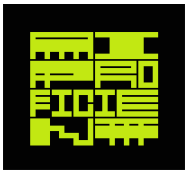
Composition du jury

Président :	Bernardetta Addis	Professor - Université de lorraine
Rapporteurs :	Emmanuel Ramasso	Associate professor HDR - Institut FEMTO-ST
	Céline Hudelot	Professor - Centralesupelec University of Paris-Saclay
Examinateurs :	Birgit Vogel-Heuser	Professor - Technische Universität München
	Raphaël Couturier	Professor - Université de Franche-Comté
Encadrants :	Christophe Cerisara	Researcher HDR (CR) - Université de Lorraine, LORIA
	Alexandre Voisin	Associate professor HDR - Université de Lorraine, CRAN

LORIA — UMR 7503, CRAN — UMR 7039

Mis en page avec la classe thesul.

Funding and resources



This work is part of the project AI-PROFICIENT which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 957391.

The Grid5000 computing resources have been used to partly train and evaluate the proposed models.

Remerciements

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Christophe Cerisara and Prof. Alexandre Voisin, for their unwavering support, insightful guidance, and the enriching discussions that significantly contributed to the successful completion of this thesis. Their encouragement and positive energy were invaluable throughout this journey.

My heartfelt thanks extend to Prof. Benoît Iung, Prof. Karen Fort, and Prof. Phuc DO for the wisdom they imparted and their invaluable guidance.

I wish to express my gratitude to the European Commission for their financial support during this period. I also want to send my thanks to all the partners of the AI-PROFICIENT project for their support and for the enjoyable times we shared during our numerous meetings.

I would also like to extend my special thanks to the PhD students, and researchers who shared this journey with me. Your collaboration, insightful discussions, and mutual support created a stimulating and inspiring research environment.

Lastly, I wish to convey my deepest gratitude to my family and friends. Their consistent encouragement, understanding, and emotional support provided the foundation upon which I could build and persevere.

Résumé

Cette thèse contribue à l'application de l'apprentissage Profond (Deep Learning) dans la prédiction de la durée de vie résiduelle (Remaining Useful Life) des équipements industriels, en traitant plusieurs défis importants. Notre recherche est motivée par des questions clés visant à développer des architectures et méthodes d'apprentissage profond pour prédire la fin de vie des équipements (RUL) sous diverses conditions operationelles, à améliorer l'interprétabilité des modèles et à faire face au manque de données en exploitant des données externes (non) labellisées. Nous avons structuré notre travail en deux parties principales.

Dans la première partie, nous explorons des architectures capables de gérer la variabilité des données résultant de différentes conditions opérationnelles, sans sélection des données d'entrée. Cela nous a mené à la proposition d'une architecture MLP-LSTM-MLP. En employant un MLP en entrée, nous avons pu normaliser les représentations, améliorant ainsi les performances de prédiction malgré plusieurs conditions opérationnelles. De plus, pour améliorer l'interprétabilité, nous avons proposé de remplacer ce premier étage de Multi Layer Perceptron (MLP) par un système de mélange d'experts (Gated mixture of experts (GMoE)), permettant une décomposition interprétable basée sur les conditions opérationnelles.

La seconde partie de la thèse aborde le problème de la rareté de données, un défi largement reconnu dans le domaine du *pronostic and health management* (Prognostics and health management (PHM)). En introduisant des adaptateurs, c'est-à-dire des couches spécifiques aux différentes tâches qui permettent le traitement de differentes structures d'entrée/sortie, nous avons proposé une approche d'entrainement auxiliaire qui exploite des données labellisées externes, présentant une méthode qui surpasse les techniques traditionnelles trouvées dans la littérature. De plus, pour exploiter les données externes non étiquetées pour l'apprentissage auxiliaire, nous avons proposé une approche de méta-apprentissage pour déterminer automatiquement les objectifs auxiliaires à partir de ces données en les pseudo-étiquetant d'une manière qui prend en compte la tâche principale. L'objectif de cette partie de la thèse était d'exploiter un spectre plus large de données disponibles pour améliorer la performance de la prédiction de la durée de vie résiduelle.

Après analyse de notre travail, nous avons également identifié certaines limites des approches que nous avons proposées et nous proposons des perspectives immédiates et à long terme pour la recherche future. Celles-ci incluent le fait de relever les défis du traitement des données séquentielles longues, d'améliorer davantage l'interprétabilité des modèles, de s'attaquer à la rareté de données avec des méthodologies de formation plus avancées, et d'explorer le potentiel de l'apprentissage fédéré et des grands modèles de langage dans les contextes industriels.

Mots-clés: Pronostic et management de la santé, prédiction de la durée de vie utile restante, Apprentissage profond, Interprétabilité, rareté des données, Apprentissage auxiliaire, Méta-apprentissage.

Abstract

This thesis contributes to the application of Deep Learning (DL) in Remaining useful life (RUL) prediction of industrial equipment, addressing significant challenges in this field. Our research is driven by the need to develop Deep Learning (DL) architectures that mitigate performance degradation under various operating conditions, to improve model interpretability, and to address data scarcity by leveraging external (un)labeled data. We structured our work into two principal parts.

In the first part, we explore architectures capable of handling data variability resulting from different operating conditions, without manual feature engineering. This led us to propose an MLP-LSTM-MLP architecture. By employing an MLP at the first stage, we were able to normalize this variability, thus improving performances under such settings. Furthermore, To enhance interpretability, we proposed to replaced the first-stage MLP stage with a Gated mixture of experts (GMoE) system, enabling interpretable decomposition based on operating conditions.

The second part of the thesis addresses the issue of data scarcity, a widely recognized challenge in the Prognostics and health management (PHM) field. Through the introduction of adapters, i.e. task-specific layers that address the challenge of handling multiple input/output structures, we proposed an auxiliary training approach that leverages external labeled data, presenting a method that surpasses traditional techniques found in the literature. Moreover, to utilize external unlabeled data in auxiliary training, We proposed a meta-learning approach to automatically derive auxiliary objectives from these data by pseudo-labeling them in an end-task aware manner. The goal of this part was to leverage broader spectrum of available data to improve Remaining useful life (RUL) prediction performances.

In reflecting upon our work, we acknowledge the limitations of the proposed approaches and suggest both immediate and long-term directions for future research. These include tackling the challenges of processing long sequence data, further improving model interpretability, addressing data scarcity with more advanced training methodologies, and exploring the potential of federated learning and large language models in industrial settings.

Keywords: Prognostics and health management (PHM), Remaining useful life (RUL) prediction, Deep Learning (DL), Interpretability, data scarcity, Auxiliary learning, Meta learning

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

AT Auxiliary training

CNN Convolutional Neural Network

CPS Cyber-Physical Systems

CV Computer Vision

DL Deep Learning

DTW Dynamic time warping

FSL Few-shot learning

GAN Generative adversarial networks

GMNN Gated modular neural network

GMoE Gated mixture of experts

GN Gating network

GRU Gated Recurrent Unit

GPU Graphics Processing Units

HI Health index

HS Health state

LIME Local interpretable model-agnostic explanations

LLM Large Language Models

LKM Large Knowledge Models

LSTM Long Short-Term Memory network

LRP Layer-Wise Relevance Propagation

MAXL Meta AuXiliary Learning

MLP Multi Layer Perceptron

MSE Mean square error

MAML Model-Agnostic Meta-Learning

MTL Multi-task learning

NLP Natural language processing

NMI Normalized mutual information

OC Operating condition

PHM Prognostics and health management

PT-FT PreTraining + FineTuning

PT-R-in-out Pre-training + retraining input and output layers

RMSE Root mean square error

RNN Recurrent Neural Network

RQ Research question

RTF Run to Failure

RUL Remaining useful life

SHAP Shapley Additive exPlanations

S-O-T-A State Of The Art

SSL Semi-supervised learning

TCN Temporal convulotional network

TL Transfer learning

VAE Variational autoencoder

XAI Explainable AI

Résumé en français (French summary)

Le développement industriel a traversé plusieurs révolutions marquantes, transformant profondément les méthodes de fabrication et de production. Cette évolution est principalement tirée par la nécessité d'adaptation rapide aux exigences changeantes du marché, notamment la personnalisation de masse (Schwab, 2017). Dès la première révolution industrielle, avec l'introduction de la machine à vapeur et la construction de chemins de fer, jusqu'à la révolution numérique caractérisée par les avancées des semi-conducteurs et de l'internet, chaque étape a apporté des innovations notables.

Actuellement, nous évoluons dans la quatrième révolution industrielle, ou Industrie 4.0, propulsée par des progrès significatifs dans les technologies de l'information et de la communication (Rojko, 2017). Cette ère est caractérisée par des avantages multiples tels que l'accélération du temps de mise sur le marché, l'amélioration de la réactivité aux besoins des clients et la possibilité de réaliser une production de masse personnalisée sans accroître de manière excessive les coûts de production. Elle promeut également un environnement de travail plus adaptable et favorise une gestion plus rationnelle des ressources naturelles et de l'énergie.

Dans ce cadre, les stratégies de maintenance et de fiabilité des systèmes ont continuellement évolué pour mieux répondre aux besoins de rentabilité et de spécialisation (Ran et al., 2019; Achouch et al., 2022). Ces demandes émergent souvent de nécessités techniques telles que la minimisation des temps d'arrêt des machines et la maximisation de la durée de vie des composants. L'effet en cascade de ces exigences s'étend aux considérations économiques, notamment la réduction des coûts de production et de maintenance, la garantie de la qualité et de la fiabilité des produits, et la sauvegarde des actifs et des services.

Le pronostic et la gestion de la santé (PHM) est un paradigme de l'industrie 4.0 et sert de catalyseur clé pour sa concrétisation, offrant une vue intégrée de l'état de santé des machines ou des systèmes. PHM met en œuvre un processus cyclique de surveillance, d'analyse et d'intervention pour maintenir la santé du système à des niveaux optimaux. Cette thèse se concentre sur la phase d'analyse, incluant l'évaluation de l'état, le diagnostic et le pronostic, avec un intérêt particulier pour ce dernier. Le pronostic de l'état de santé consiste à prédire

l'évolution future de l'état de santé du système. Plus spécifiquement, ce travail se concentre sur le pronostic en tant que prédiction de la durée de vie utile résiduelle (RUL), i.e. temps restant avant que le système ne soit pas en mesure d'assurer ses fonctions correctement, indispensable à la maintenance prédictive, permettant ainsi de prévenir les défaillances et d'anticiper les besoins du système.

Les méthodes de prédiction de la durée de vie utile résiduelle (RUL) peuvent être classifiées en trois grandes catégories : les approches basées sur des modèles physiques, les méthodes fondées sur les données, et les techniques hybrides. Notre recherche se concentre sur les méthodes basées sur les données, et plus spécifiquement sur les techniques d'apprentissage profond, qui ont montré un potentiel significatif pour cette tâche. Cependant, leur mise en œuvre effective dans des environnements industriels réels reste limitée par plusieurs défis.

Alors que nous approfondissons l'application de cette technologie, certaines limitations deviennent apparentes. Une gamme d'architectures avancées a été proposée dans la littérature pour la prédiction de la durée de vie utile résiduelle. Cependant, ces architectures sont souvent moins performantes en cas de fonctionnement sous différentes conditions opérationnelles. Une condition opérationnelle (OC) peut être définie comme les circonstances dans lesquelles un équipement fonctionne, et dans de nombreuses applications d'ingénierie, les OC changent avec l'environnement ou les modes de fonctionnement, entraînant souvent des divergences dans les données, ce qui nuit aux performances des approches de la littérature. Pour relever ce défi, notre question de recherche pour ce challenge est la suivante : "Quelle architecture d'apprentissage profond est capable de gérer une telle variabilité des données et de fournir des prédictions précises de la durée de vie restante ?"

Un autre problème important se pose lorsqu'on essaie d'utiliser des modèles d'apprentissage profond pour estimer la durée de vie résiduelle : la nature "boîte noire" de ces modèles, car les résultats de ces modèles seront utilisés pour prendre des décisions critiques et il est nécessaire de comprendre comment ils fonctionnent. Les chercheurs ont introduit plusieurs approches pour les interpréter, mais des recherches supplémentaires sont nécessaires. En conséquence, notre question de recherche est : "Comment les modèles d'apprentissage profond peuvent-ils être conçus pour être plus interprétables ?"

De plus, les modèles d'apprentissage profond sont généralement entraînés à l'aide de grandes quantités de données étiquetées et peuvent obtenir des résultats impressionnants sur un large éventail de tâches. Cependant, la construction de tels modèles pour la pronostic peut être difficile, notamment lorsque les données sont limitées à un cas d'utilisation spécifique. En effet, dans de nombreux cas, les données pour un équipement ou un système spécifique peuvent être rares pour

plusieurs raisons. Les équipements industriels sont souvent entretenus de manière préventive, ce qui réduit l'occurrence des pannes. De plus, bien qu'il puisse y avoir une grande quantité de données disponibles provenant de la surveillance du système, la plupart des données représentent le fonctionnement normal. Les états de dégradation et de pannes du système industriel, car ils conduisent à des produits indésirables, sont la plupart du temps largement sous-représentés.

En outre, malgré l'abondance de données connexes provenant de machines similaires ou différentes dans les usines, leur potentiel reste inexploité principalement parce qu'elles ne sont pas étiquetées, manquent de trajectoires complètes du cycle de vie ou ne sont pas directement applicables au développement de modèles, souvent en raison des pratiques de maintenance et des coûts élevés de l'étiquetage. Enfin, il est souvent impossible d'obtenir des données de "fonctionnement jusqu'à la panne" d'un processus "réel", où le système est autorisé à fonctionner jusqu'à ce qu'il tombe en panne, car cela est coûteux et prend du temps (Eker et al., 2012). Contrairement à des domaines tels que la vision par ordinateur ou le traitement du langage naturel où des modèles pré-entraînés sur de grands ensembles de données servent de base à de nouvelles applications, le pronostic manque de ces ressources fondamentales. Les ensembles de données disponibles sont fragmentés, provenant de divers types d'équipements avec des caractéristiques d'entrée différentes, avec des durées de vie différentes, et souvent orientés vers la recherche. Cette dispersion crée un obstacle à l'avancement des applications de l'apprentissage profond dans le domaine du pronostic.

Étant donné ces paramètres, lorsqu'on doit développer un modèle pour un cas particulier, le processus doit souvent recommencer à zéro. Néanmoins, nous pouvons disposer de jeux de données externes étiquetés ou non étiquetés (publics ou privés) qui pourraient être exploités. Cette opportunité nous amène à cette question de recherche : "Comment mieux exploiter les données externes pour développer des modèles de prédiction de la RUL ?"

Pour répondre à ces questions de recherche et contribuer à l'adoption de cette technologie par les industriels, quatre contributions principales sont fournies dans cette thèse :

- Proposition d'un modèle d'apprentissage profond de bout en bout pour améliorer la prédiction de la durée de vie utile résiduelle.

- Proposition d'une méthode qui aide à l'interprétation des modèles avec un système de mélange d'experts.

- Proposition d'une approche de pronostic de la durée de vie restante par entraînement auxiliaire avec des données externes).

- Proposition d'une approche basée sur le méta-apprentissage auxiliaire, dont l'objectif est de créer des objectifs auxiliaires à partir d'ensembles de données hétérogènes non supervisés.

Cette thèse est organisée comme suit, autour des quatre contributions principales :

- Le chapitre 1 présente une introduction à l'industrie 4.0, au pronostic et à la gestion de la santé, ainsi qu'à l'apprentissage profond. En outre, les principaux problèmes liés à l'application de telles technologies pour la prédiction de la durée de vie résiduelle sont également discutés, ce qui permet d'identifier les questions de recherche abordées dans cette thèse.

- Le chapitre 2 présente la revue de la littérature relative aux questions de recherche identifiées dans l'application de l'apprentissage profond pour la prédiction de la durée de vie résiduelle. En particulier, les architectures utilisées pour cette tâche, l'interprétabilité de tels modèles, et leur utilisation dans des environnements avec des données limitées. L'état de l'art sur ces aspects permet de mettre en évidence les questions scientifiques.

- Le chapitre 3 propose une architecture MLP-LSTM-MLP (Figure 1 (a)) qui intègre un simple réseau de neurones multicouche (MLP) à l'étape initiale pour normaliser la variabilité des données d'entrée causée par les conditions opérationnelles. Il est suivi d'une couche avec des cellules à mémoire à long terme et à court terme (LSTM) pour capturer les dépendances temporelles et, enfin, d'un autre ensemble de couches MLP pour prédire la durée de vie résiduelle. Cette approche s'est révélée plus performante que les modèles proposés dans la littérature pour gérer les multiples conditions opérationnelles.

- Le chapitre 4 propose une approche active pour améliorer l'interprétabilité du modèle proposé dans le chapitre précédent. Les couches MLP initiales ont été remplacées par un système de mélange d'experts (GMoE) (Figure 1 (b)). Ce système décompose les données d'entrée en fonction des conditions opérationnelles, pour avoir un ensemble de paramètres spécifiques à chaque condition opérationnelle.

- Le chapitre 5 propose une approche d'entraînement auxiliaire qui exploite des données étiquetées externes grâce à l'utilisation d'adaptateurs - des couches spécifiques à une tâche, conçues pour gérer diverses structures d'entrée/sortie (Figure 1 (c)). Cette méthode a surpassé les techniques traditionnelles proposées dans la littérature en tirant des connaissances issues d'ensembles de données connexes.

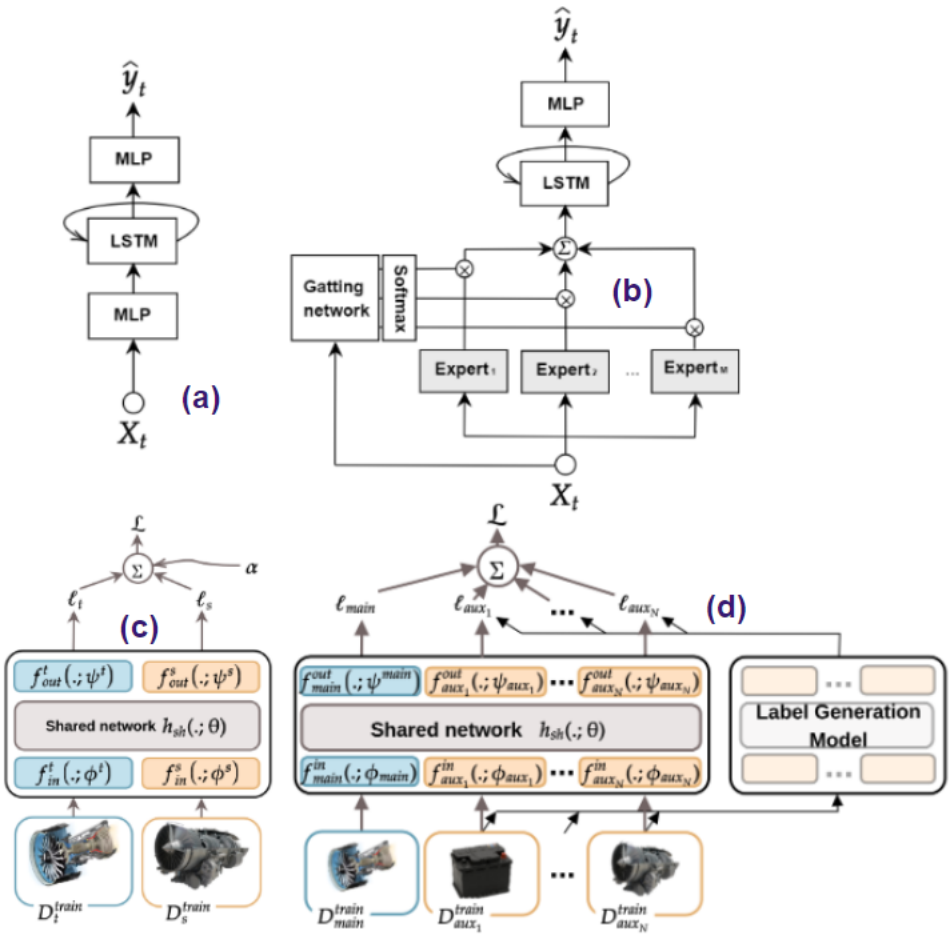


Figure 1: les approches proposées dans le cadre de cette thèse.(a) MLP-LSTM-MLP, (b) GMoE-LSTM-MLP, (c) Apprentissage auxiliaire, (d) Méta-apprentissage auxiliaire

- Le chapitre 6 traite le problème des données externes non étiquetées en proposant une approche basée sur le méta-apprentissage et l'entraînement auxiliaire. L'objectif est d'étiqueter les bases de données externes non étiquetées de manière à améliorer les performances sur cette tâche principale (Figure 1 (d)).

En réfléchissant à ce travail, certaines limites et pistes de recherche futures méritent d'être soulignées. L'architecture MLP-LSTM-MLP proposée, bien qu'efficace pour gérer la variabilité des données, rencontre des difficultés avec les séquences longues en raison des limitations inhérentes aux couches LSTM. Les recherches futures pourraient explorer des modèles de séquences plus avancés capables de gérer les dépendances à long terme. De plus, le domaine bénéficierait du développement de jeux de données plus étendus couvrant une gamme plus large de types d'équipements et de conditions opérationnelles. L'intégration de l'apprentissage fédéré et des grands modèles de langage dans les applications industrielles présente également une piste

prometteuse pour les recherches futures.

Introduction and Literature review

Chapter 1

Introduction

1.1 Industry 4.0 and Key Technologies

The trajectory of industrial development is characterized by significant revolutions that significantly transform manufacturing and production processes. This evolution (Figure 1.1) has been driven by global competition and the need for fast adaptation of production to the ever-changing market requests, i.e. mass customization. As outlined by (Schwab, 2017), the first industrial revolution spanned from about 1760 to around 1840. Triggered by the construction of railroads and the invention of the steam engine, it ushered in mechanical production. The second industrial revolution, which started in the late 19 th century and into the early 20th century, made mass production possible, fostered by the advent of electricity and the assembly line. The third industrial revolution began in the 1960s. It is usually called the computer or digital revolution because it was catalysed by the development of semiconductors, mainframe computing (1960s), personal computing (1970s and 80s) and the internet (1990s).

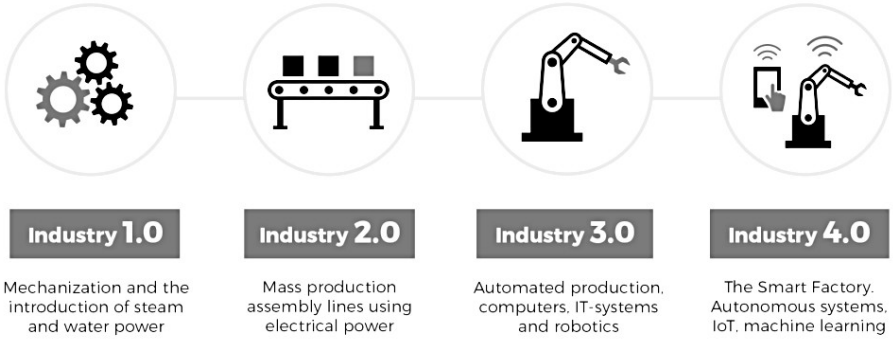


Figure 1.1: the four industrial revolutions

Currently, we are navigating the Fourth Industrial Revolution, colloquially known as Industry 4.0, a transformation ignited by advancements in Information and Communications Technologies (ICT) (Rojko, 2017). The definition of I4.0 depends on the application context and the research domain. The German digital association Bitkom reports that there are more than 100 relevant explanations for the term Industry 4.0 (Bauer et al., 2014). A more encompassing definition is offered by (Lu, 2017), which regards this concept as "the integration of complex physical machinery and devices with networked sensors and software, deployed to predict, control, and strategize for enhanced business and societal outcomes." Industry 4.0 has garnered considerable interest

among various stakeholders, primarily due to the multitude of benefits it offers. These advantages include facilitating time-to-market for new products, enhancing customer responsiveness, enabling custom mass production without a significant rise in overall production costs, fostering a more flexible and friendly working environment, and promoting more efficient usage of natural resources and energy (Rojko, 2017). Additionally, it has the potential to curtail production costs, logistic costs, and quality management costs by 10-30%, 10-30%, and 10-20%, respectively (Bauernhansl et al., 2016).

Industry 4.0 emerges not as a single, standalone entity, but as a complex network of interconnected technologies. It integrates a combination of established and emerging technologies within a complex architectural framework (Martinelli et al., 2021), such as:

- **Internet of Things (IoT):** This is a key component of smart factories, facilitating the interconnection of machines on the factory floor through IP-enabled sensors. The connection among these web-enabled devices allows for the collection, analysis, and exchange of large volumes of valuable data.
- **Cloud Computing:** A vital pillar of Industry 4.0, cloud computing enables the integration of diverse aspects of smart manufacturing, ranging from engineering and supply chain to sales, distribution, and service. Cloud computing accommodates the storage and efficient processing of vast quantities of data, potentially reducing startup costs for small and medium manufacturers by allowing them to scale their needs as they grow.
- **Artificial Intelligence (AI) and Machine Learning:** These technologies allow manufacturing companies to capitalize on the information generated across their business units, partners, and third-party sources. AI and machine learning can generate insights for improved visibility, predictability, and automation of operations and business processes. For example, predictive maintenance based on machine learning algorithms can enhance uptime and efficiency by using data collected from industrial machines.
- **Edge Computing:** For real-time production operations, data analysis needs to be performed at the "edge"—where data is created—to minimize latency. This is crucial when immediate response to equipment conditions is necessary.
- **Cybersecurity:** In the age of Industry 4.0, the interconnectedness that enables efficient manufacturing processes also creates new avenues for cyber threats. Therefore, a robust cybersecurity approach encompassing both IT and operational technology (OT) is imperative during digital transformation.
- **Digital Twins:** Enabled by Industry 4.0's digital transformation, digital twins are virtual replicas of processes, production lines, factories, and supply chains. Created by pulling data from IoT sensors, devices, and other internet-connected objects, digital twins can be used to enhance productivity, improve workflows, and develop new products.

While the principal components of I4.0 may vary depending on the research perspectives (Rojko, 2017; Dalmarco et al., 2019; B. Chen et al., 2017), there is a prevalent agreement centering around these foundational technologies. However, the significant progress made in these technological areas doesn't translate into an effortless implementation within industrial environments. The unique characteristics and inherent complexities of these settings underscore the necessity for ongoing research and innovation. The ability to effectively leverage these technologies hinges on the development of practical, tailored solutions that address the specific demands of industrial

applications, ensuring robustness and trustworthiness in their deployment. Existing models and algorithms, for instance, may not adequately account for the dynamics and variability within industrial settings, complicating the effective application of these technologies. This emphasizes the need for systems that are not only technically proficient but also resilient and reliable, fostering confidence among stakeholders in their adoption and use.

1.2 Prognostics and health management

Within the broader framework of Industry 4.0, the strategies for maintenance and system reliability have continually evolved (Ran et al., 2019; Achouch et al., 2022), responding to the increasing demands of profitability, specialization, and mass customization. This evolution towards more specialized maintenance strategies, such as predictive maintenance, is a paradigm of Industry 4.0 and serves as a key enabler for its realization. These demands often emerge from technical necessities such as minimizing machine downtime and maximizing the lifespan of components. The cascading effect of these requirements stretches into economic considerations, including reducing production and maintenance costs, ensuring the quality and reliability of products, and safeguarding assets and services.

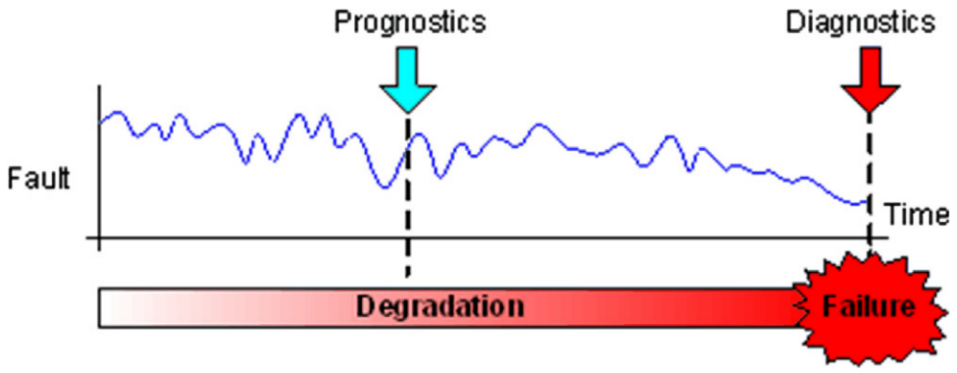


Figure 1.2: Diagnostics and Prognostics.(Jay Lee, F. Wu, et al., 2014)

PHM emerges as a pivotal discipline and is one of the key enabler of Industry 4.0 that aims to deliver an integrated perspective on the health status of machinery or an entire system.

At its core, PHM is a cyclical process, encapsulating the continuous nature of monitoring, analysis, and intervention to maintain system health at optimal levels. Illustrated in Figure 1.3, the PHM cycle typically involves three critical steps: observe, analyze, and act (Javed et al., 2013; Atamuradov et al., 2017).

- **Observe:** This phase includes data acquisition and data processing. During the data acquisition stage, an array of sensors collects a wealth of information about system conditions, including temperature, vibration, pressure, and more. Following this is the data processing stage. Depending on the complexity of the system and the subsequent analysis or modelling methods employed, data processing can be challenging and may require a significant amount of manual work and time. This stage involves organizing, cleaning, and preparing the collected data for the ensuing analysis which could include features selection and features engineering. The observe stage is critical for setting the foundation for effective PHM, as the accuracy and comprehensiveness of the data collected directly impact the subsequent analysis and decision-making processes.

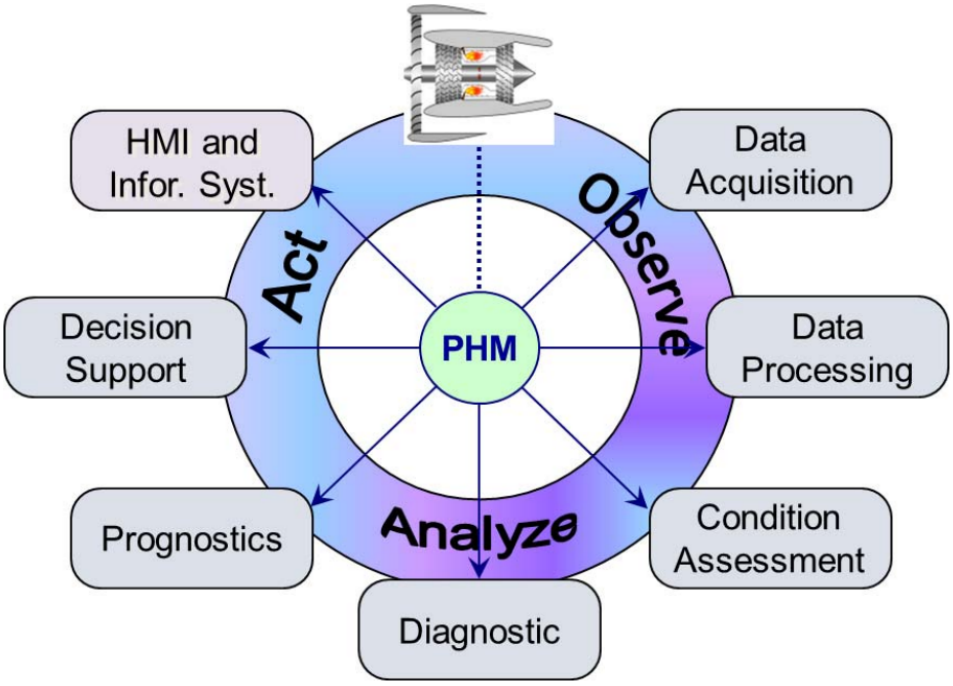


Figure 1.3: Prognostics and health management cycle (Javed et al., 2013)

- **Analyze:** This step involves condition assessment, diagnostics, and prognostics. Condition assessment is where the current status of the system or equipment is determined. Diagnostics (Figure 1.2), a component of PHM, can be summarized as the process of identifying and determining the relationship between cause and effect in that its function is to isolate faults and identify failure root causes. Prognostics (Figure 1.2), another key component, can be interpreted as the process of health prediction, which includes detecting incipient failure and predicting the Remaining useful life (RUL). The RUL, refers to the length of time a machine or system is expected to perform its intended function before it requires maintenance or replacement (Si et al., 2011; Youdao Wang et al., 2020). The RUL prediction is especially vital for strategic maintenance planning, allowing for the anticipation of system needs and the prevention of unexpected failures.

- **Act:** This stage integrates decision support systems and Human-Machine Interfaces (HMI), facilitating the execution of health management strategies. Health management represents a fundamental objective of PHM, defined as the process of implementing timely and appropriate maintenance actions. These actions are informed by the careful analysis of outputs from diagnostics and prognostics, consideration of available resources, and operational demands, with the goal of mitigating the impact of potential failures and minimizing associated losses (Jay Lee, F. Wu, et al., 2014). This process is enabled by decision support tools that analyse the outputs of previous PHM phases and by the integration of an intuitive HMI that further enhances this stage by providing a user-friendly interface for operators that simplify interactions with the PHM system.

Transitioning from the foundational aspects of Prognostics and Health Management (PHM) to the practicalities of its implementation within Industry 4.0, it becomes imperative to address

the evolution of industrial systems into more complex entities, often referred to as Cyber-Physical Systems (CPS). In these advanced systems, the integration of a vast array of sensors becomes crucial for gathering real-time data across multiple operational points, a development that significantly enhances the observability and controllability of industrial processes. Sensors, strategically embedded within these complex systems, are pivotal in acquiring a multidimensional understanding of a system's condition, covering aspects such as temperature, pressure, flow rate, vibration levels, and more. This approach to multi-sensor monitoring marks a considerable advancement in industrial operations, allowing for the creation of a highly comprehensive representation of the system's conditions. Such depth in monitoring is fundamental to the further development and effective application of PHM strategies (S. Cheng et al., 2010). The comprehensive data collected by these sensors provides the input for diagnosing current system states, prognosticating future conditions, and informing the management actions required to maintain system health. The ability to leverage data from various sensors ensures that PHM applications can offer more precise predictions and informed decisions, ultimately facilitating the proactive management of complex industrial systems.

However, this revolution in industrial monitoring is not without its challenges. The generated data from various types of sensors is profoundly heterogeneous and is captured at different time scales, which raises new issues concerning data management and interpretation. Furthermore, the relevance of each sensor's data varies across different types of faults, and the correlation between them is rarely straightforward, as pointed out by (Fink et al., 2020). In addition, as industries continue to grow and become more interconnected, we are faced with an expanding diversity of equipment, each with its unique operational dynamics and failure modes. This diversity adds a new layer of complexity in the application of Prognostics and Health Management solutions. Therefore, there is a pronounced need for dynamic, robust, and adaptive solutions that can maintain its applicability and adaptability across a wide array of cases, equipment or machinery.

1.3 Deep learning for RUL prediction

1.3.1 Deep learning

Deep Learning (DL), a specialized branch of machine learning, leverages artificial neural networks to analyze and learn from vast amounts of data. Comprising multiple layers of interconnected nodes, these neural networks are designed to process complex, high-dimensional information, making them exceptionally good at uncovering intricate, non-linear relationships hidden within data (Goodfellow et al., 2016). These capabilities are particularly advantageous in the context of Industry 4.0, where systems generate vast amounts of varied data that reflect the operational state of machinery. By automatically extracting meaningful features from raw data, DL offers a more adaptable and swiftly implementable solution across various types of machinery and equipment (Fink et al., 2020).

The evolution of DL has been primarily driven by several key factors: the availability of large data sets, advancements in computational power, the development of innovative network architectures, and the success of multiple research in this field. A notable catalyst in this evolution was the ImageNet dataset (J. Deng et al., 2009), introduced in 2009 and rapidly evolving into the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015). This challenge was instrumental in demonstrating the effectiveness of deep learning, in 2012 (Krizhevsky et al., 2012) proposed a deep convolutional neural networks that performed 41% better than the next best competitor, highlighting that deep learning was a viable strategy for machine learning and arguably triggering the explosion of deep learning in ML research. It also allowed a breakthrough

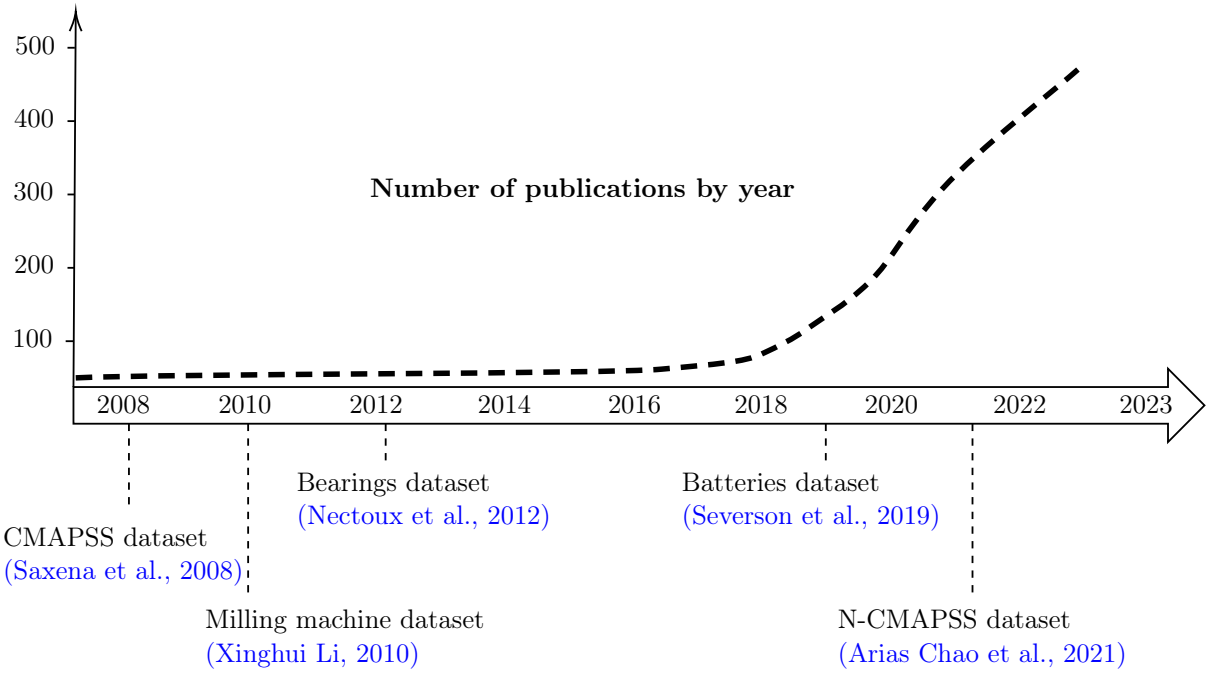


Figure 1.4: Examples of open source data sets for RUL prediction with a rough estimate of the number of papers having DL and RUL prediction in their abstract (numbers sourced from Web of Science (Clarivate, 2023)).

of similar importance in transfer learning: researchers soon realized that the weights learned in state of the art models for ImageNet could be used to initialize models for completely other datasets and improve performance significantly (Donahue et al., 2014). Similarly, 2018 marked the "ImageNet moment" for Natural language processing (NLP), as attention-based networks appeared (Vaswani et al., 2017) and the pre-training of large models (Devlin et al., 2018; Radford et al., 2018) began successfully achieving new state-of-the-art results on multiple natural language processing tasks. The collective impact of these research efforts cannot be overstated; by showcasing the success of DL and demonstrating the reduced need for large datasets for downstream tasks, they have significantly contributed to its widespread adoption.

