uc3m | Universidad Carlos III de Madrid

Master's Degree in Computer Science and Technology

2022-2023

Master's Thesis

Continually Adaptive Machine Learning applied to a Healthcare Platform for the Detection of Tuberculosis

Simón E. Sánchez Viloria

Jesús Carretero PhD.

Lara Visuña Perez

Leganés, September 2023

ABSTRACT

Keywords: tuberculosis, machine learning, data analysis, era4tb, healthcare

DEDICATION

Special thanks to my advisor, for guiding me and supporting me at every step in the making of this thesis . . .

> “You must know no one rejects, dislikes, or avoids pain because it is pain, but because occasionally circumstances occur in which toil and pain can procure great pleasure [. . .] In a time of freedom, when our power of choice is untrammeled and when nothing prevents us from doing what we like best, every pleasure is to be welcomed and every pain avoided [. . .] But the wise man should always hold himself to the following principle of selection: *Reject pleasure to secure greater pleasures, or else endure pains to avoid worse pains.*”
>
> *- Marcus Tullius Cicero, 45 BC*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

As health information becomes increasingly digitized, machine learning (ML) algorithms play a crucial role in deriving meaningful insights from complex, multi-modal data that is often difficult to interpret by humans. Conventional machine-learning approaches, however, may be inadequate in the face of rapidly evolving health technology, shifting patient needs, and the increasing availability of large amounts of uncertain and noisy data that is deemed too unreliable for use in clinical settings.

Modern techniques aim to address these limitations by refining models in a continual loop, prioritizing data acquisition, labeling, and feedback from experts to dynamically adapt to the data stream reliably. A field of research known in the relevant literature as *continual learning* (CL) [1] This work explores the potential of these techniques in a medical context.

In particular, we consider the problem of developing a machine-learning platform to aid in the research of *tuberculosis* (TB), an infectious disease that affects millions of people worldwide, we incorporate continual and active learning methods into this platform to improve the performance of computer vision models used to detect TB.

The platform proposed here is meant to be integrated into a Healthcare and Data Portal that is being developed as part of an ongoing European project for the research of Tuberculosis, and it might serve as a tool to test and validate the use of machine learning models for its diagnosis.

As such, in an effort to research the competence of continual adaptation methods in that context, the platform incorporates a system that is designed to continually adapt to new data as it becomes available and to ease labeling efforts by prioritizing the acquisition of the most informative data samples, including a front-end interface for the visualization of the detection, labeling, and interaction with the models.

We evaluate the system's performance on a real dataset and compare it to a baseline model that doesn't use adaptation techniques. Our results show that the use of these methods outperforms the baseline in terms of robustness and sample efficiency while maintaining a similar level of accuracy on the test set. Furthermore, the system proposed can automatically adapt to new data and improve its performance over time.

These results demonstrate the potential of incorporating these techniques into designing real-world systems for healthcare applications and other high-stake domains. The final chapter of this work includes a discussion about possible future work and improvements to the system that might facilitate its integration into other projects and a broader discussion about the potential of these techniques, going beyond the scope of this thesis.

1.1. Context and Motivation

Tuberculosis (TB) is an infectious pulmonary disease that has affected humankind for well over 4,000 years [2] and still affects millions of people worldwide. According to the World Health Organization, until the arrival of the COVID-19 pandemic, TB was the leading cause of widespread death from a single infectious agent, even above HIV [3].

This work is done in the context of the European Regimen Accelerator for Tuberculosis (ERA4TB). ERA4TB is a public-private initiative that started in 2020 and aims to create an open European platform to accelerate the development of new treatment regimens for tuberculosis (TB). The project is integrated by over 31 organizations from the European Union and the United States, including academic institutions, research centers, non-profit organizations, and other public and private entities [4].

ERA4TB's mission statement aligns itself with (and is in response to) the United Nation's (UN) Sustainable Development Goals (SGD) to end the TB epidemic by 2030 [5]. The project's website reads, 'The goal of ERA4TB is to deliver an innovative and differentiated combination regimen for the treatment of TB, which can play a key role in the TB elimination agenda' [4].

To address some of the challenges of TB drug development and clinical trial design, one of the project's objectives is developing a data-science-specific platform to enable the efficient use of machine learning methods from the collaborative platform. The platform is meant to be used by researchers and other project stakeholders to facilitate the use, development, and evaluation of ML models that can aid in the research of TB.

Thus, the motivation behind this work comes from the idea of incorporating novel methods into this data-science-specific platform to support researchers and collaborators in their mission to end TB by 2030. The techniques described in this work are designed to improve the performance of supervised models while prioritizing their overall robustness and reliability, aspects that are crucial in the context of healthcare applications.

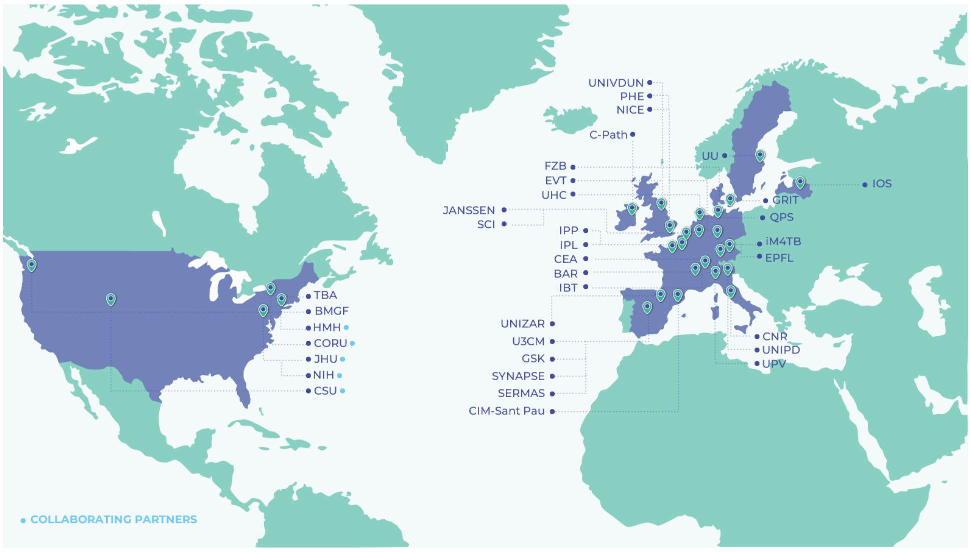

Fig. 1.1. ERA4TB's Consortium of Partners and Collaborators. Source: [4]

1.2. Background Concepts

The following section provides a brief introduction to some of the concepts and techniques that are relevant to this work. It is meant to provide the reader with the necessary high-level information to understand the context behind the ideas proposed that were used to inform every design decision and experiment conducted. For a more in-depth overview of the same concepts, the reader is referred to the literature review in Chapter 2.

1.2.1. Tuberculosis Treatment and Diagnosis

Tuberculosis is caused by the bacillus (bacteria) *Mycobacterium tuberculosis* (MTB), which is transmitted when people who are sick with TB expel the bacteria into the air by coughing, sneezing, or spitting. The disease is preventable with the administration of a vaccine and curable with the use of antibiotics over a significant period (although drug-resistant strains of the bacteria are becoming increasingly common) [6].

Nonetheless, the disease is often underdiagnosed and undertreated, especially in low-resource settings (i.e., developing countries, rural areas, and marginalized/impoverished communities), where the disease is more prevalent, calling for the development of more efficient and cost-effective methodologies to diagnose and treat the disease [3], [6], [7].

Some of the diagnosis techniques to detect TB include chest X-rays, sputum smear microscopy, and molecular tests. *Chest X-rays* are a common procedure to diagnose any signs of tuberculosis due to the wide availability of radiology devices, but they are often inconclusive and require expert radiologists to interpret the results [8].

Sputum smear microscopy is another widely available technique that requires a trained clinician to identify the bacteria under images taken with a microscope of a patient's sputum (a mixture of saliva and mucus from the respiratory tract).

This technique is relatively inexpensive but requires a high concentration of bacteria in the sample to be effective. Additionally, studies argue that in conditions with limited resources and a significant number of samples, there have been reports of poor sample observation and quality control measures, which can result in false-negative results [9].

Molecular tests based on *nucleic acid amplification* (NAATs), similar to the ones popularized to detect COVID-19, are another technique to diagnose TB [10]. These tests identify the presence of bacilli by amplifying the genetic material of the bacteria in a patient's sputum sample (if any is present) and using a chemical solution to react to it.

NAATs are by far the most reliable method to diagnose TB and have the advantage of being rapid and fully automated. However, they are also expensive to produce and require specialized equipment, making them less accessible in low-resource settings [11], [12]. Indeed, a study conducted in 2018 showed that the ratio of smear microscopy tests to NAATs in countries with a high burden of TB was 6:1 [12], [13].

Each of these techniques presents its own benefits and limitations. Recently, there have been efforts to develop machine-learning models that can aid in the diagnosis of TB as a way to reduce the need (or provide a first/second opinion) for expert clinicians in the process and improve the speed and/or accuracy of the diagnosis [1].

[1] A literature review on the topic of TB diagnosis using ML methods is presented in section 2.1.2

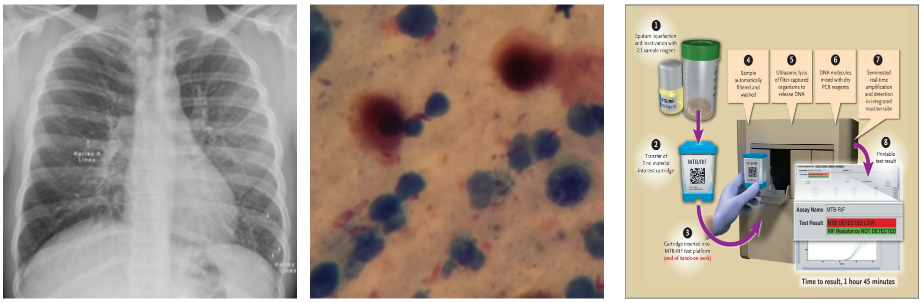
1
Sputum liquefaction and inactivation with 2:1 sample reagent
2
Transfer of 2 ml material into test cartridge
3
Cartridge inserted into MTB-RIF test platform (end of hands-on work)
4
Sample automatically filtered and washed
5
Ultrasonic lysis of filter-captured organisms to release DNA
6
DNA molecules mixed with dry PCR reagents
7
Seminested real-time amplification and detection in integrated reaction tube
8
Printable test result
MTB/RIF
Assay Name MTB-RIF
Test Result MTB DETECTED LOW; RIF Resistance NOT DETECTED
Time to result, 1 hour 45 minutes

Fig. 1.2. Examples of common techniques to diagnose Tuberculosis. Left: Chest X-ray of a patient with TB [14]. Middle: Sputum smear with tuberculosis bacilli [15]. Right: Example of a molecular test for MTB [16].

1.2.2. Supervised and Semi-Supervised Machine-Learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that studies the design of algorithms that can learn from data and make predictions based on it. ML algorithms are fed a set of data samples (often called the training set) and learn a function that maps the input data to a desired output. The goal is then to learn a function that can generalize well to unseen data and make accurate predictions.

Supervised learning is a paradigm of ML where the goal is to fit a function $f : x \rightarrow \hat{y}$ that maps a given input, x, to a 'prediction' output, $\hat{y}$, based on an available finite set of input-output pairs (x_i, y_i) that are passed to a learnable model as 'training' data. The function is learned by minimizing a loss function $L(y, \hat{y})$ that measures the difference between the 'predicted' output and the actual one and returns a value that represents the error of the prediction, which is then used to update the model's parameters.

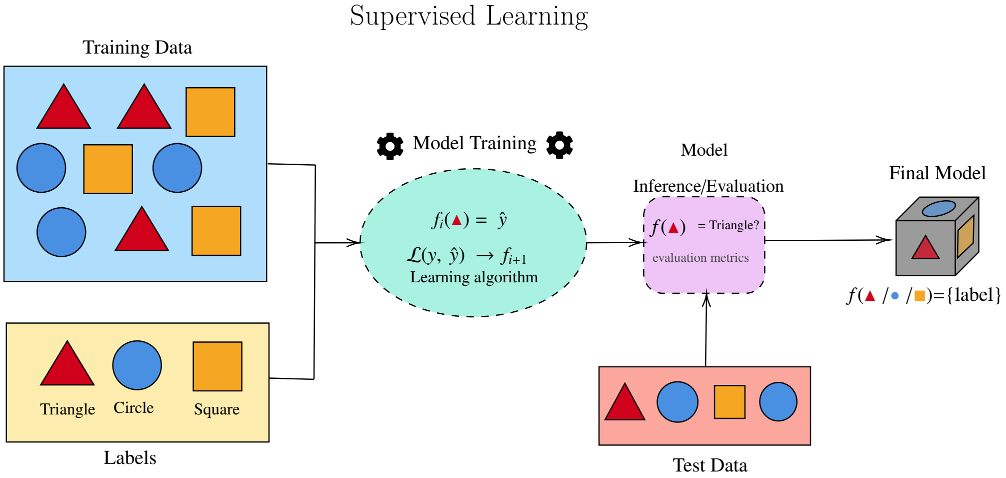

Fig. 1.3. Illustrative example of a supervised machine-learning pipeline.

