EXPLORATORY RESEARCH

ARTIMATION

sesar JOINT UNDERTAKING

D 5.1: Lifelong Machine Learning framework with integration of causality

Deliverable ID:	**[5.1]**
Dissemination Level:	**[CO]**
Project Acronym:	**[ARTIMATION]**
Grant:	**[894238]**
Call:	**[H2020-SESAR-2019-2]**
Topic:	**[Digitalisation and Automation principles for ATM]**
Consortium Coordinator:	**[MDU]**
Edition date:	**[1st September 2022]**
Edition:	**[00.03.00]**
Template Edition:	**02.00.05**

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Authoring & Approval

Authors of the document

Name / Beneficiary	Position / Title	Date
Shaibal Barua	Senior Lecturer at MDU	2022-05-23
Mir Riyanul Islam	Doctoral Student at MDU	2022-05-26
Waleed Jmoona	Research Assistant at MDU	2022-06-03
Mobyen Uddin Ahmed	Prof. at MDU	2022-07-02

Reviewers internal to the project

Name / Beneficiary	Position / Title	Date
Augustin Degas	PhD at ENAC	2022-07-08
Nicola Cavagnetto	Deep Blue	2022-07-08
Ana Ferreira	Deep Blue	2022-07-19

Reviewers external to the project

Name / Beneficiary	Position / Title	Date

Approved for submission to the SJU By - Representatives of all beneficiaries involved in the project

Name / Beneficiary	Position / Title	Date
Mobyen Uddin Ahmed	Professor at MDH	2022-07-15
Christophe Hurter	Prof at ENAC	2022-07-20
Pietro Arico'	PhD at UNISAP	2022-07-19
Stefano Boneli	WP leader at Deep Blue	2022-07-18

Rejected By - Representatives of beneficiaries involved in the project

Name and/or Beneficiary	Position / Title	Date

Document History

Edition	Date	Status	Name / Beneficiary	Justification
00.00.01	2022-05-23	Draft	Shaibal Barua	With SJU Template

EUROPEAN PARTNERSHIP

Co-funded by the European Union

00.00.02	2022-05-31	Draft	Shaibal Barua	Update template with the structure of the document
00.00.03	2022-06-07	Draft	Shaibal Barua	Writing section 2.1, 2.2, 2.3 and 3.1
00.00.04	2022-06-28	Draft	Waleed Jmoona	2.2.3 Long Short-Term Memory Networks (LSTM)
00.00.05	2022-06-28	Draft	Mir Riyanul Islam	Chapter 3 casual modelling
00.00.06	2022-07-07	Draft	Mobyen Uddin Ahmed	Introduction
00.00.07	2022-07-08	Draft	Waleed	Conclusion
00.00.08	2022-07-08	Draft	Shaibal Barua	Complete version for review
00.00.09	2022-07-21	Draft	Shaibal Barua	Update after internal review
00.00.10	2022-07-21	Draft	Mobyen Uddin Ahmed	Revised and added abastract
00.01.00	2022-07-21	Final	Mobyen	Ready for submission
00.01.01	2022-08-22	Draft	Mobyen	Fixed the review comments
00.01.02	2022-08-23	Draft	Mir Riyanul Islam	Reviewing
00.02.00	2022-08-23	Final	Mobyen	Ready for submission
00.02.01	2022-08-31	Draft	Mobyen	Added Executive Summary prepared by deep blue
00.03.00	2022-09-01	Final	Mobyen	Ready for submission

[ARTIMATION]

[TRANSPARENT ARTIFICIAL INTELLIGENCE AND AUTOMATION TO AIR TRAFFIC MANAGEMENT SYSTEMS]

This [D5.1: Lifelong Machine Learning framework with integration of causality] is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No [894238] under European Union's Horizon 2020 research and innovation programme.