1.3.2 The rise of DL for RUL prediction

The realm of RUL prediction is broadly split into two categories: physics model-based methods and data-driven methods (Sheppard et al., 2008; Jay Lee, F. Wu, et al., 2014; Lei et al., 2018). The former relies on understanding the core principles of the equipment, including failure mechanisms, and is notably accurate at the component level. However, it struggles with complex systems due to the intricate interactions within these systems that are challenging to capture with physical models. Furthermore, these methods must be tailored for each specific system and degradation, limiting their applicability. On the other hand, DL methods are focused on extracting insights from empirical data. DL has proven to be highly effective for tasks like RUL prediction while having a lower technical barrier compared to former approaches (Liangwei Zhang et al., 2019).

Figure (1.4) presents an estimate of the annual number of publications featuring both DL

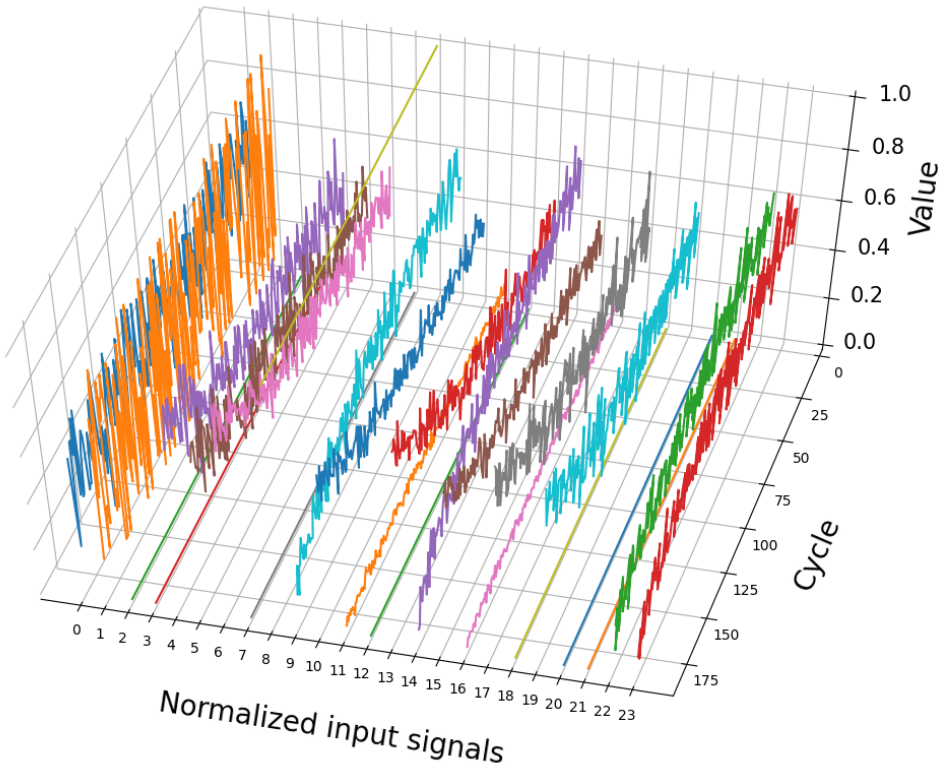


Figure 1.5: Example of a Run to failure trajectory of a Turbofan engine. This figure shows readings from 24 sensors throughout the engine's life, from the beginning (cycle = 0) to the end of its useful life. Each cycle represents a single flight data, providing a visual representation of the engine's usage over time. The variability in sensor readings highlights the complexity of predicting the engine's Remaining Useful Life visually.

and RUL in their abstracts, indicating a growing trend in this research domain. This surge is attributed to the increasing availability of public data sets essential for training DL models (Fink et al., 2020). Comprehensive literature reviews underscore DL's transformative role in prognostics, introducing diverse RUL prediction models and catering to various equipment types (Liangwei Zhang et al., 2019; Lei et al., 2018; Rezaeianjouybari and Shang, 2020).

1.3.3 Data for RUL prediction

As mentioned before, one of the key factors of success of DL is the availability of large data sets. In the context of RUL prediction, the collection of Run to Failure (RTF) trajectories is essential. These trajectories are detailed records that capture the operational data of equipment, spanning from the point of a maintenance operation to the system's failure. Alternatively, RTF trajectories can also be understood as a set of time series of Condition Monitoring data, each paired with corresponding time-to-failure labels (Arias Chao et al., 2021). These trajectories typically includes readings from various sensors (Figure 1.5), such as temperatures, vibration levels, and others. And it can be very heterogeneous, gathered at different time scales, representing different operating conditions and noisy due to measurements and transmission (Fink et al., 2020).

In this context, the labels, referred to as RUL values, are expressed as a linear countdown

of cycles or time remaining until failure (Lei et al., 2018). However, it's important to note that often equipment operates under normal conditions for a significant period before any signs of degradation begin. To accurately reflect such scenarios, the linear countdown can be adjusted or 'cropped' (G. Hou et al., 2020; H. Li et al., 2020; C. Zhao et al., 2022) to only account for the period when the equipment begins to show measurable signs of wear or decline, aligning the values RUL more closely with the actual onset of deterioration. The role of the DL models is to map input data to the remaining time until failure by leveraging historical data to make these predictions as accurate as possible.

To provide additional context on the extent of data available for analysis, it's important to highlight that the scope often extends beyond individual pieces of equipment or units, it's common to have a network of various systems and machines, each equipped with their corresponding sensors that continuously gather operational data. This creates a scenario where, alongside the targeted data from a specific equipment, we also have the opportunity to access other ones.

1.3.4 Challenges

DL as shown is a promising approach in the realm of RUL prediction, however, its widespread adoption and application in real industrial settings have been limited. This can be attributed to various constraints and challenges inherent to the industrial environment, for example:

Size of data sets: in many use cases, RTF data for specific equipment or systems may be scarce for several reasons. Most of industrial system are quite reliable by design. Furthermore, these system are often maintained in a preventive manner which reduces the occurrence of failures (Fink et al., 2020; Gay et al., 2022; Jiaxian Chen et al., 2023). Additionally, even though there may be a large amount of data available from monitoring of the system, most of the data represents normal operation and doesn't include features representative of the degradation. The degraded and failure states of the industrial system, as they lead to unwanted product are most of the time largely under represented. Also, the labeling of the data, or identifying which data corresponds to failures, is a time-consuming manual process that requires reviewing maintenance reports and expert knowledge (Liangwei Zhang et al., 2019). Finally, it is often not possible to obtain "run-to-failure" data from a "real" process, where the system is allowed to run until it fails, because it's costly and time-consuming (Eker et al., 2012).

Heterogeneity of available data sets: As mentioned before, we could have access to data sets from other equipment from the same factory or others. However, these data sets can be very heterogeneous in terms of operating conditions, inputs features, output values, which represent a bottleneck for leveraging this knowledge, for instance, both the CMAPSS (Saxena et al., 2008) and the N-CMAPSS data set (Arias Chao et al., 2021) represent RUN-to-failure trajectories of turbofan engines, but the input features, operating conditions and fault modes are different.

Absence of Pre-trained models: One of the possible solutions that is exploited in other fields such as NLP and computer vision is to fine tune large pre-trained models on the target task in hand. This reduces the need for large data sets in new application contexts. However in the industrial context, there is no large pre-trained model, and this can be attributed to companies not sharing their data, and to the fact that the open source data sets suffer from heterogeneity and are tremendously smaller compared to the ones used in other fields. While Several studies have employed transfer learning (Moradi and Groth, 2020; F. Yang et al., 2020), pre-training often relies on smaller datasets compared to those in other domains. Alternatively, some studies leverage pre-trained models from different fields, provided their data adhere to the required domain. An example of this approach is seen in the work by (El Hachem et al., 2021), where pre-trained Convolutional Neural Network (CNN) was applied for industrial quality control.

Interpretability of DL models: While developing performing models is still a major challenge in real world RUL prediction cases, interpretability of such models adds another layer of complexity and constrains the adoption and investment for these approach. Acting based on black-box models can raise a lot of concerns for users and domain experts about transparency and trust (Fink et al., 2020), also, about ethical questions towards users responsibilities.

1.4 Objectives and research questions

The majority of recent research on DL for RUL prediction has focused on the development of advanced DL architectures. Yet, the current research landscape (performances on benchmarks) and the practical demands indicates that the enhancements from intricate architectural designs are minimal compared to the impacts of data quality and algorithmic learning strategies. Recognizing the practical challenges identified and current trends in the literature, we choose in this thesis to focus on training algorithms, utilizing straightforward DL architectures.

1.4.1 Objectives

In this thesis, we focus on proposing approaches that could facilitate and accelerate the adoption of deep learning for RUL prediction. Our work is structured under two distinct scenarios:

- **Deep learning models development in data-rich environments:**

aim:

  - Develop an end-to-end deep learning model (cf. Chapter 3) for accurate RUL prediction across various operating conditions (a challenge highlighted in chapter 2, section 2.2).
  - Propose a strategy for improving interpretability of deep learning models for RUL prediction. (cf. Chapter 4).

- **Overcoming Data Scarcity using auxiliary data-sets:**

aim:

  - Propose an approach to leverage, integrate external data sources into deep learning models, which may include heterogeneous (cf. Chapter 5), unlabeled data from diverse industrial machines (cf. Chapter 6).

1.4.2 Research questions

This thesis addresses the following Research question (RQ):

(A) Deep learning approaches for RUL prediction: State of the Art

- What are the DL models applied to RUL prediction? what are their documented performances and limitations ?
- How are deep learning models for RUL prediction currently interpreted or explained? Are traditional Explainable AI (XAI)/interpretability methods used in industrial contexts? Are they adapted and useful in these contexts?

- What strategies are employed to address data scarcity in RUL prediction using deep learning? What are the primary challenges and issues identified in applying these approaches to real-world scenarios?

(B) End-to-End deep learning models and more interpretability for RUL prediction in data rich environment.

In the current literature, common practice in deep learning applications for RUL prediction involves pre-processing steps such as feature selection or engineering prior to model input. While this can improve model performance, it can also limit the model's ability to learn complex patterns autonomously. In addition, this pre-processing often requires domain-specific expertise. Deep learning models are also often perceived as "black boxes", their decision-making processes lacking clarity. This lack of transparency can hinder their practical application in real-life scenarios, raising ethical and trust issues. To address these challenges, we propose an end-to-end deep learning model capable of handling data variability and producing reliable results without extensive pre-processing. In addition, we aim to improve the interpretability of this model, particularly with regard to its decision-making process, by aligning it with key industrial concepts such as operating conditions.

- Which deep learning architectures are best suited for end-to-end training, can handle data variability, and still deliver accurate RUL predictions?

- How can deep learning models be designed to provide interpretability that aligns with industrial concepts that are present in the data?

(C) External data to improve generalization and to help solve the problem of data scarcity

In real-world industrial settings, acquiring run-to-failure trajectories for specific equipment can be challenging due to the rarity of failures and operational constraints. A potential solution is utilizing data sets from different equipment or machines. However, common approaches in literature often face issues like over-fitting and struggle with integrating heterogeneous and unlabeled data sets, common in industrial contexts. This part addresses the challenge of effectively incorporating diverse data sets, including those that are unlabeled.

- How can external data be leveraged to develop RUL prediction models in a way that minimizes the risk of over fitting?

- How to leverage unlabeled and heterogeneous data sets to develop DL models for RUL prediction?

1.5 Organization of the manuscript

The organization of this thesis is presented in figure (1.6) After the general introduction (see chapter 1), we will answer the research questions of this study. Thus, Chapter 2 carries out an in-depth literature review to frame deep learning in the context of remaining useful life (RUL) prediction. The document is then divided into two parts. Part 1, comprising Chapters 3 and 4, investigates deep learning applications in data-rich environments. Chapter 3 presents a new

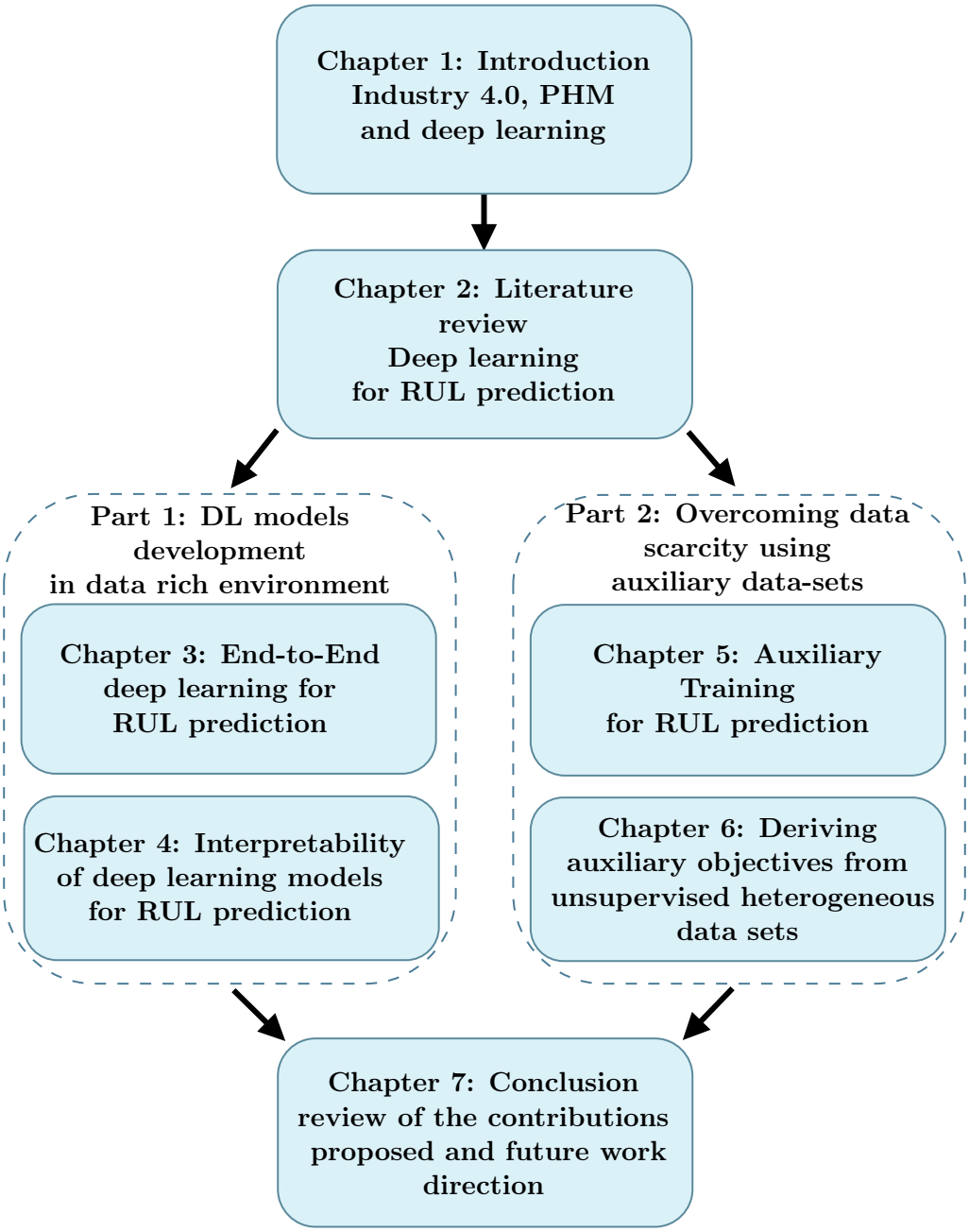


Figure 1.6: General diagram of the organization of the manuscript

deep learning architecture designed for RUL prediction, which is trained end-to-end to address data variability. Chapter 4 proposes the development of deep learning models that are inherently interpretable. Part 2, comprising Chapters 5 and 6, addresses the issue of data sparsity. Chapter 5 presents auxiliary training methods for enhancing RUL prediction, while Chapter 6 explores a methodology for deriving auxiliary objectives from unlabeled and heterogeneous data sets, with the aim of improving RUL predictive capability. Finally, chapter 5 synthesizes the research contributions, reviews the proposed methodologies and defines future work directions.

1.6 Publications

This section presents the scientific articles published or under review.

- Chaoub, A., Voisin, A., Cerisara, C., & Iung, B. (2021). Learning Representations with End-to-End Models for Improved Remaining Useful Life Prognostic. PHM Society European Conference, 6(1), 8. https://doi.org/10.36001/phme.2021.v6i1.2785

- A. Chaoub, C. Cerisara, A. Voisin and B. Iung, "Towards interpreting deep learning models for industry 4.0 with gated mixture of experts," 2022 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, 2022, pp. 1412-1416, doi: 10.23919/EUSIPCO55093.2022.9909884.

- Chaoub, A., Cerisara, C., Voisin, A., & Iung, B. (2022). Deep Learning Representation Pre-training for Industry 4.0. PHM Society European Conference, 7(1), 571–573. https://doi.org/10.36001/phme.2022.v7i1.2784

- Chaoub, A., Cerisara, C., Voisin, A., & Iung, B. (2024). Few-shot remaining useful life prognostics through auxiliary training with related data-set. Neural Computing and Applications, (under review)

- Chaoub, A., Cerisara, C., Voisin, A., & Iung, B. (2024). Enhancing RUL prediction with meta-learning: Deriving auxiliary objectives from unsupervised heterogeneous data sets. IEEE Transactions on Industrial Informatics, (under review)

Chapter 2

Literature Review

Contribution

This chapter focuses on examining recent advances in deep learning techniques in the context of prognostics, particularly in their application to RUL prediction. It critically assesses the contribution of these methods to the field, and highlights the challenges and basic research directions emerging from current advances.
Emphasizing the specificity of our work, the chapter refrains from general discussions of deep learning applications in other fields. Instead, it maintains a focused perspective, reserving broader comparative discussions for later chapters where our unique contributions are detailed. This approach ensures a focused review on RUL prediction, establishing a solid foundation for our methodologies introduced subsequently.

2.1 Introduction

The fast-growing field of PHM has seen a significant resurgence of interest in deep learning models, largely attributed to their power in handling complex, high-dimensional data. This chapter explores the complex?? relationship between deep learning and RUL prediction, a crucial aspect of PHM. To structure our talk, let us thus recall the three research questions proposed at the end of Chapter 1:

- **RQ1 :** What are the DL models applied to RUL prediction? what are their documented performances and limitations ?
- **RQ2 :** How are deep learning models for RUL prediction currently interpreted or explained? Are traditional XAI/interpretability methods used in industrial contexts? Are they adapted and useful in these contexts?
- **RQ3 :** What strategies are employed to address data scarcity in RUL prediction using deep learning? What are the primary challenges and issues identified in applying these approaches to real-world scenarios?

Recently, several review related to DL application in PHM have been published (Fink et al., 2020; Youdao Wang et al., 2020; Yangyang Zhang et al., 2023; Liangwei Zhang et al., 2019). These works provided a summary of the main prognostics approaches, the current trends in prognostics and diagnostics, and also discussed the challenges faced in this field. This chapter seek to enrich the material presented in the previous papers, by presenting the latest DL techniques

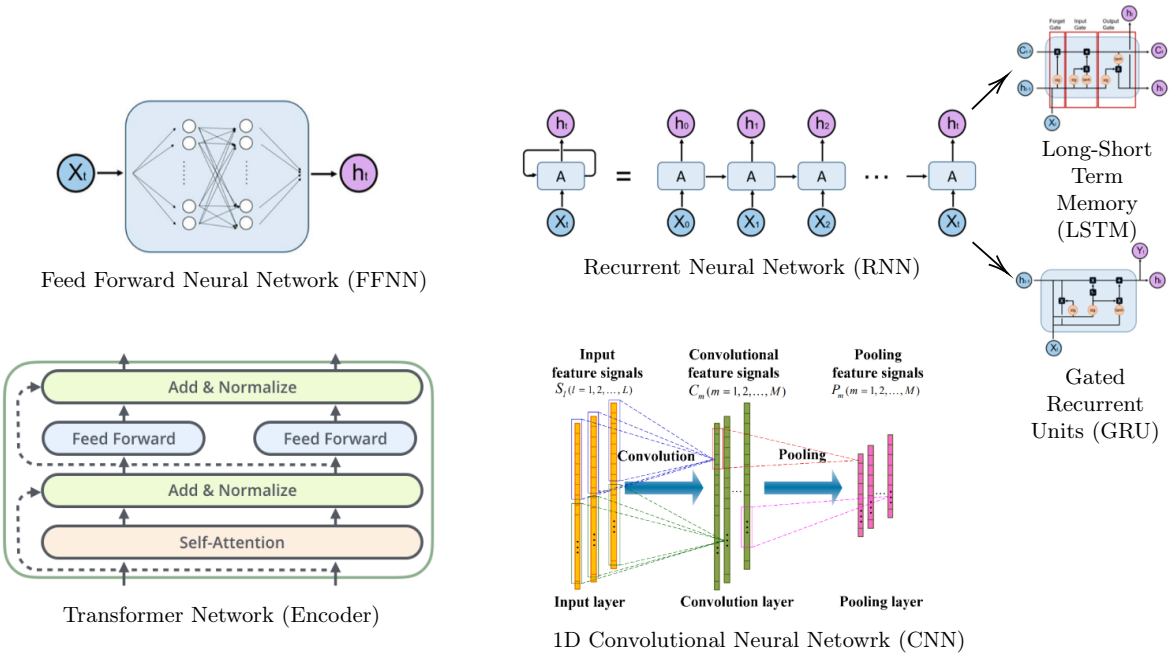


Figure 2.1: State-of-the-art neural network models used for RUL prediction: It displays the architectures of Feed Forward Neural Networks (FFNN), Recurrent Neural Networks (RNN) (Williams and Zipser, 1989), Long Short-Term Memory networks (LSTM) (Hochreiter and Schmidhuber, 1997), Gated Recurrent Units (GRU) (Cho et al., 2014), Convolutional Neural Netowrks and the Transformer Network (Encoder) (Vaswani et al., 2017).

and approaches used exclusively in RUL, offering fresh perspectives and insights to answer these research questions.

In pursuit of this goal, the chapter is structured as follows: Section 2.2 presents a comprehensive overview of the current architectures utilized for RUL prediction. Section 2.3 explores the interpretability of prognostic DL models in the industrial context. Section 2.4 examines the strategies employed to confront the challenge of data scarcity in this field. Section 2.5 introduces the datasets employed in this thesis for model development and evaluation. Finally, Section 2.6 synthesizes the key findings and insights from the chapter, setting the stage for the subsequent chapters of this thesis.

2.2 Deep Learning Architectures for RUL Prediction

In the evolving landscape of predictive maintenance, deep learning architectures have emerged as pivotal tools for RUL prediction, offering sophisticated approaches to deciphering the complexities of equipment health and usage data. This section critically analyzes various deep learning models that have been effectively employed for RUL prediction (Figure 2.1). The discussion encompasses an overview of the architectures used in the literature (Table 2.1), their empirical performance on established benchmark data-sets, and their limitations.

In the initial exploration of neural network applications for RUL prediction, (Heimes, 2008) developed an Recurrent Neural Network (RNN) based approach, followed by a foundational study by (Sateesh Babu et al., 2016) in 2016 undertook a detailed examination of various neural

Table 2.1: Examples of papers from the literature.

Base Architecture	**Papers**
MLP	(Sateesh Babu et al., 2016)
CNN/CapsNets/TCN	(D. Huang et al., 2022)(Sateesh Babu et al., 2016)(C. Cheng et al., 2020) (Ruiz-Tagle Palazuelos et al., 2020)(R. Jin, M. Wu, et al., 2022) (D. Zhou et al., 2020) (Jungan Chen et al., 2021) (Wenqiang et al., 2019)
RNN/LSTM/GRU	(Heimes, 2008)(Sateesh Babu et al., 2016)(Zheng et al., 2017) (Z. Liu et al., 2023) (C.-G. Huang et al., 2019) (Jinglong Chen et al., 2019) (Yuchen Song et al., 2018)(J. Zhou et al., 2022) (J. Wang et al., 2019)(Sayah et al., 2020)(R. Jin, Z. Chen, et al., 2022)
Attention mechanism	(Y. Chen et al., 2020)(D. Chen et al., 2022)(D. Xiao et al., 2022) (Zhizheng Zhang et al., 2022)
hybrid	(Yi'An et al., 2023)(Al-Dulaimi et al., 2019)(K. Zhao et al., 2023)(C. Zhao et al., 2021) (C. Zhao et al., 2022)(Z. Chen et al., 2020)(H. Liu et al., 2020)(Yan Song et al., 2020) (Ragab, Z. Chen, M. Wu, C.-K. Kwoh, et al., 2021)(Zeng et al., 2021)(X. Su et al., 2021) (L. Liu et al., 2022)

network architectures, encompassing MLP, RNN, and CNN. This work was instrumental in illustrating the efficacy of CNNs for RUL forecasting. By employing convolution and pooling layers over temporal sequences of multi-channel sensor data, which enabled the automatic learning of features from raw signals, marking CNNs as a powerful approach for the application.

Expanding upon these findings, (Zheng et al., 2017) implemented LSTM layers followed by MLP layers, which empirically demonstrated superior performance over the models scrutinized by (Sateesh Babu et al., 2016). The LSTM networks' ability to capture and process long-term dependencies in time series data significantly refined the accuracy of RUL predictions, thereby establishing a new benchmark for the quality of predictive models in the domain. Subsequent research by (Sayah et al., 2020) and (Z. Liu et al., 2023) adopted similar architectures, incorporating additional feature engineering techniques or applying them to diverse data sets, while (C.-G. Huang et al., 2019) and (R. Jin, Z. Chen, et al., 2022) adapted the architecture by integrating Bidirectional LSTM cells, aiming to assimilate temporal information from both directions over time. Further contributions in the realm of RUL prediction have been made through the application of Gated Recurrent Unit (GRU)-based models, as presented in works such as (Jinglong Chen et al., 2019), (Yuchen Song et al., 2018), (J. Zhou et al., 2022), and (J. Wang et al., 2019). GRUs are posited as an alternative to LSTMs, offering a reduction in parameter count without compromising the model's capacity to manage sequential data.

In an effort to address the computational demands and complexity associated with training LSTM networks, as well as to tackle the limitations of local feature representations inherent in CNNs, recent studies have turned to attention-based architectures. Works by (D. Xiao et al., 2022) and (Zhizheng Zhang et al., 2022) have proposed models leveraging the attention mechanism, renowned for its parallelization and adeptness in handling long sequences more efficiently than traditional LSTMs. Drawing inspiration from their success in NLP, attention mechanisms and Transformer architectures have gained traction for their capacity to capture global interdependencies between input and output sequences, thereby enhancing the model's ability to emphasize degradation patterns critical for RUL estimation. Additionally, the investigation into Temporal convulotional network (TCN)-based architectures, as delineated in (Jungan Chen et

al., 2021), presents an alternative methodology characterized by an extended receptive field. This feature is particularly advantageous for RUL prediction, as it allows for the model to integrate information over longer time spans, which can be crucial in capturing the progressive deterioration patterns of machinery. Furthermore, research such as that conducted by (C. Zhao et al., 2022) and (Ruiz-Tagle Palazuelos et al., 2020) has introduced capsule network-based approaches for RUL prediction. Capsule networks offer a solution to the challenge of spatial information loss that can occur with the pooling layers in CNNs. By preserving hierarchical relationships between different types of features, capsule networks have shown potential in enhancing model performance for RUL prediction.

As the field of neural network applications in RUL prediction continues to evolve, a notable trend is the development of hybrid architectures, designed to surmount the limitations inherent in single architecture systems. These innovative approaches capitalize on the synergy of combining different neural network layers, harnessing their collective strengths to create more effective predictive models.

A prime example of this approach is the work by (Al-Dulaimi et al., 2019), which employs a dual-path structure. This model features parallel paths, one utilizing LSTM and the other CNN, which then converge into a singular MLP that acts as the prediction head. The rationale behind this design is to simultaneously enhance the extraction of temporal and spatial features, thereby providing a more comprehensive analysis of equipment health data. Further emphasizing the benefits of hybrid models, (K. Zhao et al., 2023) demonstrated an architecture where a CNN first extracts deep features from the input data, which are subsequently processed by an LSTM network. This sequential processing enables the model to more effectively learn and represent potential degradation patterns. Empirical results from this study indicate a significant performance improvement over architectures that feed input data directly into an LSTM layer. Similarly, (Yi'An et al., 2023) introduced a TCN-Transformer architecture, leveraging the local feature extraction ability of convolutional neural network and the superiority of Transformer network in long time series information processing.

After outlining some of the various deep learning architectures, it is crucial to evaluate their performance in RUL prediction. To gain a thorough, fair and representative overview, we choose to use the widely used NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) turbofan engine data set (Saxena et al., 2008). There is a number of reasons for the choice of the C-MAPSS data set: C-MAPSS is recognized as the benchmark data set for RUL prognostics (Ramasso and Saxena, 2014b; Ramasso and Saxena, 2014a; Zschech et al., 2019; Vollert and Theissler, 2021). Furthermore, its properties allow to relate the findings to a variety of other use cases in research and industry: (a) the data set contains multivariate time series complemented by metadata with operational settings, (b) there are multiple fault modes and operating conditions, and (c) different lifespans and natural failures are considered. We argue that by using the C-MAPSS data set as an example domain, we derive general challenges that apply to RUL prediction.

While several other datasets exist for testing RUL prediction models (Hagmeyer et al., 2021), such as the bearings dataset (Nectoux et al., 2012) and the Batteries dataset (Severson et al., 2019), the C-MAPSS dataset has been more widely used as a benchmark in DL-oriented research, as it contains sufficient run-to-failure samples and was one of the first public data sets in this field (Eker et al., 2012).

Thus, our main focus is not on the data set, but on its properties and their implications for DL approaches.

Table 2.2 presents a comparative analysis of results from multiple papers, highlighting a notable trend: a performance decline is observed in models when confronted with multiple operating

Table 2.2: Performance of multiple research work on the C-MAPSS benchmark. The table show the RMSE on the 4 sub-data sets.

Models	DATA SETS			
	FD001	FD002	FD003	FD004
DCNN (Xiang Li et al., 2018)	13.86	18.91	13.55	23.32
DAG-CL (Jialin Li et al., 2019)	11.96	20.34	12.46	22.43
GA-RBM-LSTM (Ellefsen et al., 2019)	12.56	22.73	12.10	22.66
HDNN (Al-Dulaimi et al., 2019)	13.02	15.24	12.22	18.15
LSTM (Pasa et al., 2019)	16.5	18.1	15.9	17.2
DA-CNN (Yan Song et al., 2020)	11.78	16.95	11.56	18.23
DCGAN (G. Hou et al., 2020)	10.71	19.49	11.48	19.71
MS-DCNN (H. Li et al., 2020)	11.44	19.35	11.67	22.22
SM-BILSTM-ED (Yu et al., 2020)	13.58	19.59	19.16	22.15
N-LSTM-CA (L. Xiao et al., 2021)	12.51	17.72	11.84	20.43
DARNN (Zeng et al., 2021)	12.04	19.24	10.18	18.02
WR-IMT (Xing et al., 2021)	12.17	14.01	11.95	15.99
DEEP & ATTENTION (Yuanjun Liu and X. Wang, 2021)	12.98	17.04	11.88	19.54
BI-LSTM based attention (Yuefeng Liu et al., 2021)	13.78	15.94	14.36	16.96
ADLDNN (Xiang et al., 2022)	13.05	17.33	12.59	16.95
Attention based (L. Liu et al., 2022)	12.25	17.08	13.39	19.86
BiGRU-TSAM (J. Zhang et al., 2022)	12.56	18.94	12.45	20.47
Res-HSA (J. Zhu et al., 2023)	11.91	17.27	11.88	17.43
SARN (W. Xu et al., 2023)	13.84	18.92	11.68	17.74
MSTformer (D. Xu et al., 2023)	-	14.48	-	15.03
CNN-LSTM (K. Zhao et al., 2023)	12.69	14.15	13.04	15.78

conditions, specifically in the FD002 and FD004 scenarios. This draws attention to a persistent challenge in RUL prediction – the models' struggle to maintain accuracy across varying operating conditions. Attempts have been made to mitigate this issue through feature engineering techniques, such as normalization adjusted to these conditions, as discussed in (Pasa et al., 2019). However, these adjustments have not yielded significant improvements in performance, indicating a need for more robust solutions. Furthermore, a common practice in many of these models is the application of feature selection methods. While this approach can streamline the model and focus on the most salient features, it also carries the risk of discarding potentially valuable information. This selective process could inadvertently eliminate subtle but crucial indicators of equipment health, thereby impacting the model's overall predictive capability.

In response to the highlighted challenges, we proposed a model architecture capable of end-to-end training. This approach obviates the requirement for manual feature engineering or selection, while handling the variability associated with various operating conditions, demonstrating strong performance in these complex scenarios.

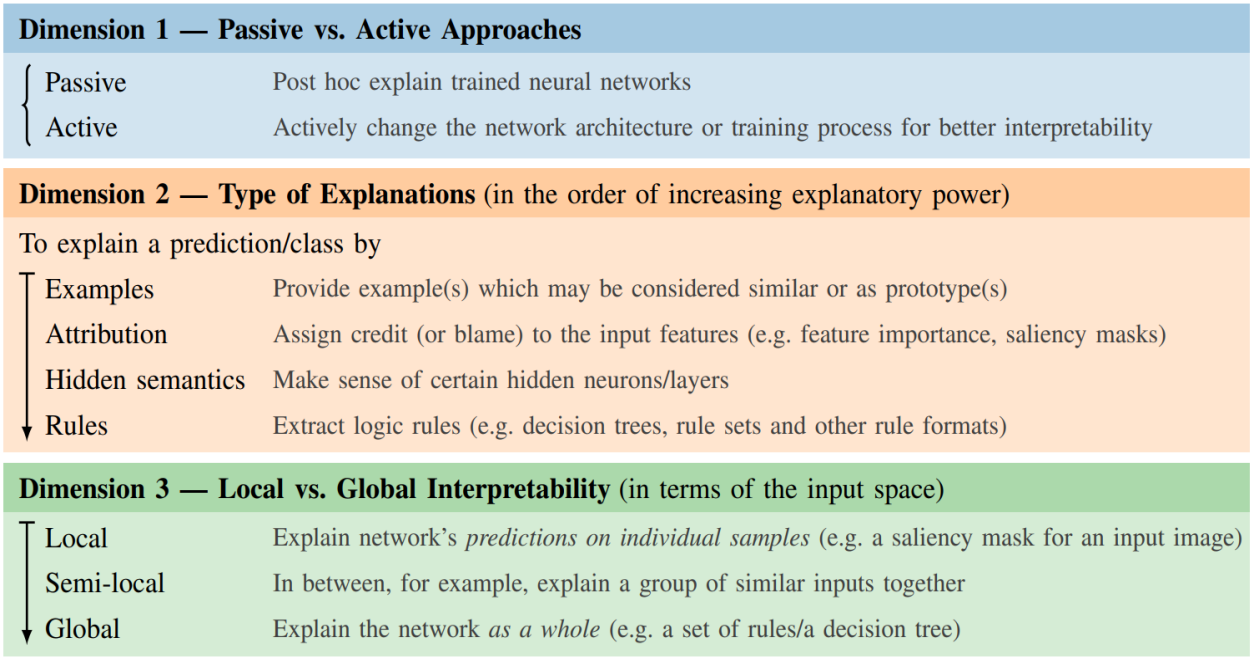


Figure 2.2: Interpretability approaches can be classified according to these three dimensions, the taxonomy being proposed by (Yu Zhang et al., 2021).

2.3 Interpretability of Deep Learning Models for RUL Prediction

As shown in the previous section, DL is getting more attention and is used more. however DL models are subject to an inherent drawback: they frequently operate in a black-box fashion. This opacity becomes a critical concern in industrial settings, where safety and reliability are paramount, and understanding the reasoning behind model predictions is not just beneficial, but can be a regulatory necessity (Hamon et al., 2022; Malgieri, 2019) if used for automated decision making. The intricate architecture of deep learning models, while powerful, often operates without offering clear insights into their decision-making processes. This lack of transparency can hinder their acceptance and trustworthiness, especially in scenarios involving critical systems where every decision can have far-reaching consequences. This section delves into the approaches used to interpret deep learning models for RUL prediction, exploring various approaches and their limitations.

(Yu Zhang et al., 2021) conducted a comprehensive review, providing a three-dimensional taxonomy for a better understanding of the distribution of research in this area (see Figure 2.2): **(a) Dimension 1** divides approaches into two categories. The first, **Passive approaches**, also known as post-hoc interpretation (Madsen et al., 2022), begins with a fully trained network and attempts to extract understandable patterns from the learned weights. Passive methods are widely applied due to their straightforward application to most existing networks. However, their generality can be a limitation, particularly in integrating specific domain knowledge or priors. The second category, **Active approaches**, also referred to as ad-hoc, involves actively altering the network architecture or training process to enhance interpretability. These active methods propose optimization strategies for networks to improve their interpretability. **(b) Dimension 2** classifies approaches based on the type of explanations they provide. This classification includes

methods like attribution, where credit is assigned to input features, and hidden semantics, where the focus is on interpreting specific hidden neurons, layers, or modules. **(c) Dimension 3** categorizes methods based on the scope of interpretability: Global (interpreting decision logic for all samples), Semi-local (for a group of samples), or Local (for individual samples).

To illustrate these dimensions in practice, let's explore some notable examples. In the category of Passive attribution approaches, commonly utilized techniques include Shapley Additive exPlanations (SHAP) (Lundberg and S.-I. Lee, 2017), Layer-Wise Relevance Propagation (LRP) (Bach et al., 2015), Integrated gradients (Sundararajan et al., 2017),and Local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016). These methods are acclaimed for effectively assigning significance to input features in the predictive process of a model, enhancing understanding of how each feature influences the output. Moving to active by example approaches, we find methods that involves computing the distance between new samples and known ones, as seen in the works of (O. Li et al., 2018) and (Koch et al., 2015). Such methods actively leverages sample comparisons to enhance the interpretability of the model's decision-making process. In the category of Global Active hidden semantics approaches, we find approaches focused on training models to learn disentangled and interpretable representations (X. Zhu et al., 2021). such as the work by (Q. Zhang et al., 2018) where they assign each filter in a high convolutional layer with one object (one filter, one concept).

After gaining insights into the various ways DL models can be interpreted, the focus shifts to their application in PHM. This shift raises a critical consideration: determining the specific requirements of interpretability that are most relevant and beneficial in industrial contexts, especially for RUL prediction models. In industrial settings, the models need to align with the operational knowledge and practical experience of industry professionals. To achieve this, several key aspects of interpretability should be considered:

- **Transparency in Decision Processes:** It is crucial to demystify the decision-making process of models. Understanding how and why certain outputs are reached enables experts to trust and effectively use the model's predictions.
  - **Concept-Specific Insights:** the approach should be capable of handling different concepts present in the data individually. Typically, models address various concepts collectively, which can obscure specific insights. Individual handling of concepts would allow experts to more effectively demystify and understand the model's outputs in relation to each specific condition.
  - **Reflect Domain-Specific Knowledge:** The method should reflect domain-specific knowledge, ensuring that the interpretations are relevant and actionable within the specific industrial context.
- **Interpretable Insights into Degradation Causes:** The approach should provide interpretable insights into the causes of degradation. This understanding is vital for making informed decisions, such as altering operational parameters to optimize performance and prolong equipment life.

These aspects are fundamental to ensuring that DL models for RUL prediction are not just tools for prediction but also partners in decision-making, providing valuable, understandable, and actionable insights for industrial practitioners.

Building upon these foundational principles of interpretability in industrial settings and the dimensions discussed before, Table 2.3 provide examples of papers that proposed approaches for interpreting DL models for PHM application.

Table 2.3: Examples of RUL prediction DL models interpretability from the literature.