The most common supervised learning tasks are classification and regression, where the goal is to predict discrete and continuous outputs, respectively. Conversely, the aforementioned loss function is often defined based on the task at hand and the type of data available (e.g., cross-entropy loss for classification tasks, mean squared error for regression tasks, etc.).

The presence of a known output (or label) for each input is what makes this paradigm 'supervised'. For example, in healthcare, supervised learning might be used to predict the presence of a disease or condition based on a set of features extracted (or learned) from the patient's data. A model can be trained to predict the presence or severity of a disease based on a set of symptoms, clinical history, or imaging data.

Besides its many benefits, the most significant disadvantage of supervised learning is that it often needs large amounts of labeled samples to produce accurate and robust results [17], approaches such as *semi-supervised learning* (sometimes also called weak supervision) aim to address this issue by leveraging both labeled and unlabeled data to train the model [18].

Semi-supervised learning works by using unlabeled samples to learn an intermediary representation of the data that can be used as a first step to train a supervised model. This approach is especially useful when the unlabeled data is abundant and easy to obtain, but their labels are scarce and expensive. This is often the case in some healthcare applications, where the labeling task can only be performed by professional workers, making it a costly and time-consuming task [19]–[21].

In the last decade, one set of algorithms that have enabled significant advances in Machine Learning, allowing to solve very difficult problems, is **deep learning**. Deep learning is a subfield of ML that studies the design of algorithms that can learn complex representations of data by hierarchically composing simpler functions in an architecture inspired by the structure of the human brain known as 'Deep Neural Networks' (DNN) [17].

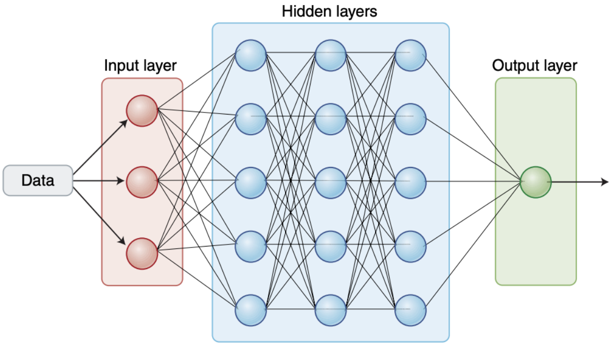

Fig. 1.4. Example of a simplified neural network architecture taken from Topol et. al (2019) [22].

1.2.3. Challenges and Potential of Adopting ML-enabled Systems in Healthcare

ML techniques have seen their adoption in many applications, from malware and spam detection to self-driving cars and environmental modeling. The healthcare field is no exception. In the 20 years between 1995 and 2015, the FDA had approved fewer than 30 algorithms for medical use. In contrast, only in the last 5 years, the total count of new approvals has reached over *10 times* that amount (see Figure 1.5).

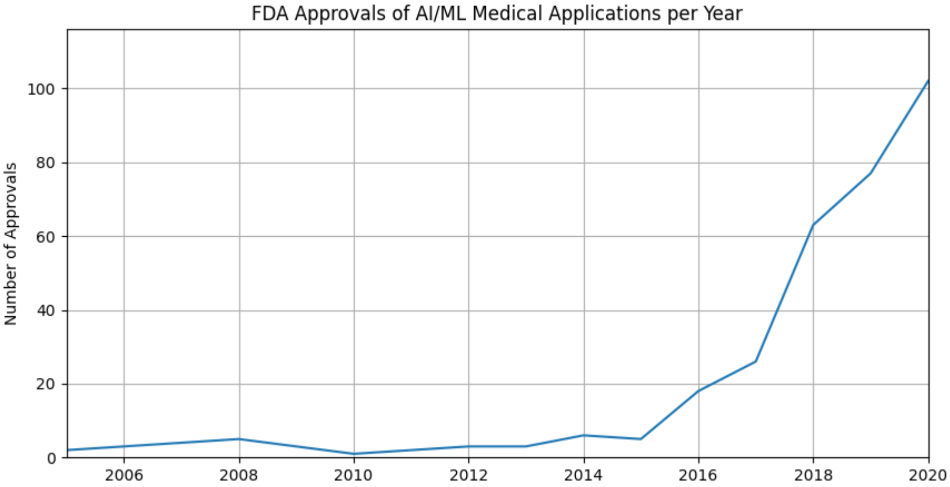

Fig. 1.5. Number of FDA-approved AI applications per year since 2005. Data Source: [23]

Indeed, the recent availability to store and process ever larger amounts of data through the use of Big Data technologies and the development of more powerful hardware and algorithmic techniques have made it possible to train models that can perform complex medical tasks with enough high accuracy and robustness to be considered for use in that industry [22].

Of such techniques, some that have recently disrupted the medical field are those based on **computer vision** (CV). CV methods try to develop algorithms that enable computers to solve visual tasks. It is a broad field that has been applied to problems like image classification, object detection, and image segmentation and has seen significant advances in the last decade thanks to the adoption of Deep Learning algorithms [17].

In the context of healthcare, CV techniques have been primarily used in radiology to aid with the diagnosis of diseases and other tasks through the analysis of medical images (e.g., X-rays, CT scans, MRIs, microscopy, etc.) [24]. Indeed, over 75% of all FDA-cleared AI applications have been for radiology use cases [23].

One study made in 2018 trained a *Convolutional Neural Network* (CNN), a common DL model popular for computer vision tasks, to detect pneumonia from X-ray images at an Indian hospital. The researchers compared the performance of the algorithm with the findings of four expert radiologists and concluded that the algorithm was comparable to - sometimes even *outperformed* - the radiologist in most cases [25].

Company	Year	Usecase	Panel
Apple	2022	Atrial Fibrillation Detection via Apple Watch	Cardiovascular
Arterys	2022	Liver and Lung Cancer Detection	Radiology
Philips Healthcare	2022	Philips Incisive CT Reconstruction	Radiology
GE Healthcare	2021	Deep Learning Image Reconstruction	Radiology
Siemens	2021	AI-Rad Companion for CT Interpretation	Radiology
Icometrix	2018	Brain MRI Analysis	Radiology
23&Me	2017	Genetic Testing for Hereditary Thrombophilia	Hematology

Table 1.1. Examples of FDA-approved AI applications for medical use.

Source: [23]

Other AI systems such as Google Deepmind's *AlphaFold* [26] have been shown to predict the 3D structure of proteins with high accuracy, solving a problem that had been considered to be one of the most challenging in computational biology for over 50 years. Such breakthrough is thought to have a significant impact in applications like drug discovery in the near future [27].

Results like these shine a light on the potential that AI techniques have in the medical field. However, the adoption of such technologies doesn't come with new challenges.

Experts have emphasized the importance of improving aspects of these models like their lack of interpretability, robustness to unseen data, and the difficulty of integrating them into existing workflows before they can be widely adopted in clinical settings [22], [24].

Thus, the design of ml-enabled systems must be mindful of the limitations of such models. Research in the healthcare and AI fields must pave the way to develop workflows that can be trusted by every stakeholder alike to improve the quality of care/research, reduce costs, and overall increase the efficiency of the healthcare system.

1.2.4. Continual Learning and Self-Adaptive Systems

To address some of the challenges of adopting ML-enabled systems in healthcare, we can take a look at the idea of incorporating *continual learning* and *self-adaptive systems* (SAS) in their design. Continual learning refers to the ability of an ML system to learn continually from new data, adapting to novel changes in the data stream that it may have not been exposed to before [1].

SAS has a similar definition, but it generalizes to any software systems that continually monitor faults in their operating environment using a closed feedback loop. The system can then modify its behavior at runtime by the execution of so-called **tactics** in an attempt to fix them, thereby reducing human efforts in the interaction. Nowadays, self-adaptivity is considered a classical concept with ample literature in fields like software engineering and robotics [28]–[30].

The two ideas are closely related, with *continual* (or *lifelong*) learning often used to refer specifically to ML systems while *SAS* to any software that adapts itself to disruptive changes. The core idea of both is the same, though, the ability of a system to 'survive' variations in its environment with or without human intervention and continue to perform its intended function [28].

In this work, we use the term **continually adaptative machine learning** to refer to this idea applied to ML models - drawing inspiration from both concepts.

Consider the scenario of a machine-learning model that has been trained with pictures from an X-ray machine of a particular hospital. The model is then deployed in a different hospital where the X-ray machines are of another and produce images with different characteristics (a particular noise, artifacts, different resolution, etc.).

Because the data distribution is different from the one the model was trained with, the model's performance may likely be affected. However, under a continual learning setting, the model would be able to **adapt to the new data** and improve its performance over time.

On the other side, the design of such systems in practice presents unique challenges that must be addressed. Some experts argue that the questionable robustness of the adaptation process [31] is one of the reasons why **the FDA has never approved a medical application based on continual learning**.

Indeed, some known problems like **catastrophic forgetting** (when the model forgets how to perform a task after learning a new one) and **bias drift** (when the model's predictions suddenly become biased towards a particular class or group of samples) are among the most studied phenomena in continual learning.

Still, the fact that ML models are often deployed as static components in a dynamic environment with the assumption that the data used to train them is representative of the data it will encounter in the real world is another source of ML degradation [31].

In Table 1.2 we describe some relevant causes of degradation of ML systems. It gives an overview of different phenomena that can affect a model's performance and the challenges that must be addressed in the design of an adaptive system.

One important thing to note is that a model experiencing one of the degradation causes listed doesn't mean that it will necessarily fail or that it should be addressed. For example, a model might suffer from catastrophic forgetting after learning new information but that might not affect the performance of the current task in any significant way.

This is why a crucial aspect of designing such systems is that of defining the *adaptation criteria* that should be monitored in order to trigger the right *tactics* that will allow the system to adapt to the specific changes in its environment that cause it to degrade.

In chapter 2 we address some state-of-the-art techniques that can be used to deal with the causes of degradation listed in the table.

Cause	Description	Example
Data Drift	**Covariate shift:** When the input distribution $P(X)$ that the model was trained with differs significantly from the one in the inference environment (i.e. the input changes over time, but the model remains the same).	A model trained on chest X-ray images from a particular dataset is deployed in a hospital where the X-ray machines are of a different brand and produce images with different characteristics.
	Label shift: When the model is trained on a dataset whose class proportions are substantially different from the ones in the inference environment ($P(Y)$).	A model trained on a dataset where the proportion of positive cases is 50% is deployed in a hospital where the proportion is 10%.
Model Unfairness	**Model bias:** The model misrepresents or produces an erroneous causal relationship between its input features and the target output, often caused when the training data is not fully representative of the real-word.	A model trained with data from patients of a hospital in one neighborhood performs poorly after being deployed in another neighborhood with very different demographics.
Learning Plasticity	**Catastrophic forgetting:** The model no longer performs well in an old task after training on new information.	A model is trained to detect a specific disease, but when adapted to detect a different disease, it ‘forgets’ how to detect the first.
	Loss of Plasticity: The model is unable to learn new information after training on a specific task (DNNs are often prone to this phenomenon [1]).	A model is trained to detect a specific disease, but can’t be adapted to detect a different one.
Adversarial Attacks	**Vulnerability to adversarial attacks:** The model is vulnerable to small perturbations to the input that cause it to make very different predictions than it would otherwise.	A bad actor makes imperceptible changes to individual pixels in an X-ray image that cause the model to misclassify the presence of a disease.

Table 1.2. Some causes of Degradation of an ML System and their characteristics.

1.3. Objectives

Main objective

The main objective of this work is to research the most relevant techniques on continual adaptation methods in Machine Learning, make a comparative analysis of the most relevant techniques and their relevance for health applications, and design and implement a system for the diagnosis of tuberculosis that incorporates these techniques into its design that can be integrated into the ERA4TB platform.

The platform should allow its users to incorporate machine-learning models into the platform, facilitate their use, and allow collaboration between researchers and other stakeholders.

Specific objectives

Auxiliary to the main objective, the following specific objectives have been defined to guide the development of the work and evaluate its success:

1. The system should be capable of automatically triggering the continual learning process when new data is available or when the model's performance degrades based on a predefined metric.
2. The platform must implement a feedback loop between the data annotation process and model training that prioritizes the acquisition of the most informative data samples to improve the model's performance (Active Learning).
3. Develop a front-end interface that allows users to interact with the machine-learning models by selecting or submitting new data samples and visualizing the model's predictions.
4. Consider the limitations of the proposed system and the ethical implications of its use and present a well-founded outline of necessary future work to address these limitations or improve the system in a way that aligns with the project's mission statement.
5. Evaluate possible future research directions that could be explored in the area of continual and dynamic adaptation in Machine Learning, highlighting the contributions that have a higher potential for impact in healthcare or other high-stake domains.

1.4. Structure of the Work

This work is divided into five chapters, including this introduction. Chapter 1 describes the context and motivation of this work and its objectives and provides the necessary background information to understand the concepts and techniques used.

Chapter 2 describes relevant work and state-of-the-art techniques. It also provides a literature review of related work in tuberculosis detection and adaptive machine learning systems, highlighting the most important contributions and their limitations and showing examples of their use in healthcare applications.