Abstract

This report presents lifelong machine learning framework and integration of a causal model for providing an explanation of the delay prediction tool. This report is the deliverable based on tasks 5.1 Lifelong ML framework, knowledge retainment and transfer model, and 5.2 Integration of causality within explainable ML of work package 5. Here, for lifelong machine learning, three methods are introduced: 1) Lifelong random forest with a genetic algorithm, 2) XGBoost-CBR framework, and 3) Long-short term Memory with elastic weight consolidation. The structural causal model is adopted to provide a causal understanding of takeoff time delay from the observational data.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

1 Executive Summary

EU-funded ARTIMATION is a Research Innovation Action, whose goal is to provide a transparent and explainable Artificial Intelligence model through visualisation, data driven storytelling and immersive analytics. This project takes advantage of human perceptual capabilities to better understand AI algorithms with appropriated data visualization, as a support for Explainable AI (XAI), exploring in the ATM field the use of immersive analytics to display information.

After the development of two specific use cases (a tool supporting Conflict Detection and Resolution and a tool supporting the prediction of delay propagation) and while performing their test and validation, the project focused on a topic relevant to AI in general: the Lifelong Machine Learning.

For specific purpose of designing an AI support for the ATM Delay Prediction task, ARTIMATION opted for a Lifelong Machine Learning-based concept. While Isolated Machine Learning needs to be "retrained" every time it is fed with a new dataset and does not allow a continuous flux of new data to be integrated with previously acquired knowledge, Lifelong Machine Learning (LML) mimics the human learning process, using gained knowledge from previous tasks seamlessly in future learning, allowing a continuous process when data is added.

The Deliverable **D5.1 - Lifelong Machine Learning framework with integration of causality** elaborates on the Lifelong Machine Learning framework that is being developed within ARTIMATION to be applied in Air Traffic Management tasks.

Specifically, this deliverable argues the application of the Lifelong Machine Learning framework to **Delay Prediction** task in ATM, and the model used to introduce causality and explainability within the LML framework. Thus, with the integration of XAI, it will be possible to create a more transparent environment for the ATCOs, who can demand more information to the AI about a given decision.

In the context of AI support of Delay Prediction in ATM, the deliverable takes into consideration three algorithms for the LML framework: **Evolutionary-based Random Forest (ERF)**, **XGBoost-CBR Framework**, and **Long Short-Term Memory Networks (LSTM)**. On the other hand, to introduce causality, the model chosen is the **Structural Causal Model (SCM)**. These algorithms and models have been identified in the relevant literature as the most suited to process sequential independent data that has time dependency, as the analysed datasets are Enhanced Tactical Flow Management (ETFMS) flight data, obtained from EUROCONTROL from the year 2019 (May - October).

In **Chapter 2 - Lifelong Machine Learning Framework**, an overview of the Lifelong Machine Learning concept and proposed methods of Lifelong Machine Learning for ATM task - specifically for the Delay Prediction task, is given. The definition of LML is provided and how it differs from "simple" Machine Learning is explained, that is, the ability to continuously learning new tasks, instead of being disrupted by catastrophic forgetting, a common feature of AIs which makes the machine forget previously acquired knowledge, when fed with a new dataset. Thus, the chapter emphasises the importance of LML for the specific task of Delay Prediction, as the AI will have to incorporate new data daily from flights.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

The algorithms taken into consideration for the LML framework are three (i.e., ERF, XGBoost-CBR, and LSTM), and each of them is analysed, explained, and assessed in the context of the task that they'll have to manage, and how LML can be achieved with these algorithms. The assessment and the literature review suggest that **Long Short-Term Memory Networks (LSTM),** in combination with **Elastic Weight Consolidation (EWC)**, is the best option for the task that the AI will have to manage and the nature of the data that will analyse (flight data).

Finally, concluding the chapter, an experimental application of the LSTM developed in ARTIMATION, designed to aid in the Delay Prediction task, is given. With the theoretical demonstration in the last paragraph, the document illustrates the data preparation, the evaluation measure used to assess the model and the setting of the specific parameters for model implementation.