	Passive	**Active**
Rule as explanation		
Explaining hidden semantics		*** Our Contribution *** VAE: (Costa and Sánchez, 2022)
Attribution as explanation	LRP (Bach et al., 2015): (F. Wang et al., 2023) SHAP (Lundberg and S.-I. Lee, 2017): (Hong et al., 2020) (Yilin Wang et al., 2021) LIME (Ribeiro et al., 2016): (Sanakkayala et al., 2022)	(F. Wang et al., 2023)
Explanation by showing examples		Siamese Network (Jang and C. O. Kim, 2021)

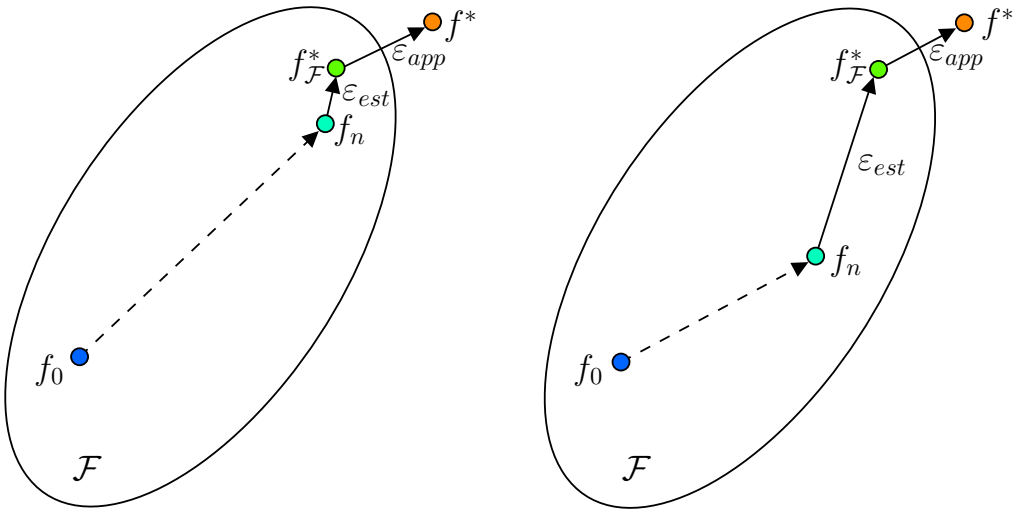
For passive approaches, (Hong et al., 2020; Yilin Wang et al., 2021) employed Shapley values to decompose the model's output into quantifiable contributions from each input feature and can also be expressed as a negative number (meaning negative impact on the expected output), offering the possibility to locating fault state points, observing the declining trend of sensor data, and evaluating the health status of subsystem, achieving partial white-boxing.

As for Active approaches, (F. Wang et al., 2023) proposed an approach based on LRP, the idea is that the LRP technique is used to calculate relevance scores of the intermediate feature map. Then, these relevance scores are embedded into optimization procedure to guide model training. By emphasizing features with higher relevance scores and diminishing the impact of less relevant ones, this approach guides the model to focus on more significant inputs, enhancing interpretability. (Costa and Sánchez, 2022) adapted a Variational autoencoder (VAE) for RUL prediction by replacing the traditional decoder with a regression model. This modification imposes an additional constraint on the VAE, resulting in a more structured and continuous latent space. In this refined space, trajectories corresponding to different RUL values are distinctly separated, facilitating clear differentiation. Such approach serves to project new samples near known samples with similar degradation pattern, this proximity in the latent space offers an explainable diagnosis, as it reveals how new samples relate to known degradation trajectories. Similarly, (Jang and C. O. Kim, 2021) employed a Siamese network architecture for predicting the RUL based on sample proximity in a representation space. By focusing on the similarities and differences between new and existing samples, the network can provide a more accurate and more interpretable predictions.

From these papers, it's evident that researchers have been actively proposing approaches that would help with several aspects of interpretability within industrial context discussed before, papers are looking to improve transparency in decision processes (Jang and C. O. Kim, 2021; Costa and Sánchez, 2022) and to provide interpretable insights into degradation causes

(Sanakkayala et al., 2022; Yilin Wang et al., 2021). However, much like in other fields of DL, there remains a need for continued research in various areas and directions. In line with this, our focus shifts towards an active approach that enhances the interpretability of DL models in relation to operating conditions. Our work involves transitioning from monolithic architecture to modular architectures, where each module in this proposed framework would specialize in a specific operating condition.

2.4 Strategies for Addressing Data Scarcity



(a) Large n. (b) Small n.

Figure 2.3: Comparison of learning with sufficient and few training samples. Small n can lead to high estimation error ε_{est}.

In the introduction, we emphasized the constraints posed by the limited availability of RTF trajectories in industrial settings. This scarcity of data is a critical bottleneck in developing reliable DL models. To contextualize this challenge, we adopt an assumption for our discussion: a specific system or equipment might only provide access to 20 or fewer RTF samples. While this assumption is not universally applicable, it mirrors a plausible scenario in various industrial environments. The reasoning for this assumption is rooted in factors such as preventive maintenance and the cost associated with letting equipment run to failure (Lv et al., 2020; Fink et al., 2020). This hypothetical yet realistic scenario offers a pertinent framework to explore the effects of data scarcity on the development and efficacy of DL models in industrial contexts. To better understand the implications of this limitation, we illustrate the issue based on the error decomposition framework in supervised machine learning. This framework allows us to dissect how the scarcity of data fundamentally impacts the model's learning.

For illustration, given a number of samples n in the training data, a hypothesis f, a specific model with a fixed form and is parameterized by a real vector over which the optimization is to be performed, $\mathcal{F}$ the family of prediction functions. Let f^* be the function that minimizes the expected risk (the true function or the target hypothesis that we are trying to approximate), $f^*_{\mathcal{F}}$ be the function in $\mathcal{F}$ that minimizes the expected risk, and f_n be the function in $\mathcal{F}$ that minimizes the empirical risk approximated over the training samples n.

Table 2.4: Approaches to deep learning with small datasets in the PHM domain.

Approach	**Selected References**
Similarity-Based	(M. Hou et al., 2020; CEYLAN and Yakup, 2021) (Aydemir et al., 2022)
Multi-task learning	(Yan et al., 2023; Yong Zhang et al., 2022) (T. Wu and T. Chen, 2023; Behera, Misra, and Sillitti, 2023) (Huaqing Wang et al., 2022; Miao et al., 2019)
Data augmentation	(Gay et al., 2022; S. Kim et al., 2020) (J. Sun and Q. Sun, 2021; Zhiyao Zhang et al., 2023) (Kwak and Jongsoo Lee, 2023; Behera and Misra, 2021)
Leveraging external data	(Ansi Zhang et al., 2018; F. Yang et al., 2020) (P. Ding and Jia, 2020; Mo et al., 2023) (Ellefsen et al., 2019; Yoon et al., 2017)

As f^* is unknown, we try to approximate it by an $f \in \mathcal{F}$. $f^*_{\mathcal{F}}$ is the best approximation for f^* in $\mathcal{F}$, while f_n is the best in $\mathcal{F}$ obtained by empirical risk minimization. if we assume that f^*, $f^*_{\mathcal{F}}$ and f_n are unique, the total error can be decomposed as (Bottou, Curtis, et al., 2018; Yaqing Wang et al., 2020):

$$\mathbb{E}[R(f_n) - R(f^*)] = \underbrace{\mathbb{E}[R(f^*_{\mathcal{F}}) - R(f^*)]}_{\varepsilon_{\text{app}}(\mathcal{H})} + \underbrace{\mathbb{E}[R(f_n) - R(f^*_{\mathcal{F}})]}_{\varepsilon_{\text{est}}(\mathcal{H},S)} \tag{2.1}$$

Where the expectation is with respect to the choice of training set (Bottou and Bousquet, 2007). The approximation error ε_{app} measures how close the functions in $\mathcal{F}$ can approximate the true function f, and the estimation error ε_{est} measures the effect of minimizing the empirical risk instead of the expected risk with $\mathcal{F}$. The total error is as shown affected by the hypothesis space $\mathcal{F}$ and n which is the number of samples. In general, ε_{est} can be minimized by having a larger number of samples (Bottou and Bousquet, 2007; Bottou, Curtis, et al., 2018), thus, when having a small number of samples, the empirical risk minimizer h_n is no longer reliable because the empirical risk may be far from being a good approximation of the expected risk (due to over-fitting). Figure 2.3 shows a comparison of learning with sufficient and few training samples.

To address the inherent challenge of data scarcity in the PHM field, a diverse array of methodologies has been explored and documented in the literature. These strategies encompass a broad spectrum (Table 2.4), including data augmentation and meta-learning, among others. In the following, we will delve into a selection of these approaches, examining their principles, applications, and effectiveness in the context of prognostics.

First, we begin our exploration with an in-depth look at **metric learning** approaches. Central to these techniques is their ability to estimate RUL by comparing new samples with existing, labeled samples. What distinguishes metric learning is that it relies solely on information inherent in the data, thus avoiding the need for external data sources or additional annotations. This is achieved by discerning and highlighting relative distances or similarities between data points within a feature space.

Among the examples showcasing the potential of Metric Learning, the study by (M. Hou et al., 2020) demonstrates the utilization of Euclidean distance on Health index (HI) to measure the

similarity between new samples and reference ones for RUL prediction. Another example of metric learning approaches is the Siamese neural network, originally proposed to learn an embedding space based on the class labels of given samples (Koch et al., 2015). In this embedding space, pairs of samples from the same class are positioned closely, while those from different classes are separated by a discernible distance. The Siamese network accomplishes this through its architecture, which consists of twin embedding networks sharing the same weights. This structure is depicted in Figure 2.4. The network is trained using a contrastive loss function, which effectively pulls together similar samples while pushing apart dissimilar ones. The Siamese network methodology has been embraced for RUL prediction in various studies, (CEYLAN and Yakup, 2021) focused on broader applications without evaluating performance on smaller datasets, whereas (Aydemir et al., 2022) provided empirical evidence of its efficacy with limited RTF trajectory data. Although the principle of sample similarity has been investigated within the realm of RUL prediction by other researchers (Khelif et al., 2014; Yingchao Liu et al., 2019), they do not fall within the scope of deep learning-based methodologies and, thus, are not the primary focus of our discussion.

Despite the considerable successes achieved by these Metric Learning approaches across various tasks (Kaya and Bilge, 2019), their performance is highly dependent on the data. This dependency may lead to a poor understanding of the similarity relationship among samples. To overcome the challenges related to this, other domains, most notably Computer Vision (CV), have leveraged pre-trained models which help achieve more discriminative learning in embedding space (Sohn, 2016).

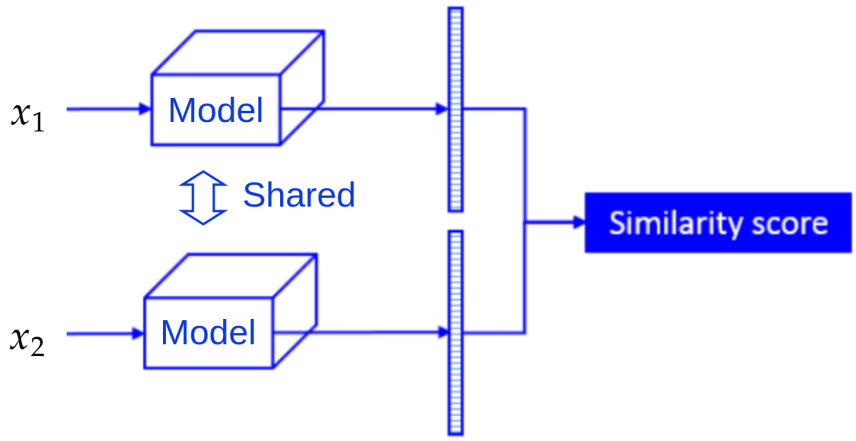


Figure 2.4: Siamese Network (Koch et al., 2015).

We now shift our attention to an alternative approach that also utilizes primary data only, but this time it incorporates an inductive bias by learning from multiple tasks simultaneously. Multi-task learning (MTL) represents a paradigm that extends beyond the sole objective of RUL prediction by undertaking multiple related tasks simultaneously. This shift enriches the model's learning process, as it leverages the inter-task relationships and concurrent task processing that encourages models to draw deeper insights and to improve generalizability and robustness (Ruder, 2017). Several studies have explored the application of MTL in RUL prediction, with research efforts focusing on various task combinations to improve predictive performance. (Yan et al., 2023; Yong Zhang et al., 2022; T. Wu and T. Chen, 2023; Behera, Misra, and Sillitti, 2023) have concentrated on jointly learning RUL prediction and Health state (HS) estimation. This

concurrent learning approach allows the model to draw parallels between the overall health state of a system and its remaining useful life, enriching the predictive accuracy. Similarly, the research presented by (Huaqing Wang et al., 2022; Miao et al., 2019; Fu et al., 2024) have applied MTL to couple RUL prediction and fault detection, degradation stages and first prediction time, respectively. (Y.-H. Lin et al., 2023) proposed to construct the added tasks in an unsupervised manner, by reconstructing the current and future inputs, they created two additional tasks to help with RUL prediction.

However, it is important to note that these MTL approaches proposed in the literature depend on the information already present in the main dataset. The additional tasks created within these models require expert knowledge for task selection and possibly manual labeling. This dependency highlights a crucial consideration: the effectiveness of MTL in RUL prediction is closely tied to the quality and comprehensiveness of the underlying data, as well as the depth of domain expertise available for task formulation and annotation.

After discussing approaches that leverage limited data directly, we now shift our focus to strategies that expand upon the existing dataset. This brings us to the concept of **Data augmentation.** The core idea of this approach is to artificially expand the size of the training dataset by creating modified versions of the existing data. Data augmentation consists in generating new samples from existing ones. It is widely used in image processing to improve performances and resilience of machine learning algorithms, see for instance (Shorten and Khoshgoftaar, 2019). Techniques to augment data range from basic manipulations such as adding noise, scaling, and rotating, as noted by (Iglesias et al., 2022), to more sophisticated approaches, such as the methods discussed by (Wen et al., 2020). A diversity of data augmentation transformations are defined in the literature, with some surveys proposing to group them in families. Following the taxonomy provided by (Iwana and Uchida, 2021), data augmentation transformations are classified in four families: (a) Random transformations which consist in altering time samples of the time series based on factors randomly picked. (b) Pattern mixing which consist in generating new trajectories as combinations of multiple original ones. (c) Generative models which consist in training deep neural networks to generate new samples. Finally, (d) Decomposition based approaches which consist in decomposing the signals in sub-signals and applying data transformations to one or more of the decomposed signals.

In the domain of RUL prediction, (Gay et al., 2022) showed empirically that some generic data augmentation techniques that do not require prior knowledge can improve performance when having small datasets, however, they highlighted the challenges related to the difficulty of choosing the adequate transformation techniques. (S. Kim et al., 2020) proposed to leverage Dynamic time warping (DTW) alongside reference data to generate virtual RTF trajectories. Similarly, (J. Sun and Q. Sun, 2021) used the segment scaling method to generate a set of synthetic sequences. Advancing into deep learning-based augmentation techniques, (Zhiyao Zhang et al., 2023) utilized informer architecture (an Auto-encoder) to generate the synthetic time series for increasing the amount of data in initial degradation stage. In a similar vein, (Kwak and Jongsoo Lee, 2023) employed a VAE to approximate the original data distribution appropriately to generate virtual data. Beyond these methods, Generative adversarial networks (GAN) have also been a focal point in data augmentation for RUL prediction. (Behera and Misra, 2021), utilized conditional GAN depending on RUL values (labels) to generate new sequences. Expanding on the versatility of GANs (Y. Ding et al., 2023) leveraged this architecture to do out-domain augmentation. Their approach focuses on developing a robust predictor capable of resisting out-of-distribution perturbations, a crucial aspect for models expected to perform under diverse and unseen operating conditions.

While the aforementioned data augmentation approaches demonstrate considerable promise,

Table 2.5: Data augmentation approaches used for RUL predicion

Family	Transformation	Selected References
Magnitude Transformation	Jittering Rotation Scaling	(Gay et al., 2022)
Time Transformation	Slicing Time Warping	(Gay et al., 2022) (S. Kim et al., 2020)
Pattern Mixing	Interpolation Extrapolation	(Gay et al., 2022)
Generative Models	GAN AE VAE	(Zhiyao Zhang et al., 2023) (Kwak and Jongsoo Lee, 2023) (Behera and Misra, 2021) (Y. Ding et al., 2023)

their practical application in RUL prediction is contingent upon specific prerequisites that can be challenging to meet in industrial contexts. Primarily, these methods often depend heavily on expert knowledge or the representativeness of the available data. For instance, developing models such as GANs and VAEs necessitates a dataset that is sufficiently representative of the equipment in question. This requirement becomes particularly problematic in scenarios characterized by extremely limited data availability. Furthermore, the reliance on expert knowledge to guide the data augmentation process introduces another layer of complexity. While domain expertise is invaluable, it also presupposes the availability of such expertise, which might not always be the case. Additionally, the subjective nature of expert-driven decisions could introduce biases or overlook nuances in the data, potentially impacting the model's accuracy and reliability.

While the discussed approaches, such as metric learning and multi-task learning, represent powerful methods to leverage existing datasets, there is also significant value in exploring strategies that incorporate external data sources. This integration is crucial, as relying solely on the primary dataset may lead to under-utilization of available information. Particularly in scenarios where access to data from similar or related equipment is feasible, the incorporation of this external data can enrich our predictive models.

In this thesis, we assume that in an industrial use case, we could work under these settings:

- We would have access to a **limited number of labeled RTF trajectories**
- We would have access to **open or private source datasets**
- These datasets may originate from **diverse types of equipment**, which could lead to **varying sensor types and numbers and trajectory lengths**, presenting a unique set of challenges and opportunities. A notable example is the disparity between the CMAPSS dataset (Saxena et al., 2008) and the N-CMAPSS dataset (Arias Chao et al., 2021), both representing RTF trajectories of turbofan engines but with differing sensor configurations. This scenario is typical in industrial settings due to varied task requirements and cost considerations.

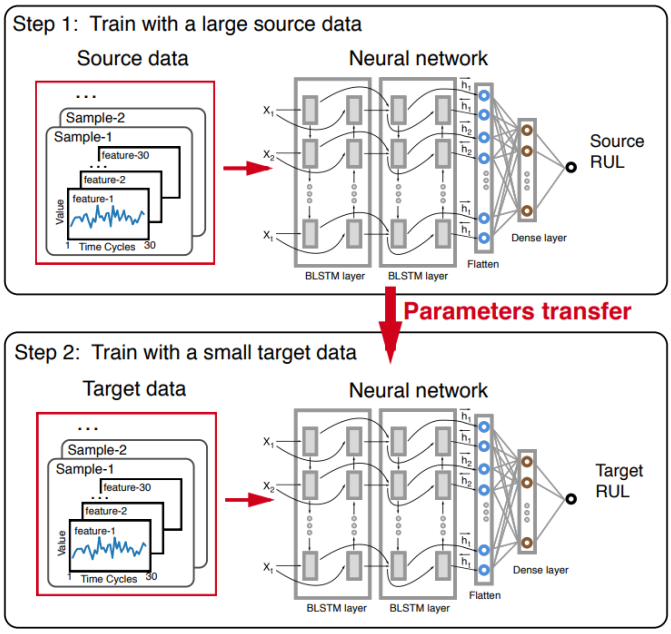


Figure 2.5: Transfer learning framework proposed by (Ansi Zhang et al., 2018).

- These datasets could be **unlabeled, do not have complete life cycle trajectories**, often due to maintenance practices and the high cost of labeling. Labeling such data is complex, time-consuming and requires considerable expertise. Furthermore, even when labeled, these labels may not be fully compatible with the main objective, as they generally represent a one-time effort that is not always relevant to the main tasks.

Historically, the **Pre-Training** of deep learning networks has addressed these challenges (Han et al., 2021). *Pre-Training* as a method, involves learning representations by training a model on a comprehensive dataset, typically of a general nature. The primary goal of this is to enable the model to assimilate a wide spectrum of features and patterns, extract generic information transferable to various applications tasks. This approach has been successful in domains like image recognition, as seen in (Krizhevsky et al., 2012), and natural language processing, demonstrated by (Vaswani et al., 2017). Representations have already been successfully explored in the fields of image recognition since 2012 (Krizhevsky et al., 2012), with convolutional architectures trained on ImageNet (J. Deng et al., 2009), and automatic natural language processing in 2018 (Vaswani et al., 2017), with attention models trained in particular on word prediction tasks on the internet. However, the industrial field has yet to see widespread adoption of such pre-trained models, due to multiple reasons such as companies not sharing their data, and to the fact that the open source data sets suffer from heterogeneity and are very small compared to the ones used in other fields, which probably explains at least in part why 'the ImageNet moment' of the industry of the future has not yet taken place. *Pre-Training* serves as an excellent foundation in various approaches, for instance, they can be used as backbones in metric learning frameworks such as Siamese networks (Figure 2.4), as discussed in (Kumar et al., 2023).

In the context of RUL prediction, the adoption of pre-trained models is yet to reach its full potential. Several studies have embarked on this path using the PreTraining + FineTuning (PT-FT) approach, where *Fine-Tuning* is employed to adapt the pre-trained model to specific tasks. This approach has been recognized for its capacity to enhance model performance, as

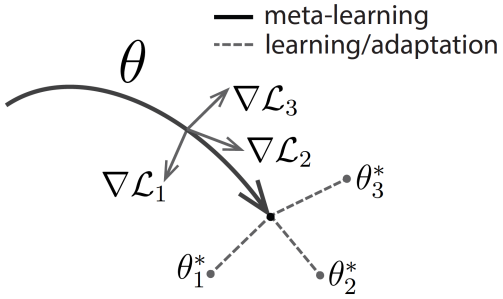


Figure 2.6: Model-agnostic meta-learning algorithm (MAML) (Finn et al., 2017).

seen in the PHM field (S. Yao et al., 2023). For instance, the studies by (Ansi Zhang et al., 2018) and (F. Yang et al., 2020) pre-trained deep learning models on related source datasets encompassing varied operating conditions and fault modes. These models are then fine-tuned on smaller, more focused datasets , as depicted in figure 2.5. Their findings indicate a marked improvement over models trained exclusively on target data. Similarly, (Ellefsen et al., 2019) propose a pre-training phase with the same data but in an unsupervised manner, follower by fine tuning for RUL prediction task.

Despite its advantages, the PT-FT approach is not without its challenges. One significant issue is the rapid over-fitting during the fine-tuning process, especially when training data for the target task is scarce. Additionally, as highlighted by (Dery, Michel, Talwalkar, et al., 2021), there is a risk that the learned representations might not be optimally aligned with the primary task. To mitigate these challenges, (W. Chen et al., 2022) proposed a novel approach of pre-training a model on both target and source data based on domain adaptation. Furthermore, a common yet unaddressed challenge in literature is the integration of multiple heterogeneous datasets, particularly when these datasets comprise varying types of sensors or have differing trajectory lengths. Such disparities lead to datasets with diverse numbers of input features and sequence lengths, complicating the process of model training and adaptation. Although works like (Ansi Zhang et al., 2018), (F. Yang et al., 2020), and (Ellefsen et al., 2019) used PT-FT, they primarily employed similar datasets during their pre-training phases. Therefore, the challenge remains significant and could lead to the under-utilization of available knowledge.

Apart from the aforementioned strategies, other approaches that are related have been proposed in the literature for scarce data problem under the umbrella name of **meta-learning**, or **Learning-to-Learn**, which is an innovative approach in tackling the scarce data problem in machine learning (Hospedales et al., 2021). This paradigm involves a bi-level optimization, where feedback from an inner-learner is utilized to optimize a meta-learner. Notably, in standard Few-shot learning (FSL) contexts, meta-learning models have demonstrated capabilities surpassing those of conventional PT-FT strategies (Yisheng Song et al., 2023). A pivotal strategy within meta-learning is episodic training. This method structures data into numerous small tasks or 'episodes,' each representing a distinct scenario. The model is trained to accomplish specific tasks within these episodes, fostering its ability to generalize from a limited set of examples. This generalization is achieved by exposing the model to a diverse array of challenges during training. An exemplary work in this sub-field is the Model-Agnostic Meta-Learning (MAML) algorithm by (Finn et al., 2017), which strives to meta-learn model parameters θ_s that can be effectively adjusted for task-specific optimization through a few gradient descent steps (Figure 2.6). Inspired by MAML, various studies have adapted meta-learning for RUL prediction. For instance, (P. Ding and Jia, 2020) extracted time-frequency images from rolling bearing vibration

signals using sliding windows. They then employed two-dimensional domain adaptation for latent subspace construction, followed by meta-learning a CNN with MAML for RUL prediction. Similarly, (Mo et al., 2023) applied MAML in turbofan RUL prediction, proposing the use of time sequence similarities to construct training tasks.

Contrasting with pre-training approaches that require extensive datasets to learn a generalized model initialization, meta-learning techniques aim for rapid adaptability with minimal data requirements. This allows models to quickly acclimate to new tasks, a fundamental difference from the extensive dataset needs of pre-training methods. However, these episodic training strategies come with inherent challenges. A primary limitation is the assumption of uniform input and output structures across different tasks, given the shared model architecture used in the optimization process (Finn et al., 2017). This assumption of data uniformity, while suitable for controlled experimental settings, is often not reflective of the diverse and varied data structures encountered in industrial settings, posing a significant limitation to the applicability of such approaches in real-world scenarios.

The **Semi-supervised learning (SSL)** paradigm represents another approach explored in the literature to address the issue of data scarcity. This paradigm offers a partial solution by incorporating unsupervised data to enhance learning, as discussed in (Duarte and Berton, 2023; Ouali et al., 2020). It has been applied in Remaining Useful Life (RUL) prediction, as seen in multiple studies (Ellefsen et al., 2019; Yoon et al., 2017). However, a significant limitation of many SSL methodologies is their reliance on the assumption that both supervised and unsupervised data sets are drawn from identical distributions. This assumption often does not hold true in various scenarios, as highlighted in (Banitalebi-Dehkordi et al., 2022), especially under the conditions we assume.

Considering the limitations of existing approaches, the unique characteristics of industrial datasets and use cases, and the absence of a large pre-trained model tailored to these types of problems, there is a definitive need for a new methodology. Such methodology should be capable of integrating multiple data sources and flexibly utilizing unlabeled data (if available) without the necessity for costly manual labeling. Our approach meets these challenges and enables the effective use of datasets to enhance the performance of prognostic models. We propose a method based on the auxiliary training paradigm, which, as demonstrated empirically, can provide improved performance and generalization capabilities for RUL prediction. Our proposed architecture is modular, allowing for efficient use of heterogeneous datasets in either pre-training or auxiliary learning contexts. Furthermore, we have developed this approach to automatically construct auxiliary objectives from unlabeled data using meta-learning.

2.5 Datasets

This section introduces the datasets that will be utilized to evaluate the prognostic models developed in this thesis. We have selected three distinct datasets (Table 2.6), each with characteristics that align with the challenges and scenarios pertinent to our research objectives. Although not an exhaustive list of all datasets available for RUL prediction, the focus here is on those that are particularly relevant to the methodologies proposed within our work. For a comprehensive survey of datasets suitable for PHM applications, (Hagmeyer et al., 2021) provides a detailed review of public datasets available in the field.

The chosen datasets for this thesis share common features that make them highly suitable for developing and testing prognostic algorithms. Below is a general overview of these shared characteristics:

Table 2.6: Data-sets

Dataset		FD001	FD002	FD003	FD004	N-CMAPSS	Batteries
Nb Train trajectories		100	260	100	249	66	100
Nb Test trajectories		100	259	100	249	43	34
Nb Operating conditions		1	6	1	6	-	-
Nb Fault modes		1	1	2	2	7	-
Trajectories length distribution	Max	360	378	525	543	100	2237
	Mean	206	206	247	245	75	817
	Min	128	128	145	128	48	149

- They capture multi-dimensional responses from complex non-linear systems, encompassing a range of behaviors of the systems under study.
- They include realistic levels of noise, reflecting the variability typically encountered in practical scenarios.
- They introduce the challenge of operational conditions that may conceal fault indications.
- They provide numerous trajectories RTF from many units, providing a rich source for DL models to learn system behaviors and effectively predict the RUL.
- Two of the three datasets include predefined independent test sets, which allows us to evaluate our results against existing literature.

The following subsections provide detailed descriptions of each dataset, describe their respective processing techniques and explain the methodologies employed for model evaluation using these datasets.

2.5.1 The C-MAPSS Dataset

The commercial modular aero-propulsion system simulation (C-MAPSS) is a turbofan engine simulation environment from NASA that provides access to health, control, and engine parameters through a graphical user interface (GUI). The C-MAPSS dataset (Saxena et al., 2008) is generated using the simulation program by monitoring the degradation of multiple Turbofan engines. Comprising four sub-datasets (FD001 to FD004), the C-MAPSS dataset offers a diverse array of operating conditions and fault modes. These datasets are extensively utilized by researchers to benchmark and validate prognostic algorithms due to their complexity and real-world relevance.

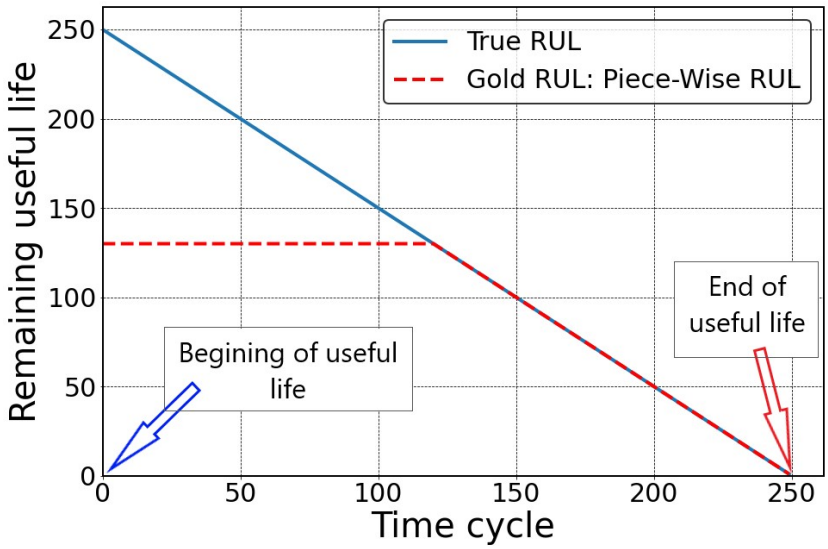


Figure 2.7: True RUL vs Gold RUL of one trajectory with a length = 250 (Piece-wise maximum RUL is 130 time cycles).

These sub-data sets were geared towards data-driven approaches where very little or no system information was made available to developers. Each sub-data set is divided into training and test subsets. The training set is composed of input time series (multivariate time series, which correspond to 24 inputs (21 sensors and 3 control signals) taken at each operating cycle (flight) of a particular simulated turbofan engine) which are assumed to go on until failure. In the test set, time series are truncated arbitrarily and the objective is to estimate the number of remaining operational cycles before the system failure occurs.

A 'gold' RUL value is established for each cycle (or equivalently, time frame) in the training set by assuming a linear decrease in RUL over time. However, engine degradation is typically negligible in the early stages and accelerates as failure approaches. To account for this non-linearity, a piece-wise linear function is adopted (illustrated in Figure 2.7), capping the maximum RUL and initiating linear degradation after a predefined threshold. This strategy, common in the literature (Heimes, 2008), allows for a more realistic modeling of RUL. We opt for a maximum RUL of 130 cycles, in line with common practice (Al-Dulaimi et al., 2019; Zheng et al., 2017).

The 'gold' RUL value for each cycle is defined by Equation 2.2:

$$Gold\ RUL = \begin{cases} 130 & \text{if } True\ RUL \geq 130 \\ True\ RUL & \text{if } True\ RUL < 130 \end{cases} \tag{2.2}$$

Setting this maximum value aims to facilitate model training by constraining the range of target values. Although this may limit the model's ability to predict very long RUL values, such scenarios are anticipated to be infrequent in the test set.

Deep learning models typically benefit from normalized input and output data. To facilitate this, we normalize all sensor readings and control signals to the $[0, 1]$ range (as depicted in Figure 2.8), using Equation 2.3 for inputs, and similarly for RUL values:

$$x_{t,i}^{j} = \frac{v_{t,i}^{j} - \min(v_i^{train})}{\max(v_i^{train}) - \min(v_i^{train})} \tag{2.3}$$

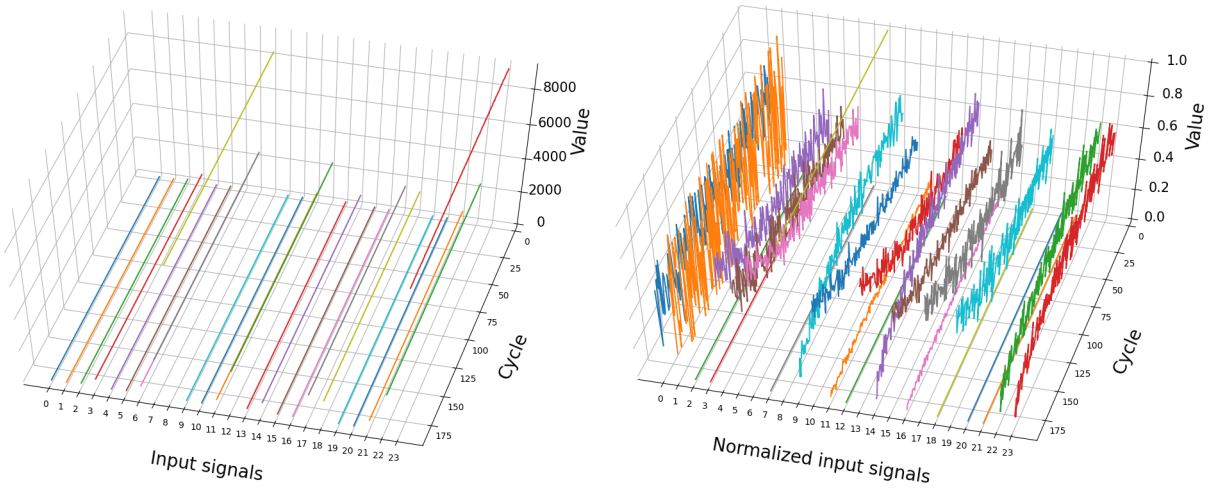


Figure 2.8: Input values with and without normalization.

Where $v_{t,i}^{j}$ is the value of the i_{th} sensor at time t from engine j and $x_{t,i}^{j}$ is the corresponding normalized value. The terms $\max(v_i^{train})$ and $\min(v_i^{train})$ represent the maximum and minimum values of the i_{th} sensor, calculated across all training dataset readings. These normalization parameters derived from the training dataset are applied consistently to the validation and test datasets, preventing any leakage of information from the validation and test datasets and ensuring that the model learning process does not acquire indirect knowledge of these datasets. Normalization helps to stabilize training of the network parameters, speeds up convergence of gradient descent, and reduces the risk of getting stuck in local optima.

2.5.2 The N-CMAPSS Dataset

The new Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) dataset (Arias Chao et al., 2021) represents further improvements and developments of the original CMAPSS data-set. This data was also synthetically generated using the CMAPSS engine simulator by using more models that simulate other important factors such as the atmosphere and the power management system. In addition, actual flight conditions recorded on board a commercial aircraft were used as input to the simulation model, providing a more realistic data-set with greater fidelity of the degradation and operating conditions.

Encompassing multivariate time series data, the N-CMAPSS dataset provides a detailed record of multiple sensor measurements throughout entire flights until engine failure. This dataset is divided by default into development and test subsets, providing a structured framework for algorithm evaluation. Each trajectory within the dataset simulates one of seven distinct failure modes, each affecting different engine sub-components, as detailed in (Arias Chao et al., 2021). Given the variability in initial conditions and operation until failure, the dataset presents a varied number of flights for each of the 109 units, with each data file containing essential information such as scenario descriptors, sensor measurements, RUL labels, and auxiliary data.

In terms of data processing, each flight's data is summarized by calculating its mean, standard deviation, minimum, and maximum values. This uniform approach is applied across all our work for fair comparison, and is performed to facilitate the training of models by transforming data into a more manageable form. In addition, we adhere to the min-max normalization method, as

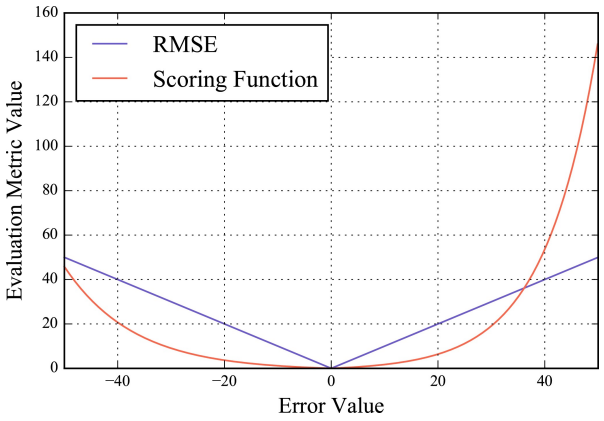


Figure 2.9: Comparison between the scoring function and RMSE with respect to different error values.

described in the previous subsection.

2.5.3 The Batteries Dataset

The batteries data-set (Severson et al., 2019) dataset provides comprehensive insights into the behavior of 134 commercial lithium-ion batteries under rapid charging conditions that pushed them to the edge of failure. This dataset originally doesn't categorize the batteries into development and test groups. However, for our analysis, we have partitioned it into two distinct subsets: 75% of the data is allocated for development, while the remaining 25% is reserved for testing.

The dataset includes two types of data: Summary and Cycle data. The Summary data, which is the focus of our analysis, provides per-cycle information such as cycle number, discharge capacity, charge capacity, internal resistance, and temperature metrics (maximum, average, and minimum), along with charge time. On the other hand, the Cycle data offers detailed within-cycle information, including time, charge capacity, current, voltage, temperature, and discharge capacity. In this thesis, we exclusively use the summary data from this dataset to build the model, as they are less complex and manageable, and as before, we use to the min-max normalization method.

2.5.4 Performance metrics

For the task of predicting the Remaining Useful Life (RUL) (regression task), we commonly employ the Root Mean Square Error (RMSE) as the primary performance metric, as it allows for easy comparison with other research results. The RMSE is calculated using the following equation (Eq. (2.4)):

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\frac{1}{T_j}\sum_{t=1}^{T_j}(\hat{y}_t^j - y_t^j)^2} \tag{2.4}$$

Where N,T_j are respectively the number, length of the trajectories, and $\hat{y}_t^j$, y_t^j represent respectively the predicted RUL and the True RUL.

In the maintenance context context, it is generally desirable to predict failures as early as possible to take preventive actions. Therefore, it's equally important to assess the models based on a scoring function that penalizes late predictions of RUL more than early predictions. This

scoring function, outlined in Eq. 2.5, was introduced in the original in the original C-MAPSS evaluation campaign by (Saxena et al., 2008):

$$S = \sum_{j=1}^{N} \sum_{t=1}^{T_j} s_t^j \text{ , Where: } s_t^j = \begin{cases} e^{-\frac{\hat{y}_t^j - y_t^j}{13}} - 1 & \text{for } (\hat{y}_t^j - y_t^j) < 0 \\ e^{\frac{\hat{y}_t^j - y_t^j}{10}} - 1 & \text{for } (\hat{y}_t^j - y_t^j) \geq 0 \end{cases} \tag{2.5}$$

Where N,T_j are respectively the number, length of the trajectories, and $\hat{y}_t^j$,y_t^j represent respectively the predicted RUL and the True RUL.

Figure (2.9) shows how both metrics penalize errors in detail. The main objective is to achieve the smallest value possible for both.

2.6 Conclusion

This chapter has offered an overview and comparative analysis of current State Of The Art (S-O-T-A) DL methodologies in the realm of RUL prediction. By critically reviewing existing challenges and highlighting potential directions for future research, we aimed to forge a valuable resource for researchers navigating this rapidly evolving field. This literature analysis, particularly focused on the application of DL in RUL prognostics, is designed to serve as a guide for further exploration and innovation in the field.