Chapter 3 describes the methodology used to design and implement the proposed system. It gives a detailed description of the data, models, techniques, and specific tools used in the implementation of the platform, experiments conducted, and relevant metrics to evaluate the system's performance.

Chapter 4 presents the results of the experiments described in the previous chapter, analyzing the performance of the system, and comparing the results with the baseline metrics.

Finally, Chapter 5 presents the conclusions of the work. It discusses the implications of the results obtained in the context of the project and the limitations of the proposed system. It also proposes possible future work that could be explored to improve the system and its integration into the ERA4TB platform and discusses emergent directions in the areas of research discussed in this work, highlighting those that have a higher potential for impact or that we consider to be of special interest.

Additionally, there are two extra chapters at the end where we discuss the regulatory framework, ethical considerations, and the socioeconomic implications of the use of the proposed system (or similar systems) in the context of the project and the healthcare field in general.

2. STATE OF THE ART

The following chapter provides an overview of the current environment and state of the art in the topics related to this work. The goal is to provide a better understanding of the techniques that can be used to address the problem introduced in the previous chapter, highlighting emerging methods and their relevance to the problem.

We begin with a review of prior studies relevant to the topics of this thesis that have achieved state-of-the-art performance or have done similar work as the one proposed here. We go into detail about the implementation of the techniques used in each study, their overall contributions to the field, and how they relate to our work.

Then, in section 2.2, we give a high-level description of known machine learning paradigms and techniques that have been proposed in the literature to address continual adaptation and the degradation of ML systems deployed to real-world scenarios, highlighting their application in the healthcare domain.

2.1. Literature Review

This section provides a detailed overview of the most relevant work in the literature related to the topics of this thesis. It is meant for readers who want to dive more into the details of the techniques studied, how they achieved state-of-the-art performance, their overall contributions, and how they relate to the problems posed in this work.

2.1.1. Computer Vision and DL-Based Object-Detection Techniques

This thesis focuses on applying computer vision (CV) techniques to detect Tuberculosis. Thus, we consider it essential to give an overview of the most critical work in the literature in CV with the goal that the discussion here also serves as a good reference for the proposed solution and the techniques that will be introduced in the upcoming sections.

CV techniques cover a vast field. Since the problem in this work relates more to the tasks of image classification and object detection using Deep Learning (DL) algorithms (to identify Tuberculosis), we will limit our scope to the most relevant work in those areas.

First, it's important to highlight the importance of DL techniques in CV. The immense popularity that DL has gained in the area of CV can be traced back to 2012 when Convolutional Neural Network (CNN) architectures like AlexNet [32] started showing breakthrough performance for solving image classification tasks in recognized competitions like the ImageNet Large Scale Visual Recognition Challenge (ISLVR) [33].

CNNs are a type of neural network that uses *convolutional layers*, which can be thought of as a set of functions that learn image filters through the use of convolutional operations, to extract features from an input image [34]. CNNs were first introduced by Yann LeCun in 1989 [35] (30 years before ImageNet), and they have since been used to achieve SOTA performance on a wide range of CV tasks, including object-detection [17], [34], [36].

Thus, bringing the focus specifically to Deep Learning for object detection tasks - and ignoring others like image segmentation, captioning, or image generation where CNNs have also achieved SOTA - we can find that most literature is divided into two main approaches to the problem: *two-stage* and *one-stage* object detectors.

Two-stage object detection methods first propose a set of candidate regions in the image (the *region proposal* stage) and then classify each region as either containing an object or not (the *classification* stage). **One-stage object detection** methods, on the other hand, directly predict the bounding boxes and class labels of the objects in the image [36]. Figure 2.1 shows a comparison between the two approaches.

Generally, the advantage of two-stage detectors is that they tend to be more accurate than one-stage detectors, as the region proposal stage allows them to focus on a smaller set of candidate regions that can be then individually discriminated by the classification stage. However, this comes at the cost of being slower than one-stage detectors since they require two inference steps through the network [36].

The SOTA status of two-stage object detection methods precedes that of their one-stage counterpart. The popularity of CNNs in the CV field in the early 2010s led to the development of **RCNN** (2014). This architecture first proposes image regions by selective search [37], which are then rescaled and fed into a CNN to extract features to finally use a linear classifier to predict their label [38].

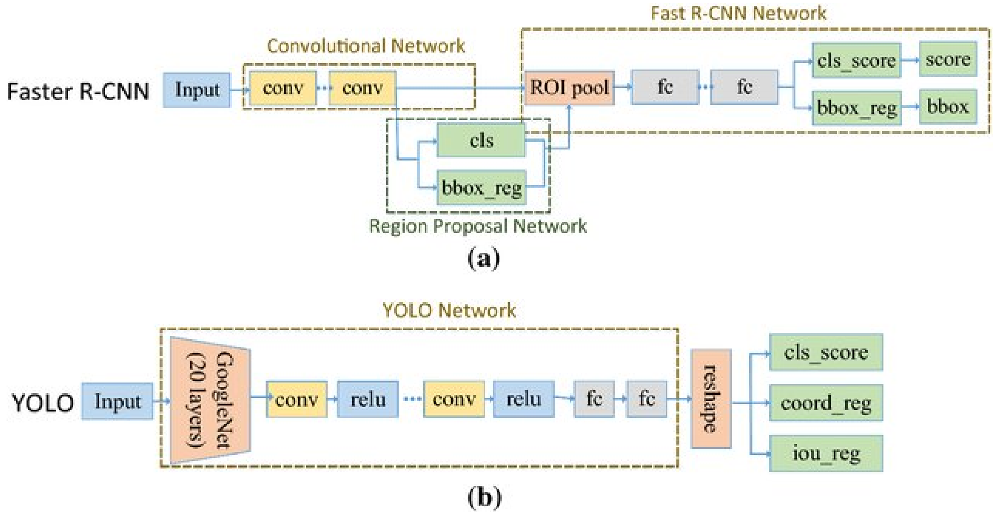

Fig. 2.1. (a) A two-stage Faster R-CNN object detector. (b) A one-stage YOLO object detector. Figure from Li et al. (2019) [39].

The success of RCNN for object detection tasks was followed by SPP-Net (2014), which introduced a Spatial Pyramid Pooling (SPP) layer to allow the model to process images of arbitrary sizes without rescaling them [40]. Then came **Fast RCNN** and **Faster R-CNN** in 2015 [41], [42], who respectively improved on RCNN by training the detector and bounding box regressor jointly, and by using a Region Proposal Network (RPN) to replace the much slower selective search algorithm used by previous models.

While two-stage detectors were achieving state-of-the-art performance in terms of accuracy, they were not fast enough to be used for real-time applications or on embedded devices such as smartphones. This led to the need to develop faster object detection methods that could be used for such purposes [36].

It wasn't until 2016 that one-stage object detectors reached this milestone with the proposal of the **YOLO** (You Only Look Once) model [43]. Unlike RCNN-based approaches, YOLO poses the task as a regression problem, where the model directly predicts the bounding boxes and class probabilities of the objects in one evaluation, allowing the system to be optimized in an end-to-end fashion.

YOLO marked a turning point in the field, a surge in the popularity of one-stage detectors led to more methods being proposed in the following years, from more adequate loss functions like RetinaNet's Focal Loss [44] to subsequent developments to YOLO's architecture like YOLO9K (2016), YOLOv3 (2018), and YOLOv8 (2023) [45]–[47] that improved on the original model's performance, allowing it to retain SOTA performance.

In this age, single-stage object detectors dominate the general benchmarks (e.g., ISLVR, COCO [48], PASCAL VOC [49]) in both accuracy and speed. More recent methods use Transformer architectures [50] to overcome the limitations of traditional CNNs in terms of parallelization and locally restrictive receptive field, showing that abandoning convolutions in favor of an attention-only approach can also achieve SOTA [51], [52].

2.1.2. Tuberculosis Detection using Machine Learning Methods

While general object detection methods have achieved outstanding performance for more common object detection tasks, like detecting cars, people, or animals from images that could be taken in a wide range of conditions (i.e., those that most people would be able to identify), they tend to perform poorly when applied to more domain-specific tasks.

This problem is because models like those described in the previous subsection are trained on big datasets of images that can be commonly found on the internet and are easier to annotate, which tend to exclude more specialized images that are hard to obtain and even harder to annotate.

Furthermore, two-stage object detections like YOLO tend to perform poorly when applied to images containing small objects (e.g., cells, bacteria, etc.) because their region proposal mechanism favors larger objects with a clear distinction from the background.

These latter issues present a problem with the type of datasets usually available for medical applications, where images are typically taken in a more controlled environment and contain objects that only experts in the specific domain can identify. Furthermore, it is likely for objects of interest in medical images to be small and difficult to distinguish, as the need for specialized devices - often noise-prone and with low spatial resolution - is a common thing in the field.

This is the case of the kind of dataset used for TB detection, which is of interest in this work. **In sputum-microscopy images**, for example, the objects of interest are the bacilli that cause tuberculosis, tiny bacteria that are small and challenging to distinguish from other organic materials that surround them [53]. This makes the detection of tuberculosis a task that requires more specialized methods to achieve good performance.

But even though the models that achieve SOTA in general benchmarks tend to fail for medical applications right out of the box, that doesn't mean that they cannot be adapted to more specialized datasets. In fact, the same methods have also been shown to perform well for medical applications when **adapted** to the task, and TB detection is no exception.

Some of the earliest examples we can find in the literature of the use of DL methods for detecting TB come from the work of Osman et al. in the 2010s [53]–[55]. Osman studied the use of techniques like multilayer perceptron (MPL) networks, K-Means clustering, and genetic algorithms, combined with more classical CV techniques like color thresholding and morphological operations, to detect and segment TB in **Ziehl-Neelsen (ZN) stained sputum smear** microscopy images.

More recently, Lakhani and Sundaram (2017) [56] used CNNs to classify Tuberculosis from chest X-ray images automatically. The authors used a deidentified dataset composed of 1007 posteroanterior chest radiographs, of which 15% were used for testing. The model achieved an AUC of 0.99 with an ensemble of AlexNet [33] and GoogLeNet [57] models.

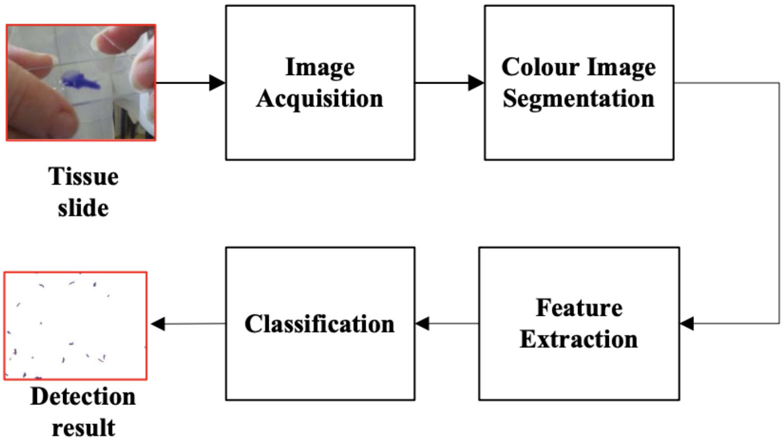

Fig. 2.2. Block diagram of the method proposed by Osman et al. (2010) [54] for automated TB bacilli detection from sputum-smear microscopy images.

Roy et al. (2020) [58] presented a deep learning-based method for the assisted diagnosis of **COVID-19** markers from lung ultrasonography (LUS) images. The researchers collected data from six Italian hospitals and used it to train a CNN architecture derived from Spatial Transformers [59] (not related to Attention-based Transformers) to classify LUS images as either healthy or pathological and segment the affected area. The model was trained on 77 LUS videos from 35 patients for a total of 58,924 image frames and achieved a semantic segmentation accuracy of 96%.

The study most close to our work comes from Visuña et al. (2023) [60], which presented a DL-based technique to localize tuberculosis present on sputum smear microscopy images. The author used a **one-stage object detection method** with a Convolutional Neural Network backbone to detect the presence of bacilli in the images.

Visuña first fragmented the image into patches of 80x80 pixels and then **classified each patch as either containing bacilli or not** for maximum spatial coverage. This study is very relevant to our use case since it uses the same dataset that we will study in this work. Their model was trained on 200 microscopy stain images and, using a 70/30 train/test split, achieved a 99.49%

2.1.3. Continually Adaptive Systems

Continually adaptive systems are systems that can adapt to changes in their environment instantly or over time. They are often used in applications where the environment is constantly changing, such as in robotics or autonomous vehicles. The thesis of this work is about considering their use case in a healthcare setting.

Casimiro et al. (2022) discusses the challenges and opportunities of self-adaptive systems (SAS) in machine learning. The authors propose a framework for the development of SAS that rely on ML components. This framework is based on the concept of **MAPE-K** loops [61], which consists of a set of modules that allow the system to **M**onitor, **A**nalyze, **P**lan, and **E**xecute changes to itself (*tactics*) with the help of a **K**nowledge base that tracks the system's behavior (see figure 2.3).