Chapter 3 - *Integration of Causality Model* elaborates on **how causality** is **integrated within** the **Machine Learning models** developed within ARTIMATION. For the scope of project, the model chosen to be integrated within the AI framework is the **Structural Causal Model (SCM)**, a model which adapts very well to the project's Delay Prediction tool, as it is the most suited to analyse observational data, in addition to be quite useful to understand the cause-effect relationship between independent and dependent variables. In the chapter, the SCM model and its three levels (*association, intervention and* counterfeit) are described, together with their specific characteristics and functions. To conclude the chapter, an overview of the SCM model developed for the specific task of Delay Prediction is given and how it has been adapted to best fit the needs of the Delay Prediction tool.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Table of Contents

EUROPEAN PARTNERSHIP

Co-funded by the European Union

List of Tables

List of Figures

EUROPEAN PARTNERSHIP

Co-funded by the European Union

2 Introduction

2.1 Purpose

This report presents the work conducted on a lifelong ML framework with integration of causality. Here, the lifelong ML concepts will be presented together within a strategic structure such that human component can be considered as an integral part of the AI and ML models. It is important that models can grow and retain knowledge in a long term, just as humans that pursuit knowledge over lifetime. Hence, different level of casual modelling will be integrated in the algorithm level. To develop the lifelong ML framework with integration of causality, two main tasks were considered: 1) Task 5.1: Lifelong ML framework, knowledge retainment and transfer mode; and 2) Task 5.2: Integration of causality within explainable. In task 5.1, the focus was to solve the phenomena in lifelong ML known as catastrophic forgetting and stability-plasticity dilemma. Again, a knowledge base is created for knowledge retainment, which is an essential part of lifelong ML architecture. The knowledge base is consisting of meta-knowledge, past information, and knowledge reasoner obtained from both domain experts and perspective users. The knowledge base is providing support for the transferring models, that is, from old model to new one, allowing the knowledge retainment. In task 5.2, a causal model e.g., Structural Causal Model (SCM), was examined for qualitative and quantitative measures and to develop an integration procedure to facilitate the explainability in the ML models. SCM provides three level of causal modelling, namely association, intervention, and counterfactual.

The present document follows up on the work reported in the ARTIMATION project-D3.2 report on the development plan based on ATM tasks with supporting AI[1] [17], in which a detailed development roadmap is defined for the proofs of concept of the lifelong ML framework with integration of causality system supporting ATM functions. The report detailed potential ATM tasks to be supported by AI algorithms at the tool elected, e.g., Delay prediction tool.

2.2 Scope of the Document

The document presents the development of a proof-of-concept lifelong ML framework with integration of a causality system. Here, the concept is developed and evaluated on the delay prediction task in the ATM domain. Three algorithms are investigated for the lifelong ML: 1) Random Forest, 2) XGBoost-CBR Framework, and 3) Long Short-Term Memory Networks (LSTM).

Generally, ML models are trained on the available data to gather knowledge on the characteristics of the data. With the addition of new data over time, the issue of forgetting previously trained knowledge is observed. This forgetting of previous knowledge is referred to as catastrophic forgetting. Mostly, the issue of catastrophic forgetting arises in lifelong ML as it tends to deal with continuously updated data which might contain different characteristics than the previously fed data to the model. To solve the catastrophic forgetting problem of random forest (RF), an evolutionary algorithm, e.g., genetic algorithm (GA), is integrated to keep the prior knowledge. During retraining of an existing RF model, mutation of the previously learned weights are retained using the GA. The XGBoost -CBR Framework consist of a combination of Extreme Gradient Boosting (XGBoost) and Case-based Reasoning (CBR)

[1] https://www.artimation.eu/wp-content/uploads/2022/03/D3.2-Report-on-the-Development-plan-based-on-ATM-tasks-with-supporting-AI_v00.02.00-1.pdf

EUROPEAN PARTNERSHIP

Co-funded by the European Union

approaches. In LSTM, the Elastic Weight Consolidation (EWC) method was considered where some parameters in the model are essential to the previous tasks and only change the unimportant parameters. To integrate the causality to the lifelong ML framework, a Structural Causal Model (SCM) was considered to infer the causality within the ML models, which consists of three levels of causal modelling: 1) association, 2) intervention, and 3) counterfactual. All the models were developed and evaluated considering the dataset of Enhanced Tactical Flow Management (ETFMS) flight data from 2019 (May - October).