Throughout this chapter, we have identified three main challenges crucial to the field of prognostics. Firstly, our review of the various DL architectures applied to prediction RUL, their performance and limitations, directly addressed our initial research question (RQ1). This analysis highlighted the need for architectures capable of maintaining consistent performance, even under multiple varying operating conditions. Next, we explored the proposed approaches to explain/interpret DL models in this field (RQ2), identifying the need for ongoing research and new approaches for industrial applications. Finally, our review of strategies to address data scarcity for RUL prediction led us to recognize their limitations (RQ3), calling for further research to exploit DL in real-world scenarios.

In subsequent chapters, we will outline our proposed approaches aimed at the challenges identified. Our methodology in addressing RQ1 and 2 will concentrate on environments with sufficient RTF data, where we aim to develop approaches that address the identified challenges. For RQ3, attention shifts to settings with limited data availability, where we propose methods to improve deep learning model development under such constraints. As we proceed, the insights gained from this review will guide our methodologies. Our goal is to ensure that our contributions meet the practical needs of the PHM field.

Deep learning models development in data-rich environments

Chapter 3

End-to-End Deep Learning Model for Improved Remaining Useful Life Prognostic

Contribution

As mentioned in the previous chapter, Deep Learning has become a major and rapidly growing research direction, redefining S-O-T-A performances in Prognostics in recent years. However, the architectures and approaches proposed in the literature suffer from a drop in performance of these models when dealing with multiple operating conditions, and often rely on feature selection/engineering.
In order to address this limitation, we propose an end-to-end deep learning model based on MLP and LSTM to predict the RUL. After normalization of all data, inputs are fed directly to an MLP layers for feature learning, then to an LSTM layer to capture temporal dependencies, and finally to other MLP layers for RUL prognostic. Despite its simplicity with respect to other recently proposed models, the model developed outperforms them with a significant decrease in errors value between the predicted and the gold value of the RUL.

3.1 Introduction

As we advance further into the exploration of DL in the field of prognostics, it becomes increasingly clear that DL is not just an emerging technology, but a transformative force redefining the S-O-T-A performances. The previous chapter highlighted the impressive strides made in applying DL to prognostics, showcasing its potential to revolutionize how we predict and manage system life cycles. Yet, this journey is not without its challenges. As we delve deeper into the application of these technologies, certain limitations become apparent, particularly when these models are subjected to complex, real-world conditions. In this chapter, we address a critical research question : **Which deep learning architectures are best suited for end-to-end training, can handle data variability, and still deliver accurate RUL predictions?**

The answer to this question lies in understanding and overcoming the hurdles posed by varying operational conditions—a common scenario in many industrial settings. An Operating condition (OC) can be defined as the circumstances under which an equipment functions, and in many engineering applications, the OCs change with the environment or operation modes (Long

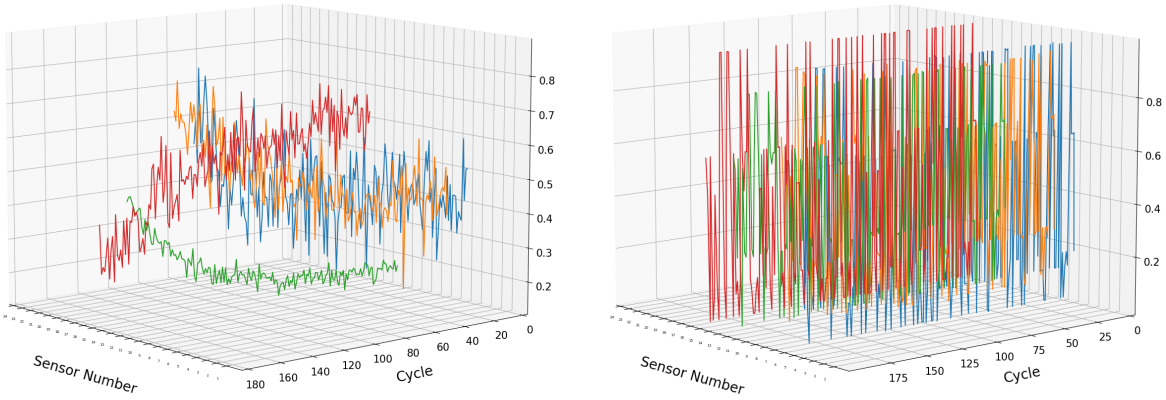


Figure 3.1: Sensor readings from run to failure trajectories under distinct operational scenarios: (left) single operating condition from the FD001 subset, (right) six operating conditions from the FD002 subset of the C-MAPSS dataset (Saxena et al., 2008).

et al., 2022) and often results in discrepancies in the data characteristics (W. Zhang et al., 2021). Figure 3.1 illustrates the sensor readings under varying OCs from the C-MAPSS datasets (Saxena et al., 2008). When examining a single OC, as depicted in the first scenario, the degradation patterns are evident, with the signals demonstrating either an increasing or decreasing trend over time. However, in the second scenario, where six OCs are present, degradation patterns become obscured. Due to the frequent changes in OCs from cycle to cycle, the data points form multiple clusters. These clusters mask the underlying degradation trend, complicating the pattern recognition that is critical for accurate RUL estimation.

For RUL prediction, a range of advanced architectures have been proposed (see section 2.2). However, as outlined in Chapter 2, these architectures often exhibit inconsistent performance across diverse operating conditions. This challenge arises from a focus on developing temporal layers such as RNNs, TCNs, attention modules, hybrids ones, or others, without adequately addressing the variations introduced by different OCs. The assumption that complex layers can simultaneously learn temporal or spatial dependencies and negate variations from OCs has proven to be overly optimistic, especially when models face multiple OCs. While some studies, such as (Pasa et al., 2019), have proposed feature engineering to mitigate these variations, these solutions often require manual intervention and domain expertise. In response to these challenges, our work introduces adding an initial MLP stage, which does not have temporal/spatial complexity, designed to perform feature selection and engineering automatically, aiming to standardize inputs across varying OCs before they are processed by layers that learn degradation patterns.
We propose an end-to-end deep learning model that employs an MLP-LSTM-MLP architecture. This model utilizes LSTM cells, chosen for their proven capability in processing sequential data and learning temporal dynamics. This choice reflects our commitment to leveraging established methods while innovating in areas where current models fall short. This architecture, while simple, aligns with the perspective that well-designed, straightforward neural networks can often rival the performance of their more intricate counterparts.

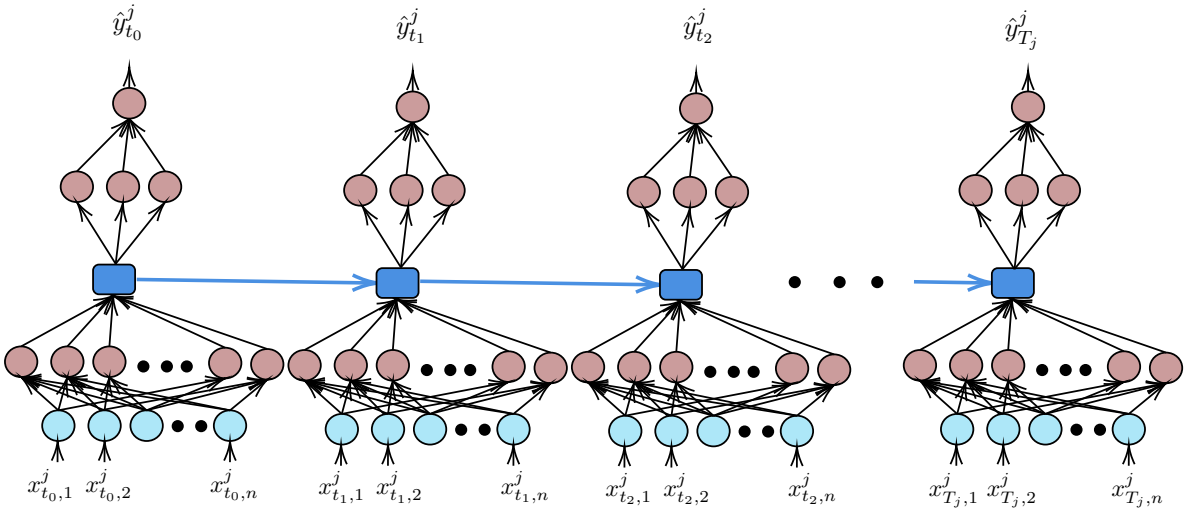


Figure 3.2: Architecture of the proposed model: it takes as input a complete sequence j of raw sensor values, encoded as a tensor x^j composed of T_j time frames with n-dimensional observations each. At training time, this sequence ranges from the first observed time frame t_0 to the last T_j just before the Turbofan j halts. At test time, a single forward pass is performed and the RUL $\hat{y}_t$ is predicted at every time step given the previous observations. To simplify the diagram, only one layer has been drawn for the MLPs.

In terms of related works, The literature showcases a variety of applications for LSTM-based architectures. For instance, (Zheng et al., 2017) proposed a model that combines multiple LSTM layers with a feed-forward neural network to predict RUL. (C.-G. Huang et al., 2019) introduced a bidirectional LSTM to harness temporal information from both past and future contexts. This cell has been further explored in studies such as (D. Huang et al., 2022), (Sateesh Babu et al., 2016), (C. Cheng et al., 2020), and (R. Jin, Z. Chen, et al., 2022). However, to the best of our knowledge, the approaches from the literature do incorporate a stage of preliminary feature selection/engineering or directly feed the raw signals into complex temporal, spatial, or attention layers, which presents fundamental differences compared to our approach.

The structure of this chapter is as follows: Section 2 describes in detail the proposed RUL prognostic architecture. Section 3 assesses the effectiveness of our model by comparing it with other prevalent methods in the field. The chapter concludes with Section 4, summarizing the key findings and implications of our research.

3.2 Proposed model architecture

The proposed model has an MLP-LSTM-MLP architecture trained in an end-to-end manner for RUL prediction. This architecture leverages the strengths of both MLP and LSTM networks, creating an approach that captures the intricate patterns in sequential data typically encountered in RUL prognostics.

LSTM networks are a variant of recurrent neural networks that are designed to overcome their limitation in capturing long-term dependencies in sequential data. LSTM address the gradient vanishing problems in Vanilla recurrent networks by introducing new gates that allow for better control of gradient flow, and better preservation of long-term dependencies, which is needed in applications like RUL prognostic. However, LSTM cells are designed to capture time

Table 3.1: Hyper-parameters of the proposed model

Hyper-parameter	Value
Learning Rate	0.0001
Number of MLP layers before LSTM	3
Number of neurons in MLP layers	100/50/50
Number of LSTM layers	1
Number of LSTM cells	60
Number of MLP layers after LSTM	3
Number of neurons in MLP layers	60/30/1
Activation function for MLP layers	Tanh()
Batch size	5
Dropout percentage	0%

dependencies but they do not have the capacity to handle complex feature processing, which has led some works in the literature to perform this task manually before the learning phase.

This limitation is where the integration of MLPs becomes crucial, as MLP are well fitted to perform such a task. We thus propose to feed all of the raw inputs into an MLP before the LSTM layers. The MLP will be in charge of processing the raw inputs and learning a good representation of each time frame, while the LSTM shall capture the dependencies through time of frame sequences. Then, a final regression head, composed of another MLP, predicts the RUL from these temporally smoothed representations.

Figure 3.2 shows the proposed architecture: each input vector x_t is processed by a first MLP layers, and the resulting sequence of feature vectors is processed by an LSTM layer. The output of each LSTM cell is finally passed to another MLP layers that outputs a scalar y_t that represents the predicted RUL. The weights of the features-MLPs are shared across all time steps, which is convenient when working with variable length sequences.

3.3 Experimental setup

To evaluate our approach, we use the C-MAPSS data sets described in chapter 2 section 2.5. The normalized data is directly fed to the network, without any feature engineering or selection. Therefore, no prior expertise on the equipment or signal processing is required for the proposed method. In order to choose the hyper-parameters of the model, we split the development set into a training subset and a validation subset, based on the ID of the equipment. The original test set is reserved for final evaluation (see 2.5).

Our model does not use neither fixed length sequences nor truncation nor window processing, each training sample is a full time series of one turbofan engine from its first cycle until failure. Henceforth, different samples have variable sequence length. We used 75% of the turbofans run to failure trajectories as training subset, and 25% as validation subset.

The Root mean square error (RMSE) and Score (section 2.5) may be used as loss functions for training. Preliminary experiments on the CMAPSS datasets show that both the score and the RMSE give similar results. So we decided to work with the RMSE because the training process is faster. Hyper-parameters have been tuned manually with a few trials and errors on

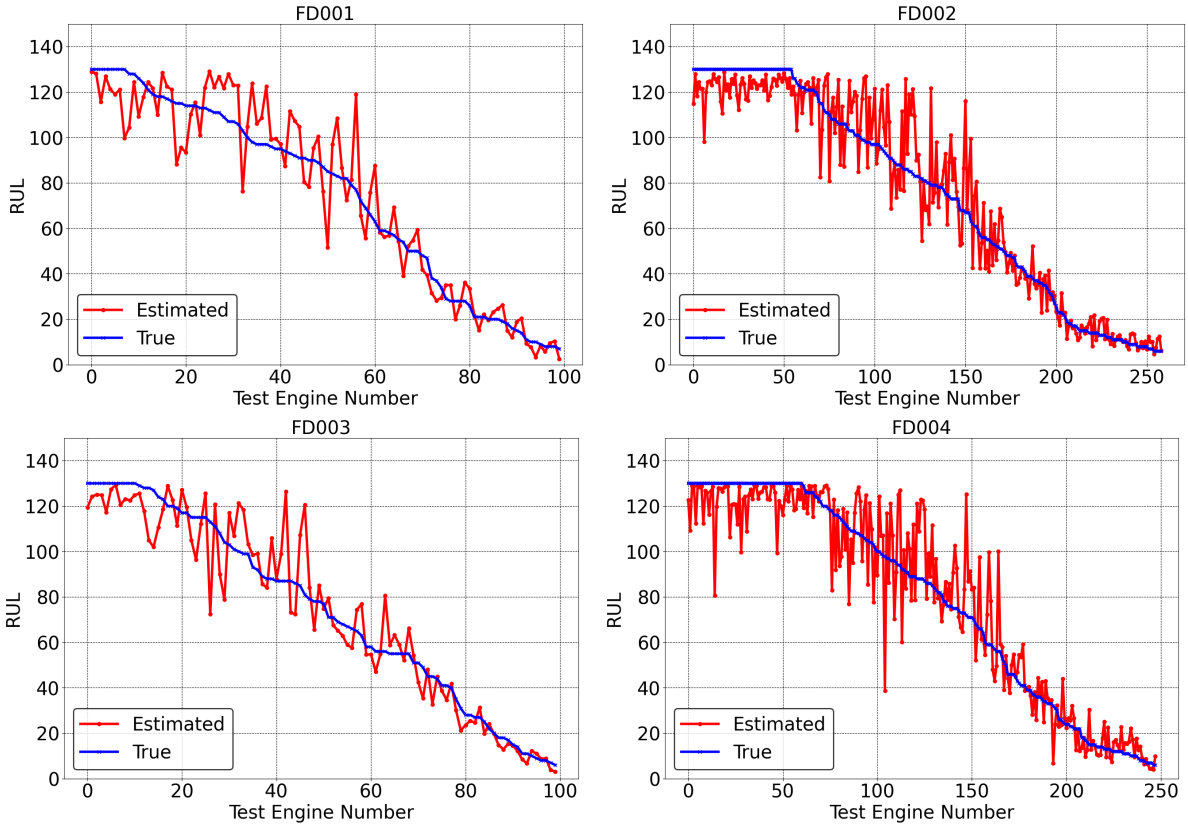


Figure 3.3: RUL Prediction on Test Sets (sorted with decreasing RUL): the figures display the predictions for test sets FD001, FD002, FD003, and FD004

the validation set. The hyper-parameters to be optimized are the learning rate, the number of layers in the input and output MLPs, the number of LSTM layers, the number of neurons/cells in each layer, the activation functions, the dropout percentages and the optimizer. The best hyper-parameters found for the proposed model are listed in Table (3.1).

Because of random initialization, the optimized model parameters values may vary across different training runs. We thus evaluate the model's performances across 10 runs, and the mean values and standard deviations are reported in results tables in the following sections.

3.4 Results and Discussion

In Figure 3.3, the performance of our model is examined across the four test sets, corresponding to FD001 to FD004. Each subplot within the figure shows the estimated and actual RUL for each engine from the test sets, arranged in a descending order based on their true RUL to improve visual clarity.

Across all four subplots, it is notable that the predicted values closely follow the actual values, particularly at the extremes of the engines' life cycles. Specifically, when the RUL is either at a higher threshold (around 120) or below a lower threshold (approximately 55 or 60), the predictions tend to be more accurate. This could potentially be attributed to the patterns of degradation being subtle or very pronounced in the sensor readings at these stages, respectively. However, during the intermediate phase of degradation, typically between a RUL of 110 and

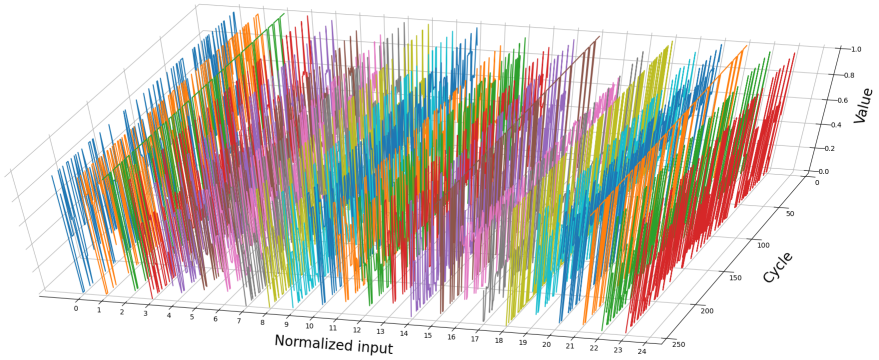


Figure 3.4: Normalized input signals that are fed directly into the model; $n = 24$ sensor measurements of the turbofan unit #13 from the beginning of its life until its failure; this engine data was taken from the 4^{th} data set (FD004) that contains 6 operating conditions and 2 fault modes; we clearly see that these normalized signals do not directly provide visible and interpretable clues for RUL estimation.

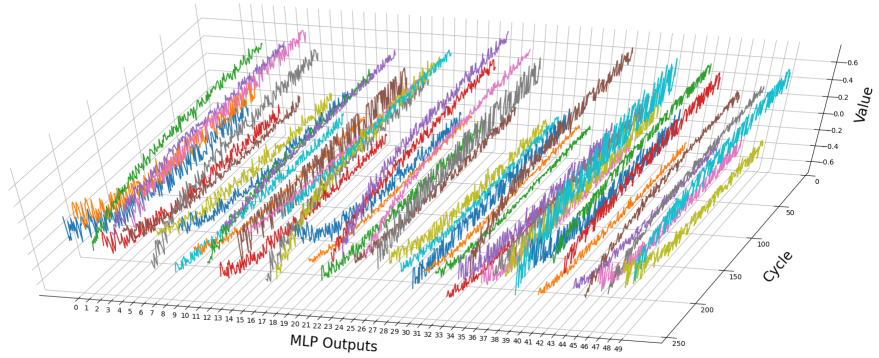


Figure 3.5: This plot presents the 50 features learned by the MLP for unit #13; we can observe trending degradation representations that have been learned from the normalized input signals. Since the first MLP is not time dependent, the learned features exhibit a relatively large variance across time cycles.

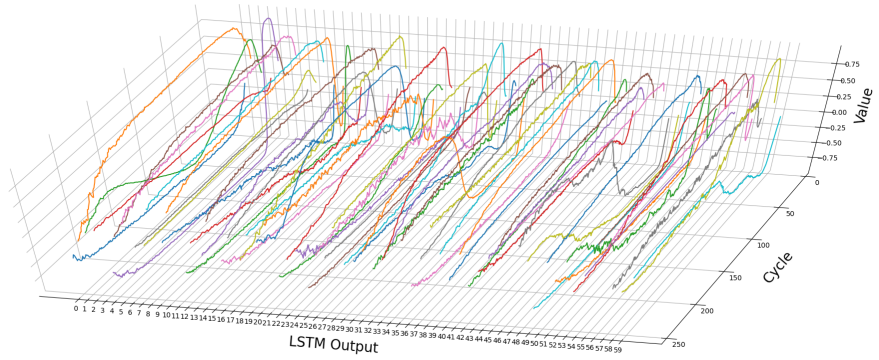


Figure 3.6: The outputs of the LSTM for unit #13 present much smother signals, due to the LSTM's ability to leverage recurrent connection from prior time steps.

Table 3.2: prognostic performance of the proposed model.

DATASET	FD001	FD002	FD003	FD004
RMSE	13.26 ± 0.57	12.49 ± 0.28	13.11 ± 1.28	13.97 ± 0.48
SCORE	284.88 ± 42.32	571.4 ± 37.45	352.39 ± 179.96	1252.32 ± 104.97

60, we observe an increase in prediction errors. This period represents the phase during which degradation patterns start to appear in certain sensors, introducing complexity into the prediction process.

For datasets FD001 and FD003, the frequency and magnitude of large prediction errors are comparatively subdued. Conversely, in FD002 and FD004, the errors are more clear, this is because they encompass larger number of test units (250) in contrast to the 100 units in FD001 and FD003. This increased sample size could contribute to the heightened variability observed in the figure for FD002 and FD004 datasets. Despite the visual differences in prediction accuracy among the datasets, the RMSE values presented in Table 3.2 suggest that our model maintains consistent performance across all datasets, regardless of the variety of operational conditions and failure modes they include.

Thanks to our end-to-end learning approach, the MLP that precedes the LSTM automatically learns a representation of the input data that is relevant to the task of RUL prediction. Figure (3.4) shows the normalized raw input signals of unit #13 from the FD004 data set, where no clear trend can be seen because of the high variance in the data, which is partly due to the operating conditions that vary from cycle to cycle. Figure (3.5) shows the output signals of the first MLP, where noticeable degradation pattern have been learned from the normalized inputs and can be observed. Feeding this learned representation to the rest of the model is more efficient than handcrafting features that require expertise and time. This first representation learning stage is particularly useful when dealing with complex data sets where no clear trend is seen, and also when inputs have a large number of dimensions. After this first MLP, the role of the LSTM layer is to capture temporal patterns and dependencies in the time series. Figure (3.6) shows the signal at the output of the LSTM. We can see that this part of the model minimizes the variance of the learned features across time cycles giving a smoother signals that can be used by the final MLP for RUL estimation.

3.5 Comparison with related works

We evaluate in Table (3.3) our proposed model by comparing its performances with the most recent methods published in the literature that give the best results on the C-MAPSS data set to the best of our knowledge. Although the previously published models are performing well on the first and third data sets (FD001, FD003), with only one operating condition, they perform poorly on the other subsets that have up to 6 operating conditions.

The proposed end-to-end architecture outperforms all other models in complex data sets (FD002 and FD004) as well as on the global results averaged over all datasets. It improves by more than 18% for the RMSE and 39% for the Score on FD002, and 18% for the RMSE and 15 % for the Score on the FD004 data set, as compared to literature results.
Indeed, we can see from the last two rows of Table (3.3) that the results improved significantly

Table 3.3: Performance comparison of related methods with our proposed model on the C-MAPSS benchmark. Methods marked with an asterisk (*) are those published subsequent to our research.

Models	DATASETS								**Average**	
	FD001		FD002		FD003		FD004			
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
DA-CNN (Yan Song et al., 2020)	11.78	229.48	16.95	1842.38	11.56	257.11	18.23	2317.32	14.63	1161.57
DCGAN (G. Hou et al., 2020)	**10.71**	**174**	19.49	2982	**11.48**	273	19.71	3874	15.34	1825.75
MS-DCNN (H. Li et al., 2020)	11.44	196.22	19.35	3747	11.67	**241.89**	22.22	4844	16.17	2257.27
HDNN (Al-Dulaimi et al., 2019)	13.017	245	15.24	1282.42	12.22	287.72	18.15	1527.42	14.65	835.64
LSTM (Pasa et al., 2019)	16.5	444	18.1	942	15.9	718	17.2	1487	16.92	897.75
ADLDNN * (Xiang et al., 2022)	13.05	-	17.33	-	12.59	-	16.95	-	14.98	-
Attention based * (L. Liu et al., 2022)	12.25	-	17.08	-	13.39	-	19.86	-	15.64	-
BiGRU-TSAM * (J. Zhang et al., 2022)	12.56	-	18.94	-	12.45	-	20.47	-	16.10	-
Res-HSA * (J. Zhu et al., 2023)	11.91	-	17.27	-	11.88	-	17.43	-	14.62	-
SARN * (W. Xu et al., 2023)	13.84	-	18.92	-	11.68	-	17.74	-	15.54	-
MSTformer * (D. Xu et al., 2023)	-	-	14.48	-	-	-	15.03	-	-	-
CNN-LSTM * (K. Zhao et al., 2023)	12.69	-	14.15	-	13.04	-	15.78	-	13.91	-
Proposed LSTM without the first MLP	14.31	337.86	17.44	1716.11	15.53	1356.36	18.86	2111.05	16.53	1380.34
Proposed LSTM with the first MLP	13.26	284.88	**12.49**	**571.4**	13.11	352.39	**13.97**	**1252.32**	**13.20**	**615.24**

after adding the first MLP to the architecture. The outputs of the first MLP shows that this part is removing a large part of the variability of the sensor signals that is due to varying operating conditions (Figure (3.5)). This greatly facilitates the work of the LSTM that can focus on temporal smoothing, and then of the final MLP, which role is to achieve prediction. This idea of facilitating the work of the LSTM can also be achieved by feature engineering, as proposed in (Pasa et al., 2019), where they did Operating Condition-specific Standardization, or as proposed in (K. Zhao et al., 2023), where they did Operating Condition-specific normalization plus having a CNN stage before the LSTM. Their results are relatively good, especially for (Pasa et al., 2019), but their approach cannot be performed when details about the operating conditions are

not known, unlike the proposed approach in this work.

The clear decomposition in our model of these three roles is the key to the increased robustness to variable input signals and better final performances. This can be also observed in Table (3.3), where all competing models suffer from a large variability of their performance between the FD001 and FD003 subsets on the one hand, and the FD002 and FD004 subsets on the other hand, while a significantly lower difference in the results between the 4 subsets can be observed with our proposed model.

3.6 Conclusion

In this chapter, we have presented an end-to-end deep learning approach for estimating the Remaining Useful Life (RUL) from multivariate time-series signals. Our method was rigorously tested on the publicly available C-MAPSS dataset, which focuses on predicting the RUL of commercial aero-engine units. Through comprehensive comparisons with several S-O-T-A approaches, our proposed architecture demonstrated superior performance, particularly in complex scenarios involving various operating conditions (RQ1). Notably, it exhibited consistent performance across all four subsets of the dataset.

We also discussed the crucial role played by the first MLP in managing the variability and clustering effects of differing operating conditions. This aspect is particularly significant from an interpretability perspective. While the MLP layers effectively mitigate variability, the potential for enhancing interpretability through operating condition-specific parameters is an exciting prospect. In the following chapter, we aim to explore this possibility by developing a modular network, thereby advancing the interpretability of our RUL prediction model.

Chapter 4

Towards interpreting deep learning models for industry 4.0 with gated mixture of experts

Contribution

In this chapter, we propose to use the GMoE to interpret RUL prediction models. Unlike monolithic deep learning models, gated modular neural networks enable to decompose parts of the models in a way that may potentially be interpreted by domain experts or users. We first propose to transform the proposed model in chapter 3 that gives S-O-T-A performances on a standard industrial benchmark according to this paradigm. Then, we experimentally validate that the performances of the transformed model are not degraded and that the resulting model segments and clusters the data streams according to an emerging concept that reflects previously published analyses by experts on the C-MAPSS dataset, even though such a concept has never been introduced at training time.

4.1 Introduction

Deep Learning (DL) models have significantly advanced the field of prognostics, yet their interpretability, particularly in industrial contexts, remains a challenge. This chapter aims to address the research question: **How can DL models be designed to provide interpretability that aligns with industrial concepts present in the data?** Addressing this question is essential for bridging the gap between advanced DL techniques and their practical, understandable application in industrial settings leading to more trustworthiness.

One system that shows promise in this regard is the Gated mixture of experts (GMoE). This architecture comprises of a set of individual neural network modules without shared parameters and a gated neural network that acts as a soft switch to determine which module will be used for each data sample. This approach has demonstrated multiple potential advantages such as enabling transfer learning (Dobre and Lascarides, 2017), leveraging domain knowledge (Pradier et al., 2021), and facilitating parallelization and distributed computing (Ryabinin and Gusev, 2020).

Most research on GMoE have mainly focused on its overall performance and rarely on its interpretability potential. Unlike monolithic neural networks, this approach is potentially inherently interpretable since the gating networks may select modules in a way that domain experts

or users could understand. Following the taxonomy (Figure 2.2) provided by the review (Yu Zhang et al., 2021) (Discussed in chapter 2, section 2.3), GMoE can be viewed as an active approach where we try to explain what the model learns and its predictions by unveiling part of the hidden semantics. The GMoE may also be viewed as enabling global interpretability when the task decomposition is perfect, but so far, according to literature results, it can be considered as semi-local only as it often fails to decompose the task perfectly. A few papers investigate these capabilities but their studies focus on classification problems or on cases where the modular gated neural network is used as a whole model.

This chapter contributes to this area of research by investigating the potential of GMoE's inherent interpretability in the context of a regression task (RUL prediction) and when the GMoE is applied to a sub part of the model. We explore whether the vanilla GMoE architecture can decompose the task in an interpretable way, and we introduce and investigate a way to incorporate human knowledge (*i.e.* through a prior distribution) into the approach. In both cases, we provide a detailed analysis of what the gating network learns, showing that this approach can indeed produce an interpretable but not perfect decomposition, and also, we show that the proposed way of integrating human knowledge into the approach could significantly improve the quality of the task decomposition.

The rest of this paper is structured as follows. Section 2 introduces related works on interpretability with GMoE. Section 3 describes the GMoE approaches for interpretability. Section 4 highlights the results of the proposed approaches in terms of interpretability and predictive performance. Finally, conclusions and discussion are provided in section 5.

4.2 Related work

Gated modular neural network (GMNN) or GMoE is an approach that has been around for three decades (Jacobs et al., 1991) that is based on the fact that dividing a task into appropriate sub-tasks seems to make it easier for users/humans to understand and debug.

GMoE systems have gained a lot of attention in the last few years, particularly in the field of NLP, where architectures like GShard (Lepikhin et al., 2020) and Switch Transformers (Fedus et al., 2022) have demonstrated significant improvements in model scalability and efficiency. These advancements have positioned this system as the preferred architecture for large-scale NLP models, not primarily for their interpretability, but for their ability to scaling models while maintaining a constant number of computational operations (Jiamin Li et al., 2023).

To the best of our knowledge, few articles investigate the interpretability potential of the GMoE approach. (Eigen et al., 2013) stacked two GMNNs into a single architecture to predict the class label on a randomly translated version of the MNIST dataset (LeCun and Cortes, 2010); their results show that this approach can learn to develop location-dependent experts at the first layer, and class-specific experts at the second layer. Using a toy example with a 2D 6-classes Gaussian mixture, (Krishnamurthy and Watkins, 2021) show that this approach may in some cases produce an interpretable task decomposition; to confirm these results, they did further experiments on a modified version of the MNIST and FMNIST datasets (LeCun and Cortes, 2010; H. Xiao et al., 2017), and to some extent, the model learns to allocate tasks among experts in an interpretable way, but this allocation remains unpredictable as it varies from one experiment to another.

Interpreting the model might be particularly important for more complex and realistic tasks. Hence, (Z. Huo et al., 2021) used a GMoE model with a sparse gating mechanism in a medical use case; by embedding and visually analysing the output of this gating network, they were

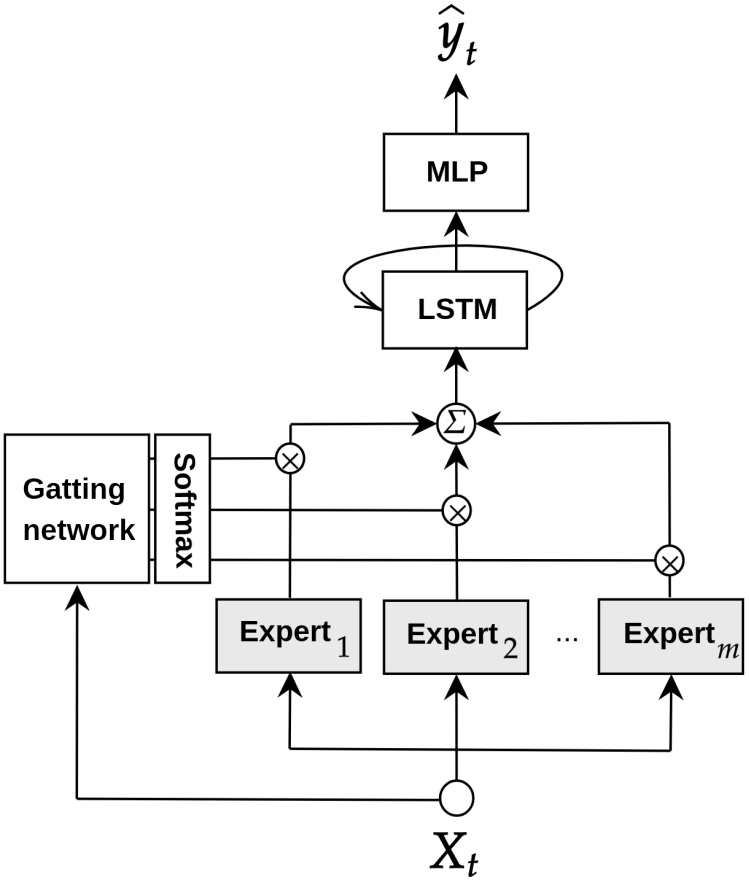


Figure 4.1: GMoE-LSTM-MLP architecture with m experts.

able to aid interpretation of patient sub-type separation. In another use case, by checking the agreement or disagreement between individual experts outputs, (Pavlitskaya et al., 2020) used the GMoE approach to gain insights into decision making process for semantic segmentation.

4.3 GMoE approaches for task decomposition

The GMoE is a system of m experts $o_i(\cdot)$ with $i \in \{1, ..., m\}$ and a Gating network (GN) $g(\cdot)$. Every expert processes the same input vector x but returns a different output vector $o_i(x)$. Typically, the gating network computes the posterior $p(i|x)$ from the same input x with a softmax:

$$g(x) = [p(1|x), \ldots, p(m|x)] \tag{4.1}$$

The final output of the system is computed as:

$$f(x) = \sum_{i=1}^{m} p(\, i \,|\, x \,) \times o_i(\, x \,) \tag{4.2}$$

The GMoE can serve as a standalone model or be integrated as a sub-layer within a larger neural network (Kirsch et al., 2018). In this work, we incorporate it into the MLP-LSTM-MLP

architecture proposed in the previous chapter 3. We substitute the initial MLP with the GMoE, where the original MLP stage addresses the variability caused by different OCs in the context of remaining useful life prediction. Consequently, the architecture is modified to GMoE-LSTM-MLP as presented in figure 4.1. This modification aims to organize the data by OCs, with each expert in the GMoE system is dedicated to processing data from a specific OC.

4.3.1 Simple GMoE

In our context, the simple GMoE involves only a change in architecture, aiming to explore how the gating network decomposes data with just an architectural adjustment. Let $x_1, \ldots, x_n \in \mathbb{R}^p$ be a training dataset consisting of n observations with p features, and $y_1, \ldots, y_n \in \mathbb{R}$ the corresponding gold values to predict (e.g., the remaining life time). The simple GMoE model $\hat{f}$ is trained with empirical risk minimization to maximize its performances with regard to the prediction objective:

$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n} \sum_{j=1}^{n} \ell(f(x_j), y_j) \tag{4.3}$$

where $\mathcal{F}$ is some model family and ℓ a loss function.

4.3.2 GMoE with constraints based on domain experts knowledge

In our context, the GMoE with constraint refers to both an architectural adjustment and the incorporation of an additional term into the loss function during training. In related works (Krishnamurthy and Watkins, 2021; Shazeer et al., 2017; Kirsch et al., 2018), and as also shown in the results section below, relying on the gated network alone often leads to the problem of low diversity in experts usage, as the gating network tends to converge to a state where it always produces large weights for the same few experts. This imbalance is self reinforcing: when favored experts are trained rapidly in the beginning of training, they will be more and more selected by the gating network, and thus the gap will increase resulting in poor data decomposition.

Multiple approaches have been introduced to solve this imbalance (Shazeer et al., 2017; Bengio et al., 2015). In this work, we choose to address this issue by integrating basic knowledge (frequency distribution of concepts present in the data) as a form of constraint to force diversity in the use of experts, this relates to the field of weakly supervised clustering where the goal is pick out the "right" way of clustering the data (Wager et al., 2015). Our idea involves adding a posterior constraint to the loss function that encourages the frequency distribution of Experts $\{p(\cdot|x_j)\}_{1 \leq j \leq n}$ to match a prior. Inspired by (T. Kim et al., 2021), we use a Mean square error (MSE) loss for this additional term. The new loss function $\mathcal{L}'$ is defined by (4.4):

$$\mathcal{L}' = \ell(f(x), y) + \lambda' * \mathrm{L2}(\hat{\Omega}, \Omega) \tag{4.4}$$

Where $\hat{\Omega}$ represent the frequency distribution of experts, Ω represents the prior distribution, and λ' is a scalar hyper-parameter controlling the strength of the constraint.

We compute an end-to-end differentiable frequency distribution $\hat{\Omega}$ of experts in two steps: first, normalizing the gated network logits with a soft-max with low temperature approximates a one-hot vector where the dominant expert (with the highest probability) has a value close to 1 and the others have values close to 0. Second, we sum these approximated one-hot vectors across the batch to obtain $\hat{\Omega}$. The soft-max operation with temperature T is defined as:

$$\sigma(z^i) = \frac{\exp(z^i/T)}{\sum_{j=1}^{m} \exp(z^j/T)} \tag{4.5}$$

Where z^i are the logits corresponding to each expert for a given input, and the temperature T has been arbitrarily set to 0.001. $\hat{\Omega}$ can be defined as:

$$\hat{\Omega} = \left\{ \sum_{k=1}^{N} \sigma(z_k^1), \ldots, \sum_{k=1}^{N} \sigma(z_k^m) \right\} \tag{4.6}$$

Where N is the batch size, and m is the number of experts. Each component of $\hat{\Omega}$ represents the aggregated activation (or usage frequency) of each expert across the batch, creating a histogram of expert utilization.

4.4 Experimental setup

Building upon the work presented in Chapter 3, our experimental approach utilizes the MLP-LSTM-MLP model architecture previously described. In this chapter, our focus shifts to interpreting the model's ability to handle varying operating conditions between cycles, a crucial aspect of RUL prediction. To this end, we have selected the FD002 subset of the C-MAPSS dataset (detailed in Section 2.5), which presents a scenario of six operating conditions and one failure mode. This subset is particularly relevant to our study as it mirrors the complexity of industrial settings we aim to analyze – multiple operating conditions and a singular fault mode. This specific combination provides a clearer context for our interpretability analysis, avoiding the potential confounding factors that multiple fault modes might introduce.

To enhance the interpretability of this stage of the model, we propose replacing the initial MLP stage with a GMoE. This adaptation aims to shed light on the clusters created by the gating network. The revised model architecture, as illustrated in Fig. 4.1, now follows a GMoE-LSTM-MLP structure. In this setup, both the experts and the gating network within the GMoE are MLPs. We maintain the same hyper-parameters as established in Chapter 3 to maintain consistency in the experimental conditions. The architecture of the initial MLP is replicated for the experts and the gating network of the GMoE.

The inclusion of GMoE in the first layers should play a key role in breaking down the various operating conditions represented in the turbofan measurements. By segmenting the data into distinct groups, GMoE will be able to find/discover that the turbofan measurements were made under several operating conditions. This will lead to a better understanding of data processing mechanisms, making model predictions more interpretable and in line with industrial concepts (OCs).