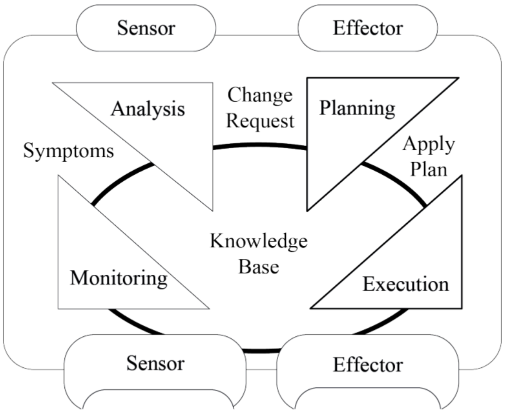

Fig. 2.3. The original MAPE-K Loop, as first introduced by IBM in 2004. Source: [62]

The authors discuss the required changes to the traditional MAPE-K loop and the challenges associated with developing such systems for ML applications. They motivate their ideas by presenting a case study of SAS in the enterprise (ES) and cyber-physical (CPS) system domains. Table 2.1 shows an example of the problems they identified in these domains along with the learning-based tactics considered to solve them.

Note how rather than focusing on how to obtain a plan to adapt the system, Casimiro et al.'s approach is to identify first the specific **causes of degradation** within each domain to then characterize the appropriate **adaptation tactics**.

Similar works such as Gheibi et al. (2020) [29], which makes a review of the literature on machine learning applied to self-adaptive systems, also focuses on the use of MAPE-K loops to develop such systems. However, the scope of the study is about using ML to support SAS rather than using the latter to support the learning process.

Note that the difference between the two approaches is subtle but important. In the first case, the problem is about using learning algorithms as a means to support the adaptive system without considering the tasks that it executes. The task of the ML model, then, is to improve one or more functions of the MAPE loop. For example, to predict the best tactic to use in a given situation (Planning) or detecting anomalies in the behavior of the system (Monitoring).

In the second case, the one we care about, the problem is about using the MAPE-K loop as a means to **improve the learning process** of an ML model. That is, we are putting the learning process at the center of the system, and the task of the MAPE-K loop is to support it with any tactic (learning-based or not) that can help improve the performance and/or efficiency of the model.

The idea of bringing MAPE-K loops to the context of this work is that much like traditional software systems, ML can also benefit from the study of SAS techniques (and the ample literature that exists on the topic) to improve their performance and robustness.

In her book 'Designing Machine Learning Systems' [63], Chip Huyen presents the challenges and patterns of deploying ML Systems. Like Casimiro et al., she stresses the importance of monitoring a model's behavior to identify causes of degradation and continually update them. She also discusses how having humans in the loop who can provide feedback to the system is essential for their reliability in high-stakes domains.

Problem	Domain	Example Situation	Applicable Tactics
Covariate shift	ES	Transaction patterns change Adversaries poison data	• Component replacement • Unlearning
	CPS	Noise/uncertainty in sensors	• Transfer learning
		Different lighting conditions for face recognition	• Component replacement
		Adversaries manipulate smart meter data	• Human-based labeling • Unlearning
Label shift	ES	Variable fraud rate	• Human-based labeling
	CPS	Unknown command for voice controller	• Human-based labeling
Concept shift	ES	New fraud strategies	• Transfer learning
	CPS	Inhabitant's living patterns	• Retrain • Unlearning

Table 2.1. Casimiro et al. (2022) [30]: Example of problems of Learning Systems within each domain and tactics to solve them.

Vokinger et al. (2021) makes further discussion about the reliability of continually-adapted systems in healthcare, arguing that the risk that such systems pose in such a domain is likely the reason why the FDA has not approved any continual-learning systems for medical use [31]. The authors highlight problems like **catastrophic forgetting** and **bias induction** as two inherent risks of continual learning systems to address.

Adaptive systems that have been deployed for health uses can be more commonly found in the remote sensing domain, where typically a set of sensors is used to monitor signals from the user's body to obtain fitness-related insights. Jha et al. (2021) [64], for example, makes an empirical analysis of continual learning applied to Human-Activity Recognition models that can learn to recognize new activities over time different from the ones they were initially trained on.

A different approach to self-adaptivity comes from the authors of MsO-KELM [65], a system that takes a kernel extreme learning machine (KELM) model [66] and introduces a swarm intelligence algorithm to optimize its hyperparameters in a self-adaptive manner. The authors show that their system can achieve better performance than other KELM-based models. This approach, however, suffers from the lack of flexibility MAPE-K loops provide, as it is not clear how this system would adapt to other problems or models.

In the next section, we will characterize some adaptation techniques and learning paradigms proposed in the literature that could be used as *tactics* of a continual-learning system that tackles the problems described in this work.

2.2. Adaptation Techniques and Learning Paradigms

Throughout this section, we consider a set of novel techniques proposed in the literature related to ML model adaptation. One of the critical aspects in selecting which methods to include in this section - besides their relevance to the topic - was the technical feasibility of implementing them as *tactics* (as described above) in a self-adaptive system.

The idea is that some of the techniques described here may be used as the basis for the adaptation process of an ML model after it has been detected (through specific monitorization) that some form of degradation has occurred. The goal would be to then use some of these techniques (or a combination of them) as a way to improve performance.

2.2.1. Continual Learning

Continual learning refers to the concept of constantly updating a model as new information arrives, allowing it to adapt to changing data [63]. This is, of course, at the core of the problem we are trying to solve in this work. Rather than a specific method or technique, continual learning is a framework that encompasses a set of techniques that allow an ML system to learn continually from a data stream.

In its most basic form, continual learning is about updating the model when new data becomes available. However, this is not as simple as it sounds. The key to a successful continual learning process is that the model is updated such that it performs well on the new data without hurting it in the task it was designed for. While obvious, this is not a trivial problem, and it is a particularly known weak point of current ML algorithms. Often called the **stability-plasticity dilemma** [67].

Richard Sutton, one of the pioneers of the field of reinforcement learning (and author of the infamous article 'The Bitter Lesson' [68]), has very recently studied this problem in *deep supervised learning models*, a subject that is very relevant to this work. In an August 2023 paper (and seminar), he makes the big claim that 'Deep learning does not work for continual learning' [69].

This statement is a bit exaggerated (by Sutton's own admission). What he argues about really is the reality that DNNs tend to eventually become very slow to learn from new data, eventually leading to a catastrophic loss of performance.

Problems with DNN plasticity have been studied before, Ash et. al's (2019) [70] was a very influential paper that discussed the failure of warm-starting neural networks, and proposed a regularization method consisting of shrinking and perturbing slightly the weights of the network at every optimization step, which improved significantly the performance of continual learning tasks.

Related to plasticity is **catastrophic forgetting**, when models forget previously learned information when trained on new data, which is another well-studied problem with ML algorithms [63], [71], [72]. Beyond that, there are other myriads of issues to face when implementing a continual learning setting: how to select the optimal samples to train the model with, dealing with class imbalance, improving hardware efficiency, how to store and manage the data and evaluate performance, to name a few.

Adding more to their complexity, we can consider two frameworks for implementing continual learning in a machine-learning system: *offline* and *online* continual learning.

Learning offline is the most typical example where the model is updated periodically using a batch of data obtained over time, i.e., the model is updated only when a certain amount of *new* data has been collected (and labeled, in the supervised case). This is the most common approach in the literature and also the easiest to implement.

Online learning is the second framework. In this case, the model is adapted as soon as new data arrives, with every new instance - or very small batch of instances - being used to update it. This approach is optimal when it is adamant for the model to adapt to the data as soon as it arrives (streaming applications, for example).

However, online learning algorithms tend to be more computationally expensive than offline learning and can be less effective than their offline counterparts, they are also likely to suffer even more from catastrophic forgetting [63].

The work in this thesis is all about adopting continual learning for healthcare applications. We'll continue this section with techniques that can be used to implement or improve a continual learning system and face the challenges described above.

2.2.2. Transfer Learning and Domain Adaptation

Transfer learning is a technique that aims to improve the performance of a model by 'transferring' the knowledge of a pre-trained model to a new task. This is achieved by first training the model on a large dataset - that was presumably easy to obtain - from which it can learn general features and patterns of the data modality, and then **fine-tuning** it on a smaller dataset specific to the new task. Avoiding the need to train the model from scratch on the new dataset allows the model to achieve better performance with less data and training time [73].

Nowadays transfer learning has been adopted widely by the Deep Learning community in parts because it allows researchers and practitioners in the industry alike to effectively 'recycle' models that have been trained and shared previously on large datasets and apply them for their own purposes. Thus saving the - often prohibitively - expensive time and resources required to train deep-learning models from scratch on such significant amounts of data [74].

Furthermore, it has been shown that initializing a neural network with pre-trained weights (i.e., using transfer learning) can help the model converge faster and achieve better generalization than training it from scratch. Showing the model can leverage the knowledge learned from previous data to learn the new task more efficiently [74].

Transfer Learning is useful for healthcare applications because it allows us to adapt pre-trained models to the medical domain - something referred to as **domain adaptation**. This is particularly relevant to our problem, we can use transfer learning to adapt models trained on general object detection datasets (e.g., COCO [48], PASCAL VOC [49]) to the task of detecting tuberculosis in microscopy images. Something that has been done before in the literature with good results [60].

We can envision this technique as part of a continual system that automatically adapts machine learning components to new tasks using transfer learning, thus reducing the need for human intervention.

2.2.3. Active Learning

Active learning strategies selectively acquire data based on their informativeness or uncertainty to the model. Its value comes from allowing the model to guide its own data acquisition process, thus potentially reducing the need for vast - or unnecessary - amounts of pre-labeled data before a model is trained or updated [20], [21], [63].

Active learning is a technique that has been used in the literature to improve the performance of ML models in medical applications. Chen et al. (2015) [20], for example, used active learning to improve the performance of a named entity recognition (NER) model to extract important entities from clinical texts (diseases, medications, procedures).

The authors used three different data sampling strategies to select the most informative instances to label. To achieve an F-measure of 0.80, their method required 66% fewer labeled instances than a baseline model trained on randomly selected data.

In the literature on active learning, we can identify three main strategies for selecting the most informative instances to label [20], [63], [75]. We describe them as follows:

- **Uncertainty sampling** is a strategy that selects the instances that the model is most uncertain about. It is usually done by selecting samples for which the model's prediction is closest to its decision boundary (i.e., where the model is most unsure about its prediction). For example, for a probabilistic binary classification model, you would sample from instances where the model is closest to 0.5 probability.

- **Expected model change** is a strategy that selects the instances for which the model's parameters are most likely to change. The way this is done depends on the model. For example, for a typical DNN trained with gradient descent methods, this can be done by sampling samples for which the gradient of the loss function might be the largest.

- **Query by committee** is a strategy used mostly in ensemble learning that selects the instances for which the model's predictions are most diverse. This is done by training multiple models on the same dataset and selecting samples that the ensemble disagrees with most.

In the context of this project, active learning enables the development of accurate models using significantly less labeled data, paving the way for more efficient and cost-effective healthcare systems. For this reason, it is a technique that we consider to be of great relevance to developing the system proposed in this work.

2.2.4. Knowledge Distillation

Knowledge distillation is a technique that aims to improve the performance of a model by transferring the knowledge of a larger model (teacher) to a smaller model (student). This is achieved by training the student model to mimic the predictions of the teacher model.

The student model is trained on the same data as the teacher model but is trained to predict the probabilities of the teacher model's predictions instead of the actual labels. This allows the student model to learn from the teacher model's mistakes and improve its performance on the given task [76].

While conceptually simple, knowledge distillation can be a very effective technique to improve the efficiency of an ML system by reducing the computational costs associated. The key, in this case, is to find the right balance between the size of the teacher and student models, as the student model needs to be small enough to be more efficient than the teacher model but large enough to be able to learn from it [76].

Knowledge distillation fits well in a continual adaptation setting. We can envision a knowledge distillation tactic that is executed when a model is too large to be constantly deployed in a resource-constrained environment.

The system may trigger the training of a smaller model that mimics the predictions of the larger one and use it in its place if it is deemed to be more efficient and - approximately - as accurate as the previous model.

2.2.5. Adversarial Training

Adversarial attacks on machine learning occur when an attacker produces inputs intentionally designed to be misclassified by a specific ML model. They are created by adding small perturbations to the input data that may be imperceptible to humans but can cause the model to make a wrong prediction.