2.3 Dataset

Dataset about Enhanced Tactical Flow Management (ETFMS) flight data obtained from the EUROCONTROL2 contains (EFD) messages for all flights during 2019 (May - October) within the Europe. The datasets are divided into basic information, the status of the flight and previous flight leg, ATFM regulations, weather, and calendar. Features extracted from the dataset are presented in the work of Ramon et al. [1]. Among the extracted features, 42 features are selected for this study and classified as categorical or numerical, and whether it changes (dynamic) or not (static) during the flight's progress, from the IFP to the ATOT. A detailed description of the processed dataset has given in D4.2 - Report on transparent AI models with explainability. Table 1 and Table 2 present the summary of the raw and processed dataset.

Table 1: Summary of raw data

Total number of instances	9509954
Total number of features	42

Table 2: A summary of the final dataset

Total number of instances	7613584
Total number Registrations	18214
Total number of Flights	609202
Minimum number of Flights per day	1 flight
Maximum number of Flights per day	152 flights
Average number of Flights per day	11.63 ≈ 12 flights
Minimum time of Delay	0 minutes
Maximum time of Delay	68 minutes 26 seconds
Average time of Delay	14 minutes 53 seconds

[2] https://www.eurocontrol.int/

Co-funded by the European Union

2.4 Structure of the document

This deliverable presents the plan of the ARTIMATION validations, and it is structured as follows:

- **Chapter 1** describes the purpose and scope of this document, introduces the dataset used in the work, and details the structure of the document.
- **Chapter 2** describes the lifelong machine learning concept, proposed methods of lifelong machine for ATM task specially for the delay prediction task.
- **Chapter 3** presents the details of structural causal model and causal model for the explanation of delay prediction.
- **Chapter 4** presents a brief conclusion of the tasks of WP5.
- **Chapter 6** provides the references.

2.5 List of Acronyms

Table 3: Acronym list

ETFMS	Enhanced Tactical Flow Management
IFP	Initial flight plan
ATOT	Actual take-off time
ETOT	Estimated take-off Time
ML	Machine Learning
SCM	Structural Causal Model
LSTM	Long Short-Term Memory Networks
GA	Genetic Algorithm
RF	Random Forest
XGBoost	Extreme Gradient Boosting
CBR	Case-based Reasoning
EWC	Elastic Weight Consolidation
LML	Lifelong machine learning
RNN	Recurrent Neural Network

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3 Lifelong Machine Learning Framework

3.1 General architecture of lifelong ML

Generally, in the machine learning (ML) paradigm, an ML model is trained using historical data (training dataset) for tasks such as prediction, classification, or clustering. If we want to update an existing ML model with a new dataset, we must repeat the whole ML training process. This paradigm is known as *isolated learning* in the ML domain since it does not consider any previously learned knowledge from the past model [2].

As humans, we often use previously learned knowledge to solve problems and make decisions, and we continually learn things that can be referred to as lifelong learning. When a mammal learns a solution to a task, the synapse's plasticity between neurons is reduced [3]. That means connections between neurons are strengthened and the knowledge learned is more likely to be remembered by the brain. The human brain can also leverage the knowledge acquired for one task to improve other tasks' learning.

The issue of *isolated learning* is that it does not retain, accumulate, and use past knowledge for future learning and decision-making as a human does. Lifelong machine learning (LML) is the learning paradigm that mimics the human learning process and capability, using gained knowledge from previous tasks seamlessly in future learning and, over time, learning more and more to become more knowledgeable. The objective of LML is to have the properties shown in Figure 1.