The GN generates a distribution over experts at each time step. The chosen expert, i.e., the one with the highest probability determined by the argmax of this distribution, represents the class for the input, thus creating a clustering of all the available data. For each input, we also have the known OC, which is the real value associated with that input. Our objective is to determine whether the clustering generated by the GN accurately represents the real OCs. Given that the number of clusters identified by the GN may differ from the actual number of OCs, we utilize the Normalized mutual information (NMI) for this comparison. The NMI is defined as follows (Kvalseth, 1987):

$$\begin{aligned} NMI(Y,C) &= \frac{2 \times I(Y,C)}{[H(Y) + H(C)]} \text{ , With:} \\ I(Y,C) &= H(Y) - H(Y \mid C) \end{aligned} \tag{4.7}$$

where, Y represents the true class labels (the real OCs), C represents the cluster labels determined by the GN, H is the entropy, and I is the mutual information between the true class labels and the cluster labels. The NMI is an external measure between 0 (no mutual information/ independent clustering) and 1 (perfect correlation/ same clustering).

We know from (Saxena et al., 2008) that six distinct OCs occur in this dataset. However, in practical applications, the exact number of OCs might not be known a priori. Often, the determination of these conditions relies on expert judgement, which can introduce variability and potential inaccuracies due to the subjective nature of human assessment. We experiment next with 6 and 9 experts to assess the robustness of our approach to an erroneous prior about the number of OCs. The choice of 9 experts, in particular, allows us to observe the model's behavior when it has a surplus of resources, thus testing its ability to decompose and interpret the data according to the task at hand (RUL prediction). We recognize that due to the stochastic elements of model training, such as random initialization of parameters, outcomes can vary across different training iterations. To account for this and provide a robust estimate of performance, each experimental configuration is replicated 20 times, enabling us to calculate statistical variance and gain insights into the consistency of the model. We employ an early stopping criterion to mitigate over-fitting, with training ceasing at 2000 epochs. This approach ensures that the model selected for testing is the one that has demonstrated the lowest validation loss.

In the following plots that present the results of our experiments,, the X-axis represents the number of experts actually used by the gated network, or in other words, the number of clusters predicted by our gated network $g(\cdot)$. Indeed, the gated network computes a posterior over the experts $g_i(x) = p(i|x)$ with $1 \leq i \leq m$, and we can thus associate to every input x a dominant expert via $\arg\max_i g_i(x)$. The number of clusters N_c is thus the total number of experts that are dominant over the whole corpus:

$$N_c = |\{\arg\max_i g_i(x_j)\}_{1 \leq j \leq n}|$$

Every plot is structured in 3 rows and 2 columns:

- A maximum of $m = 6$ (resp. $m = 9$) experts is set in the left (resp. right) column.
- The top row shows the histograms of the number of predicted clusters N_c over the 20 experimental runs realized.
- The middle row shows the mean and standard deviation of the NMI between the predicted clusters and the operating conditions.
- The bottom row shows the root mean square prediction error values (RMSE) on the test data; for comparison, a green rectangle presenting the mean and standard deviation of the state-of-the-art RMSE from chapter 3 is also shown.

4.5 Results and discussion

4.5.1 Simple GMoE results

Figure 4.2 presents an evaluation of the clustering capability of the simple GMoE-LSTM-MLP architecture. The results indicate a tendency for the gating network to predominantly rely on one or two experts during most runs, accounting for more than 80% of all cases. With six experts available (left column), a single expert is used 40% of the time and two experts 40% of the time,

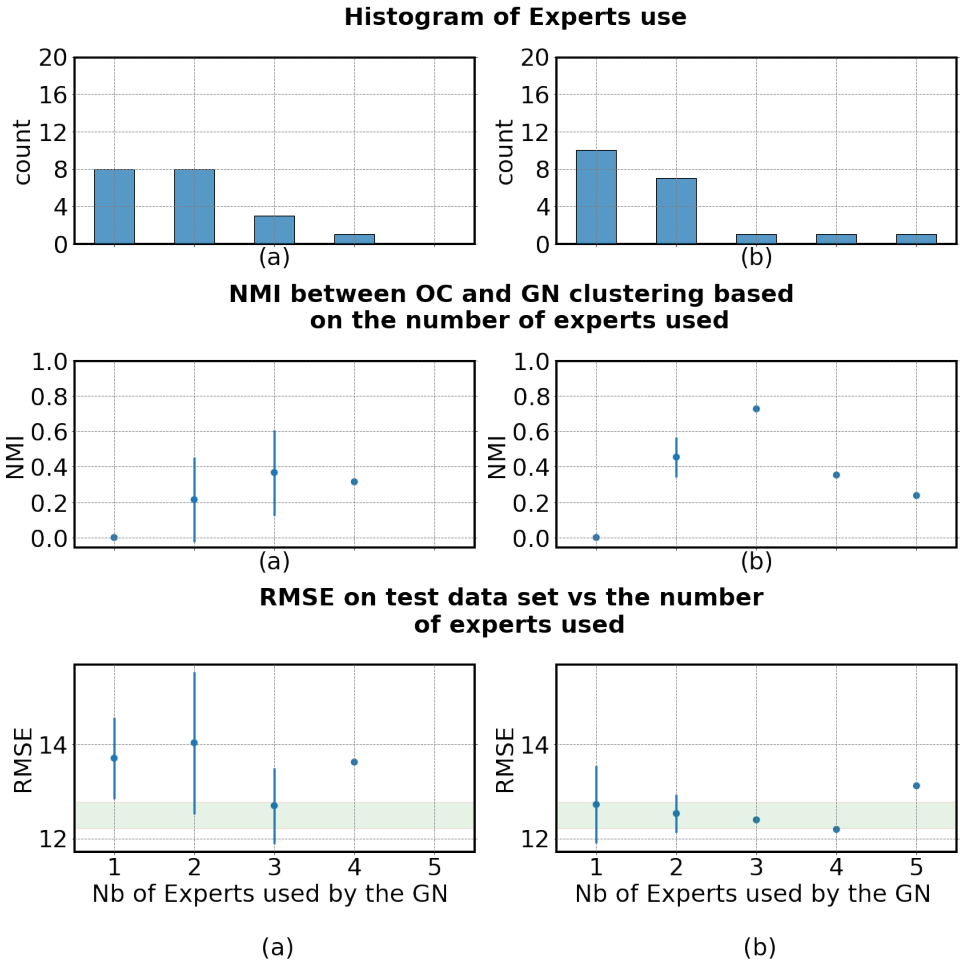


Figure 4.2: Clustering evaluation with the simple GMoE-LSTM-MLP; left column (a): $m = 6$; right column (b): $m = 9$

while three experts are utilized in 15% of the cases and four experts in 5%. The NMI score peaks at a moderate 0.5, exhibiting an ascending trend from one to three experts used, followed by a small decline when four experts are engaged. The performance of the model, as indicated by the RMSE values, shows a slight decrease in performance when compared to the S-O-T-A results. However, with three experts used, the RMSE values are competitive with those achieved without the GMoE system. When the GMoE has access to nine experts (right column), it may utilize up to $N_c = 5$, although such instances are infrequent (5% of runs). One expert is used 50% of the time and two experts 35% of the time, while three to five experts are used in 5% of cases. The NMI score approaches 0.8, showing an ascending trend from one to three experts used, with a subsequent descending trend when more than three experts are employed. The RMSE values are close to those obtained without the GMoE system, except for a small decrease in performance when five experts are used.

These observations suggest that the model can identify distinct operational regimes within the data, but it may not fully capture the complex interplay of conditions that characterize the dataset. Despite the model's ability to employ multiple experts, the finer nuances that distinguish between six unique operating conditions are not as sharply defined by the clustering performed by the gating network. The results demonstrate that while the GMoE can partially

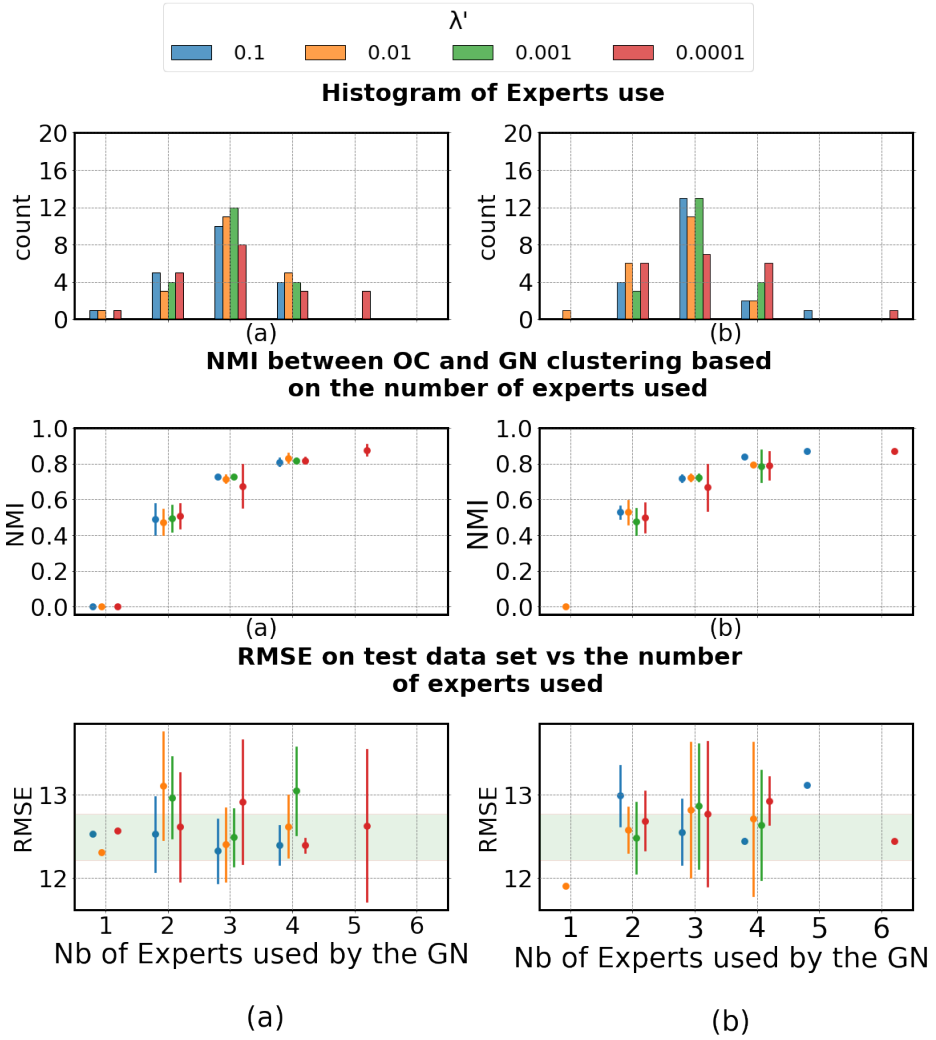


Figure 4.3: GMoE-LSTM-MLP with knowledge based constraint results, the constraint is not used for best model selection; left column (a): $m = 6$; right column (b): $m = 9$

retrieve the concepts present in the data, it still suffers from a low diversity in expert usage. When a significant number of experts are used, the NMI does not reach high values, leading to a problem of poor interpretability. Therefore, the model might benefit from a more sophisticated mechanism that encourages a more balanced and diverse expert utilization to improve both performance and interpretability.

4.5.2 GMoE with knowledge-based constraint results

We assume a uniform prior frequency distribution Ω for the constrained loss in Equation (4.4). This choice is informed by the empirical distribution of the OCs, which is approximately uniform for the majority of categories, with all OCs showing similar frequencies, except for a particular OC that exhibits a frequency double that of its counterparts. While our prior does not perfectly mirror the observed distribution, it can offer a partial solution to the low diversity in experts usage.

This loss is utilized to train the model parameters to optimize the main objective, RUL

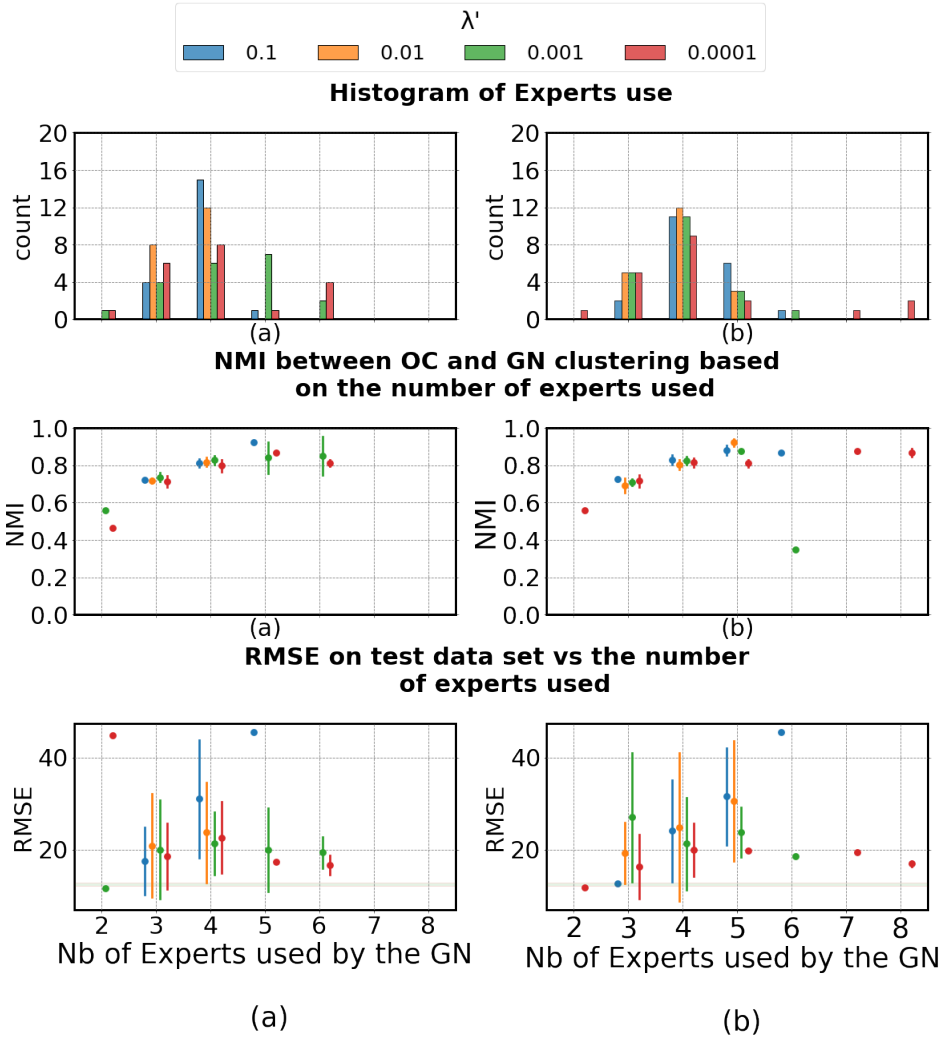


Figure 4.4: GMoE-LSTM-MLP with knowledge based constraint results, the constraint is used for best model selection; left column (a): $m = 6$; right column (b): $m = 9$

prediction, while also inciting the model to decompose the process into independent experts according to the prior Ω. The prior Ω is dependent on the number of experts m utilized within the architecture. Specifically, for $m = 6$ experts, the model is encouraged to use all experts equally due to the uniform distribution assumption. A similar encouragement is applied when $m = 9$ experts are present.

Results when using unconstrained loss for best model selection

Fig. 4.3 illustrates the clustering quality and performance of the models when we employ the unconstrained loss in Equation (4.3) (the primary supervised task objective) for selecting the best model. We observe from figure 4.3 that three clusters are predominantly predicted, regardless of the number of experts incorporated into the architecture. These clusters align more accurately with the OCs, achieving an NMI of 0.7. Notably, as the number of experts increases, the NMI also rises in a logarithmic fashion, diverging from our initial experiments without constraints. Thus, training with this constraint appears to further encourage the model to decompose the data into

an interpretable format, particularly when a higher number of experts is employed. Moreover, the number of predicted clusters does not exceed the number of OCs, even when the model is encouraged to utilize all nine experts, as shown in Fig. 4.3(b). This suggests a harmonious scaling of the model's complexity with the increasing expert count without over-complicating the clustering beyond the actual number of OCs.

In terms of model predictive performance, we note a consistency across different conditions, with RMSE values that are in close proximity to the S-O-T-A benchmarks denoted by the green rectangle. Additionally, alterations in the strength of the constraint seem to have a negligible effect on the model's interpretability or performance. This stability indicates that the model's predictive capabilities are robust to changes in constraint strength, maintaining efficacy in its primary task of RUL prediction while accommodating the interpretability aspect via clustering.

Results when using constrained loss for best model selection

Experimental results derived from employing the knowledge-based constraint, as defined in Equation 4.4, for the selection of the best model are discussed herein. This process involves analyzing models that are chosen based on their capability to predict the RUL while adhering to the constraints imposed by prior knowledge encapsulated in the loss function. Fig. 4.4 presents the outcomes of this experimental approach, highlighting the implications for model interpretability. A noteworthy observation is that the utilization of the knowledge-based constraint yields a model that predominantly predicts four clusters, which corresponds to a heightened NMI of approximately 0.8. Additionally, the prevalent issue of a low diversity in expert usage observed in previous experiments is ameliorated. The frequency of utilizing two or less is low, while the frequency of using three to five experts is high, and notably, the NMI score exhibits a logarithmic increase, underscoring a superior correlation between the model's clusters and the actual OCs. However, we remark that in some cases, specifically, when the model is trained with $m = 9$ experts and best model selection is guided by a loss that motivates the use of all experts, the model may use more than six experts. This phenomenon suggests that the model is over-complicating the clustering beyond the actual number of OCs.

This enhanced interpretability, however, comes at the expense of predictive performance. A significant decline is observed across most conditions evaluated, with the RMSE on the test set considerably exceeding the S-O-T-A benchmarks. This trade-off indicates that while the integration of the knowledge-based constraint enhances the model's ability to dissect the data into meaningful clusters, it may simultaneously limit the model's predictive performance.

Effect of the constraint on best model selection

We examine the impact of applying the constraint loss on model selection. Figure 4.5 presents the RMSE scores and the epochs at which the optimal model is identified, with and without the constraint loss. The results clearly demonstrate that the inclusion of the constraint loss significantly influences both the performance of the selected model and the timing of its selection. When using only the primary objective for model selection, defined by Equation (4.3), we observe that the best models are typically identified around epoch 500. In this scenario, the RMSE performance is competitive, falling within the range of $11.0 <$ RMSE < 15.0. Conversely, when incorporating the constraint into the criteria for best model selection, the epoch at which the optimal model is determined becomes considerably more irregular, ranging from the very first epoch to the maximum allowed training duration of 2000 epochs,regardless of the weight of the constraint. This variation implies that the convergence during training is highly sensitive to

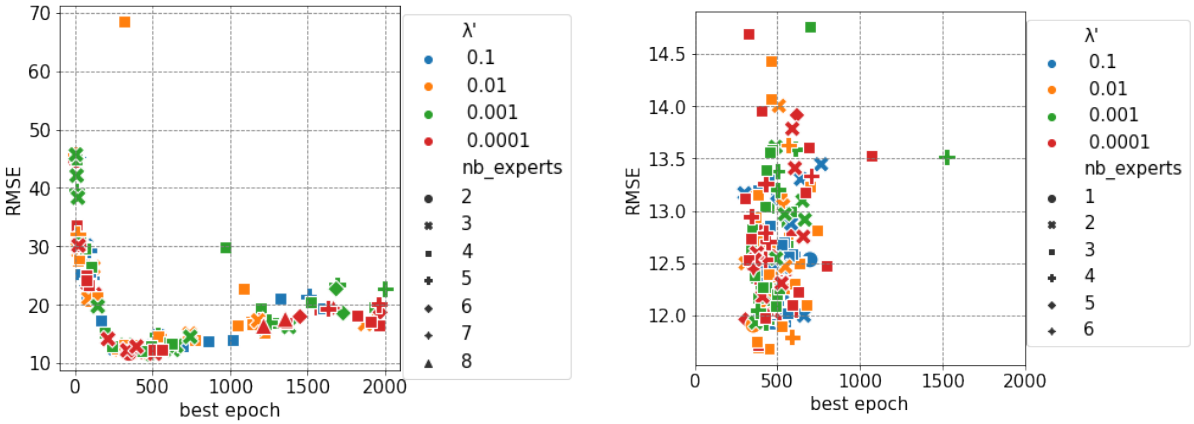


Figure 4.5: Performances of the GMoE with the knowledge based constraint based on the epoch where the best model is found, the constraint strength and the number of predicted clusters, the constraint value is either part (left) or not (right) of the validation loss.

the initial conditions set by the random initialization. The inclusion of the constraint term not only affects the stability of the convergence but also introduces significant variability in the RUL prediction performance, with RMSE values spanning a broad spectrum from 11.0 to 50.0.

Based on these observations, we recommend using the constraint term strictly for learning model parameters. It should not be used for selecting the best model, where the focus should be on optimising the main prediction task. This approach mitigates the risk of the constraint dominating the main loss, which could otherwise lead to sub-optimal predictive performance, as indicated by the increased variability of the RMSE values.

4.5.3 RUL prediction Performance comparison with related works

Table 4.1 illustrates the prediction performance of our model, which has been trained using the knowledge-based constraint defined in Equation (4.4). Notably, for the selection of the best model, only the primary loss is considered. Despite the incorporation of the constraint within the GMoE framework, we observe a marginal performance decrement; specifically, the mean RMSE experiences a slight increase of 0.10 and 0.23 when utilizing 6 and 9 experts, respectively. Also in terms of standard deviation, we remark an important increase, this can be attributed to working with GMoE system, which can lead to more variability due to the number of experts used and the number of parameters added. This performance is juxtaposed with the results presented in Chapter 3, thereby establishing a benchmark for comparison.

Moreover, when positioned against findings from other research contributions considered to be S-O-T-A, our constrained GMoE model maintains a superior performance on the C-MAPSS FD002 Dataset. This is evidenced by lower RMSE values compared to HDNN (Ruiz-Tagle Palazuelos et al., 2020) and CapsNet (Al-Dulaimi et al., 2019) papers, which represent the leading performances in the current literature to the best of our knowledge.

The slight differences in performance observed when integrating the interpretability constraint are instructive. They suggest that the constraint's impact on the model's ability to predict RUL is minimal, thereby allowing the model to retain a high level of accuracy while also gaining the benefits of increased interpretability.

Table 4.1: Performance comparison with related methods: RMSE on the C-MAPSS FD002 test data. Standard deviations are given when available. (Methods marked with an asterisk (*) are those published subsequent to our research.)

Approach	RMSE
HDNN (Al-Dulaimi et al., 2019)	15.24
CapsNet (Ruiz-Tagle Palazuelos et al., 2020)	16.30 ± 0.23
MSTformer *(D. Xu et al., 2023)	14.48
CNN-LSTM *(K. Zhao et al., 2023)	14.15
MLP-LSTM-MLP (Chapter 3)	12.49 ± 0.28
GMoE-LSTM-MLP (m = 6) with constraint	12.59 ± 0.54
GMoE-LSTM-MLP (m = 9) with constraint	12.72 ± 0.62

4.6 Conclusion

In this study, we implemented the Gated mixture of experts (GMoE) system to enhance the interpretability of our DL model for RUL prediction. This approach directly contributes to the research question posed earlier. Through comprehensive evaluations of various GMoE configurations, we gained insights into the effect of different initial conditions on the training process's convergence and the clarity of the resulting clusters. Our proposed methodology has proven capable of achieving more interpretability, which importantly does not compromise the predictive performances. On the contrary, our approach have succeeded in forming interpretable clusters that are aligned with the core predictive task. However, it is crucial to acknowledge that achieving perfect alignment consistently between the clustering made by the GN and the real operating conditions is still not achieved and further research is needed.

While interpretability remains a major challenge, the difficulty of developing models with limited data is equally critical, posing a significant barrier to the wider application of these techniques in industry. The next section of the thesis will focus on tackling this issue. Building on the concept of modular networks discussed in this chapter, we plan to design models that can effectively utilize datasets with variable input and output structures.

Overcoming Data Scarcity using auxiliary data-sets

Chapter 5

Auxiliary training for prognostics

Contribution

In this chapter, we propose an auxiliary training approach that integrates auxiliary objectives from related but distinct data sets. This approach enriches the learning process, utilizing knowledge from a broader data range and acting as a regularization mechanism to improve generalization from limited data. Our methodology introduces task-specific projection layers designed to handle the complexities of diverse industrial data-sets.
The effectiveness of the proposed method is demonstrated by experiments on two well-known public data sets, CMAPSS and N-CMAPSS, across eight distinct settings, and is shown to outperform state-of-the-art approaches such as single-task learning and pre-training followed by fine-tuning.

5.1 Introduction

DL models are typically trained using large amounts of labeled data and can achieve impressive results on a wide range of tasks. However, building such models for prognostics can be challenging, particularly when data is limited on a specific use case. Indeed, in many use cases, data for specific equipment or systems may be scarce for several reasons. Industrial equipment are often maintained in a preventive manner which reduces the occurrence of failures. Additionally, even though there may be a large amount of data available from monitoring of the system, most of the data represents normal operation. The degraded and failures states of the industrial system, as they lead to unwanted product are most of the time largely under represented. Finally, it is often not possible to obtain "run-to-failure" data from a "real" process, where the system is allowed to run until it fails, because it's costly and time-consuming (Eker et al., 2012). Unlike fields such as computer vision or natural language processing where pre-trained models on large datasets lay the groundwork for new applications, PHM lacks these foundational resources. The available datasets are fragmented, emanating from various equipment types with different input features, lengths of RTF trajectories, and often tailored more for research than practical field use. This fragmentation leads to a bottleneck in advancing DL applications within the PHM domain.

Given these settings, when faced with developing a DL model for a particular case, the process often must begin anew. Nevertheless, we may have at our disposal external labeled datasets (public or private) that could be harnessed. This opportunity leads us to this research question: **How to better leverage external data to develop RUL prediction models?**

Addressing this question, we propose an approach based on Auxiliary training (AT). Auxiliary training (AT) is a learning paradigm focused on improving the generalization of a single primary task through the use of additional objectives. The role of auxiliary tasks is to assist the primary task, and at the time of testing, only the primary task is considered. we perform AT by adding auxiliary objectives that are based on related data sets, the goal is to allow the model to learn broader patterns from all tasks and apply this enriched knowledge on the primary one.

This chapter is structured as follows: section 5.2 provides a summary of related work, Section 5.3 introduces the proposed method of utilizing an auxiliary data-set for learning, Section 5.4 showcases the experimental setup and performance evaluation results for the proposed approach, and finally, Section 5.6 concludes the article.

5.2 Related work

Multi-task learning (MTL) is a technique that aims to extract shared feature representations or modules for related tasks. Unlike learning separate networks for each task independently, MTL allows for the extraction of correlated information from multiple tasks, leading to substantial enhancement in network performance for each individual task. One such approach is the multi-task deep neural network proposed by (Xiaodong Liu et al., 2019), which combines MTL and language model pre-training to achieve S-O-T-A results in various natural language understanding tasks compared to the original single-task deep neural network setting. In terms of RUL prediction, (Yan et al., 2023) proposed a method for learning RUL prediction alongside health state estimation, while (Huaqing Wang et al., 2022) focused on the joint learning of RUL prediction and fault detection. In the context of few-shot learning, (Weller et al., 2022) demonstrated that MTL can outperform intermediate fine-tuning in natural language processing tasks when the main task is smaller than the supporting task. Furthermore, (Haoxiang Wang et al., 2021) provided theoretical and empirical evidence to support the claim that, under certain conditions, MTL can compete with S-O-T-A gradient-based meta-learning algorithms in few-shot image classification benchmarks.

In contrast to MTL, **Auxiliary training** is focused on improving the generalization of a single primary task by utilizing additional tasks. The role of the auxiliary tasks is to assist the primary task, and at test time, only the primary task is considered. Auxiliary training methods can use simple auxiliary tasks that are based on the primary task data, or entirely different ones. For example, (L. Xu et al., 2021) trained a primary task of semantic segmentation alongside two auxiliary tasks, multi-label image classification, and saliency detection, using only the main task image-level ground-truth labels. In a similar use of primary task data, (Linfeng Zhang et al., 2020) employed augmented data as auxiliary tasks to enhance the accuracy and robustness of image classifiers, (T. Lin et al., 2021) proposed a fault classification-assisted RUL prediction network based on multi-task learning and auxiliary training, (S. Liu et al., 2019) proposed a self-supervised approach based on meta-learning and auxiliary training to enhance the performances on multiple image classification benchmarks. Their approach involves training a multi-task network that performs the primary task and the auxiliary task in parallel, along with a label generation network that generates labels for the auxiliary task to improve primary task performance. In another approach to auxiliary training, (Watanabe et al., 2022) proposed using multiple data-sets as auxiliary training tasks for named entity recognition.

Pre-training followed by fine-tuning has been shown to be effective in a variety of applications. It involves copying the weights from a pre-trained network and tuning all/part of them on a downstream task. In the Prognostics and health management field, several studies (Y.

Deng et al., 2021; Ansi Zhang et al., 2018; S. Yao et al., 2023) have demonstrated the usefulness of this approach, using it to improve the performance of a DL model on the prognosis and diagnosis of multiple industrial equipment. For instance, (Couture and X. Lin, 2022) employed a pre-trained Convolutional Neural Network (CNN) for feature extraction, opting to retrain just the final layer to align with their target outputs. In a similar vein, (Behera and Misra, 2023) utilized three different pre-trained CNNs in a multi-modal fashion to enhance the performance of RUL prediction. This is related to our work in terms of information transfer between support (auxiliary) data-set and the target (main) one. However, the features learned during pre-training are not always tailored to the target task. Furthermore, tasks with insufficient training data often encounter rapid over-fitting during the fine-tuning process.

The characteristics of industrial data-sets and use cases, as well as the lack of a large pre-trained model for this kind of problems, motivated us to propose a solution under the auxiliary training paradigm. Our approach is described next.

5.3 Proposed approach

This chapter presents a method that leverages auxiliary training to augment model performance, i.e. the use of related data as auxiliary task, alongside the limited number of samples from the main task. The aim is to achieve improved performances on the main task by leveraging the insights and patterns captured from the auxiliary data.

For simplicity and also to compare the approach with others, we consider that there is one single related auxiliary data-set. Let $D_A{=}(x_i^A, y_i^A)_{i=1}^{I}$ and $D_M{=}(x_i^M, y_i^M)_{i=1}^{I'}$ be the auxiliary and main sets, containing I and I' data samples respectively, where $I' < 20$. The choice of having less than 20 samples in the main set, aligns with the common practice in Few-Shot Learning (Z. Li et al., 2017; Qi et al., 2018). Furthermore, this choice is designed to simulate the conditions often encountered in many real-world scenarios for RUL prediction. Let f be a neural network model parameterized by θ, capable of processing an input data sample x from either dataset and producing a corresponding output y. The loss function $\mathcal{L}$ for auxiliary training can be defined as (P. Wu and Dietterich, 2004):

$$\mathcal{L} = \ell(y^A, f(x^A)) + \ell(y^M, f(x^M)) \tag{5.1}$$

where ℓ is the task-specific loss. The first term in the loss function corresponds to the auxiliary task, and the second term corresponds to the main task. In the process of jointly optimizing this loss function for both tasks, our primary concern is not the performance on the auxiliary task. Instead, we are interested on the outcomes of the main task. This focus guides our best model selection process, aiming to identify the optimal model parameters θ^* for the main task during training.

When developing a model for auxiliary learning with distinct data sets, several problems can arise. One is that the auxiliary task may have different input characteristics to the main task, making it difficult to use the same model/parameters for both tasks. In addition, the operating conditions of different manufacturing processes or machines may change, leading to variations in data set distributions. These variations can hamper the development of a model capable of learning efficiently from both data sets, particularly if the differences are large. In addition, inconsistency in the length of RTF or outputs between data sets adds complexity to the creation of a model capable of exploiting the available knowledge.

One way to exploit them is through task-specific parts of the model that act as adapters to project the features or representations of different data-sets into a common space. These modules

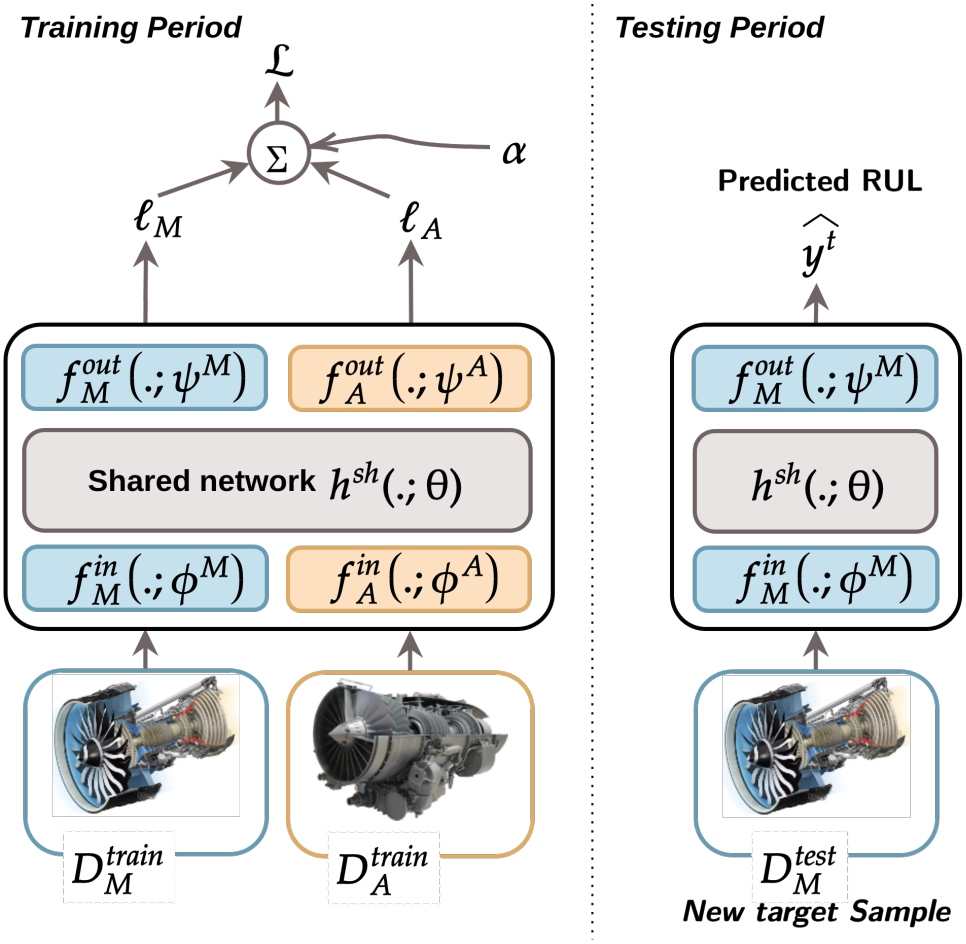


Figure 5.1: Illustration of the proposed auxiliary training approach. (a) Training period: The data involved in the training include run to failure trajectories from the main set D_M^{train} and the trajectories from the auxiliary data-set D_A^{train}. (i) the samples from each task are fed to their respective input adapter f^{in} to obtain similar dimensions and related features that can be used by the shared layers after. (ii) The features of both tasks are fed into the same layers h^{sh} where knowledge from both data-sets is learned. (iii) The output of the shared network for each task are fed into their main output adapter f^{out}. (vi) losses from both tasks are combined using a weighting parameter, α, to get a total loss $\mathcal{L}$ which is used for back-propagation. (b) Testing period: Prediction of the RUL of a new sample from main task is done by the main adapters, and the auxiliary adapters can be dropped to reduce model parameters.

can be used for the input, projecting different features from different tasks into the same space, thus solving the problem of varying input sizes and/or operating conditions between data-sets. In addition, they can also be used as task-specific prediction heads to address the issue of varying lengths of trajectories i.e., useful lifetimes of equipment.

Mathematically, we can represent the model with multiple branches as follows (Eq. 5.2):

$$\hat{y}_k = f_k^{out}(h^{sh}(f_k^{in}(x_k))) \tag{5.2}$$

where $k \in \{M, A\}$ represents the task index. For each task k: $(x_k, \hat{y}_k)$ stand for the input and output values, f_k^{in} and f_k^{out} represent the input and output adapters parameterized by ϕ

and ψ, respectively. Furthermore, h^{sh} denotes the shared backbone parameterized by θ used for both tasks. A schematic illustration of this structure is depicted in Figure 5.1.

To help solve the few shot learning problem, our approach consists in weighting the two losses of auxiliary and main tasks during training in a way that aims to prevent over-fitting on the few samples from main task. To do so, we propose a linear combination of the two losses that significantly reduces the weight of the main one; we add a main loss weight parameter $\alpha << 1$ to the joint loss $\mathcal{L}$ as shown in Equation 5.3 :

$$\mathcal{L} = \mathcal{L} = \ell(y^A, \hat{y}^A) + \alpha * \ell(y^M, \hat{y}^M) \tag{5.3}$$

This approach might initially seem counter intuitive to our primary objective; nevertheless, empirical evidence from our experiments substantiates its validity. Specifically, by empirically demonstrating the advantages through our results, we show that reducing the weight of the main loss, denoted by a smaller alpha, the model becomes less prone to over-fitting to the samples of the main task. This allows the auxiliary task to serve effectively as a regularization mechanism during training. Our empirical findings contrast this method with the conventional two-step approach of pre-training and fine-tuning, which tends to make the model more susceptible to over-fitting on the main task samples, leading to potential loss of auxiliary information. Furthermore, our joint learning approach allows the shared network to learn and generalize mainly from the auxiliary task while the representations learned are biased/tailored towards the main task without over-fitting. Consequently, the main task adapters benefit by utilizing this shared knowledge, which guides the optimization process and improves overall performance.

Algorithm 1 Algorithm of the proposed auxiliary training approach

Input: main training data-set D_{M}, auxiliary training data-set D_{A}
$Model \leftarrow$ initialize model
for $i = 1$ to $EPOCH$ **do**
    for $j = 1$ to $ITERATION$ **do**
        $Batch_{\mathrm{M}} \leftarrow \mathrm{extract}(D_{\mathrm{M}}, BatchSize)$
        $Batch_{\mathrm{A}} \leftarrow \mathrm{extract}(D_{\mathrm{A}}, BatchSize)$
        $\ell_M, \ell_A \leftarrow \quad \mathrm{Compute}_{\mathrm{Loss}}(Model, [Batch_{\mathrm{M}}, Batch_{\mathrm{A}}])$
        $\mathcal{L} \leftarrow \ell_A + \alpha \times \ell_M$
        $Model \leftarrow$ update model parameters
    end for
end for
$select\_best\_model_{\mathrm{M}}(Model)$
Output: trained $Model$

The proposed auxiliary training Algorithm 1 begins by initializing the model parameters, and the main and auxiliary data-sets are iterated over for a specified number of epochs. Within each epoch, the data is partitioned into batches and fed into the model. Next, the algorithm computes the main loss and auxiliary loss for each batch. The two losses are then combined into a total loss using the hyper-parameter α, and the model's parameters are updated via back-propagation.

Once all the epochs have been completed, the $select\_best\_model_{\mathrm{t}}$ function selects the model with the lowest error on the main task validation set, which represent 20% of the data available for model development. To accomplish this, the function evaluates the model's performance on the validation set after each epoch and saves the model's parameters if its performance is better than the previous best model.

5.4 Experimental setup

5.4.1 Model architecture

In the present study, we utilize a modified version of the MLP-LSTM-MLP architecture described in chapter 3 as the foundation of our approach. The modification involves the addition of supplementary branches into the base architecture (Figure 5.1) as discussed in section 5.3.

5.4.2 Baselines

We consider the following baselines to assess the proposed approach (See Figure 5.2).