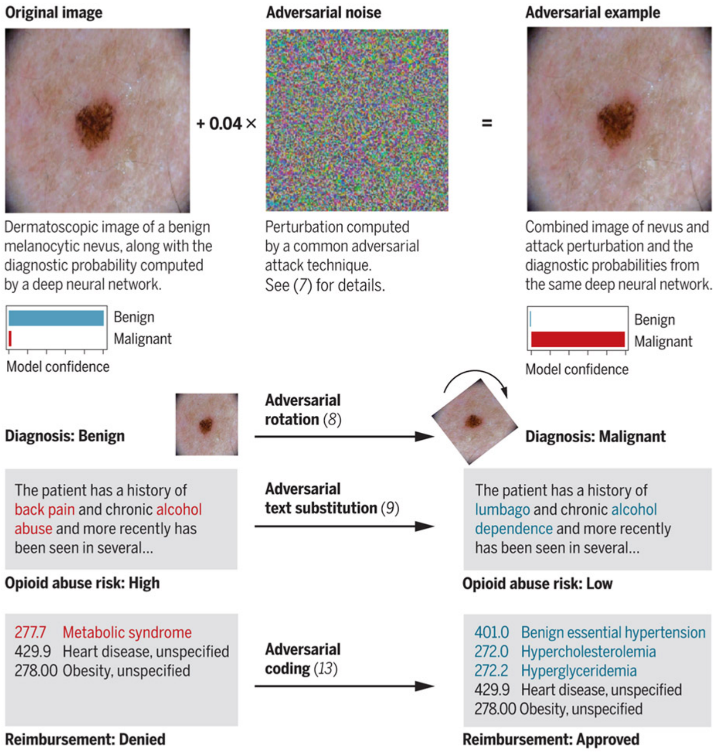
Original image
+ 0.04 ×
Adversarial noise
=
Adversarial example
Dermatoscopic image of a benign melanocytic nevus, along with the diagnostic probability computed by a deep neural network.
Perturbation computed by a common adversarial attack technique. See (7) for details.
Combined image of nevus and attack perturbation and the diagnostic probabilities from the same deep neural network.
Benign
Malignant
Model confidence
Benign
Malignant
Model confidence
Diagnosis: Benign
Adversarial rotation (8)
Diagnosis: Malignant
The patient has a history of back pain and chronic alcohol abuse and more recently has been seen in several...
Adversarial text substitution (9)
The patient has a history of lumbago and chronic alcohol dependence and more recently has been seen in several...
Opioid abuse risk: High
Opioid abuse risk: Low
277.7 Metabolic syndrome
429.9 Heart disease, unspecified
278.00 Obesity, unspecified
Adversarial coding (13)
401.0 Benign essential hypertension
272.0 Hypercholesterolemia
272.2 Hyperglyceridemia
429.9 Heart disease, unspecified
278.00 Obesity, unspecified
Reimbursement: Denied
Reimbursement: Approved

Fig. 2.4. Adversarial examples in health from Finlayson et al. (2019) [77].

Adversarial attacks are a major concern in healthcare applications. An attacker may purposely create these types of examples to fool an ML system into making incorrect decisions, which can have severe consequences like misdiagnosing a patient or prescribing them unnecessary treatment.

Furthermore, adversarial examples shine a light on the black-box nature of several types of ML models. The fact that these models can be fooled by small changes that are not perceptible by any expert raises questions about their general **reliability and trustworthiness.** This is why this phenomenon is an active area of research in the field [77].

One way to deal with adversarial attacks is by choosing models that are more **robust** to these examples. Goodfellow et al. 2015, for example, favor the use of nonlinear model families like Radial Basis Function Networks or using regularization strategies like dropout, weight decay, or gradient masking to improve their robustness [78].

Another popular approach to make ML models more robust to these attacks is to feed adversarial examples directly into their training data. This is known as **adversarial training**. Methods like the Fast Gradient Sign Method [78] (FGSM) or the more recent Projected Gradient Descent (PGD) [79] can be used to generate adversarial examples and train the model on them.

For use cases where the vulnerability of healthcare systems is considered important, one could try to integrate adversarial training into the continual learning process, for example, by incorporating adversarial samples into a fine-tuning dataset, as a way to improve the robustness of the model to adversarial attacks.

2.2.6. Dynamic Quantization and Network Pruning

We consider the idea of integrating more hardware-related optimization techniques that aim to reduce the size of a DNN and/or reduce its computational cost. Quantization and Network pruning are some interesting methods to accomplish this. These two techniques have been shown to be promising in significantly reducing the computational cost of Deep Neural Networks without significantly affecting their performance [80]–[82].

The way they go about doing that is different. Network pruning attempts to remove redundant parameters of a neural network to **optimize its size and computational cost**. Quantization, on the other hand, reduces the precision of the weights and activations to **reduce its memory footprint** (e.g., converting all 32-bit floating-point numbers to 16-bit to cut memory size in half [81], [82]).

It is important to note, however, that there's generally a **tradeoff** from using these techniques as they tend to reduce the model's performance. Like in the case of knowledge distillation, one should assess a balance between task performance and computational costs to determine whether the use of these techniques is worth it.

The way we would propose such an adaptation tactic would be to continually evaluate different versions (quantized, not quantized, pruned, not pruned) of the model to determine which one is the most suitable for the given task, and under what circumstances.

For example, we could have a model that is trained on a particular task and then quantized to reduce its computational cost. The system would then evaluate the performance of the quantized model on the task under all monitored aspects and determine whether the performance drop is relevant enough to warrant the use of the quantized model and reduce the computational cost of the overall system.

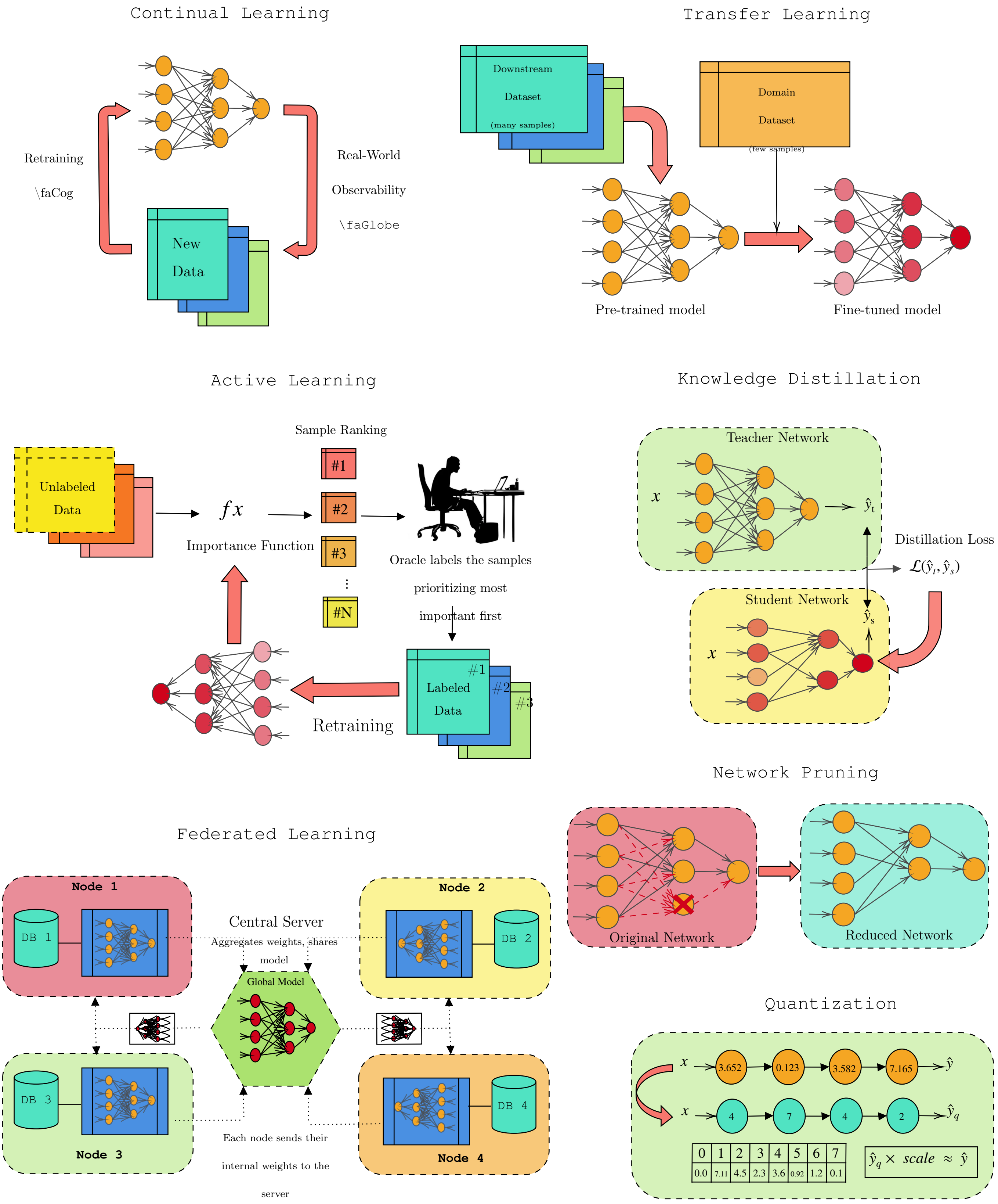

Fig. 2.5. Illustrations of the relevant Machine Learning Paradigms and Techniques

Reference	**Domain**	**Technique(s)**	**Summary of Contributions**
Visuña et al. (2023) [60]	Tuberculosis detection	Object detection, Image preprocessing, CNN fine-tuning, NASNetMobile	Tested novel DL-based method on 200 microscopy stain images of sputum. Reached 99.49% precision and 92.86% recall

Table 2.2. Summary of the most relevant works in the literature.

3. DESIGN OF THE SOLUTION

This work is driven by the problem of designing platforms primarily used to perform Machine-Learning inference (only obtaining the output predictions from a model) while ensuring that the obtained outputs are as reliable and robust as possible when encountering new data. This is important in scenarios where a model is deployed in high-stake environments where their output causes significant downstream impact, and inaccurate predictions may be costly to the relevant stakeholders.

We consider that these types of platforms have the following characteristics:

- The (already trained) models are uploaded to the platform primarily to obtain predictions from new data. We refer to the bag of models available to the platform as the *model repository*.
- The data used to train and evaluate the models is available. This means the platform can have prior knowledge about the data distribution the models were trained on.
- The platform has enough hardware resources available to perform inference with the models that it has available.
- The platform constantly receives new data for inference, which is assumed to be independent of the data used to train the models.
- Labeling capabilities are limited. That is, annotating new data is costly and unfeasible in large amounts. This is common in real-world applications, notably in the healthcare sector, where the labeling task is performed by professional workers, making it a costly and time-consuming task [19]–[21].

Furthermore, for the purpose of this work, we consider that the inference environment is deployed alongside an *experimental* environment. This environment is assumed to be detached from the training environment and is set up to continuously evaluate the models' robustness on new data, evaluate its performance after retraining using the aforementioned techniques, and compare the results with the deployed model. If the experimental model

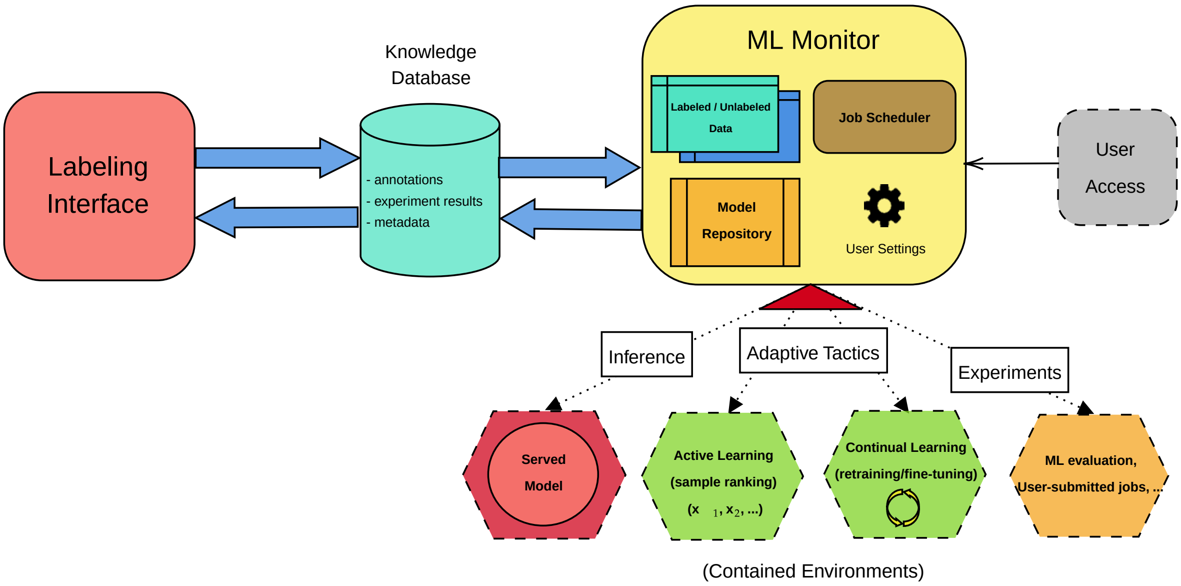

Fig. 3.1. Diagram of the Proposed System

outperforms the deployed model, the latter is updated with the new model. This process is repeated for the entire lifetime of the application.

The role of the human annotator in designing machine-learning models is also an important aspect of this work...

4. RESULTS

In this chapter, we present a rundown of the results obtained from evaluating all the experiments. First, an analysis of the results of testing our system...

4.1. Analysis of the results

.

4.2. Comparison of the different methods

4.3. Discussion

5. CONCLUSIONS

9-10 páginas

Discuss the results obtained and implications in the project context, limitations of the proposed system, future work, etc.