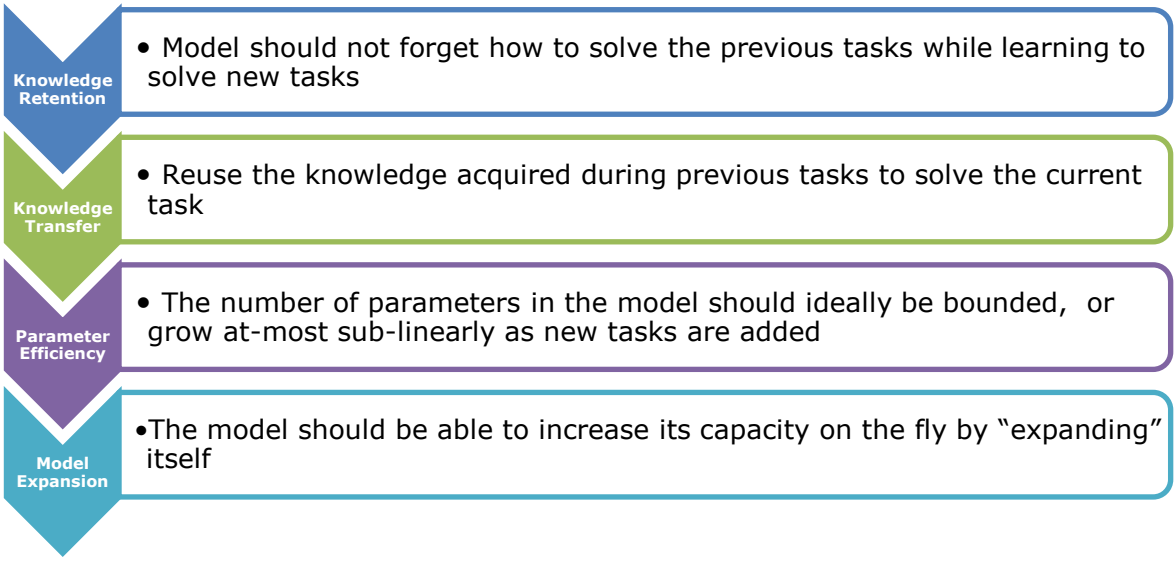

Figure 1: Properties of lifelong machine learning

Definition of Lifelong Machine Learning

LML is a continuous learning process. The general idea of LML is that a system has performed n tasks. When the system is doing $(n+1)$-th task, it uses the knowledge obtained from the n task to complete the $(n+1)$-th task. Here the system needs to accumulate the knowledge in a knowledge base and, when required, retain that knowledge from previous learning. We can define LML as follows:

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Consider at a given time a learning is performed n sequence of learning tasks, $T_1, T_2, \cdots \cdots T_n$. These tasks are called previous tasks and each of them have corresponding datasets $D_1, D_2, \cdots \cdots D_n$. Here tasks can be same or different type and can come from same or different domains. The learner utilizes and takes the leverage of previous knowledge from the knowledge base (KB) when the learner must solve a new or current task T_{n+1} with the new dataset D_{n+1}. The goal of LML is to optimize the performance on the new task T_{n+1}, but it can optimize on any task by treating the rest of the tasks as the previous tasks. KB maintains the knowledge learned and accumulated from learning the prior tasks. After completing learning T_{n+1}, KB is updated with the knowledge (e.g., intermediate and the results) gained from learning T_{n+1}. The updating can involve consistency checking, reasoning, and meta-mining of additional higher-level knowledge.

The key characteristics of a LML are:

- Should have continuous learning process
- Accumulation and maintenance of the knowledgebase
- Ability to use past knowledge in future learning
- Ability to discover new tasks
- Ability to learn while working on the task