- **Single task training** (**Single**) simply train the network on the **few samples** from the main data-set independently (Figure 5.2 (a)).
- **PreTraining + FineTuning (PT-FT).** The second baseline is derived from the popular pre-training fine-tuning paradigm. It involves pre-training the model on the auxiliary data-set, and then copying all the model parameters to the fine-tuning stage on the main task, except the first MLP layers (adapter input) which will be initialized from scratch. This is done because the input features between the auxiliary and main data-sets could be different. During fine-tuning, the pre-trained model is re-trained on the small number of samples from the target set (Figure 5.2 (b)). This is one of the standard approaches in the literature used for transfer learning for RUL prediction (Ansi Zhang et al., 2018; S. Yao et al., 2023).
- **Pre-training + retraining input and output layers (PT-R-in-out).** It involves pre-training the model on the auxiliary data-set, freezing the hidden layers parameters, then re-initializing the parameters of the first and last layers (the adapters), and finally training them on the main set (Figure 5.2 (c)).

All approaches except single task learning use the same quantity of samples (see Table 5.1), we study how different ways of leveraging this knowledge affects the performance on the main task.

Table 5.1: Comparison of approaches in terms of their use of information from primary and auxiliary data sources. The quantity of primary data samples is variable and depends on the specific experiment conducted.

Approach	Number of primary samples	Number of auxiliary samples
Single Task learning	$\{3, 5, 10, 20\}$	0
PT-FT	$\{3, 5, 10, 20\}$	100
PT-R-in_out	$\{3, 5, 10, 20\}$	100
Our approach (AT)	$\{3, 5, 10, 20\}$	100

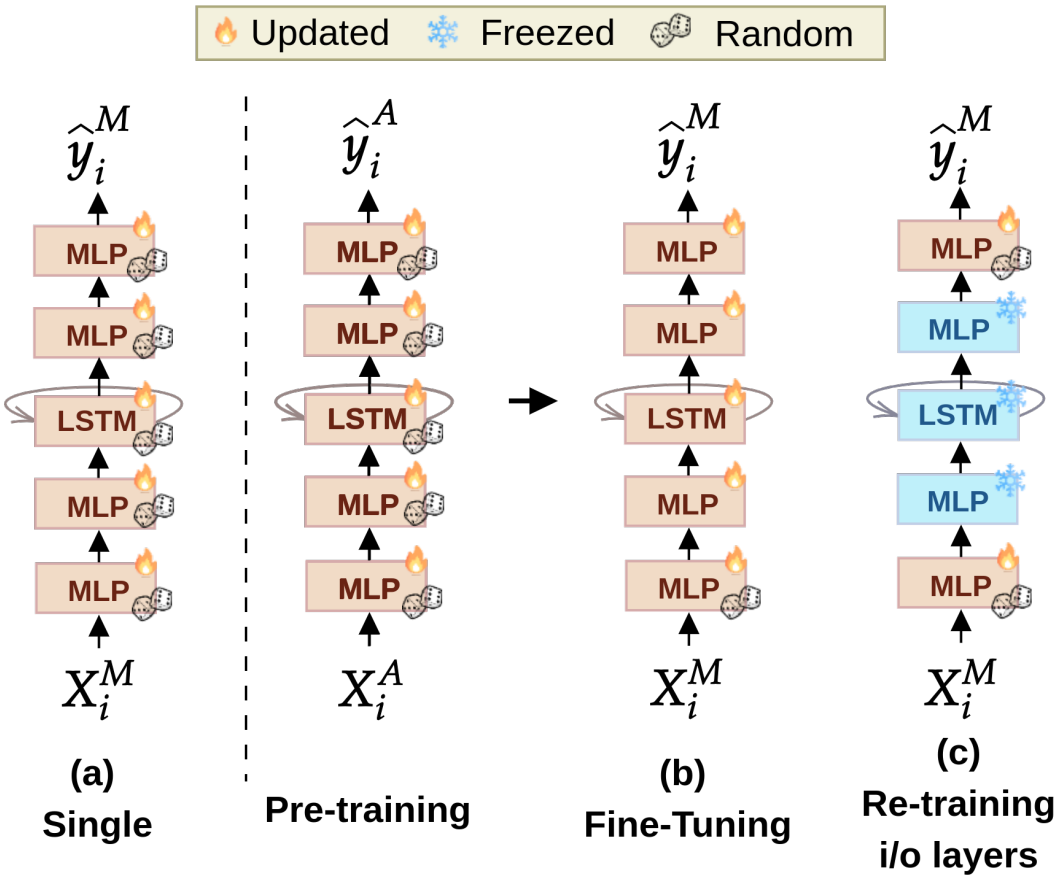


Figure 5.2: baseline approaches used for comparison. Single task training (a), Pre-training followed by Fine-tuning (b), and Pre-training followed by Retraining Input and Output Layers (c). Random icon indicate layers initialized from scratch. In (b) and (c), input adapters are also initialized from scratch to accommodate different input structure.

5.4.3 Settings

To verify the performance of the proposed approach, we conduct a series of experiments on the data-sets presented in Table 5.2. The auxiliary data-set used is a CMAPSS data-set (FD001) and the main task data can be either sub-sampling of FD004 or N-CMAPSS. To be in an FSL configuration, we randomly select a limited number (3,5,10,20) of Run-to-Failure trajectories from the selected main data-set. We chose these configurations for the auxiliary and main sets because FD001 data-set is the simplest, thus evaluating the approach on the most challenging configurations possible using these data-sets (Table 5.2).

Due to the random selection of Run-to-Failure trajectories from the main set and also due to random initialization, the performance may vary across different training runs and sub-sampled D_M's. In order to assess the variability in our results, we first conduct random sub-sampling five separate times. Each instance of sub-sampling (selection) is treated as a unique experiment. Within each experiment, we train using the four different approaches (AT, Single, PT-FT and PT-R-in-out), repeating the process five times to provide an array of results that reflects the performance variability from both the auxiliary training and baseline approaches. To quantify our findings, we report two statistical measures: the average (mean), which indicates the central

Table 5.2: experimental configurations

Configuration	Auxiliary set	Main set	Operating conditions	Fault modes
C_{FD004}	FD001	FD004	1→6	1→2
$C_{N-CMAPSS}$		N-CMAPSS	1→ -	1→7

Table 5.3: Hyper-parameters used in grid search for each approach

Hyper-parameter	Approaches	
	Single/PT-FT/PT-R-in-out	**AT**
learning rate	{ 5×10^{-4}, 1×10^{-4}, 1×10^{-5} }	{ 5×10^{-4}, 1×10^{-4}, 1×10^{-5} }
number of epochs	{ 30, 100, 300 }	{ 100, 300, 500 }
dropout	{ 0.0, 0.4, 0.6 }	{ 0.0 }
α	-	{ 1×10^{-3}, 1×10^{-5}, 1×10^{-7} }

tendency of each approach, and the standard deviation, which shows result variability.

5.4.4 Training details

We chose the best hyper-parameters based on the lowest RMSE obtained on the validation set, which was 20% of our development subset. The rest, 80%, was used for training. We ensured the train-validation split was the same for all tests to keep our results consistent. We used a grid search to fine-tune hyper-parameters, which essentially performs a brute force testing of different combinations. These hyper-parameters include the learning rates, dropouts, epochs, and main loss weight, and evaluated their impact on the model's performance. Refer to Table 5.3 for specifics on each approach.

We maintained a consistent model architecture throughout our testing, consisting of input adapters projecting features to a 10-dimensional space, a 3-layer MLP with 50 neurons per layer, a single-layer LSTM with 50 cells, two additional MLP layers with 50 and 10 neurons, and output adapters consisting of a one-layer MLP with one neuron. This consistent architecture allowed us to isolate the impact of hyper-parameters on performance, helping identify the best settings for each selection. In our study, comparing hyper-parameters might not be useful because their effectiveness depends on the training run and data subset. The choice of hyper-parameters is closely linked to each experiment's specific conditions, making general comparisons less valuable.

5.5 Results and discussion

This section presents the results of the four different approaches: Single task training (Single), Pre-training + Fine-tuning (PT-FT), Pre-training + retraining input and output layers (PT-R-in-out), and Auxiliary Training (AT). The performance evaluation was carried out as follows: we selected various amounts of samples, ranging from 3 to 20, from the development sets of the main

Table 5.4: Few-Shot Remaining Useful Life (RUL) Prediction on FD004 (top) and on N-CMAPSS (bottom): RMSE on Test Data shows that Auxiliary Training (AT) predominantly outperforms Single task training (Single), Pre-Training + Fine-Tuning (PT-FT), and Pre-Training + Retraining Input/Output Layers (PT-R-in-out). The standard deviation across multiple selections and runs is represented by ±.

Approach	Number of samples			
	3	5	10	20
Single	46.11 ± 5.78	38.36 ± 7.37	44.41 ± 5.89	36.01 ± 8.19
PT-FT	49.40 ± 7.39	41.23 ± 7.68	46.48 ± 6.53	39.27 ± 7.49
PT-R-in-out	48.07 ± 7.64	38.37 ± 7.90	44.15 ± 7.25	39.95 ± 8.38
AT	$\mathbf{36.58 \pm 5.93}$	$\mathbf{31.94 \pm 4.93}$	$\mathbf{27.92 \pm 3.73}$	$\mathbf{24.82 \pm 2.93}$

Approach	Number of samples			
	3	5	10	20
Single	$\mathbf{17.74 \pm 2.71}$	17.12 ± 4.99	13.94 ± 2.44	14.20 ± 1.85
PT-FT	19.67 ± 5.42	16.29 ± 4.80	14.72 ± 3.93	15.46 ± 2.29
PT-R-in-out	18.54 ± 6.58	$\mathbf{15.79 \pm 4.80}$	13.85 ± 2.96	14.47 ± 1.75
AT	18.64 ± 5.17	16.14 ± 6.10	$\mathbf{12.82 \pm 1.91}$	$\mathbf{12.80 \pm 1.54}$

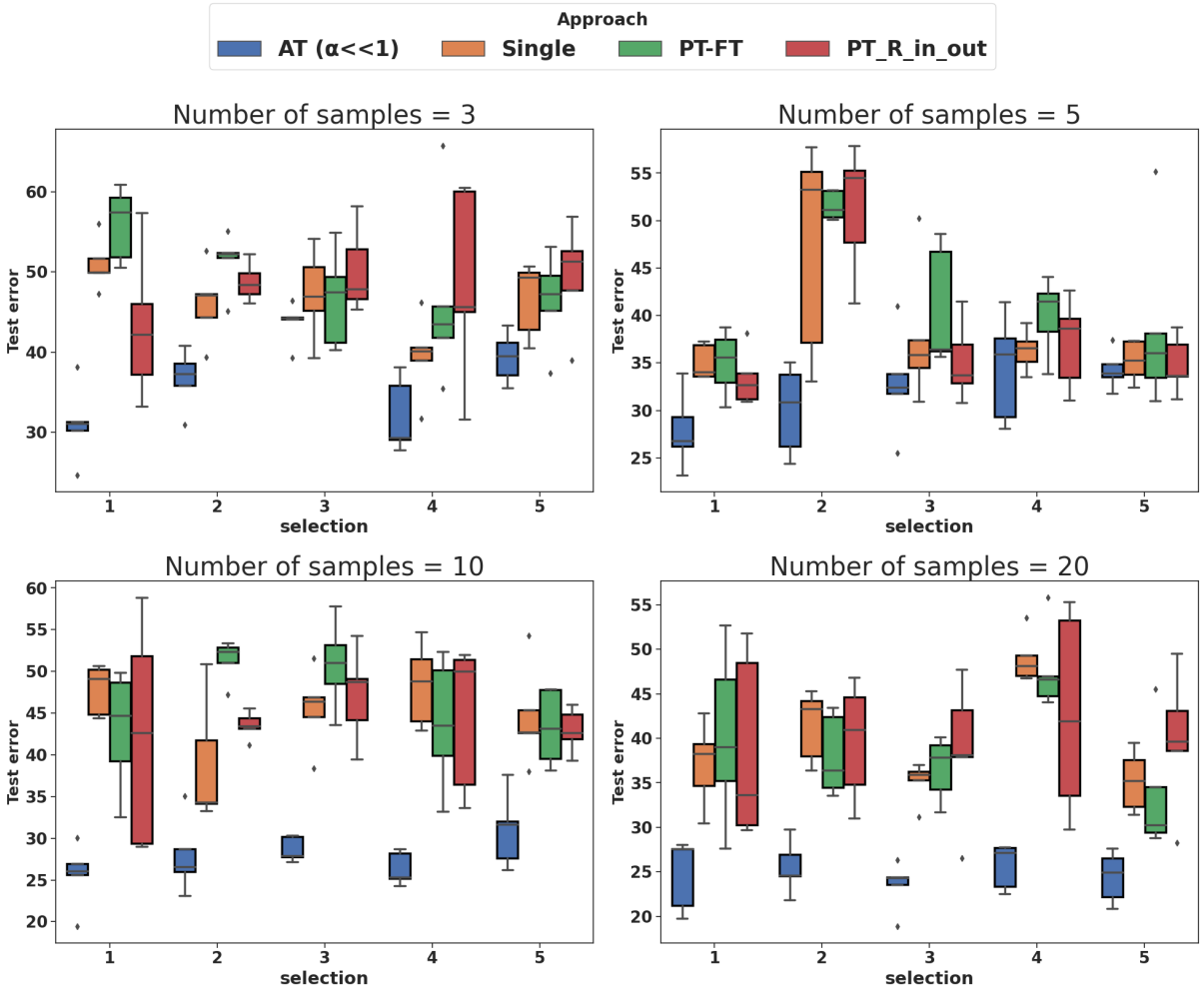


Figure 5.3: Boxplots illustrating the RMSE distributions across the various approaches applied to the FD004 data-set. Each plot corresponds to a different subset size—3, 5, 10, and 20 samples—across five selections. The approaches are color-coded allowing for an evaluative assessment of each method's predictive performances.

data. These selected samples were used to train the models using the approaches discussed. After the training phase, the approaches were then tested using the test data from the corresponding main data-set, and the results were collected.

Table 5.4 (top) and figure 5.3 shows the results on the FD004 test set, demonstrating that the Auxiliary Training approach consistently outperforms the other methods across all shot settings. AT achieved the lowest error values, with mean RMSE of 36.58 ± 5.93 for 3-shot, 31.94 ± 4.93 for 5-shot, 27.92 ± 3.73 for 10-shot, and 24.82 ± 2.93 for 20-shot. In contrast, the Single task training approach consistently achieved the highest error values, with mean RMSE of 46.11 ± 5.78 for 3-shot, 38.36 ± 7.37 for 5-shot, 44.41 ± 5.89 for 10-shot, and 36.01 ± 8.19 for 20-shot. The PT-FT and PT-R-in-out approaches showed intermediate results, with mean RMSE values ranging from 39.27 ± 7.49 to 46.48 ± 6.53, but did not achieve significant improvements compared to Single. These findings suggest that fine-tuning and retraining the input/output layers alone may not be enough to enhance the model's performance in few-shot learning scenarios. Auxiliary Training, however, significantly enhances model generalization.

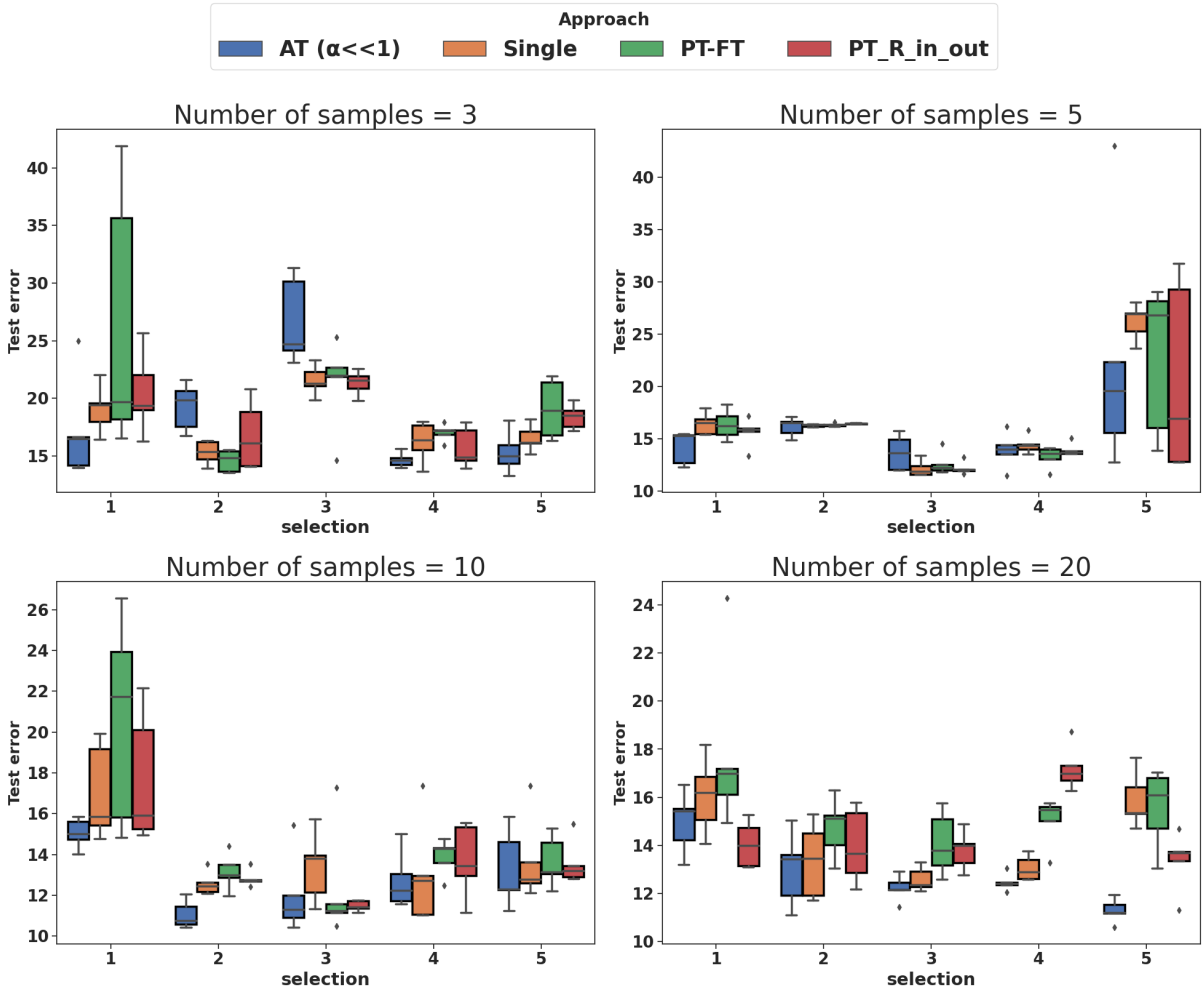


Figure 5.4: Boxplots illustrating the RMSE distributions across the various approaches applied to the N-CMAPSS data-set. Each plot corresponds to a different subset size—3, 5, 10, and 20 samples—across five selections. The approaches are color-coded allowing for an evaluative assessment of each method's predictive performances.

Direct comparisons with existing studies are challenging due to unique constraints and settings of our research. Nonetheless, similar works using the same datasets provide reference points. For instance, (Ansi Zhang et al., 2018) investigate transfer learning methodologies on the same data (CMAPSS). One relevant result from this study employed the FD002 subset for pre-training, which contains six operational conditions (OC), and select samples from FD004 for fine-tuning. Despite the fact that FD002 has a higher correlation with the main data, given the greater number of OCs compared to FD001, the results yield an RMSE of 29.21 and 29.14 respectively when utilizing 10 and 20 samples. Another relevant study by (Ragab, Z. Chen, M. Wu, C. K. Kwoh, et al., 2020) proposed an adversarial transfer learning method to deal with the problem of unlabeled main data. Their work, despite being tangentially related to ours, demonstrated an RMSE of 31.78, achieved by using the entirety of the unlabeled FD004 dataset and FD001. These performances, though commendable, fall short when compared to our approach.

The results on the N-CMAPSS test set, as presented in Table 5.4 (bottom), show that AT achieves the lowest RMSE values of 12.82 ± 1.91 and 12.80 ± 1.54 in the 10-shot and 20-

shot settings, respectively. AT also delivers competitive performance in the 3-shot and 5-shot scenarios, with RMSE values of 18.64 ± 5.17 and 16.14 ± 6.10, respectively. Further analysis on the sub-optimal outcomes in the 3 and 5 shot scenarios, detailed in Table 5.5 and Figure 5.4, shows the mean RMSE over five runs for each selection. For the 3-shot scenario, AT was superior in three selections, while PT-FT had the lowest mean error in two. In the 5-shot scenario, AT outperformed baseline methods in two selections, with other approaches leading in the rest. These variances might be due to a few reasons. The N-CMAPSS (main) and FD001 (auxiliary) datasets are not closely related, suggesting a need for more or similar run-to-failure trajectories to better align the tasks. Alternatively, the simplicity of the auxiliary task's adapters (one layer) could lead to less relevant representations for the main task under these specific conditions.

Table 5.5: 3 and 5-shot RUL prediction over multiple selections of the few samples from N-CMAPSS Data-set.

RMSE values					
Selection	1	2	3	4	5
Approach	3 samples				
Single	19.09	15.30	21.55	16.23	16.53
PT-FT	26.39	**14.58**	**21.30**	17.00	19.07
PT-R-in-out	20.47	16.77	21.33	15.70	18.40
AT	**17.2**	19.27	26.70	**14.65**	**15.32**
	5 samples				
Single	16.46	16.28	**12.19**	14.45	26.21
PT-FT	16.38	16.37	12.64	**13.28**	22.81
PT-R-in-out	15.63	16.46	12.20	13.95	**20.72**
AT	**14.23**	**16.16**	13.68	13.93	22.68

Let's now delve into the impact of reducing the alpha parameter on predictive performance. The results presented in Figure 5.5 underscores the efficacy of this adjustment across the two configurations, FD004 (top) and N-CMAPSS (bottom). Lowering alpha, indicated by blue boxes, generally results in lower or similar RMSE compared to when alpha is set to 1 (orange boxes), with 80% of cases showing improved performance with the reduced alpha. This highlights the benefit of a more regularized model. The consistent improvement across different sample sizes and selections indicates that a smaller alpha helps prevent over-fitting, thereby improving the model's generalization capabilities and its performance.

While acknowledging the contributions of other works in transfer learning using the same datasets, our study stands out by operating within a specific set of constraints and achieving superior outcomes. Overall, these findings suggest that the proposed approach improves the model's generalization ability by utilizing related auxiliary data.

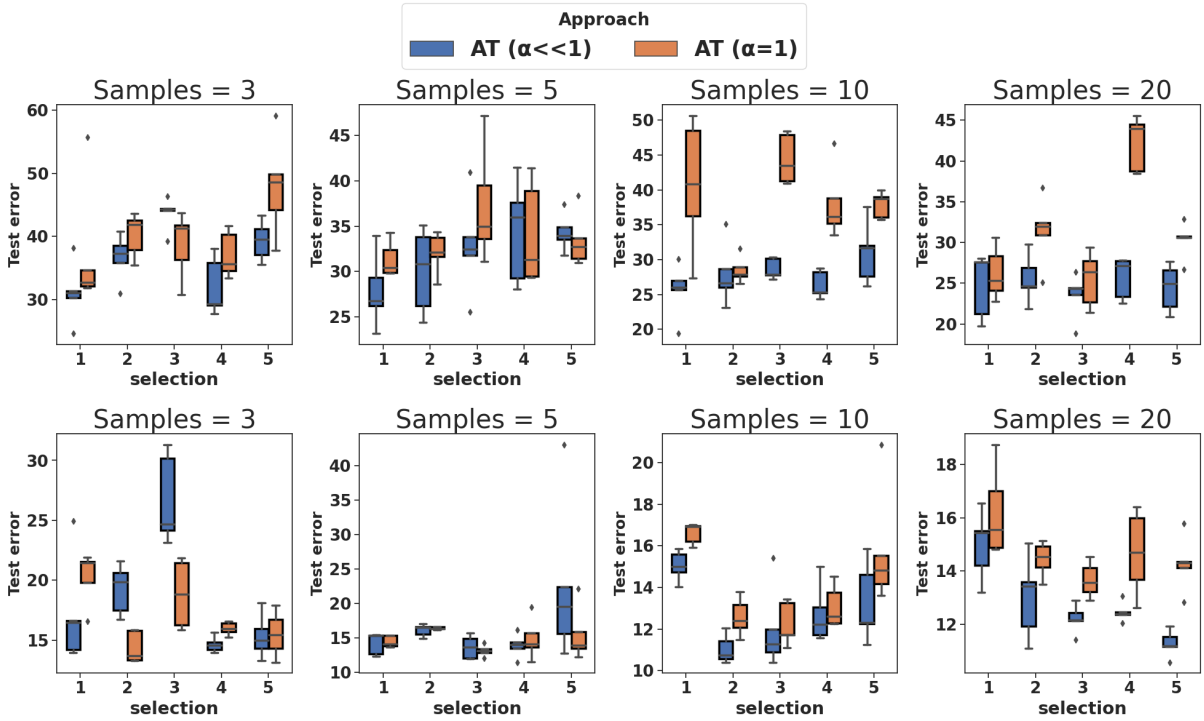


Figure 5.5: Boxplots illustrating the RMSE distributions on FD004 (top) and N-CMAPSS (bottom). The results contrast using a main task loss weight of 1 with our proposed approach, where the weight is significantly reduced to a value much less than 1.

5.6 Conclusion

This chapter presents a method that leverages auxiliary training to enhance the Remaining Useful Life prediction. By utilizing related data-sets, while reducing the weight of the main task loss, the proposed approach leverages knowledge from other data sets. The application of task-specific layers is a notable contribution of this chapter, which projects inputs/outputs from various tasks to relevant spaces, demonstrating its utility in dealing with dissimilar data-sets for auxiliary training. This is also beneficial for other approaches such as pre-training using multiple data sources. In comparison to existing methodologies, Auxiliary Training demonstrated lower susceptibility to over-fitting under the few-shot learning paradigm. Empirical evidence in this chapter shows that it outperforms other baseline approaches from existing literature, such as Pre-Training followed by Fine-Tuning.

However, the assumption of accessible labeled data may not always hold true in industrial settings. Often, large volumes of unlabeled data are generated, representing machines in unknown operational states due to preventative maintenance practices. Labeling such data demands considerable expertise and time, often impractical in many scenarios. This leads an important question: **How can external, unlabeled data be leveraged without necessitating manual labeling?** To address this question, the next chapter proposes an approach that automatically derive auxiliary objectives from heterogeneous unlabeled data.

Chapter 6

Deriving auxiliary objectives from unsupervised heterogeneous data sets

Contribution

In the evolving landscape of Prognostics and Health Management, a significant amount of heterogeneous, unsupervised data remains under-exploited, mainly due to preventive maintenance operation arriving before the development of incipient degradation as well as the high costs and labor involved in manual labeling. To address this challenge, our work proposes an approach based on auxiliary learning that exploits these untapped heterogeneous data reserves through an automated pseudo-labeling process. This process involves the automatic assignment of pseudo labels to the unsupervised data sets, transforming them into a valuable resource for improving the performance of the primary task.
Our extensive validation across three different datasets and twelve experimental settings confirms the effectiveness of our method. The results showcase that our approach outperforms traditional strategies while also delivering results comparable to auxiliary learning with labeled data.

6.1 Introduction

The Auxiliary training approach can involve auxiliary tasks derived from the main task data, extracting additional layers of complexity and relationships in the data to facilitate the learning process, or exploit entirely separate data sets (Chapter 5), as illustrated in Table (6.1), which can introduce a richer feature diversity and potentially improved generalization compared to relying on the same primary input data. While using external data is potentially more promising, the accessibility of such labeled data remains a significant challenge, especially in domains where data labeling is inherently difficult or costly. This chapter is dedicated to tackling this crucial challenge by addressing the question: **"How can external, unlabeled data be leveraged for auxiliary training without necessitating manual labeling?"**

Semi-supervised learning (SSL) partially addresses this by integrating unsupervised data to enhance learning (Duarte and Berton, 2023). Nonetheless, most SSL methods assume that both supervised and unsupervised sets originate from the same distributions, an assumption that often does not hold in all scenarios (Banitalebi-Dehkordi et al., 2022). A step forward has been made by (Banitalebi-Dehkordi et al., 2022), which formalized the concept of Auxiliary semi-supervised learning. This problem is predicated on leveraging unsupervised data from

Table 6.1: This table categorizes state-of-the-art (SOTA) methods based on the source of auxiliary information and the nature of task labeling. Two main sources of auxiliary information are proposed in the literature: the main input data, which may be augmented or contextualized, and heterogeneous auxiliary data sets. Additionally, auxiliary tasks are differentiated by their labeling approach. Our work focuses on the pseudo-labeling of unsupervised heterogeneous datasets.

Auxiliary tasks derived from \ Auxiliary objectives based on	Pseudo labels	True labels/Unsupervised
Main data	(S. Liu et al., 2019) (L. Xu et al., 2021)	(T. Lin et al., 2021) (Dery, Michel, Khodak, et al., 2022) (Liebel and Körner, 2018) (Linfeng Zhang et al., 2020) (Hwang et al., 2023)
Heterogeneous Auxiliary data sets	Our work	(Watanabe et al., 2022) (Dery, Michel, Khodak, et al., 2022) (S. Liu et al., 2019)

distributions that differ from the main task to improve the performance of the primary objective, a broader assumption than traditional SSL. Their definition, however, is tailored to classification tasks and hinges on the notion that the labeled and unlabeled data sets do not share the same class distribution. Our research builds on this, dealing with varied unlabeled auxiliary data sources that differ in distribution, feature spaces, and output values, extending to regression tasks with heterogeneous data.

This problem manifests itself in a variety of applications, such as RUL prediction, which necessitates leveraging different datasets rich in informative features yet underutilized due to lack of labels. In industrial settings, despite the abundance of related data from similar or different machinery across plants, its potential remains untapped primarily because it's unlabeled, lacks complete life-cycle trajectories, or isn't directly applicable to model development, often due to maintenance practices and the extensive costs of labeling. Furthermore, even when labeled, these labels may not be fully compatible with the main objective, as they generally represent a one-time effort that is not always relevant to the main task. For challenges like RUL prediction or others within similar contexts, there is a need for an approach capable of incorporating multiple data sources, flexible to automatically define the auxiliary objectives without the need for manual expensive labeling.

In this study, we leverage primary data to inform the usage of additional unsupervised data through a bidirectional interaction. Our method employs a label generation network and an auxiliary learning network, both trained end-to-end by a meta-learning algorithm. The idea is that the label generation model labels auxiliary tasks, which are then used to inform the auxiliary learning model's training. The label generation network is trained based on the auxiliary model's performance on the primary task's validation set, aiming to optimize labeling of auxiliary data sources in a way to enhance generalization on the main task. While a similar approach was explored by (S. Liu et al., 2019) for image classification, where the goal is to find optimal

auxiliary objectives from the same primary data set, our research as shown in Table 6.1 extends this by incorporating multiple data sources and provide empirical evidence to demonstrate its effectiveness.

This chapter is structured as follows: section 6.2 provides a summary of related work, Section 6.3 introduces the proposed method of utilizing an auxiliary data-set for learning, Section 6.4 showcases the experimental setup and performance evaluation results for the proposed approach, and finally, Section 6.5 concludes the chapter.

6.2 Related Work

Meta Learning Also known as "Learning to Learn" is a learning paradigm that typically involves a bi-level optimization process where the inner-learner provides feedback for optimization of the meta-learner. Meta learning approaches have been applied in a variety of applications, multiple works focused on learning good initialisation for few-shot learning (Finn et al., 2017) and on learning optimizer/finding optimal hyper-parameters (Z. Li et al., 2017). For label generation, (Pham et al., 2021) used meta learning to optimize a pseudo label generator for better semi-supervised knowledge distillation, while (Ng and Q. Wang, 2022) proposed similar approach but without the teacher model. For Auxiliary training, majority of works used meta learning for adaptive weighting of tasks or even samples under this paradigm (Dery, Michel, Khodak, et al., 2022; Hwang et al., 2023; Dery, Michel, Talwalkar, et al., 2021). As for auxiliary label generation, there exists an approach that leverages multi-task framework to automatically generate auxiliary labels to the same input data (S. Liu et al., 2019), which bears relevance to our research. However, this method confines itself to the knowledge inherent in the main input data, which may be insufficient for certain applications. Our work seeks to extend this boundary by proposing to learn to generate labels for disparate unsupervised data-sets from multiple data sources. Additionally, we propose a more general algorithm that avoids the meta-learner and inner-learner mismatch inherent in (S. Liu et al., 2019) iterative optimization, ensuring a more effective learning process.

Semi supervised learning has seen great strides when labeled data is scarce but unlabeled data is abundant, multiple works proposed algorithms to supplement small labeled data with a larger unlabeled examples during training (Pham et al., 2021; Sohn et al., 2020), and showed it can greatly improve generalization when properly used (Duarte and Berton, 2023). This relates to our work in the idea of using unsupervised auxiliary data to boost performance. However, in SSL, the majority of work assumes that the unlabeled data is drawn from the same distribution as the labeled data; conversely, our work does not follow this assumption. Recent work from (Banitalebi-Dehkordi et al., 2022) formalized a similar problem to ours as auxiliary semi-supervised learning, where the unlabeled data can come from unconstrained distributions.

6.3 Labeling auxiliary data sources with meta learning

Previous approaches have successfully utilized auxiliary learning to boost neural network performance by creating auxiliary objectives by leveraging domain-specific knowledge, further exploiting the main data, or by leveraging labeled data. However, these methods often overlook the vast amounts of unsupervised heterogeneous datasets, particularly in areas where obtaining labeled data is challenging and costly, like RUL prediction (Fink et al., 2020).

Intuitively, we assume that there exists a latent generic degradation process z that is common to both the primary and auxiliary datasets: the primary observations $x^M \sim p(x|z, C^M)$ as well as the auxiliary observations $x^A \sim p(x|z, C^A)$ are sampled from this underlying process conditioned

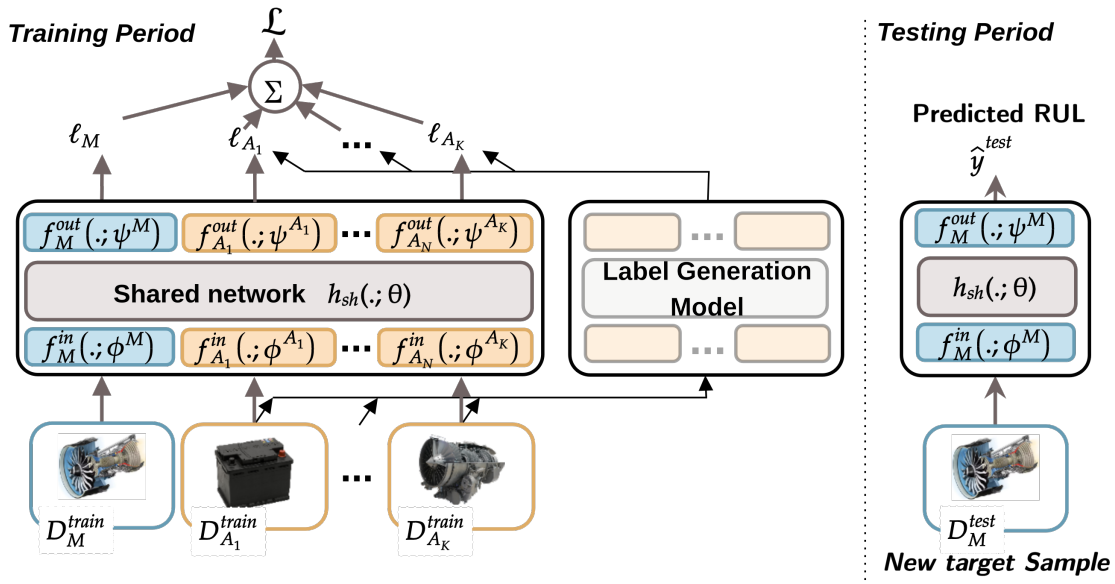


Figure 6.1: Illustration of the proposed approach. (a) Training period: The base model processes the main and auxiliary input data to generate corresponding output values for each task. These outputs are then used to calculate the loss, by comparing them with the true labels for the main task and the pseudo-labels generated for the auxiliary tasks. Simultaneously, a label generation model is trained to iteratively adjust the pseudo-labels for the unsupervised auxiliary tasks in order to improve the learning process for the main task. (b) Test period: The main objective is to predict the remaining useful life of the new samples for the main task. During this phase, the framework uses only the main adapters and shared parameters of the base model. Auxiliary adapters and the label generation model are eliminated, reducing the total number of parameters and simplifying the model structure.

on their respective task specific contexts C^M and C^A. We do not aim at explicitly modeling this latent process z, but we rather rely on the capacity of deep neural networks to selectively "remove" the specific contexts and only capture z. Even though the auxiliary datasets are not labeled with their respective tasks, their observations contain novel information with regard to the degradation trajectories z that is not present in the primary dataset, because of its limited size. Hence, we propose to automatically label the auxiliary datasets based on the primary task corpus in order to guide the deep model towards extracting the most relevant information for the target task.

6.3.1 problem Setting

Let $D_M = \{(x_i^M, y_i^M)\}_{i=1}^{N_M}$ be the main data-set consisting of N_M labeled instances, where $x_i^M \in \mathcal{X}^M$ represents the input features and $y_i^M \in \mathcal{Y}^M$ corresponds to the associated true labels. Additionally, we consider K auxiliary data-sets, expressed as $D_{A_k} = \left\{x_{j,k}^A\right\}_{j=1}^{N_{A_k}}$ for $k = 1, 2, ..., K$, each comprising N_{A_K} instances, only represented by input features $x_{j,k}^A \in \mathcal{X}^{A_k}$ due to the absence of labels. $\mathcal{X}^{A_k}$ represents the feature space for the k-th auxiliary dataset. Here, each $\mathcal{X}^{A_k}$ could be different from $\mathcal{X}^M$ and also from each other $\mathcal{X}^{(A_{k'})}$ for any $k' \neq k$, highlighting the heterogeneous nature of auxiliary data.

Our methodology focuses on generating labels for these unlabeled auxiliary datasets to

improve the primary task's performance. This involves the training of two key networks: a base model, which is trained on both primary and auxiliary tasks as part of a conventional auxiliary learning framework, and a Label Generation Model, which goal is to generate labels for the auxiliary tasks. We denote the label generation model G, parameterized by θ_G, which generates the labels for the multiple auxiliary tasks D_{A_k}. Given an auxiliary input $x^A_{j,k}$, it outputs the pseudo labels $\hat{y}^G_{j,k} = G(x^A_{j,k})$, thereby creating pseudo-labeled datasets $D_{GA_k} = \{(x^A_{j,k}, \hat{y}^G_{j,k})\}_{j=1}^{N_{A_k}}, k = 1, 2, ..., K$.

For the base Model B, parameterized by θ_B to solve the issue of varying input/output heterogeneity between tasks, we apply hard parameter sharing approach (Ruder, 2017) with branches at the input and outputs layers as discussed in the previous chapter and as shown in figure (6.1).

6.3.2 Model objectives

Our goal is to train the label generation model G, to generate pseudo-labels for auxiliary tasks by adopting a "learn to label" strategy (Pham et al., 2021; Ng and Q. Wang, 2022), aiming to enhance the performance on the primary task. This "learning to label" paradigm follows the bi-level optimization framework recognized in meta learning literature (Hospedales et al., 2021), where we have an inner-learner, the base model in our case, that provides feedback to optimize the meta-learner which corresponds to the label generation network in our case.