5.1. Main Implications

0.65 pág

5.2. Limitations of the System

0.5 pág

5.3. Future Work

2.75 pág

The following section. . .

5.3.1. Implementing new adaptation strategies to the system

0.5 pág

5.3.2. Improving the explainability of the continual learning process

0.5 pág

Anytime a model is trained on a particular dataset, any biases present on that dataset are also introduced to the model. In any high-stakes environment, the researchers who design these models should be aware of the biases present in the data and how they might affect the model's predictions and consider them in the decision-making process.

Furthermore, this aspect is especially crucial in the healthcare sector, where legislation and ethical guidelines stress the importance of AI systems being transparent [83], [84]. Healthcare professionals have the moral and legal obligation to be able to explain the decisions made by the AI systems they use to the relevant stakeholders, and the latter also has the right to know how the decisions that affect them are made.

In that regard, one of the main limitations of the system proposed here is the lack of explainability of the continual learning process, where a model that is designed initially to be as transparent as possible might gradually lose explainability power as it adapts to new data distributions over time.

Methods such as TRAK [85] have been proposed as a way to improve the explainability of DNNs by providing a way to trace the predictions of a model to individual instances of their training data in a concept known as *data attribution*, which has been proven helpful in improving sample selection for active learning approaches [85]–[87].

5.3.3. Potential to run the system as a federated learning platform

1.75 pág

Federated learning is the concept of training a model using data from multiple sources without having to share the data itself. This is achieved by training the model on each source separately and combining the results to obtain a final model. This technique benefits healthcare applications where data privacy is a significant concern.

Indeed, efforts such as Joshi et al.'s (2022) [88] have demonstrated the benefits and potential of taking a federated learning approach to training machine learning models in the healthcare industry to protect the privacy of patients and/or give more utility to highly-valuable datasets that might require significant research effort to collect.

Thus, a future proposal to improve the system presented in this work would be to possibly adapt the current platform to one where the models are trained in a federated fashion.

We would propose a system where each research laboratory trains an instance of a global model - previously agreed to train - with its own local data - privately held - and shares the trained weights with a central server. The new platform might be mainly used to perform analysis or research on public data. At the same time, each client could have an instance of the annotation frontend and backend, including the active learning framework, running locally to annotate their data.

The motivation for this comes from the fact that it is challenging to convince partners (even within the same project organization) to share their data. In the healthcare industry, data is considered a very valuable asset, with a high cost to obtain and annotate, which is why it is often used as a bargaining chip in negotiations between companies and research organizations .

include citation

Show a diagram of the proposed federated learning system.

5.4. Further Research Directions

4.5 pág

The following are some of the recent and upcoming research directions in machine learning that we consider relevant to the work presented in this thesis.

Unlike the previous section, which focused on the limitations of the proposed system and detailed ideas about how to address or improve them in the immediate future, this section takes a broader view, focusing on gaps in the current state-of-the-art and what might be considered to be the next steps in the field.

The idea is that these directions could used as a starting point for future research (i.e., a Ph.D. thesis) on the topics presented in this thesis, either as a continuation of some of the work here or as a completely new approach to the problem. We make no claims about the feasibility of these ideas but rather present why I consider them interesting for further research and/or discussion.

5.4.1. Future directions in Tuberculosis AI Research

<1 pág

Detection will soon be a solved problem thanks to NAATs (or rather, more of a money problem than a technical one). Thus, the emphasis should be less on CV-based detectors and more to advances in drug discovery, finding new biomarkers, etc. . . .

5.4.2. Scalable Adaptability through Mixtures of Experts

1.5 pág

Mixture of experts (MoE) systems is a type of ensemble model that combines the predictions of multiple models to obtain a final prediction. The difference between MoEs and other ensemble models is that the predictions of the individual ‘experts’ are combined using a **gating/routing function** - typically a neural network - that adapts to the given data point and dynamically determines the weight of each model in the final prediction [89].

MoEs are really powerful systems. They have been shown to be able to learn complex multimodal distributions and have been used in a wide variety of applications, including object detection, language modeling, machine translation, and even multiomics [90]–[93].

But by far, the most significant advantage of MoEs is that each expert model can be deployed independently, allowing for a more flexible, modular system capable of being distributed and data-parallelized among different hardware resources.

The computational sparsity of these systems has shown the capacity of MoEs to scale DNN models to outrageous amounts of parameters. Recently, researchers at Google Brain presented a MoE architecture called Switch Transformers [94] that allows language models (AI systems that can generate text) to **scale to even a trillion parameters** at a constant computational cost.

Furthermore, systems that take advantage of MoEs to scale LLMs (Language Models with a Large number of parameters) already exist and have been deployed to production for applications as big as OpenAI's ChatGPT (GPT-4 is thought to be a mixture of 8 experts, each with over 220 *billion* parameters [95]).

In the context of the area of this work, we consider that MoEs could be used to create a more **scalable and robust adaptable system**. The idea is that it would be composed of a set of 'expert' models, each specialized in a particular aspect of the input data. The system would then adapt to new tasks by autonomously learning a new expert model or retraining an existing one when the need arises.

The main advantage of this approach is that it could potentially allow the system to **scale to a large number of tasks and data distributions**, only needing to retrain the gating function continually instead of entire models.

Another advantage of MoEs is the potential for more **failsafe systems**. By having each model deployed independently in a distributed way, one could devise a mechanism that detects when one of the expert models fails, either due to a hardware/software error or due to a significant performance drop, and automatically replaces the gating function of the current MoE with one (previously trained) that excludes the failing model.

One could even use such a mechanism to save operational costs. The system might monitor the number of instances being routed to each expert model, and if one of the models is not being used, it could be automatically shut down to save resources.

Figure 5.1 shows a diagram illustrating the concept of a mixture of expert system.

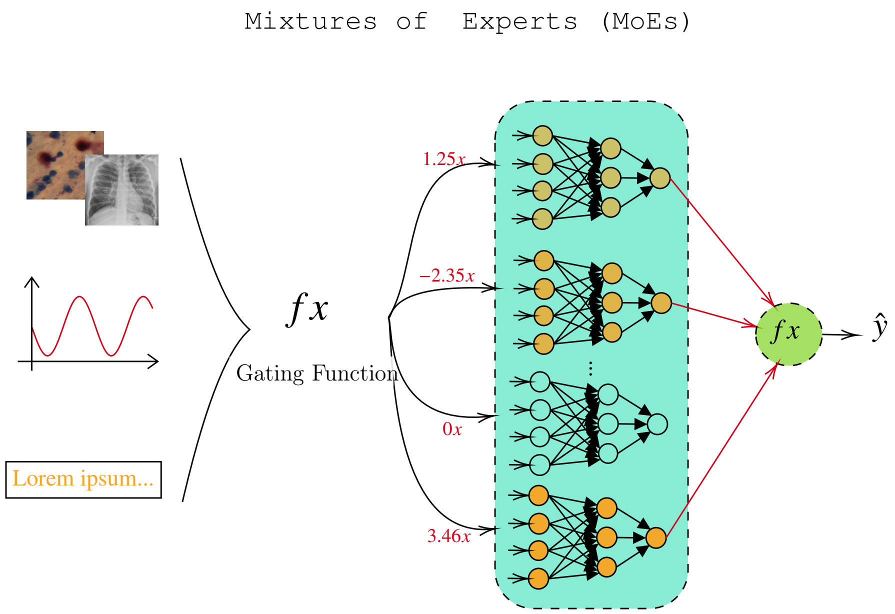

Fig. 5.1. Diagram of a mixture of experts (MoE) system.

5.4.3. Meta-Learning and L2L Systems

1 pág

Much like human learners, who, building from previous knowledge, continuously seek and filter information that could be useful to learn new concepts and skills, an area of research in machine learning concerns the design of programs/systems that can efficiently improve their learning process without the need for explicit human intervention. This area of research is known as **meta-learning**, and it is a very active area of research in AI .

add citation

Meta-learning is a technique that aims to improve the performance of machine-learning models by ‘learning to learn’ (L2L) a certain task. Such ideas have been successfully applied to a wide range of problems, including computer vision, natural language processing, robotics, video games, and more [96].

The way meta-learning is formulated is by training a model on a variety of tasks and then using the knowledge gained from those to improve its performance on new tasks or learn it faster / more sample-efficiently than if it had been trained only for that task [96].

This idea is regarded to have been first introduced by Dr. Jurgen Schmidhuber in 1987 with his thesis ‘Evolutionary Principles in Self-Referential Learning’. In his work, Schmidhuber proposed an algorithm that adaptively improves its learning skills by recursively applying genetic programming to itself and ensuring that only ‘useful’ modifications (made by the program to itself) ‘survive’ in an evolutionary fashion [97].

Recently, Finn et al. (2017) [98] propose a model-agnostic framework for meta-learning that can be applied to any deep-learning architecture and learning task. The framework consists of . . .

The idea behind researching this area is that, by adopting this L2L framework, we could design a system that can continually improve its learning process over time.

As a low-hanging fruit, we can envision the design of a system similar to the one proposed in this work that integrates and builds upon concepts from meta-learning, unlearning, knowledge distillation, and transfer learning [2], that enables model that can more sophisticatedly adapt to new problems.

This self-adaptive process would necessarily be based on evaluating the model's performance and a metric of the 'necessity' of adaptation/learning that task better, but rather than relying on simple heuristics and a model-agnostic approach, the system would trigger a more complex adaptation process.

5.5. Final Remarks

0.75 pág

. . .

Unlike many scientific advances, breakthroughs such as a true self-improving model that adapts itself continually in an online fashion will have immediate applications in every domain, from healthcare to education, to economics, to scientific discovery itself . . .

[2] See section 2.2

REGULATORY FRAMEWORK

The material written in this thesis is published under a Creative Commons license CC BY-NC. The work here can be shared and distributed for non-commercial purposes as long as attribution is properly credited. Adaptations of the work are also permitted as long as they are shared under the same license. The sole copyright and intellectual property belongs to the author.

Similarly, the software developed for this work is openly available [3] and licensed under an Apache License 2.0, which grants permission to the use, distribution, and modification of the code for commercial and non-commercial purposes (restrictions apply, see license for details).

add url to github repo

Any desire to use this work or any material derived from it for commercial and/or monetary purposes should be communicated to the author and made with explicit written permission. Finally, note that the author is not liable for any direct or indirect consequential damages of any kind that arise from the use of any material in this work or from any derivatives of it in which the author is not directly involved.

Possible references to current or possible future legislation/regulations about the use of AI for healthcare applications, healthcare data, or other related topics (e.g., GDPR, HIPAA, EU AI Act, etc.)

Ethical Considerations

The EU published in 2019 their ‘Ethic guidelines for trustworthy AI’ [83], which are based on seven key requirements that AI systems should meet in order to be considered trustworthy. These requirements are human agency and oversight, technical robustness and safety, privacy and data governance, transparency, diversity with regard to non-discrimination and fairness, environmental and societal well-being, and accountability.

Any AI system that is developed and deployed should meet these requirements, and

[3] All code developed in this work can be found under the following url: https://github.com/simonsanvil/..., for any questions, concerns, or comments, please contact the author at simonsviloria@gmail.com

this work is no exception . . .

Furthermore, the more recent EU AI Act [84]. . .

SOCIO-ECONOMIC ENVIRONMENT

Budget

The estimated costs of the realization of this project include those related to the human labor and material costs associated with it.

In terms of human resources, the student author of the work, and the advisor and co-advisor of the thesis are considered as the main contributors to the project. The student author is considered to have worked part-time on the project for the duration of the work, while the advisor and co-advisor contributed a few hours per week. The salary per hour of each contributor is estimated to be €18, €30, and €60, respectively, to reflect the estimated salaries of a junior researcher, senior researcher, and university professor.

By analyzing the time planning of the project throughout its duration , the project is estimated to have lasted approximately eight months. During this time, the author contributed an estimated total of 530 hours, while the advisor and co-advisor contributed 50 and 35 hours, respectively.

include Gannt chart?

Table 5.1 shows a summary of the estimated costs of human resources. Thus, we calculate a **total budget of €13140.00** associated with the labor costs of this work.

	# of Hours	Salary per hour (€)	Total Salary (€)
Author	530	18	9540
Co-Advisor	35	60	2100
Advisor	50	30	1500
		Total Cost:	**€13140.00**

Table 5.1. Estimated Costs of Human Resources

In terms of non-human resource costs, the only relevant ones are those associated with the material costs utilized throughout the work. This is because only open-source software tools (Python, LaTeX, Google Colab, etc.) were used. These tools are freely available or under permissive licenses, thus, no costs are associated with them.

In terms of material costs, we consider the following:

- The personal computer from which the thesis was written and experiments designed and executed. The device consisted of a 14" Apple Macbook Pro (2021) valued at €2200 at the time of writing **apple_comprar_2022**. With a 3-year depreciation of €733.33 divided by the duration of the project, this would add €489 to the material costs associated with this work.
- Other equipment, lab materials (e.g., paper, pens, etc.), and other miscellaneous costs are considered negligible and are not included in the budget.