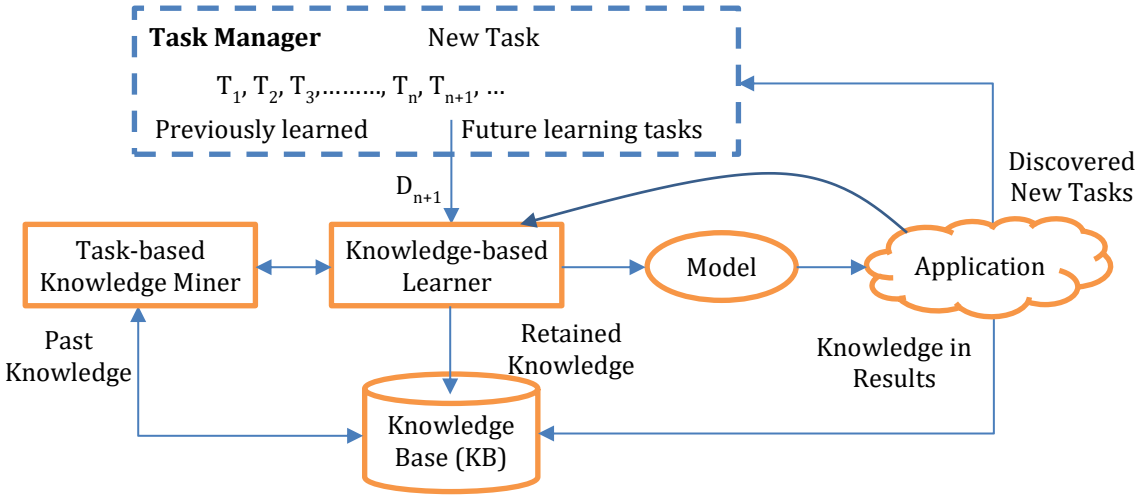

Figure 2: Lifelong Machine learning system adapted from [2]

Figure 2 shows a schematic architecture of a LML system that consist of six components, i.e., task manager, task-based knowledge miner, knowledge-based learner, Knowledge Base, model, and the application domain. The task manager is responsible for receiving and managing tasks in the system. Tasks can come sequential manner or can be discovered in the ATM application domain. The centre part of LML architecture is the Knowledge Base that stores previous knowledge learned by a machine learning algorithm. The learned knowledge can be the past information obtained from the learning, such as the outcome of a machine learning model, patterns identified from data etc. The knowledge base can also include meta-knowledge and meta knowledge miner that store suitable knowledge representation schemes. Here, the application is the ATM domain for which the model learns new knowledge and discover new tasks. The ATM application domain also sends feedback to the

EUROPEAN PARTNERSHIP
Co-funded by the European Union

knowledge-based learner for future improvements of the LML model. Two other essential parts, i.e., task-based knowledge miner and knowledge-based learner, mine knowledge and continually update and improve the LML model utilising the knowledge base.

Typically, the pipeline of an ML model development process includes data processing, training a model, tuning, and validation. After validation, if the model's performance is satisfactory, it is deployed on a target system for the intended use. During the model development process, historical data, i.e., the training dataset, is used to train a model, and the model's performance is tested with some unseen data, i.e., the test dataset. This approach is known as learning in isolation. In the future, if it needs to train the existing model again with a new dataset, the existing model will forget all the trained information. The issue of forgetting previously trained knowledge is referred to as catastrophic forgetting. In LML, researchers are working toward solving the catastrophic forgetting problem. In recent years, many different approaches have been proposed to address catastrophic forgetting and continuous learning. Machine learning researchers are trying to imitate various aspects of how mammals' brains behave during learning. However, this research is still in its infancy.

Most state-of-the-art methods of LML are developed for Deep Learning and neural networks. One of the most popular approaches to LML is **Elastic Weight Consolidation (EWC)** which is inspired by how the human brain addresses continual learning, as suggested by Kirkpatrick et al. [3]. The idea is to identify the weights and biases necessary for a specific task and constrain them not to change too much.

3.2 Algorithms

Three algorithms are adopted for the lifelong machine learning framework. These algorithms are random forest, XGBoost and LSTM. The rationale for choosing these methods was based on the literature study for the takeoff delay prediction reported in deliverable 3.1 Report on State of Art -AI support in ATM.

3.2.1 Random Forest

One of the popular ensemble algorithms of machine learning is the Random Forest (RF) algorithm which consists of a series of randomized decision trees [4]. The main idea of RF is the bootstrapping process, where each decision tree is trained using bootstrap data samples with replacement. During the bootstrapping process, a subset of data referred to as out-of-bag-data is selected for training. The out-of-bag data are used to find the generalization or out-of-bag errors. A generic architecture for a random forest classifier is shown in Figure 3.