Specifically, starting from the current θ_B and θ_G, we first start by pre-updating θ_B by optimizing the auxiliary training objective using training samples from all tasks, resulting in pre-updated parameters θ'_B:

$$\theta'_B = \theta_B - \eta \nabla_{\theta_B} \left[\ell(y^{M,train}, \hat{y}^{M,train}(\theta_B)) + \sum_{k=1}^{K} \ell(\hat{y}^G_k(\theta_G), \hat{y}^A_k(\theta_B)) \right] \tag{6.1}$$

where η is the step size, ℓ denotes the training loss (MSE in our case).

Then the label generation network undergoes an "end task aware" update (Dery, Michel, Talwalkar, et al., 2021). This implies that this network is trained in such a manner that, when the base model uses these labels for training, performance on the main task would be maximized on the validation data. Thus, the meta learner parameters are updated as

$$\theta_G \leftarrow \theta_G - \mu \nabla_{\theta_G} \ell(y^{M,val}, \hat{y}^{M,val}(\theta'_B)) \tag{6.2}$$

Here, μ is the step size, θ'_B represents the base model' parameters after one epoch update, by using a retained computational graph, we can compute the derivatives with respect to the meta learner's parameters θ_G. The choice of using validation data in the meta objective is to mitigate over-fitting as shown in multiple works applying similar look a head approach (Dery, Michel, Talwalkar, et al., 2021; Dery, Michel, Khodak, et al., 2022).

After that, we update the base model original parameters θ_B with the updated label generation network θ_G:

$$\theta_B \leftarrow \theta_B - \eta \nabla_{\theta_B} \left[\ell(y^{M,train}, \hat{y}^{M,train}(\theta_B)) + \sum_{k=1}^{K} \ell(\hat{y}^G_k(\theta_G), \hat{y}^A_k(\theta_B)) \right] \tag{6.3}$$

This is the same equation as before (6.1), but we now use the updated parameters of the label generation network to train the base model.

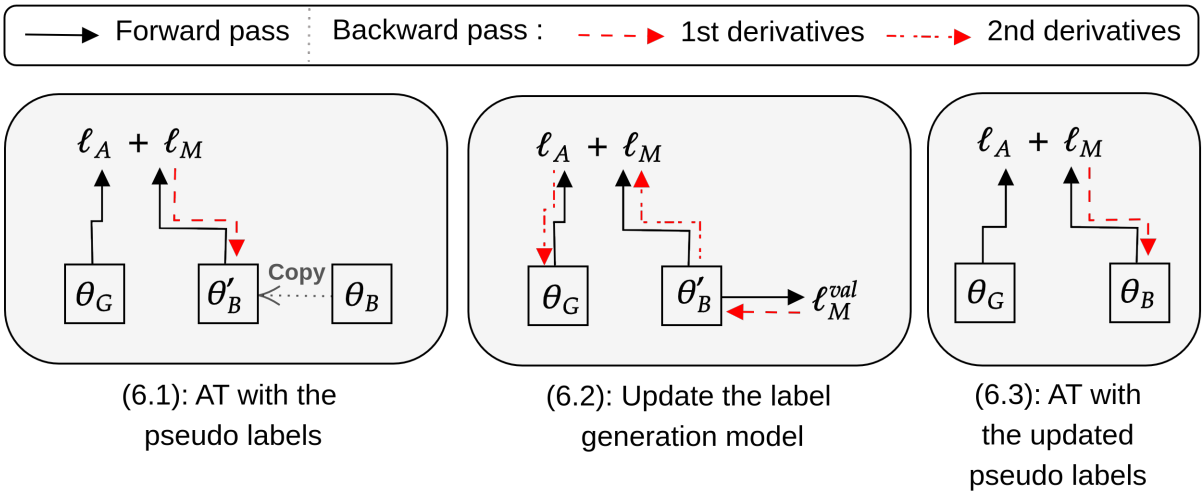


Figure 6.2: Steps of the proposed training methodology, detailing the iterative optimization of the base and label generation models across main and auxiliary datasets.

Algorithm 2 MetaAuxiLabeler Algorithm

Require: Base model θ_B, Label generation model, θ_G

Require: η, μ: learning rates for the two networks.

1: **while** not done **do**

2: **for** each training iteration i **do**

3: Update θ'_B : $\theta'_B = \theta_B - \eta \nabla_{\theta_B} \left[\ell(y^{M,train}, \hat{y}^{M,train}(\theta_B)) + \sum_{k=1}^{K} \ell(\hat{y}_k^G(\theta_G), \hat{y}_k^A(\theta_B)) \right]$

4: **end for**

5: Update θ_G : $\theta_G \leftarrow \theta_G - \mu \nabla_{\theta_G} \ell(y^{M,val}, \hat{y}^{M,val}(\theta'_B))$

6: **for** each training iteration i **do**

7: Update the original θ_B with the updated labels:

θ_B : $\theta_B \leftarrow \theta_B - \eta \nabla_{\theta_B} \left[\ell(y^{M,train}, \hat{y}^{M,train}(\theta_B)) + \sum_{k=1}^{K} \ell(\hat{y}_k^G(\theta_G), \hat{y}_k^A(\theta_B)) \right]$

8: **end for**

9: **end while**

The Algorithm 2 (Figure 6.2) represents a cooperative effort between the label generation model and the base model. The first learns to generate better labels based on feedback from the latter's performance, while the latter benefits from the refined labels to improve its own performance on the main task.

Our approach above is similar to that of (S. Liu et al., 2019) with two key differences: First, we leverage entirely separate data sources instead of reusing input from the main task in a multi-task setup. This allows us to leverage additional information, beyond what is available in the main dataset. Second, the meta learner in their approach is optimized for the original inner-learner and then applied to update the inner-learner in the next iteration, which may be sub-optimal as there is a mismatch between the inner-learner and the meta-learner. In our algorithm, we update the meta model after one step of training of the base model, which removes this mismatch during learning.

6.4 Experiments

6.4.1 Experimental setup

Models architecture

The base model and the label generation network follow a unified architecture consisting of TCN (Bai et al., 2018) layers and linear layers with Tanh activation functions, with adapter modules as needed. This architecture differs from the one discussed in previous chapters. The shift to TCN was driven by the practical challenges associated with meta-training LSTM layers on Graphics Processing Units (GPU), second derivative of LSTM layers is not supported on GPUs, but are supported for TCN which makes the training much faster.

Settings

In our experiments, we aimed to test our approach in different conditions, simulating scenarios with scarce labeled data for the main task but abundant unlabeled data for auxiliary tasks. We chose a limited number of Run-to-Failure trajectories (3, 5, 10, or 20) from the target set to reflect these conditions.

We utilized four distinct datasets: three Turbofan engine degradation datasets, FD001 and FD004, from C-MAPSS dataset, and N-CMAPSS dataset and one Battery degradation dataset (See section 2.5). We included the Battery dataset despite it focuses on a different type of equipment because it shares key similarities with the others, particularly in having multiple RTF trajectories and showing degradation through amplitude changes. This similarity allowed us to use the same model architecture/adapters avoiding adjustments for different degradation patterns, such as frequency changes seen in bearings data for instance. We designed three test configurations to cover a wide range of scenarios, as detailed in Table 6.2. The primary task was based on either the Turbofan (FD004 or N-CMAPSS) or the Battery data-set. Intentionally, we excluded other C-MAPSS data-sets like FD001, FD002, and FD003 from being the main task, given their relatively simpler nature compared to FD004. The aim was to challenge our approach by focusing on the most complex configurations possible. The auxiliary data used are based on the inputs of FD001 + Batteries or FD001 + N-CMAPSS datasets, depending on the configuration. The RUL values, referred to as labels, have not been utilized, as the aim is to generate them using the label generation model.

Table 6.2: Experimental configurations

Configuration	**Auxiliary set**	**Main set**	**Number of samples**
C_{FD004}	FD001 + Batteries	FD004	3,5,10,20
$C_{N-CMAPSS}$	FD001 + Batteries	N-CMAPSS	3,5,10,20
C_{Batt}	FD001 + N-CMAPSS	Batteries	3,5,10,20

Baselines

To assess our approach, we compared it against several baseline methods:

Table 6.3: Comparison of baseline approaches in terms of their use of information from primary and auxiliary data sources. The quantity of primary data samples is variable and depends on the specific experiment conducted. Similarly, the number of samples from heterogeneous auxiliary data is also subject to variation, depending on the configuration.

Approach	Number of primary task samples (labeled)	Number of auxiliary tasks samples (labeled)	Number of auxiliary tasks samples (unlabeled)	Label generator network
True labels	$\{3, 5, 10, 20\}$	$\{234, 166\}$	0	No
Single Task learning	$\{3, 5, 10, 20\}$	0	0	No
Our approach	$\{3, 5, 10, 20\}$	0	$\{234, 166\}$	Yes
MAXL	$\{3, 5, 10, 20\}$	0	0	Yes
Random labels	$\{3, 5, 10, 20\}$	0	$\{234, 166\}$	No

- **Single task learning :** This baseline involves training the model solely on the main task. The comparison helps determine if integrating auxiliary datasets improves performance or not.
- **True labels :** Considering our method relies on unlabeled auxiliary datasets, It's essential to compare it to auxiliary training with actual labeled datasets, i.e. with RUL values. This comparison aims to see if our meta labeling strategy can match or surpass the performances achieved when actual labels are employed.
- **Meta AuXiliary Learning (MAXL) :** Described in (S. Liu et al., 2019), employs a generator to create auxiliary tasks based on the same main dataset in a multi-task framework. We adapted this concept, tailoring it to the context of RUL prediction by introducing a single auxiliary task through our algorithm 2 and our architecture. By contrasting our approach with MAXL, we compare using cross-dataset auxiliary training over intra-dataset multitasking.
- **Random :** Evaluating our method against a scenario where auxiliary tasks are labeled randomly helps gauge the effectiveness of our meta labeling strategy.

Training details

For our experiments, we maintained a consistent base model configuration, adding adapter branches as necessary. We exclusively used Adam as the optimizer for the label generation network model and kept the batch size constant at 16. A grid search was performed based on the validation set performance to determine the optimal hyper-parameters. Specifically: base model learning rates η were tested across $\{0.01, 0.001, 0.0005, 0.0001\}$, label generation network-specific learning rates μ spanned the set $\{0.1, 0.01, 0.001\}$, base model optimizers were evaluated between "Adam" and "SGD". The weighting for the main task loss was explored across

$\{1, 0.001, 0.0001\}$. To ensure robustness in our results, each configuration underwent multiple runs. For a given number of samples in the main task, we executed a random selection of that number of samples five times. Each of these selections was then subjected to five individual experiments. Training was consistently conducted over 1,000 epochs, employing the mean squared error as the loss function.

6.4.2 Experimental Results

We present our findings through comparative analyses between the proposed method and established benchmarks. These analyses are structured according to the configurations detailed in Table 6.2. Performance metrics are provided in Table 6.4, 6.6, and 6.5, as well as in Figure 6.3. [1]

Comparison with Single Task Learning

Our comparison with the single-task learning baseline (Table 6.4) shows that using meta-labeled auxiliary training generally improves model performance. Our method produced superior results in over 80% of tests across three experimental setups. Specifically, for C_{FD004} and C_{Batt}, our approach outperformed single-task learning in 15 out of 20 cases, and in 19 out of 20 cases for $C_{N-CMAPSS}$. The instances where our method under-performed, mostly occurring in 30% of cases with 3 or 5 samples, coincided with the presence of a single trajectory in the validation set. This suggests that, in some cases, a single validation trajectory may not be sufficiently representative, leading to less accurate pseudo labels and reduced performance. When our method showed improvement over Single, the average increase in performance was 18%, with some cases showing enhancements up to 38% (more than 50% in one case). In scenarios where our approach was worst than single-task learning (about 20% of the time) the average decrease in performance was 12%, with decreases of up to 28%. Overall, incorporating meta-labeled auxiliary data tends to yield better results in most cases, indicating that it improves model performance.

Comparison to labeled Auxiliary task learning (True Labels)

Following the initial evaluation, we contrasted our approach with a scenario wherein the auxiliary objectives are based on True labels of the auxiliary data sets (Table 6.6), we aimed to measure performance differences between actual labels and those generated by our meta-labeling technique.

In the cases of C_{FD004} and C_{Batt}, auxiliary learning with true labels surpassed our meta-labeling method in 65% of instances, with an average performance gap of 8%. Conversely, in the scenarios where our approach did outperform true labels, the average improvement was 4.7%. However, for the second configuration, $C_{N-CMAPSS}$, our method exceeded the performance of true labels in 80% of cases, achieving an average enhancement of 14%, while the performance difference in the other 20% of cases was 4.8%.
These findings indicate that true labels provide better outcomes than pseudo labels in two out of three configurations. Nevertheless, in the $C_{N-CMAPSS}$ configuration, our approach demonstrated superior results, suggesting that the efficacy of each method could be influenced by the specific characteristics of the main dataset at hand, such as the similarity between the data among other potential factors.

[1] the results of this chapter are not comparable with those of the previous one because we have modified the architecture of the model.

Table 6.4: Comparative performance evaluation of our approach against **Single task learning** method on the three experimental configurations. The values presented as RMSE mean over 5 runs. For ease of reading, C_{Batt} values are scaled by 1/23.

		C_{FD004}		$C_{N-CMAPSS}$		C_{Batt} ($\div 23$)	
Number of samples	Selection	Single	**Our approach**	Single	**Our approach**	Single	**Our approach**
3	1	34.45	**31.98**	22.97	**19.35**	23.14	**19.32**
	2	38.22	**36.94**	39.41	**18.31**	**17.60**	18.62
	3	**41.94**	44.49	**24.39**	31.37	**20.70**	23.51
	4	**33.39**	37.58	26.24	**18.02**	25.67	**24.06**
	5	44.03	**41.99**	22.59	**19.04**	27.45	**18.46**
5	1	36.89	**29.55**	29.92	**16.12**	26.08	**20.13**
	2	34.04	**33.39**	24.33	**20.88**	**22.44**	25.43
	3	**33.98**	39.85	16.99	**15.73**	21.01	**20.71**
	4	**33.41**	39.02	22.53	**14.95**	17.46	**16.88**
	5	38.18	**34.75**	25.65	**18.24**	**19.60**	22.18
10	1	36.28	**27.18**	19.21	**15.25**	19.93	**16.76**
	2	30.32	**27.53**	18.70	**13.22**	22.54	**16.97**
	3	**32.15**	36.36	17.20	**15.39**	23.87	**16.79**
	4	38.78	**33.62**	20.70	**14.60**	**22.59**	23.04
	5	41.11	**30.81**	18.85	**14.19**	21.07	**19.29**
20	1	31.73	**26.13**	17.63	**16.06**	18.37	**15.84**
	2	34.76	**30.77**	16.21	**13.58**	19.73	**18.81**
	3	30.64	**26.05**	20.72	**15.25**	22.15	**14.85**
	4	32.55	**30.77**	14.52	**13.28**	21.56	**15.12**
	5	31.05	**27.43**	17.75	**12.54**	20.47	**15.86**

Overall, our methodology presents a viable alternative when access to labeled data is limited, aiming to make use of unlabeled data. It shows an improvement over single-task learning and, in certain cases, offers comparable performance with labeled auxiliary data. The variability in performance across different configurations may point to underlying dataset-specific elements that warrant further investigation, rather than reflecting limitations of our approach.

Comparison to intra-data-set Auxiliary label generation (MAXL)

Our study also investigated whether integrating unsupervised data from heterogeneous domains into auxiliary objectives is beneficial. To this end, we compare our approach with adding auxiliary objectives based on the same primary input data in a multi-task framework, an approach proposed by (S. Liu et al., 2019). we adapted the idea of intra-data set based objectives to the context of RUL prediction, introducing a single auxiliary task based on our algorithm 2.

The comparative analysis (table 6.5) shows that our approach outperformed MAXL in over

Table 6.5: Comparative performance evaluation of our approach against **True labels** method on the three experimental configurations. The values presented as RMSE mean over 5 runs. For ease of reading, C_{Batt} values are scaled by 1/23.

		C_{FD004}		$C_{N-CMAPSS}$		C_{Batt} ($\div 23$)	
Number of samples	Selection	True labels	**Our approach**	True labels	**Our approach**	True labels	**Our approach**
3	1	33.71	**31.98**	29.84	**19.35**	20.07	**19.32**
	2	**35.32**	36.94	24.89	**18.31**	20.95	**18.62**
	3	**44.16**	44.49	**30.38**	31.37	23.52	**23.51**
	4	**33.67**	37.58	25.56	**18.02**	**23.55**	24.06
	5	**41.13**	41.99	19.10	**19.04**	20.26	**18.46**
5	1	**26.68**	29.55	**15.55**	16.12	**19.20**	20.13
	2	**31.23**	33.39	**19.04**	20.88	**24.16**	25.43
	3	**33.66**	39.85	19.24	**15.73**	**20.49**	20.71
	4	**37.18**	39.02	18.15	**14.95**	17.62	**16.88**
	5	**34.06**	34.75	21.39	**18.24**	**19.59**	22.18
10	1	**25.01**	27.18	18.98	**15.25**	**16.31**	16.76
	2	27.98	**27.53**	**12.87**	13.22	17.57	**16.97**
	3	**26.91**	36.36	18.18	**15.39**	17.90	**16.79**
	4	**30.38**	33.62	18.46	**14.60**	**21.07**	23.04
	5	**30.05**	30.81	16.23	**14.19**	**18.28**	19.29
20	1	27.15	**26.13**	16.94	**16.06**	**14.74**	15.84
	2	**26.30**	30.77	15.09	**13.58**	**18.80**	18.81
	3	27.12	**26.05**	15.67	**15.25**	15.37	**14.85**
	4	**25.01**	30.77	13.67	**13.28**	**16.21**	15.12
	5	**26.75**	27.43	12.99	**12.54**	**15.26**	15.86

80% of cases, where the average improvement was 15% (median 15%). Conversely, in the remaining cases where it under-performed, the average difference was 12% (median 9%). This superiority suggests that even though both methods generate pseudo-labels from the main task's validation set, the incorporation of heterogeneous data sets can enhance generalization on the primary task. We attribute this improvement to several factors: First, the diversity of heterogeneous data sets introduces a broader spectrum of features and examples, avoiding excessive adaptation to the particularities of the training data for the main task. Second, the label generation network is able to generate labels that capture concepts and knowledge from heterogeneous data sets, which can then be transferred to the main task, possibly discovering hidden patterns in the main data set that were not initially captured. In addition, heterogeneous auxiliary tasks can serve as a stronger form of regularization, encouraging the model to develop more generalizable representations compared to what's achievable through multi-task learning within a singular dataset.

Table 6.6: Comparative performance evaluation of our approach against **MAXL** method on the three experimental configurations. The values presented as RMSE mean over 5 runs. For ease of reading, C_{Batt} values are scaled by 1/23.

		C_{FD004}		$C_{N-CMAPSS}$		C_{Batt} ($\div 23$)	
Number of samples	Selection	MAXL	**Our approach**	MAXL	**Our approach**	MAXL	**Our approach**
3	1	39.57	**31.98**	**17.39**	19.35	**18.86**	19.32
	2	38.30	**36.94**	22.51	**18.31**	**17.85**	18.62
	3	**41.29**	44.49	**29.24**	31.37	24.02	**23.51**
	4	40.46	**37.58**	**16.12**	18.02	25.19	**24.06**
	5	55.20	**41.99**	19.43	**19.04**	24.20	**18.46**
5	1	34.29	**29.55**	18.84	**16.12**	24.89	**20.13**
	2	38.33	**33.39**	23.3	**20.88**	**22.88**	25.43
	3	**37.35**	39.85	18.36	**15.73**	22.05	**20.71**
	4	**30.65**	39.02	18.12	**14.95**	19.40	**16.88**
	5	36.48	**34.75**	24.11	**18.24**	**16.58**	22.18
10	1	33.83	**27.18**	22.98	**15.25**	20.29	**16.76**
	2	30.19	**27.53**	17.84	**13.22**	23.16	**16.97**
	3	40.74	**36.36**	18.29	**15.39**	22.65	**16.79**
	4	40.44	**33.62**	18.78	**14.60**	24.85	**23.04**
	5	41.26	**30.81**	18.94	**14.19**	21.13	**19.29**
20	1	30.72	**26.13**	16.36	**16.06**	18.23	**15.84**
	2	33.83	**30.77**	16.69	**13.58**	20.90	**18.81**
	3	30.46	**26.05**	19.76	**15.25**	21.97	**14.85**
	4	32.24	**30.77**	15.12	**13.28**	20.62	**15.12**
	5	33.72	**27.43**	16.87	**12.54**	19.07	**15.86**

Essentially, although both approaches rely on the performance of the validation set for the meta updates, leveraging heterogeneous data sets with pseudo-labels can improve generalization by introducing a richer set of features, more patterns, and learning scenarios, making the model more performing and robust to the conditions in the test set.

Comparison with Random Labeling

To assess our meta-learning loop's effectiveness, we compared it against a random labeling approach, focusing on the validation set for a more direct analysis of pseudo label impact. Our pseudo labels are optimized to minimize loss on the validation set, unlike random labeling, which doesn't account for primary task performance and is applied without iteration.

The comparison, illustrated through box-plots in Figure (6.3), clearly demonstrated our method's superior validation performance. Our approach resulted in tighter inter-quartile ranges, lower medians, and a reduced spread in RMSE values, indicating that our meta-learning loop

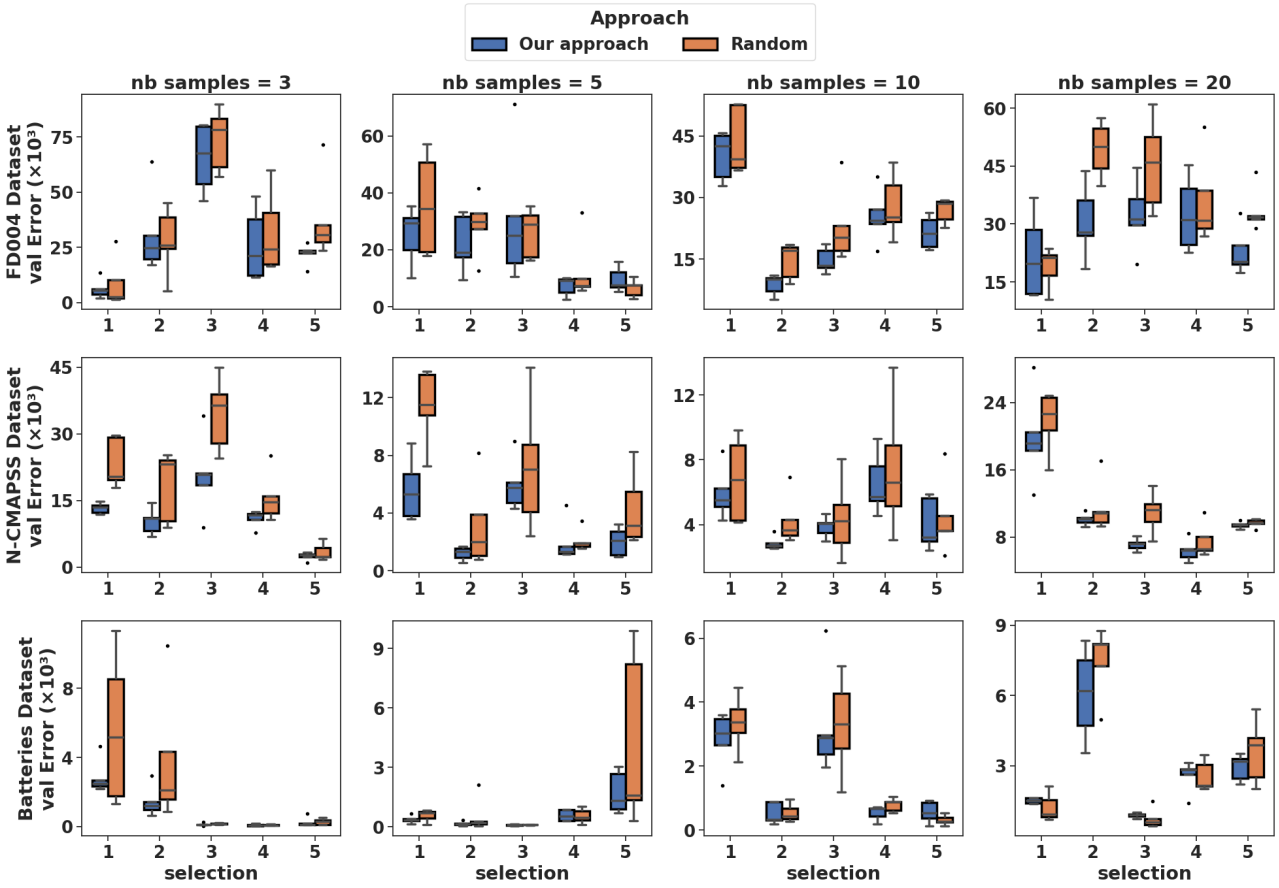


Figure 6.3: Boxplot comparison of RMSE on the validation set across different main data-sets, sample sizes (denoted by nb samples), and selections, contrasting our method method with Random labeling.

successfully y guides the model towards more optimal parameters, thereby enhancing its predictive accuracy. Moreover, random labeling displayed greater volatility, evident from a higher occurrence of outliers, reflecting its unstructured nature. Conversely, our approach demonstrated fewer outliers, indicating a more stable and reliable learning process that effectively utilizes the data's inherent structure.

Evaluating the Pseudo-Labeling Approach on Whole vs. Truncated Trajectories

In order to best represent real-life conditions in PHM, we study the application of our approach to truncated trajectories. In practical industrial scenarios, letting equipment run until failure is often avoided due to the associated high costs and risks. To reflect this reality, we implemented a truncation process on the auxiliary data, which randomly shortens the input data lengths to between 40%-80% of their original trajectory length.

The results, detailed in Table 6.7, shows the impact of utilizing either whole or truncated trajectories on performance. The results indicate a preference for full trajectories, although by a narrow margin. Each strategy outperforms the other in nearly the same number of cases (48%, 52%) , and the observed average performance difference between the two is around 6-7% (median of 5%). Despite the reduction in data length and, by extension, in the information available for

Table 6.7: Comparative performance evaluation of Pseudo-Labeling Whole vs. Truncated Trajectories. The values presented as RMSE mean over 5 runs. For ease of reading, C_{Batt} values are scaled by 1/23.

		C_{FD004}		$C_{N-CMAPSS}$		C_{Batt} ($\div 23$)	
Number of samples	Selection	**Truncated Traj**	**Whole Traj**	**Truncated Traj**	**Whole Traj**	**Truncated Traj**	**Whole Traj**
3	1	32.09	**31.98**	**17.35**	19.35	20.63	**19.32**
	2	36.81	**36.94**	18.45	**18.31**	22.08	**18.62**
	3	**41.03**	44.49	**21.64**	31.37	23.54	**23.51**
	4	40.00	**37.58**	18.10	**18.02**	**23.98**	24.06
	5	46.16	**41.99**	20.95	**19.04**	20.35	**18.46**
5	1	**28.89**	29.55	16.33	**16.12**	**18.98**	20.13
	2	34.35	**33.39**	**18.73**	20.88	**23.83**	25.43
	3	**36.04**	39.85	16.99	**15.73**	21.57	**20.71**
	4	39.09	**39.02**	18.72	**14.95**	**16.18**	16.88
	5	35.54	**34.75**	24.33	**18.24**	**20.19**	22.18
10	1	**26.05**	27.18	15.33	**15.25**	**16.38**	16.76
	2	31.02	**27.53**	13.60	**13.22**	18.43	**16.97**
	3	39.31	**36.36**	16.99	**15.39**	**15.94**	16.79
	4	**30.22**	33.62	**13.79**	14.60	**21.30**	23.04
	5	35.95	**30.81**	14.26	**14.19**	**18.22**	19.29
20	1	**25.22**	26.13	**15.79**	16.06	**14.83**	15.84
	2	31.22	**30.77**	14.76	**13.58**	19.74	**18.81**
	3	**24.90**	26.05	**15.19**	15.25	15.73	**14.85**
	4	**28.51**	30.77	**12.74**	13.28	**15.07**	15.12
	5	**25.84**	27.43	12.97	**12.54**	**14.31**	15.86

learning, the RMSE values indicate only a minor degradation in model performance, and in some instances, it even outperformed the performance obtained with complete RTF trajectories. This demonstrates the robustness of our approach, highlighting its suitability for real-world PHM applications where complete RTF data may not be readily available.

6.4.3 Discussion and Limitations

Unsupervised heterogeneous data are often utilized during pre-training phases to learn generic representations. However, as noted by (Dery, Michel, Talwalkar, et al., 2021), the drawback of this pre-training approach is that the learned representations may not be optimally tailored to the primary task. Our approach addresses this by deriving auxiliary objectives through meta pseudo labeling unsupervised external data, making the learning process 'end-task aware' and tailored to the primary task's needs. This is particularly useful when labeled data for the primary task is limited, as direct fine-tuning on such data often leads to over-fitting.

Our empirical findings confirm that leveraging separate, unsupervised heterogeneous data sets presents a promising solution when faced with a scarcity of labeled data in the primary task. However, there are limitations to this method particularly concerning the use of the validation set. The methodology relies heavily on the validation set for two critical functions: updating the meta-learner and selecting the optimal model during training. This dependency can compromise generalizability if pseudo-labels optimized for validation performance do not represent a wide enough range of scenarios.

While some research recommends employing dual validation sets to address this limitation, our study, constrained by the number of available trajectories, does not incorporate this approach. Consequently, the potential knowledge gained from unsupervised auxiliary data may be underutilized or skewed, depending on the representativeness of the validation set.

Despite these challenges, it is noteworthy that our method consistently outperformed other techniques in our tested configurations, suggesting that these limitations, though present, do not significantly detract from the overall efficacy of our approach in the explored scenarios.

6.5 Conclusion

This chapter introduced a meta-learning-based method for automatically generating auxiliary objectives from unsupervised data, demonstrating its efficacy in enhancing the generalization of the main task through end-task aware pseudo-labeling. Applied within the context of remaining useful life prediction, our approach addresses scenarios characterized by limited labeled data for the main task and the availability of other diverse unsupervised datasets.

Comparative analyses against various baselines revealed that our method offers a viable alternative in situations where acquiring labeled datasets is challenging. It outperforms single task learning and other benchmarks, proving to be competitive against auxiliary learning with labeled auxiliary data. Our experiments show that our approach is a promising step towards automating auxiliary learning with heterogeneous unsupervised data.

Conclusion and Perspectives

Chapter 7

Summary of Findings and Contributions, and Potential Areas for Future Research

7.1 Conclusion

The main objective of this thesis was to advance the Deep Learning (DL) applications in prognostic domain, focusing on Remaining useful life (RUL) prediction for industrial equipment. A review of the existing literature in Chapter 2 identified several areas for improvement in this research field, leading to the formulation of focused research questions intended to guide this work:

- Which deep learning architectures are best suited for end-to-end training, can handle data variability, and still deliver accurate RUL predictions?
- How can deep learning models be designed to provide interpretability that aligns with industrial concepts present in the data?
- How to better leverage external data to develop RUL prediction models?
- How can external, unlabeled data be leveraged for auxiliary training without necessitating manual labeling?

In this thesis, we have addressed these key questions through a two-part investigation, in the first part of the thesis, where data availability was not a limiting factor, we worked on the two first questions: Chapter (3) presented an MLP-LSTM-MLP architecture. This architecture was designed to address the variability introduced by the system functioning under multiple operating conditions, a common issue that often degrades performance of models in literature. This architecture, through end-to-end training without manual feature selection/engineering, demonstrated improved performance on datasets characterized by multiple operating conditions.

Subsequently, Chapter (4) delved into enhancing the explainability of the previously developed model. By replacing the initial MLP layers with a gated mixture of experts, the gating network will allocate an expert per operating condition, thereby enhancing the model's interpretability. This approach, integrated during the model's conception and training, can be used in combination with other post-hoc interpretability techniques.

The second part of the thesis addressed the reality of limited run-to-failure trajectories specific to individual equipment. Assuming access to additional datasets, which may be related to the primary equipment and could be labeled or unlabeled, Chapter (5) introduced an auxiliary training approach that leverages labeled external data. The proposed use of task-specific projection layers addressed input/output variability across datasets, which helped to leverage the knowledge present in them. Through extensive testing, we demonstrated that our approach outperforms conventional methods that leverage external data like Pre-training followed by Fine-Tuning.

Nevertheless, the issue of exploiting unlabeled data, common in industrial settings due to the infrequency of failures and the prohibitive costs of labeling, remained unaddressed. To this end, in Chapter (6), we proposed a meta-learning approach that derive auxiliary objectives from such unlabeled data, where a label generation network meta-trained to produce pseudo labels that enhance the performance on the main validation set. Comparative analyses across a spectrum of configurations confirmed the viability of our method as a practical alternative in scenarios where labeled data is scarce and labeling is neither economically nor logistically feasible.

7.2 Perspectives

7.2.1 Criticism and short-term perspectives

Reflecting on this thesis's contributions highlights the necessity to address certain limitations and chart immediate paths for future research.

The proposed MLP-LSTM-MLP architecture, effective in managing variability caused by multiple operating conditions, struggles with long sequence lengths, a known limitation of LSTM models due to the exploding and vanishing gradient problem (Arpit et al., 2018). Moreover, this architecture sometimes falls behind other models from the literature in scenarios characterized by a single operating condition. An immediate research direction could explore advanced sequence models capable of processing long-range dependencies, while integrating an MLP in the suitable location.

In terms of interpretability, the current implementation of the gated mixture of experts represents an initial step towards creating a modular network wherein each module represents a discrete concept from the dataset. While progress has been made in mapping experts to operating conditions, perfect separation remains elusive, indicating the need for continued research in this area. Additionally, future work could extend the application of gated mixtures of experts to other parts of the model, potentially developing experts tailored to specific fault modes.

Regarding data scarcity, the auxiliary training approach introduced in the second part of the thesis relies on a single validation set for both base and meta model learning, which might amplify biases from the data split, potentially leading to over-fitting. A cross-validation method, which involves periodically changing the training and validation sets throughout the learning process, could address this by ensuring a more balanced learning process (Yoa et al., 2021).

Finally, to address data scarcity more generally, our approach could be combined with data augmentation. Given the unique challenges of data augmentation in the context of prognostics, future work could evaluate combining random, non-expert-driven data augmentations with the meta-training of a label generation model, to label/select augmented inputs, potentially improving the generalization capabilities of the model and offering an automatic approach for enriching the training data.

These short-term prospects aim to consolidate the progress made in this thesis, providing a basis for follow-up efforts to enhance the application of DL models for RUL prediction.

7.2.2 Long-term Perspectives

Advancing industrial prognostics through Deep Learning (DL) demands scientific research across various domains, key areas for future research should include:

1. **Datasets:** Acknowledging the essential role of data in DL, there's a significant need for an expanded collection of datasets. Future research should focus on expanding the range of datasets to include a broader spectrum of equipment types and longer RTF trajectories. alongside data that accurately reflects actual maintenance practices, such as repairs and replacements. Additionally, the development of datasets that contain new types of faults/-operating conditions in test sets is critical for enabling open set learning which can reflect real cases. Furthermore, integrating operational data with textual information, like maintenance logs and failure reports, is also needed to leverage language models. Creating such data sets will help researchers develop new approaches to tackle challenges faced in real world scenarios, and also develop new architectures/approaches that could have a similar impact on the field as CNNs for CV and Large Language Models (LLM)s for NLP.

2. **Application of Federated Learning:** In response to the increasing digitization of industries and the need for privacy-preserving methodologies, it's important to research federated learning. This approach allows for the development of DL models from decentralized data sources without compromising proprietary information, which could lead to learning other types of faults not present in certain factories but are present in others. By demonstrating federated learning's potential to address these challenges, research can pave the way for its broader acceptance and implementation, fostering a more collaborative and secure environment for implementing DL in industrial settings, such as developing large pre-trained models, or other approaches beneficial for various applications.

3. **Industrial Large Knowledge Models (LKM):** proposed by (Jay Lee and Hanqi Su, 2023), envisions a synergy between a comprehensive knowledge library containing both machine and human data (e.g., manuals, documents), domain knowledge LLMs, and other AI techniques. Scientific research about this framework would allow to provide precise solutions to specific problems, to facilitate human-in-the-loop interactions, which is a design principle of industry 4.0 (Vogel-Heuser and Hess, 2016), and to improve the human's ability to achieve a holistic understanding of problems and minimise errors arising from narrow focus.

These research areas align with the objectives of advancing DL applications for Prognostics and Health Management, showing a promising potential next generation of industrial systems.

7.3 Epilogue

This thesis was funded by the European project "AI-proficient", and it seeks to summarize the research experiences accumulated over the past three years, particularly in the domain of deep learning for failure prognostics, an interesting topic that still leaves a lot of challenges to be solved. The principal goal of this thesis was to advance the field of DL application to PHM. Initially, our research operated under conditions where data scarcity was not a concern, aiming to deepen our understanding of the research landscape and make contributions in terms of model architecture and interpretability. The focus shifted to the issue of data scarcity in the latter part of the research. The approach proposed to mitigate data scarcity involves the use of external,

heterogeneous, unlabeled data, which is directly related to the demands observed across various industries and applications. Note that the methodologies developed in this study are applicable across different fields and should be considered in future research.

Bibliography

Achouch, Mounia, Mariya Dimitrova, Khaled Ziane, Sasan Sattarpanah Karganroudi, Rizck Dhouib, Hussein Ibrahim, and Mehdi Adda (2022). "On predictive maintenance in industry 4.0: Overview, models, and challenges". In: *Applied Sciences* 12.16, p. 8081.

Arias Chao, Manuel, Chetan Kulkarni, Kai Goebel, and Olga Fink (2021). "Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics". In: *Data* 6.1, p. 5.

Arpit, Devansh, Bhargav Kanuparthi, Giancarlo Kerg, Nan Rosemary Ke, Ioannis Mitliagkas, and Yoshua Bengio (2018). "h-detach: Modifying the LSTM gradient towards better optimization". In: *arXiv preprint arXiv:1810.03023*.

Atamuradov, Vepa, Kamal Medjaher, Pierre Dersin, Benjamin Lamoureux, and Noureddine Zerhouni (2017). "Prognostics and health management for maintenance practitioners-Review, implementation and tools evaluation." In: *International Journal of Prognostics and Health Management* 8.3, pp. 1–31.

Aydemir, Gurkan, Kamran Paynabar, and Burak Acar (2022). "Robust Feature Learning for Remaining Useful Life Estimation Using Siamese Neural Networks". In: *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 1432–1436.

Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek (2015). "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7, e0130140.

Bai, Shaojie, J Zico Kolter, and Vladlen Koltun (2018). "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". In: *arXiv preprint arXiv:1803.01271*.

Banitalebi-Dehkordi, Amin, Pratik Gujjar, and Yong Zhang (2022). "AuxMix: semi-supervised learning with unconstrained unlabeled data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3999–4006.

Bauer, Wilhelm, Sebastian Schlund, Dirk Marrenbach, and Oliver Ganschar (2014). "Studie: Industrie 4.0–Volkswirt schaftliches Potenzial für Deutschland". In.

Bauernhansl, Thomas, Jörg Krüger, Gunther Reinhart, and Günther Schuh (2016). "WGP-Standpunkt Industrie 4.0". In.

Behera, Sourajit and Rajiv Misra (2021). "Generative adversarial networks based remaining useful life estimation for IIoT". In: *Computers & Electrical Engineering* 92, p. 107195.

Behera, Sourajit and Rajiv Misra (2023). "A multi-model data-fusion based deep transfer learning for improved remaining useful life estimation for IIOT based systems". In: *Engineering Applications of Artificial Intelligence* 119, p. 105712.

Behera, Sourajit, Rajiv Misra, and Alberto Sillitti (2023). "GAN-Based Multi-Task Learning Approach for Prognostics and Health Management of IIoT". In: *IEEE Transactions on Automation Science and Engineering*.

Bengio, Emmanuel, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup (2015). "Conditional computation in neural networks for faster models". In: *arXiv preprint arXiv:1511.06297*.

Bottou, Léon and Olivier Bousquet (2007). "The tradeoffs of large scale learning". In: *Advances in neural information processing systems* 20.