	Software & Licenses	Personal Computer	Miscellaneous	**Total**
Cost (€)	0	489	0	**€489**

Table 5.2. Estimated Material Costs

Thus, the total material costs associated with this work are estimated to be €489.00. If we add that to the cost of human resources, we can calculate a **total budget of €13,629.00** associated with its realization.

Socio-Economic Impact

BIBLIOGRAPHY

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual Lifelong Learning with Neural Networks: A Review," *Neural Networks*, vol. 113, pp. 54–71, May 2019, arXiv:1802.07569 [cs, q-bio, stat]. DOI: 10.1016/j.neunet.2019.01.012.. [Online]. Available: http://arxiv.org/abs/1802.07569 (visited on 08/27/2023).

[2] CDCTB, *World TB Day History*, en-us, Feb. 2023. [Online]. Available: https://www.cdc.gov/tb/worldtbday/history.htm (visited on 08/19/2023).

[3] WHO, *Global Tuberculosis Report 2022*, en, 2022. [Online]. Available: https://www.who.int/teams/global-tuberculosis-programme/tb-reports/global-tuberculosis-report-2022 (visited on 08/19/2023).

[4] *European Accelerator of Tuberculosis Regime Project*, en-US. [Online]. Available: https://era4tb.org/ (visited on 04/24/2023).

[5] World Health Organization. Regional Office for Europe, "Tuberculosis: Fact sheet on Sustainable Development Goals (SDGs): Health targets," en, World Health Organization. Regional Office for Europe, Tech. Rep. WHO/EURO:2017-2388-42143-58059, 2017, number-of-pages: 8. [Online]. Available: https://apps.who.int/iris/handle/10665/340885 (visited on 08/19/2023).

[6] WHO, *Tuberculosis (TB)*, en, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/tuberculosis (visited on 08/20/2023).

[7] IMI, *IMI Innovative Medicines Initiative | ERA4TB | European regimen accelerator for tuberculosis*, en, Jan. 2020. [Online]. Available: http://www.imi.europa.eu/projects-results/project-factsheets/era4tb (visited on 08/20/2023).

[8] P. Escalante, "Tuberculosis," en, *Annals of Internal Medicine*, vol. 150, no. 11, ITC6–1, Jun. 2009. DOI: 10.7326/0003-4819-150-11-200906020-01006. [Online]. Available: http://annals.org/article.aspx?doi=10.7326/0003-4819-150-11-200906020-01006 (visited on 08/20/2023).

[9] P. Desikan, “Sputum smear microscopy in tuberculosis: Is it still relevant?” *The Indian Journal of Medical Research*, vol. 137, no. 3, pp. 442–444, Mar. 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3705651/ (visited on 08/19/2023).

[10] C. C. for Disease Control and Prevention, Governmental, *TB Diagnostic Tool: Xpert MTB/RIF Assay Fact Sheet | TB | CDC*, en-us, Aug. 2016. [Online]. Available: https://www.cdc.gov/tb/publications/factsheets/testing/xpert_mtb-rif.htm (visited on 08/19/2023).

[11] H. Albert *et al.*, “Development, roll-out and impact of Xpert MTB/RIF for tuberculosis: What lessons have we learnt and how can we do better?” eng, *The European Respiratory Journal*, vol. 48, no. 2, pp. 516–525, Aug. 2016. DOI: 10.1183/13993003.00543-2016.

[12] E. MacLean *et al.*, “Advances in Molecular Diagnosis of Tuberculosis,” *Journal of Clinical Microbiology*, vol. 58, no. 10, 10.1128/jcm.01582–19, Sep. 2020, Publisher: American Society for Microbiology. DOI: 10.1128/jcm.01582-19. [Online]. Available: https://journals.asm.org/doi/10.1128/jcm.01582-19 (visited on 08/19/2023).

[13] D. Cazabon *et al.*, “Market penetration of Xpert MTB/RIF in high tuberculosis burden countries: A trend analysis from 2014 - 2016,” eng, *Gates Open Research*, vol. 2, p. 35, 2018. DOI: 10.12688/gatesopenres.12842.2.

[14] B. A. A. Ubaidi, “The Radiological Diagnosis of Pulmonary Tuberculosis (TB) in Primary Care,” en-US, Mar. 2018, Publisher: clinmed journals. DOI: 10.23937/2469-5793/1510073. [Online]. Available: https://www.clinmedjournals.org/articles/jfmdp/journal-of-family-medicine-and-disease-prevention-jfmdp-4-073.php?jid=jfmdp (visited on 08/22/2023).

[15] M. I. Shah *et al.*, “Ziehl–Neelsen sputum smear microscopy image database: A resource to facilitate automated bacilli detection for tuberculosis diagnosis,” *Journal of Medical Imaging*, vol. 4, no. 2, p. 027 503, Apr. 2017. DOI: 10.1117/1.JMI.4.2.027503. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5492794/ (visited on 08/22/2023).

[16] C. C. Boehme *et al.*, “Rapid Molecular Detection of Tuberculosis and Rifampin Resistance,” *New England Journal of Medicine*, vol. 363, no. 11, pp. 1005–1015, Sep. 2010, Publisher: Massachusetts Medical Society _eprint: https://doi.org/10.1056/NEJMoa0907847. DOI: 10.1056/NEJMoa0907847. [Online]. Available: https://doi.org/10.1056/NEJMoa0907847 (visited on 08/22/2023).

[17] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” en, *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, Number: 7553 Publisher: Nature Publishing Group. DOI: 10.1038/nature14539. [Online]. Available: https://www.nature.com/articles/nature14539 (visited on 05/06/2022).

[18] X. Zhu, “Semi-Supervised Learning Literature Survey,” *Comput Sci, University of Wisconsin-Madison*, vol. 2, Jul. 2008.

[19] A. Yakimovich, A. Beaugnon, Y. Huang, and E. Ozkirimli, “Labels in a haystack: Approaches beyond supervised learning in biomedical applications,” en, *Patterns*, vol. 2, no. 12, p. 100 383, Dec. 2021. DOI: 10.1016/j.patter.2021.100383. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666389921002506 (visited on 04/24/2023).

[20] Y. Chen, T. A. Lasko, Q. Mei, J. C. Denny, and H. Xu, “A study of active learning methods for named entity recognition in clinical text,” en, *Journal of Biomedical Informatics*, vol. 58, pp. 11–18, Dec. 2015. DOI: 10.1016/j.jbi.2015.09.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046415002038 (visited on 04/24/2023).

[21] R. L. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. H. Ngo, “Predicting sample size required for classification performance,” *BMC Medical Informatics and Decision Making*, vol. 12, no. 1, p. 8, Feb. 2012. DOI: 10.1186/1472-6947-12-8. [Online]. Available: https://doi.org/10.1186/1472-6947-12-8 (visited on 04/24/2023).

[22] E. J. Topol, “High-performance medicine: The convergence of human and artificial intelligence,” en, *Nature Medicine*, vol. 25, no. 1, pp. 44–56, Jan. 2019, Number: 1 Publisher: Nature Publishing Group. DOI: 10.1038/s41591-018-0300-7. [Online]. Available: https://www.nature.com/articles/s41591-018-0300-7 (visited on 08/25/2023).

[23] C. f. D. a. R. Health, “Artificial Intelligence and Machine Learning (AI/ML)-Enabled Medical Devices,” en, *FDA*, Oct. 2022, Publisher: FDA. [Online]. Available: https://www.fda.gov/medical-devices/software-medical-device-samd/artificial-intelligence-and-machine-learning-aiml-enabled-medical-devices (visited on 08/26/2023).

[24] A. Esteva *et al.*, “Deep learning-enabled medical computer vision,” *NPJ Digital Medicine*, vol. 4, p. 5, Jan. 2021. DOI: 10.1038/s41746-020-00376-2. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7794558/ (visited on 08/25/2023).

[25] X. Wang *et al.*, “ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, arXiv:1705.02315 [cs], Jul. 2017, pp. 3462–3471. DOI: 10.1109/CVPR.2017.369. [Online]. Available: http://arxiv.org/abs/1705.02315 (visited on 08/25/2023).

[26] Z. Yang, X. Zeng, Y. Zhao, and R. Chen, “AlphaFold2 and its applications in the fields of biology and medicine,” en, *Signal Transduction and Targeted Therapy*, vol. 8, no. 1, pp. 1–14, Mar. 2023, Number: 1 Publisher: Nature Publishing Group. DOI: 10.1038/s41392-023-01381-z. [Online]. Available: https://www.nature.com/articles/s41392-023-01381-z (visited on 08/26/2023).

[27] J. Jumper *et al.*, “Highly accurate protein structure prediction with AlphaFold,” en, *Nature*, vol. 596, no. 7873, pp. 583–589, Aug. 2021, Number: 7873 Publisher: Nature Publishing Group. DOI: 10.1038/s41586-021-03819-2. [Online]. Available: https://www.nature.com/articles/s41586-021-03819-2 (visited on 08/26/2023).

[28] F. D. Macías-Escrivá, R. Haber, R. del Toro, and V. Hernandez, “Self-adaptive systems: A survey of current approaches, research challenges and applications,” *Expert Systems with Applications*, vol. 40, no. 18, pp. 7267–7279, Dec. 2013. DOI: 10.1016/j.eswa.2013.07.033. (visited on 08/30/2023).

[29] O. Gheibi, D. Weyns, and F. Quin, “Applying Machine Learning in Self-adaptive Systems: A Systematic Literature Review,” *ACM Transactions on Autonomous and*

Adaptive Systems, vol. 15, no. 3, pp. 1–37, Sep. 2020. DOI: 10.1145/3469440. (visited on 08/31/2023).

[30] M. Casimiro *et al.*, "Self-adaptive Machine Learning Systems: Research Challenges and Opportunities," in Aug. 2022, pp. 133–155. DOI: 10.1007/978-3-031-15116-3_7.

[31] K. N. Vokinger, S. Feuerriegel, and A. S. Kesselheim, "Continual learning in medical devices: FDA's action plan and beyond," *The Lancet Digital Health*, vol. 3, no. 6, e337–e338, Jun. 2021. DOI: 10.1016/S2589-7500(21)00076-5. (visited on 08/30/2023).

[32] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012. (visited on 08/28/2023).

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press, 2016.

[35] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989. DOI: 10.1162/neco.1989.1.4.541.

[36] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023. DOI: 10.1109/JPROC.2023.3238524.

[37] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," *International Journal of Computer Vision*, vol. 104, 2013.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, Comment: Extended version of our CVPR 2014 paper; latest update (v5) includes results using deeper networks (see Appendix G. Changelog), Oct. 2014. DOI: 10.48550/arXiv.1311.2524. arXiv: 1311.2524 [cs]. (visited on 08/28/2023).

[39] C. Li, B. Zhang, H. Hu, and J. Dai, "Enhanced Bird Detection from Low-Resolution Aerial Image Using Deep Neural Networks," *Neural Processing Letters*, vol. 49, Jun. 2019. DOI: 10.1007/s11063-018-9871-z.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in vol. 8691, Comment: This manuscript is the accepted version for IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2015. See Changelog, 2014, pp. 346–361. DOI: 10.1007/978-3-319-10578-9_23. arXiv: 1406.4729 [cs]. (visited on 08/28/2023).

[41] R. Girshick, *Fast R-CNN*, Comment: To appear in ICCV 2015, Sep. 2015. DOI: 10.48550/arXiv.1504.08083. arXiv: 1504.08083 [cs]. (visited on 08/28/2023).

[42] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Comment: Extended tech report, Jan. 2016. DOI: 10.48550/arXiv.1506.01497. arXiv: 1506.01497 [cs]. (visited on 08/28/2023).

[43] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, May 2016. DOI: 10.48550/arXiv.1506.02640. arXiv: 1506.02640 [cs]. (visited on 08/28/2023).

[44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal Loss for Dense Object Detection*, Feb. 2018. DOI: 10.48550/arXiv.1708.02002. arXiv: 1708.02002 [cs]. (visited on 08/28/2023).

[45] J. Redmon and A. Farhadi, *YOLO9000: Better, Faster, Stronger*, Dec. 2016. DOI: 10.48550/arXiv.1612.08242. arXiv: 1612.08242 [cs]. (visited on 08/28/2023).

[46] J. Redmon and A. Farhadi, *YOLOv3: An Incremental Improvement*, Comment: Tech Report, Apr. 2018. DOI: 10.48550/arXiv.1804.02767. arXiv: 1804.02767 [cs]. (visited on 08/28/2023).

[47] G. Jocher, A. Chaurasia, and J. Qiu, *YOLO by Ultralytics*, version 8.0.0, Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[48] T.-Y. Lin *et al.*, *Microsoft COCO: Common Objects in Context*, Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list, Feb. 2015. DOI: 10.48550/arXiv.1405.0312. arXiv: 1405.0312 [cs]. (visited on 08/28/2023).

[49] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010. DOI: 10.1007/s11263-009-0275-4. (visited on 08/28/2023).

[50] A. Vaswani *et al.*, *Attention Is All You Need*, Comment: 15 pages, 5 figures, Jun. 2017. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762 [cs]. (visited on 08/28/2023).

[51] N. Carion *et al.*, *End-to-End Object Detection with Transformers*, May 2020. DOI: 10.48550/arXiv.2005.12872. arXiv: 2005.12872 [cs]. (visited on 08/28/2023).

[52] X. Zhu *et al.*, *Deformable DETR: Deformable Transformers for End-to-End Object Detection*, Comment: ICLR 2021 Oral, Mar. 2021. DOI: 10.48550/arXiv.2010.04159. arXiv: 2010.04159 [cs]. (visited on 08/28/2023).

[53] M. K. Osman, M. Y. Mashor, and H. Jaafar, "Tuberculosis bacilli detection in Ziehl-Neelsen-stained tissue using affine moment invariants and Extreme Learning Machine," *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, pp. 232–236, Mar. 2011, Conference Name: its Applications (CSPA) ISBN: 9781612844145 Place: Penang, Malaysia Publisher: IEEE. DOI: 10.1109/CSPA.2011.5759878. [Online]. Available: http://ieeexplore.ieee.org/document/5759878/ (visited on 08/22/2023).

[54] M. K. Osman, M. Y. Mashor, and H. Jaafar, "Detection of mycobacterium tuberculosis in Ziehl-Neelsen stained tissue images using Zernike moments and hybrid multilayered perceptron network," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2010, pp. 4049–4055. DOI: 10.1109/ICSMC.2010.5642191.

[55] F. Ahmad, N. A. Mat-Isa, Z. Hussain, R. Boudville, and M. K. Osman, "Genetic Algorithm-Artificial Neural Network (GA-ANN) Hybrid Intelligence for Cancer Diagnosis," in *2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks*, Jul. 2010, pp. 78–83. DOI: 10.1109/CICSyN.2010.46.

[56] P. Lakhani and B. Sundaram, “Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks,” *Radiology*, vol. 284, no. 2, pp. 574–582, Aug. 2017. DOI: 10.1148/radiol.2017162326. (visited on 08/29/2023).

[57] C. Szegedy *et al.*, *Going Deeper with Convolutions*, Sep. 2014. DOI: 10.48550/arXiv.1409.4842. arXiv: 1409.4842 [cs]. (visited on 08/29/2023).

[58] S. Roy *et al.*, “Deep Learning for Classification and Localization of COVID-19 Markers in Point-of-Care Lung Ultrasound,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2676–2687, Aug. 2020, [TLDR] A novel deep network, derived from Spatial Transformer Networks, is presented, which simultaneously predicts the disease severity score associated to a input frame and provides localization of pathological artefacts in a weakly-supervised way. DOI: 10.1109/TMI.2020.2994459. (visited on 08/29/2023).

[59] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, *Spatial Transformer Networks*, Feb. 2016. DOI: 10.48550/arXiv.1506.02025. arXiv: 1506.02025 [cs]. (visited on 08/29/2023).

[60] L. Visuña, J. Garcia-Blas, and J. Carretero, “Novel Deep Learning-Based Technique for Tuberculosis Bacilli Detection in Sputum Microscopy,” in *International Conference on Interactive Collaborative Robotics*, Springer, 2023, pp. 269–279.

[61] J. Kephart and D. Chess, “The Vision Of Autonomic Computing,” *Computer*, vol. 36, pp. 41–50, Feb. 2003. DOI: 10.1109/MC.2003.1160055.

[62] I. B. M. Redbooks, *A Practical Guide to the IBM Autonomic Computing Toolkit*, en. IBM, International Support Organization, 2004, Google-Books-ID: XHeoSgAACAAJ.

[63] C. Huyen, *Designing Machine Learning Systems*, en. O’Reilly Media, Inc., May 2022, ISBN: 9781098107963. [Online]. Available: https://www.oreilly.com/library/view/designing-machine-learning/9781098107956/ (visited on 04/24/2023).

[64] S. Jha, M. Schiemer, F. Zambonelli, and J. Ye, “Continual learning in sensor-based human activity recognition: An empirical benchmark analysis,” *Information Sci-*

ences, vol. 575, pp. 1–21, Oct. 2021. DOI: 10.1016/j.ins.2021.04.062. (visited on 08/30/2023).

[65] Z. Hao, J. Ma, and W. Sun, "The Technology-Oriented Pathway for Auxiliary Diagnosis in the Digital Health Age: A Self-Adaptive Disease Prediction Model," *International Journal of Environmental Research and Public Health*, vol. 19, no. 19, p. 12 509, Sep. 2022. DOI: 10.3390/ijerph191912509. (visited on 08/30/2023).

[66] J. Xia *et al.*, "Evolving kernel extreme learning machine for medical diagnosis via a disperse foraging sine cosine algorithm," *Computers in Biology and Medicine*, vol. 141, p. 105 137, Feb. 2022. DOI: 10.1016/j.compbiomed.2021.105137.

[67] M. Mermillod, A. Bugaiska, and P. BONIN, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in Psychology*, vol. 4, 2013. (visited on 08/30/2023).

[68] R. Sutton, "The bitter lesson," *Incomplete Ideas (blog)*, vol. 13, no. 1, 2019.

[69] S. Dohare, J. F. Hernandez-Garcia, P. Rahman, R. S. Sutton, and A. R. Mahmood, *Loss of Plasticity in Deep Continual Learning*, Aug. 2023. arXiv: 2306.13812 [cs]. (visited on 08/30/2023).

[70] J. T. Ash and R. P. Adams, *On Warm-Starting Neural Network Training*, Dec. 2020. DOI: 10.48550/arXiv.1910.08475. arXiv: 1910.08475 [cs, stat]. (visited on 08/31/2023).

[71] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," in *Psychology of Learning and Motivation*, G. H. Bower, Ed., vol. 24, Academic Press, Jan. 1989, pp. 109–165. DOI: 10.1016/S0079-7421(08)60536-8. (visited on 08/30/2023).

[72] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual Lifelong Learning with Neural Networks: A Review," *Neural Networks*, vol. 113, pp. 54–71, May 2019. DOI: 10.1016/j.neunet.2019.01.012.. arXiv: 1802.07569 [cs, q-bio, stat]. (visited on 08/27/2023).

[73] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. DOI: 10.1109/TKDE.2009.191.

[74] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, *How transferable are features in deep neural networks?* Comment: To appear in Advances in Neural Information Processing Systems 27 (NIPS 2014), Nov. 2014. DOI: 10.48550/arXiv.1411.1792. arXiv: 1411.1792 [cs]. (visited on 08/30/2023).

[75] B. Settles, *Active Learning* (Synthesis Lectures on Artificial Intelligence and Machine Learning). Cham: Springer International Publishing, 2012. DOI: 10.1007/978-3-031-01560-1. (visited on 09/04/2023).

[76] G. Hinton, O. Vinyals, and J. Dean, *Distilling the Knowledge in a Neural Network*, arXiv:1503.02531 [cs, stat], Mar. 2015. DOI: 10.48550/arXiv.1503.02531. [Online]. Available: http://arxiv.org/abs/1503.02531 (visited on 08/20/2023).

[77] S. G. Finlayson *et al.*, “Adversarial attacks on medical machine learning,” *Science (New York, N.Y.)*, vol. 363, no. 6433, pp. 1287–1289, Mar. 2019. DOI: 10.1126/science.aaw4399. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7657648/ (visited on 04/25/2023).

[78] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and Harnessing Adversarial Examples*, Mar. 2015. DOI: 10.48550/arXiv.1412.6572. arXiv: 1412.6572 [cs, stat]. (visited on 05/16/2023).

[79] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards Deep Learning Models Resistant to Adversarial Attacks*, Comment: ICLR’18, Sep. 2019. DOI: 10.48550/arXiv.1706.06083. arXiv: 1706.06083 [cs, stat]. (visited on 05/16/2023).

[80] M. Á. Carreira-Perpiñán and Y. Idelbayev, *Model compression as constrained optimization, with application to neural nets. Part II: Quantization*, arXiv:1707.04319 [cs, math, stat], Jul. 2017. DOI: 10.48550/arXiv.1707.04319. [Online]. Available: http://arxiv.org/abs/1707.04319 (visited on 08/22/2023).

[81] S. Han, H. Mao, and W. J. Dally, *Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding*, arXiv:1510.00149 [cs], Feb. 2016. [Online]. Available: http://arxiv.org/abs/1510.00149 (visited on 08/22/2023).

[82] M. A. Carreira-Perpinan and Y. Idelbayev, “"Learning-Compression" Algorithms for Neural Net Pruning,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ISSN: 2575-7075, Jun. 2018, pp. 8532–8541. DOI: 10.1109/CVPR.2018.00890.

[83] European Commission, *Ethics guidelines for trustworthy AI | Shaping Europe’s digital future*, en, Apr. 2019. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai (visited on 08/23/2023).

[84] European Parliament, *EU AI Act: First regulation on artificial intelligence | News | European Parliament*, en, Aug. 2023. [Online]. Available: https://www.europarl.europa.eu/news/en/headlines/society/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence (visited on 08/23/2023).

[85] S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Madry, *TRAK: Attributing Model Behavior at Scale*, arXiv:2303.14186 [cs, stat], Apr. 2023. DOI: 10.48550/arXiv.2303.14186. [Online]. Available: http://arxiv.org/abs/2303.14186 (visited on 05/18/2023).

[86] D. Holzmüller, V. Zaverkin, J. Kästner, and I. Steinwart, *A Framework and Benchmark for Deep Batch Active Learning for Regression*, arXiv:2203.09410 [cs, stat], Aug. 2023. DOI: 10.48550/arXiv.2203.09410. [Online]. Available: http://arxiv.org/abs/2203.09410 (visited on 08/23/2023).

[87] Z. Liu *et al.*, *Influence Selection for Active Learning*, arXiv:2108.09331 [cs], Aug. 2021. DOI: 10.48550/arXiv.2108.09331. [Online]. Available: http://arxiv.org/abs/2108.09331 (visited on 08/23/2023).

[88] M. Joshi, A. Pal, and M. Sankarasubbu, “Federated Learning for Healthcare Domain - Pipeline, Applications and Challenges,” en, *ACM Transactions on Computing for Healthcare*, vol. 3, no. 4, pp. 1–36, Oct. 2022. DOI: 10.1145/3533708. [Online]. Available: https://dl.acm.org/doi/10.1145/3533708 (visited on 08/22/2023).

[89] Z. Chen, Y. Deng, Y. Wu, Q. Gu, and Y. Li, *Towards Understanding Mixture of Experts in Deep Learning*, arXiv:2208.02813 [cs, stat], Aug. 2022. DOI: 10.48550/

arXiv.2208.02813. [Online]. Available: http://arxiv.org/abs/2208.02813 (visited on 08/22/2023).

[90] C. Hwang *et al.*, *Tutel: Adaptive Mixture-of-Experts at Scale*, arXiv:2206.03382 [cs], Jun. 2023. [Online]. Available: http://arxiv.org/abs/2206.03382 (visited on 07/14/2023).

[91] B. Mustafa, C. Riquelme, J. Puigcerver, R. Jenatton, and N. Houlsby, *Multimodal Contrastive Learning with LIMoE: The Language-Image Mixture of Experts*, arXiv:2206.02770 [cs], Jun. 2022. DOI: 10.48550/arXiv.2206.02770. [Online]. Available: http://arxiv.org/abs/2206.02770 (visited on 08/22/2023).

[92] N. Shazeer *et al.*, *Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer*, en, Jan. 2017. [Online]. Available: https://arxiv.org/abs/1701.06538v1 (visited on 08/23/2023).

[93] K. Minoura, K. Abe, H. Nam, H. Nishikawa, and T. Shimamura, *ScMM: Mixture-of-Experts multimodal deep generative model for single-cell multiomics data analysis*, en, Pages: 2021.02.18.431907 Section: New Results, Feb. 2021. DOI: 10.1101/2021.02.18.431907. [Online]. Available: https://www.biorxiv.org/content/10.1101/2021.02.18.431907v1 (visited on 08/22/2023).

[94] W. Fedus, B. Zoph, and N. Shazeer, *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*, Comment: JMLR, Jun. 2022. DOI: 10.48550/arXiv.2101.03961. arXiv: 2101.03961 [cs]. (visited on 08/31/2023).

[95] M. Rickard, *Mixture of Experts: Is GPT-4 Just Eight Smaller Models?* https://matt-rickard.com, Jun. 2023. (visited on 09/04/2023).

[96] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, *Meta-Learning in Neural Networks: A Survey*, arXiv:2004.05439 [cs, stat], Nov. 2020. DOI: 10.48550/arXiv.2004.05439. [Online]. Available: http://arxiv.org/abs/2004.05439 (visited on 08/20/2023).

[97] J. Schmidhuber, “Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook,” Diploma Thesis, Technische Universitat Munchen, Germany, Mar. 1987. [Online]. Available: http://www.idsia.ch/~juergen/diploma.html.

[98] C. Finn, P. Abbeel, and S. Levine, *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*, arXiv:1703.03400 [cs], Jul. 2017. DOI: 10.48550/arXiv.1703.03400. [Online]. Available: http://arxiv.org/abs/1703.03400 (visited on 08/20/2023).

APPENDIX A