Bottou, Léon, Frank E Curtis, and Jorge Nocedal (2018). "Optimization methods for large-scale machine learning". In: *SIAM review* 60.2, pp. 223–311.

CEYLAN, Uğur and GENÇ Yakup (2021). "Siamese Inception Time Network for Remaining Useful Life Estimation". In: *Journal of Artificial Intelligence and Data Science* 1.2, pp. 165–175.

Chen, Baotong, Jiafu Wan, Lei Shu, Peng Li, Mithun Mukherjee, and Boxing Yin (2017). "Smart factory of industry 4.0: Key technologies, application case, and challenges". In: *Ieee Access* 6, pp. 6505–6519.

Chen, Daoquan, Weicong Hong, and Xiuze Zhou (2022). "Transformer network for remaining useful life prediction of lithium-ion batteries". In: *Ieee Access* 10, pp. 19621–19628.

Chen, Jiaxian, Ruyi Huang, Zhuyun Chen, Wentao Mao, and Weihua Li (2023). "Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective". In: *Mechanical Systems and Signal Processing* 193, p. 110239.

Chen, Jinglong, Hongjie Jing, Yuanhong Chang, and Qian Liu (2019). "Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process". In: *Reliability Engineering & System Safety* 185, pp. 372–382.

Chen, Jungan, Danjiang Chen, and Gaoping Liu (2021). "Using temporal convolution network for remaining useful lifetime prediction". In: *Engineering Reports* 3.3, e12305.

Chen, Wenbai, Weizhao Chen, Huixiang Liu, Yiqun Wang, Chunli Bi, and Yu Gu (2022). "A RUL prediction method of small sample equipment based on DCNN-BiLSTM and domain adaptation". In: *Mathematics* 10.7, p. 1022.

Chen, Yuanhang, Gaoliang Peng, Zhiyu Zhu, and Sijue Li (2020). "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction". In: *Applied Soft Computing* 86, p. 105919.

Chen, Zhenghua, Min Wu, Rui Zhao, Feri Guretno, Ruqiang Yan, and Xiaoli Li (2020). "Machine remaining useful life prediction via an attention-based deep learning approach". In: *IEEE Transactions on Industrial Electronics* 68.3, pp. 2521–2531.

Cheng, Cheng, Guijun Ma, Yong Zhang, Mingyang Sun, Fei Teng, Han Ding, and Ye Yuan (2020). "A deep learning-based remaining useful life prediction approach for bearings". In: *IEEE/ASME transactions on mechatronics* 25.3, pp. 1243–1254.

Cheng, Shunfeng, Michael H. Azarian, and Michael G. Pecht (2010). "Sensor Systems for Prognostics and Health Management". In: *Sensors* 10.6, pp. 5774–5797. ISSN: 1424-8220. DOI: 10.3390/s100605774. URL: https://www.mdpi.com/1424-8220/10/6/5774.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078*.

Clarivate (2023). Citation Report graphic is derived from Clarivate Web of Science. All rights reserved.

Costa, Nahuel and Luciano Sánchez (2022). "Variational encoding approach for interpretable assessment of remaining useful life estimation". In: *Reliability Engineering & System Safety* 222, p. 108353.

Couture, Jonathan and Xianke Lin (2022). "Image-and health indicator-based transfer learning hybridization for battery RUL prediction". In: *Engineering Applications of Artificial Intelligence* 114, p. 105120.

Dalmarco, Gustavo, Filipa R Ramalho, Ana C Barros, and Antonio L Soares (2019). "Providing industry 4.0 technologies: The case of a production technology cluster". In: *The journal of high technology management research* 30.2, p. 100355.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Deng, Yafei, Delin Huang, Shichang Du, Guilong Li, Chen Zhao, and Jun Lv (2021). "A double-layer attention based adversarial network for partial transfer learning in machinery fault diagnosis". In: *Computers in Industry* 127, p. 103399.

Dery, Lucio M, Paul Michel, Mikhail Khodak, Graham Neubig, and Ameet Talwalkar (2022). "AANG: Automating Auxiliary Learning". In: *arXiv preprint arXiv:2205.14082*.

Dery, Lucio M, Paul Michel, Ameet Talwalkar, and Graham Neubig (2021). "Should we be pre-training? an argument for end-task aware training as an alternative". In: *arXiv preprint arXiv:2109.07437*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Ding, Peng and Minping Jia (2020). "Intelligent health evaluation of rolling bearings based on subspace meta-learning". In: *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*. Vol. 1. IEEE, pp. 750–754.

Ding, Yifei, Minping Jia, Yudong Cao, Peng Ding, Xiaoli Zhao, and Chi-Guhn Lee (2023). "Domain generalization via adversarial out-domain augmentation for remaining useful life prediction of bearings under unseen conditions". In: *Knowledge-Based Systems* 261, p. 110199.

Dobre, Mihai Sorin and Alex Lascarides (2017). "Combining a mixture of experts with transfer learning in complex games". In: *2017 AAAI Spring Symposium Series*.

Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell (2014). "Decaf: A deep convolutional activation feature for generic visual recognition". In: *International conference on machine learning*. PMLR, pp. 647–655.

Duarte, José Marcio and Lilian Berton (2023). "A review of semi-supervised learning for text classification". In: *Artificial Intelligence Review*, pp. 1–69.

Al-Dulaimi, Ali, Soheil Zabihi, Amir Asif, and Arash Mohammadi (2019). "Hybrid deep neural network model for remaining useful life estimation". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, pp. 3872–3876.

Eigen, David, Marc'Aurelio Ranzato, and Ilya Sutskever (2013). "Learning factored representations in a deep mixture of experts". In: *arXiv preprint arXiv:1312.4314*.

Eker, Omer Faruk, Faith Camci, and Ian K Jennions (2012). "Major challenges in prognostics: Study on benchmarking prognostics datasets". In: *PHM Society European Conference*. Vol. 1. 1.

El Hachem, Charbel, Gilles Perrot, Loic Painvin, and Raphaël Couturier (2021). "Automation of quality control in the automotive industry using deep learning algorithms". In: *2021 International Conference on Computer, Control and Robotics (ICCCR)*. IEEE, pp. 123–127.

Ellefsen, André Listou, Emil Bjørlykhaug, Vilmar Æsøy, Sergey Ushakov, and Houxiang Zhang (2019). "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture". In: *Reliability Engineering & System Safety* 183, pp. 240–251.

Fedus, William, Barret Zoph, and Noam Shazeer (2022). "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity". In: *The Journal of Machine Learning Research* 23.1, pp. 5232–5270.

Fink, Olga, Qin Wang, Markus Svensen, Pierre Dersin, Wan-Jui Lee, and Melanie Ducoffe (2020). "Potential, challenges and future directions for deep learning in prognostics and health management applications". In: *Engineering Applications of Artificial Intelligence* 92, p. 103678.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *CoRR* abs/1703.03400. arXiv: 1703.03400. URL: http://arxiv.org/abs/1703.03400.

Fu, Song, Lin Lin, Yue Wang, Feng Guo, Minghang Zhao, Baihong Zhong, and Shisheng Zhong (2024). "MCA-DTCN: A novel dual-task temporal convolutional network with multi-channel attention for first prediction time detection and remaining useful life prediction". In: *Reliability Engineering & System Safety* 241, p. 109696.

Gay, Antonin, Alexandre Voisin, Benoit Iung, Phuc Do, Rémi Bonidal, and Ahmed Khelassi (2022). "Data Augmentation-based Prognostics for Predictive Maintenance of Industrial System". In: *CIRP Annals* 71.1, pp. 409–412.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

Hagmeyer, Simon, Fabian Mauthe, and Peter Zeiler (2021). "Creation of publicly available data sets for prognostics and diagnostics addressing data scenarios relevant to industrial applications". In: *International Journal of Prognostics and Health Management* 12.2.

Hamon, Ronan, Henrik Junklewitz, Ignacio Sanchez, Gianclaudio Malgieri, and Paul De Hert (2022). "Bridging the gap between AI and explainability in the GDPR: towards trustworthiness-by-design in automated decision-making". In: *IEEE Computational Intelligence Magazine* 17.1, pp. 72–85.

Han, Xu, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. (2021). "Pre-trained models: Past, present and future". In: *AI Open* 2, pp. 225–250.

Heimes, Felix O (2008). "Recurrent neural networks for remaining useful life estimation". In: *2008 international conference on prognostics and health management*. IEEE, pp. 1–6.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hong, Chang Woo, Changmin Lee, Kwangsuk Lee, Min-Seung Ko, Dae Eun Kim, and Kyeon Hur (2020). "Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction". In: *Sensors* 20.22, p. 6626.

Hospedales, Timothy, Antreas Antoniou, Paul Micaelli, and Amos Storkey (2021). "Meta-learning in neural networks: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 44.9, pp. 5149–5169.

Hou, Guisheng, Shuo Xu, Nan Zhou, Lei Yang, and Quanhao Fu (2020). "Remaining Useful Life Estimation Using Deep Convolutional Generative Adversarial Networks Based on an Autoencoder Scheme". In: *Computational Intelligence and Neuroscience* 2020.

Hou, Mengru, Dechang Pi, and Bingrong Li (2020). "Similarity-based deep learning approach for remaining useful life prediction". In: *Measurement* 159, p. 107788.

Huang, Cheng-Geng, Hong-Zhong Huang, and Yan-Feng Li (2019). "A bidirectional LSTM prognostics method under multiple operational conditions". In: *IEEE Transactions on Industrial Electronics* 66.11, pp. 8792–8802.

Huang, Dezhi, Gang Yu, Jian Zhang, Jian Tang, and Jian Su (2022). "An accurate prediction algorithm of RUL for bearings: time-frequency analysis based on MRCNN". In: *Wireless Communications and Mobile Computing* 2022.

Huo, Zepeng, Lida Zhang, Rohan Khera, Shuai Huang, Xiaoning Qian, Zhangyang Wang, and Bobak J Mortazavi (2021). "Sparse gated mixture-of-experts to separate and interpret patient

heterogeneity in ehr data". In: *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, pp. 1–4.

Hwang, Dasol, Sojin Lee, Joonmyung Choi, Je-Keun Rhee, and Hyunwoo J Kim (2023). "Robust auxiliary learning with weighting function for biased data". In: *Information Sciences* 628, pp. 307–319.

Iglesias, Guillermo, Edgar Talavera, Ángel González-Prieto, Alberto Mozo, and Sandra Gómez-Canaval (2022). "Data augmentation techniques in time series domain: A survey and taxonomy". In: *arXiv preprint arXiv:2206.13508*.

Iwana, Brian Kenji and Seiichi Uchida (2021). "An empirical survey of data augmentation for time series classification with neural networks". In: *Plos one* 16.7, e0254841.

Jacobs, Robert A, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton (1991). "Adaptive mixtures of local experts". In: *Neural computation* 3.1, pp. 79–87.

Jang, Jaeyeon and Chang Ouk Kim (2021). "Siamese network-based health representation learning and robust reference-based remaining useful life prediction". In: *IEEE Transactions on Industrial Informatics* 18.8, pp. 5264–5274.

Javed, Kamran, Rafael Gouriveau, and Noureddine Zerhouni (2013). "Novel failure prognostics approach with dynamic thresholds for machine degradation". In: *IECON 2013-39th annual conference of the IEEE industrial electronics society*. IEEE, pp. 4404–4409.

Jin, Ruibing, Zhenghua Chen, Keyu Wu, Min Wu, Xiaoli Li, and Ruqiang Yan (2022). "Bi-LSTM-based two-stream network for machine remaining useful life prediction". In: *IEEE Transactions on Instrumentation and Measurement* 71, pp. 1–10.

Jin, Ruibing, Min Wu, Keyu Wu, Kaizhou Gao, Zhenghua Chen, and Xiaoli Li (2022). "Position encoding based convolutional neural networks for machine remaining useful life prediction". In: *IEEE/CAA Journal of Automatica Sinica* 9.8, pp. 1427–1439.

Kaya, Mahmut and Hasan Şakir Bilge (2019). "Deep metric learning: A survey". In: *Symmetry* 11.9, p. 1066.

Khelif, Racha, Simon Malinowski, Brigitte Chebel-Morello, and Noureddine Zerhouni (2014). "RUL prediction based on a new similarity-instance based approach". In: *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*. IEEE, pp. 2463–2468.

Kim, Seokgoo, Nam Ho Kim, and Joo-Ho Choi (2020). "Prediction of remaining useful life by data augmentation technique based on dynamic time warping". In: *Mechanical Systems and Signal Processing* 136, p. 106486.

Kim, Taehyeon, Jaehoon Oh, NakYil Kim, Sangwook Cho, and Se-Young Yun (2021). "Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation". In: *arXiv preprint arXiv:2105.08919*.

Kirsch, Louis, Julius Kunze, and David Barber (2018). "Modular networks: Learning to decompose neural computation". In: *Advances in neural information processing systems* 31.

Koch, Gregory, Richard Zemel, Ruslan Salakhutdinov, et al. (2015). "Siamese neural networks for one-shot image recognition". In: *ICML deep learning workshop*. Vol. 2. 1. Lille.

Krishnamurthy, Yamuna and Chris Watkins (2021). "Interpretability in gated modular neural networks". In: *eXplainable AI approaches for debugging and diagnosis.*

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.

Kumar, C Ranjeeth, N Saranya, M Priyadharshini, Derrick Gilchrist, et al. (2023). "Face recognition using CNN and siamese network". In: *Measurement: Sensors* 27, p. 100800.

Kvalseth, Tarald O (1987). "Entropy and correlation: Some comments". In: *IEEE Transactions on Systems, Man, and Cybernetics* 17.3, pp. 517–519.

Kwak, Myeongsun and Jongsoo Lee (2023). "Diagnosis-based domain-adaptive design using designable data augmentation and Bayesian transfer learning: Target design estimation and validation". In: *Applied Soft Computing* 143, p. 110459.

LeCun, Yann and Corinna Cortes (2010). *MNIST handwritten digit database*. URL: http://yann.lecun.com/exdb/mnist/.

Lee, Jay and Hanqi Su (2023). "A Unified Industrial Large Knowledge Model Framework in Smart Manufacturing". In: *arXiv preprint arXiv:2312.14428*.

Lee, Jay, Fangji Wu, Wenyu Zhao, Masoud Ghaffari, Linxia Liao, and David Siegel (2014). "Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications". In: *Mechanical systems and signal processing* 42.1-2, pp. 314–334.

Lei, Yaguo, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin (2018). "Machinery health prognostics: A systematic review from data acquisition to RUL prediction". In: *Mechanical systems and signal processing* 104, pp. 799–834.

Lepikhin, Dmitry, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen (2020). "Gshard: Scaling giant models with conditional computation and automatic sharding". In: *arXiv preprint arXiv:2006.16668*.

Li, Han, Wei Zhao, Yuxi Zhang, and Enrico Zio (2020). "Remaining useful life prediction using multi-scale deep convolutional neural network". In: *Applied Soft Computing* 89, p. 106113.

Li, Jialin, Xueyi Li, and David He (2019). "A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction". In: *IEEE Access* 7, pp. 75464–75475.

Li, Jiamin, Qiang Su, Yitao Yang, Yimin Jiang, Cong Wang, and Hong Xu (2023). "Adaptive gating in mixture-of-experts based language models". In: *arXiv preprint arXiv:2310.07188*.

Li, Oscar, Hao Liu, Chaofan Chen, and Cynthia Rudin (2018). "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Li, Xiang, Qian Ding, and Jian-Qiao Sun (2018). "Remaining useful life estimation in prognostics using deep convolution neural networks". In: *Reliability Engineering & System Safety* 172, pp. 1–11.

Li, Xinghui (2010). *2010 PHM Society Conference Data Challenge*. DOI: 10.21227/jdxd-yy51. URL: https://dx.doi.org/10.21227/jdxd-yy51.

Li, Zhenguo, Fengwei Zhou, Fei Chen, and Hang Li (2017). "Meta-sgd: Learning to learn quickly for few-shot learning". In: *arXiv preprint arXiv:1707.09835*.

Liebel, Lukas and Marco Körner (2018). "Auxiliary tasks in multi-task learning". In: *arXiv preprint arXiv:1805.06334*.

Lin, Tianjiao, Huaqing Wang, Liuyang Song, Bo Ma, and Zuoyi Dong (2021). "Multi-task learning based classified-assisted prediction network for remaining useful life prediction". In: *2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD)*. IEEE, pp. 1–6.

Lin, Yan-Hui, Lu-Xin Guan, Liang Chang, and Enrico Zio (2023). "A Semi-supervised Deep Hybrid Multi-Task Model for RUL Prediction". In: *IEEE Transactions on Instrumentation and Measurement*.

Liu, Hui, Zhenyu Liu, Weiqiang Jia, and Xianke Lin (2020). "Remaining useful life prediction using a novel feature-attention-based end-to-end approach". In: *IEEE Transactions on Industrial Informatics* 17.2, pp. 1197–1207.

Liu, Lu, Xiao Song, and Zhetao Zhou (2022). "Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture". In: *Reliability Engineering & System Safety* 221, p. 108330.

Liu, Shikun, Andrew Davison, and Edward Johns (2019). "Self-supervised generalisation with meta auxiliary learning". In: *Advances in Neural Information Processing Systems* 32.

Liu, Xiaodong, Pengcheng He, Weizhu Chen, and Jianfeng Gao (2019). "Multi-task deep neural networks for natural language understanding". In: *arXiv preprint arXiv:1901.11504*.

Liu, Yingchao, Xiaofeng Hu, and Wenjuan Zhang (2019). "Remaining useful life prediction based on health index similarity". In: *Reliability Engineering & System Safety* 185, pp. 502–510.

Liu, Yuanjun and Xingang Wang (2021). "Deep & attention: A self-attention based neural network for remaining useful lifetime predictions". In: *2021 7th International Conference on Mechatronics and Robotics Engineering (ICMRE)*. IEEE, pp. 98–105.

Liu, Yuefeng, Xiaoyan Zhang, Wei Guo, Haodong Bian, Yingjie He, and Zhen Liu (2021). "Prediction of remaining useful life of turbofan engine based on optimized model". In: *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1473–1477. DOI: 10.1109/TrustCom53373.2021.00210.

Liu, Zixuan, Chaobin Tan, Yuxin Liu, Hao Li, Beining Cui, and Xuanzhe Zhang (2023). "A study of a domain-adaptive LSTM-DNN-based method for remaining useful life prediction of planetary gearbox". In: *Processes* 11.7, p. 2002.

Long, Junqi, Chuanhai Chen, Zhifeng Liu, Jinyan Guo, and Weizheng Chen (2022). "Stochastic hybrid system approach to task-orientated remaining useful life prediction under time-varying operating conditions". In: *Reliability Engineering & System Safety* 225, p. 108568.

Lu, Yang (2017). "Industry 4.0: A survey on technologies, applications and open research issues". In: *Journal of industrial information integration* 6, pp. 1–10.

Lundberg, Scott M and Su-In Lee (2017). "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30.

Lv, Haixin, Jinglong Chen, and Tongyang Pan (2020). "Sequence adaptation adversarial network for remaining useful life prediction using small data set". In: *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*. Vol. 1. IEEE, pp. 115–118.

Madsen, Andreas, Siva Reddy, and Sarath Chandar (2022). "Post-hoc interpretability for neural nlp: A survey". In: *ACM Computing Surveys* 55.8, pp. 1–42.

Malgieri, Gianclaudio (2019). "Automated decision-making in the EU Member States: The right to explanation and other "suitable safeguards" in the national legislations". In: *Computer law & security review* 35.5, p. 105327.

Martinelli, Arianna, Andrea Mina, and Massimo Moggi (2021). "The enabling technologies of industry 4.0: examining the seeds of the fourth industrial revolution". In: *Industrial and Corporate Change* 30.1, pp. 161–188.

Miao, Huihui, Bing Li, Chuang Sun, and Jie Liu (2019). "Joint learning of degradation assessment and RUL prediction for aeroengines via dual-task deep LSTM networks". In: *IEEE Transactions on Industrial Informatics* 15.9, pp. 5023–5032.

Mo, Yu, Liang Li, Biqing Huang, and Xiu Li (2023). "Few-shot RUL estimation based on model-agnostic meta-learning". In: *Journal of Intelligent Manufacturing* 34.5, pp. 2359–2372.

Moradi, Ramin and Katrina M Groth (2020). "On the application of transfer learning in prognostics and health management". In: *arXiv preprint arXiv:2007.01965*.

Nectoux, Patrick, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Noureddine Zerhouni, and Christophe Varnier (2012). "PRONOSTIA: An experimental platform for bearings accelerated degradation tests." In: *IEEE International Conference on Prognostics and Health Management, PHM'12*. IEEE Catalog Number: CPF12PHM-CDR, pp. 1–8.

Ng, Kei-Sing and Qingchen Wang (2022). "Self Meta Pseudo Labels: Meta Pseudo Labels Without The Teacher". In: *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 1405–1411.

Ouali, Yassine, Céline Hudelot, and Myriam Tami (2020). "An overview of deep semi-supervised learning". In: *arXiv preprint arXiv:2006.05278*.

Pasa, Gabriel Duarte, Ivo Medeiros, and Takashi Yoneyama (2019). "Operating condition-invariant neural network-based prognostics methods applied on turbofan aircraft engines". In: *Annual conference of the phm society*. Vol. 11, pp. 1–10.

Pavlitskaya, Svetlana, Christian Hubschneider, Michael Weber, Ruby Moritz, Fabian Huger, Peter Schlicht, and Marius Zollner (2020). "Using mixture of expert models to gain insights into semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 342–343.

Pham, Hieu, Zihang Dai, Qizhe Xie, and Quoc V Le (2021). "Meta pseudo labels". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11557–11568.

Pradier, Melanie F, Javier Zazo, Sonali Parbhoo, Roy H Perlis, Maurizio Zazzi, and Finale Doshi-Velez (2021). "Preferential mixture-of-experts: Interpretable models that rely on human expertise as much as possible". In: *AMIA Summits on Translational Science Proceedings* 2021, p. 525.

Qi, Hang, Matthew Brown, and David G Lowe (2018). "Low-shot learning with imprinted weights". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5822–5830.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training". In.

Ragab, Mohamed, Zhenghua Chen, Min Wu, Chee Keong Kwoh, and Xiaoli Li (2020). "Adversarial transfer learning for machine remaining useful life prediction". In: *2020 IEEE international conference on prognostics and health management (ICPHM)*. IEEE, pp. 1–7.

Ragab, Mohamed, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Ruqiang Yan, and Xiaoli Li (2021). "Attention-based sequence to sequence model for machine remaining useful life prediction". In: *Neurocomputing* 466, pp. 58–68.

Ramasso, Emmanuel and Abhinav Saxena (2014a). "Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets." In: *International Journal of Prognostics and Health Management* 5.2, pp. 1–15.

Ramasso, Emmanuel and Abhinav Saxena (2014b). "Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset". In: *Annual Conference of the Prognostics and Health Management Society 2014*.

Ran, Yongyi, Xin Zhou, Pengfeng Lin, Yonggang Wen, and Ruilong Deng (2019). "A survey of predictive maintenance: Systems, purposes and approaches". In: *arXiv preprint arXiv:1912.07383*.

Rezaeianjouybari, Behnoush and Yi Shang (2020). "Deep learning for prognostics and health management: State of the art, challenges, and opportunities". In: *Measurement* 163, p. 107929.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). """ Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

Rojko, Andreja (2017). "Industry 4.0 concept: Background and overview." In: *International journal of interactive mobile technologies* 11.5.

Ruder, Sebastian (2017). "An overview of multi-task learning in deep neural networks". In: *arXiv preprint arXiv:1706.05098*.

Ruiz-Tagle Palazuelos, Andres, Enrique López Droguett, and Rodrigo Pascual (2020). "A novel deep capsule neural network for remaining useful life estimation". In: *Proceedings of the In-*

stitution of Mechanical Engineers, Part O: Journal of Risk and Reliability 234.1, pp. 151–167.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115, pp. 211–252.

Ryabinin, Max and Anton Gusev (2020). "Towards crowdsourced training of large neural networks using decentralized mixture-of-experts". In: *Advances in Neural Information Processing Systems* 33, pp. 3659–3672.

Sanakkayala, Deva Chaitanya, Vijayakumar Varadarajan, Namya Kumar, Karan, Girija Soni, Pooja Kamat, Satish Kumar, Shruti Patil, and Ketan Kotecha (2022). "Explainable AI for bearing fault prognosis using deep learning techniques". In: *Micromachines* 13.9, p. 1471.

Sateesh Babu, Giduthuri, Peilin Zhao, and Xiao-Li Li (2016). "Deep convolutional neural network based regression approach for estimation of remaining useful life". In: *Database Systems for Advanced Applications: 21st International Conference, DASFAA 2016, Dallas, TX, USA, April 16-19, 2016, Proceedings, Part I 21*. Springer, pp. 214–228.

Saxena, Abhinav, Kai Goebel, Don Simon, and Neil Eklund (2008). "Damage propagation modeling for aircraft engine run-to-failure simulation". In: *2008 international conference on prognostics and health management*. IEEE, pp. 1–9.

Sayah, Mohamed, Djillali Guebli, Noureddine Zerhouni, and Zeina Al Masry (2020). "Towards distribution clustering-based deep LSTM models for RUL prediction". In: *2020 Prognostics and Health Management Conference (PHM-Besançon)*. IEEE, pp. 253–256.

Schwab, Klaus (2017). *The fourth industrial revolution*. Currency.

Severson, Kristen A, Peter M Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H Chen, Muratahan Aykol, Patrick K Herring, Dimitrios Fraggedakis, et al. (2019). "Data-driven prediction of battery cycle life before capacity degradation". In: *Nature Energy* 4.5, pp. 383–391.

Shazeer, Noam, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean (2017). "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer". In: *arXiv preprint arXiv:1701.06538*.

Sheppard, John W, Mark A Kaufman, and Timothy J Wilmering (2008). "IEEE standards for prognostics and health management". In: *2008 IEEE AUTOTESTCON*. IEEE, pp. 97–103.

Shorten, Connor and Taghi M Khoshgoftaar (2019). "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1, pp. 1–48.

Si, Xiao-Sheng, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou (2011). "Remaining useful life estimation–a review on the statistical data driven approaches". In: *European journal of operational research* 213.1, pp. 1–14.

Sohn, Kihyuk (2016). "Improved deep metric learning with multi-class n-pair loss objective". In: *Advances in neural information processing systems* 29.

Sohn, Kihyuk, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li (2020). "Fixmatch: Simplifying semi-supervised learning with consistency and confidence". In: *Advances in neural information processing systems* 33, pp. 596–608.

Song, Yan, Shengyao Gao, Yibin Li, Lei Jia, Qiqiang Li, and Fuzhen Pang (2020). "Distributed attention-based temporal convolutional network for remaining useful life prediction". In: *IEEE Internet of Things Journal* 8.12, pp. 9594–9602.

Song, Yisheng, Ting Wang, Puyu Cai, Subrota K Mondal, and Jyoti Prakash Sahoo (2023). "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities". In: *ACM Computing Surveys*.

Song, Yuchen, Lyu Li, Yu Peng, and Datong Liu (2018). "Lithium-ion battery remaining useful life prediction based on GRU-RNN". In: *2018 12th international conference on reliability, maintainability, and safety (icrms)*. IEEE, pp. 317–322.

Su, Xuanyuan, Hongmei Liu, Laifa Tao, Chen Lu, and Mingliang Suo (2021). "An end-to-end framework for remaining useful life prediction of rolling bearing based on feature pre-extraction mechanism and deep adaptive transformer model". In: *Computers & Industrial Engineering* 161, p. 107531.

Sun, Jun and Qiao Sun (2021). "Bearing Prognostics: An Instance-Based Learning Approach with Feature Engineering, Data Augmentation, and Similarity Evaluation". In: *Signals* 2.4, pp. 662–687.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic attribution for deep networks". In: *International conference on machine learning*. PMLR, pp. 3319–3328.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Vogel-Heuser, Birgit and Dieter Hess (2016). "Guest Editorial Industry 4.0–Prerequisites and Visions". In: *IEEE Transactions on Automation Science and Engineering* 13.2, pp. 411–413. DOI: 10.1109/TASE.2016.2523639.

Vollert, Simon and Andreas Theissler (2021). "Challenges of machine learning-based RUL prognosis: A review on NASA's C-MAPSS data set". In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, pp. 1–8.

Wager, Stefan, Alexander Blocker, and Niall Cardin (2015). "Weakly supervised clustering: Learning fine-grained signals from coarse labels". In.

Wang, Fujin, Zhibin Zhao, Zhi Zhai, Zuogang Shang, Ruqiang Yan, and Xuefeng Chen (2023). "Explainability-driven model improvement for SOH estimation of lithium-ion battery". In: *Reliability Engineering & System Safety* 232, p. 109046.

Wang, Haoxiang, Han Zhao, and Bo Li (2021). "Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation". In: *International conference on machine learning*. PMLR, pp. 10991–11002.

Wang, Huaqing, Tianjiao Lin, Lingli Cui, Bo Ma, Zuoyi Dong, and Liuyang Song (2022). "Multi-task learning-based self-attention encoding atrous convolutional neural network for remaining useful life prediction". In: *IEEE Transactions on Instrumentation and Measurement* 71, pp. 1–8.

Wang, Jinjiang, Jianxing Yan, Chen Li, Robert X Gao, and Rui Zhao (2019). "Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction". In: *Computers in Industry* 111, pp. 1–14.

Wang, Yaqing, Quanming Yao, James T Kwok, and Lionel M Ni (2020). "Generalizing from a few examples: A survey on few-shot learning". In: *ACM computing surveys (csur)* 53.3, pp. 1–34.

Wang, Yilin, Yuanxiang Li, Yuxuan Zhang, Yongsheng Yang, and Lei Liu (2021). "RUSHAP: A Unified approach to interpret Deep Learning model for Remaining Useful Life Estimation". In: *2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing)*. IEEE, pp. 1–6.

Wang, Youdao, Yifan Zhao, and Sri Addepalli (2020). "Remaining useful life prediction using deep learning approaches: A review". In: *Procedia manufacturing* 49, pp. 81–88.

Watanabe, Taiki, Tomoya Ichikawa, Akihiro Tamura, Tomoya Iwakura, Chunpeng Ma, and Tsuneo Kato (2022). "Auxiliary Learning for Named Entity Recognition with Multiple Auxiliary Biomedical Training Data". In: *Proceedings of the 21st Workshop on Biomedical Language Processing*, pp. 130–139.

Weller, Orion, Kevin Seppi, and Matt Gardner (2022). "When to use multi-task learning vs intermediate fine-tuning for pre-trained encoder transfer learning". In: *arXiv preprint arXiv:2205.08124*.

Wen, Qingsong, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu (2020). "Time series data augmentation for deep learning: A survey". In: *arXiv preprint arXiv:2002.12478*.

Wenqiang, Ji, Cheng Jian, and Chen Yi (2019). "Remaining useful life prediction for mechanical equipment based on temporal convolutional network". In: *2019 14th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*. IEEE, pp. 1192–1199.

Williams, Ronald J and David Zipser (1989). "A learning algorithm for continually running fully recurrent neural networks". In: *Neural computation* 1.2, pp. 270–280.

Wu, Pengcheng and Thomas G Dietterich (2004). "Improving SVM accuracy by training on auxiliary data sources". In: *Proceedings of the twenty-first international conference on Machine learning*, p. 110.

Wu, Tong and Tengpeng Chen (2023). "A Gated Multiscale Multitask Learning Model Using Time-Frequency Representation for Health Assessment and Remaining Useful Life Prediction". In: *Sensors* 23.4, p. 1922.

Xiang, Sheng, Yi Qin, Fuqiang Liu, and Konstantinos Gryllias (2022). "Automatic multi-differential deep learning and its application to machine remaining useful life prediction". In: *Reliability Engineering & System Safety* 223, p. 108531.

Xiao, Dengyu, Chengjin Qin, Jianwen Ge, Pengcheng Xia, Yixiang Huang, and Chengliang Liu (2022). "Self-attention-based adaptive remaining useful life prediction for IGBT with Monte Carlo dropout". In: *Knowledge-Based Systems* 239, p. 107902.

Xiao, Han, Kashif Rasul, and Roland Vollgraf (2017). *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*.

Xiao, Lei, Junxuan Tang, Xinghui Zhang, Eric Bechhoefer, and Siyi Ding (2021). "Remaining useful life prediction based on intentional noise injection and feature reconstruction". In: *Reliability Engineering & System Safety* 215, p. 107871.

Xing, Haibo, Fei Xiao, and Jianxun Li (2021). "WR-IMT: A time series predictive model for Remaining Useful Life Prediction". In: *2021 33rd Chinese Control and Decision Conference (CCDC)*. IEEE, pp. 5127–5133.

Xu, Dan, Xiaoqi Xiao, Jie Liu, and Shaobo Sui (2023). "Spatio-temporal degradation modeling and remaining useful life prediction under multiple operating conditions based on attention mechanism and deep learning". In: *Reliability Engineering & System Safety* 229, p. 108886.

Xu, Lian, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu (2021). "Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6984–6993.

Xu, Weiyang, Quansheng Jiang, Yehu Shen, Qixin Zhu, and Fengyu Xu (2023). "New RUL prediction method for rotating machinery via data feature distribution and spatial attention residual network". In: *IEEE Transactions on Instrumentation and Measurement* 72, pp. 1–9.

Yan, Jianhai, Zhen He, and Shuguang He (2023). "Multitask learning of health state assessment and remaining useful life prediction for sensor-equipped machines". In: *Reliability Engineering & System Safety* 234, p. 109141.

Yang, Fan, Wenjin Zhang, Laifa Tao, and Jian Ma (2020). "Transfer learning strategies for deep learning-based PHM algorithms". In: *Applied Sciences* 10.7, p. 2361.

Yao, Siya, Qi Kang, Meng Chu Zhou, Muhyaddin J. Rawa, and Abdullah Abusorrah (Apr. 2023). "A survey of transfer learning for machinery diagnostics and prognostics". English (US). In: *Artificial Intelligence Review* 56.4, pp. 2871–2922. ISSN: 0269-2821. DOI: 10.1007/s10462-022-10230-4.

Yi'An, Zhuo, Hao Ziming, Chai Yi, Ma Le, et al. (2023). "Bearing Remaining Useful Life Prediction based on TCN-Transformer Model". In: *2023 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)*. IEEE, pp. 1–6.

Yoa, Seungdong, Minkyu Jeon, Youngjin Oh, and Hyunwoo J Kim (2021). "Learning to balance local losses via meta-learning". In: *IEEE Access* 9, pp. 130834–130844.

Yoon, Andre S, Taehoon Lee, Yongsub Lim, Deokwoo Jung, Philgyun Kang, Dongwon Kim, Keuntae Park, and Yongjin Choi (2017). "Semi-supervised learning with deep generative models for asset failure prediction". In: *arXiv preprint arXiv:1709.00845*.

Yu, Wennian, II Yong Kim, and Chris Mechefske (2020). "An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme". In: *Reliability Engineering & System Safety* 199, p. 106926.

Zeng, Fuchuan, Yiming Li, Yuhang Jiang, and Guiqiu Song (2021). "A deep attention residual neural network-based remaining useful life prediction of machinery". In: *Measurement* 181, p. 109642.

Zhang, Ansi, Honglei Wang, Shaobo Li, Yuxin Cui, Zhonghao Liu, Guanci Yang, and Jianjun Hu (2018). "Transfer learning with deep recurrent neural networks for remaining useful life estimation". In: *Applied Sciences* 8.12, p. 2416.

Zhang, Jiusi, Yuchen Jiang, Shimeng Wu, Xiang Li, Hao Luo, and Shen Yin (2022). "Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism". In: *Reliability Engineering & System Safety* 221, p. 108297.

Zhang, Liangwei, Jing Lin, Bin Liu, Zhicong Zhang, Xiaohui Yan, and Muheng Wei (2019). "A review on deep learning applications in prognostics and health management". In: *Ieee Access* 7, pp. 162415–162438.

Zhang, Linfeng, Muzhou Yu, Tong Chen, Zuoqiang Shi, Chenglong Bao, and Kaisheng Ma (2020). "Auxiliary training: Towards accurate and robust models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 372–381.

Zhang, Quanshi, Ying Nian Wu, and Song-Chun Zhu (2018). "Interpretable convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8827–8836.

Zhang, Wei, Xiang Li, Hui Ma, Zhong Luo, and Xu Li (2021). "Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions". In: *Reliability Engineering & System Safety* 211, p. 107556.

Zhang, Yangyang, Liqing Fang, Ziyuan Qi, and Huiyong Deng (2023). "A Review of Remaining Useful Life Prediction Approaches for Mechanical Equipment". In: *IEEE Sensors Journal*.

Zhang, Yong, Yuqi Xin, Zhi-wei Liu, Ming Chi, and Guijun Ma (2022). "Health status assessment and remaining useful life prediction of aero-engine based on BiGRU and MMoE". In: *Reliability Engineering & System Safety* 220, p. 108263.

Zhang, Yu, Peter Tiňo, Aleš Leonardis, and Ke Tang (2021). "A survey on neural network interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5, pp. 726–742.

Zhang, Zhiyao, Pengpeng Chen, Chenguang Xing, Bo Liu, Ruo Wang, Longxiao Li, Xiaohui Chen, and Enrico Zio (2023). "A Data Augmentation Boosted Dual Informer Framework for the Performance Degradation Prediction of Aero-engines". In: *IEEE Sensors Journal*.

Zhang, Zhizheng, Wen Song, and Qiqiang Li (2022). "Dual-aspect self-attention based on transformer for remaining useful life prediction". In: *IEEE Transactions on Instrumentation and Measurement* 71, pp. 1–11.

Zhao, Chengying, Xianzhen Huang, Yuxiong Li, and Shangjie Li (2021). "A novel cap-LSTM model for remaining useful life prediction". In: *IEEE Sensors Journal* 21.20, pp. 23498–23509.

Zhao, Chengying, Xianzhen Huang, Yuxiong Li, and Shangjie Li (2022). "A novel remaining useful life prediction method based on gated attention mechanism capsule neural network". In: *Measurement* 189, p. 110637.

Zhao, Kaisheng, Jing Zhang, Shaowei Chen, Pengfei Wen, Wang Ping, and Shuai Zhao (2023). "Remaining Useful Life Prediction Method Based on Convolutional Neural Network and Long Short-Term Memory Neural Network". In: *2023 Prognostics and Health Management Conference (PHM)*, pp. 336–343. DOI: 10.1109/PHM58589.2023.00068.

Zheng, Shuai, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta (2017). "Long short-term memory network for remaining useful life estimation". In: *2017 IEEE international conference on prognostics and health management (ICPHM)*. IEEE, pp. 88–95.

Zhou, Danhua, Zhanying Li, Jiali Zhu, Haichuan Zhang, and Lin Hou (2020). "State of health monitoring and remaining useful life prediction of lithium-ion batteries based on temporal convolutional network". In: *IEEE Access* 8, pp. 53307–53320.

Zhou, Jianghong, Yi Qin, Jun Luo, and Tao Zhu (2022). "Remaining useful life prediction by distribution contact ratio health indicator and consolidated memory GRU". In: *IEEE Transactions on Industrial Informatics*.

Zhu, Junjun, Quansheng Jiang, Yehu Shen, Fengyu Xu, and Qixin Zhu (2023). "Res-HSA: Residual hybrid network with self-attention mechanism for RUL prediction of rotating machinery". In: *Engineering Applications of Artificial Intelligence* 124, p. 106491.

Zhu, Xinqi, Chang Xu, and Dacheng Tao (2021). "Where and what? examining interpretable disentangled representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5861–5870.

Zschech, Patrick, Jonas Bernien, and Kai Heinrich (2019). "Towards a Taxonomic Benchmarking Framework for Predictive Maintenance: The Case of NASA's Turbofan Degradation." In: *ICIS*.

Back to Table of Contents