The tree generation process works as follows: for the k^{th} tree, a random vector υ_k containing subset of data is generated, which is drawn from the same data distribution but independent of previous random vectors $\upsilon_1, \dots \dots, \upsilon_{k-1}$ (i.e., subsets of data sample). For a given training dataset, the tree grows using the random vectors and creates a predictor $h(\chi, \mathrm{X}_k, \upsilon_k)$, where is χ is the input data, X_k is the bootstrap sample, and υ_k consists of several independent random variables m between 1 and K. We can achieve different generalizations by varying the number of variables, and it is recommended to start the search from $m = \lfloor \log_2 K + 1 \rfloor$ or $m = \sqrt{K}$ number of variables [4, 5]. After generating many trees, the majority voting approach is often used to decide from the outputs of all these trees. One important characteristic of RF is that as the forest grows by adding more trees, it will converge to a limiting value that reduces the risk of overfitting and does not assume feature independence. Thus, RF

EUROPEAN PARTNERSHIP

Co-funded by the European Union

is implemented using bagging (i.e., majority voting), which is the process of bootstrapping the data plus using aggregation to plan.

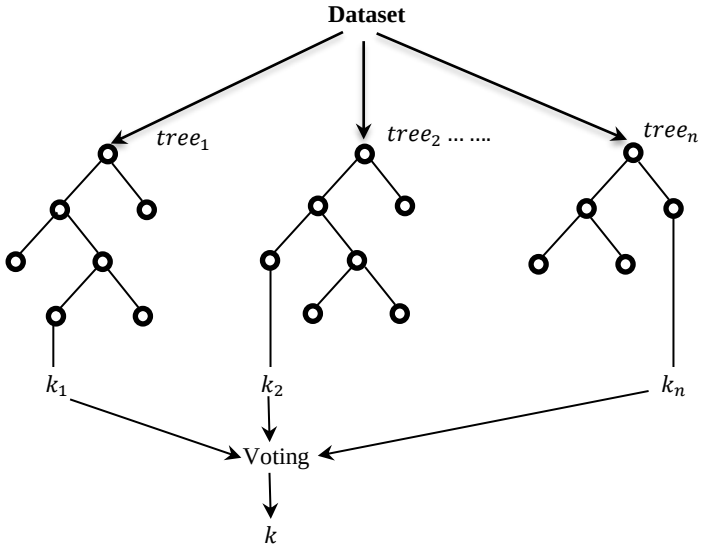

Figure 3: Generic structure of random forest

3.2.1.1 Lifelong learning for Random Forest

In a typical RF model, catastrophic forgetting appears when the model is retrained with a new dataset, where the distribution of target classes is changed. That means once new classes are given, the previously learned model no longer has knowledge of learned knowledge and fails to make the correct prediction. In literature, extensive works have been done for neural networks, and study has been on RF [6], but this work is limited to streaming data. In RF catastrophic forgetting manifests mainly in three ways:

- By excluding older classes from a meaningful contribution to the best split criterion.
- By disabling the conditional classification, which is responsible for decision making.
- By erasing priors (e.g., node weights) which leads to complete class forgetting at a given node.

Here we propose an evolutionary, i.e., GA algorithm-based approach which is a modification of [6] for the lifelong learning RF model that updates the weights and split criterion, capable of adding new knowledge and retaining previously learned concepts. The GA-RF lifelong algorithm approach is depicted in Figure 4.

The approach is divided into three main parts: preventing forgetting, keeping prior knowledge, and updating and retaining new knowledge. The proposed method adopted the rough class-conditional attribute estimation and propagation of these class-conditional attribute estimators to children of nodes as presented in [6] to prevent forgetting. The novelty of our method is keeping the priors (i.e., weights estimated by tree entropy) and mutation of the previously learned weights during retraining using the GA algorithm.

Co-funded by the European Union

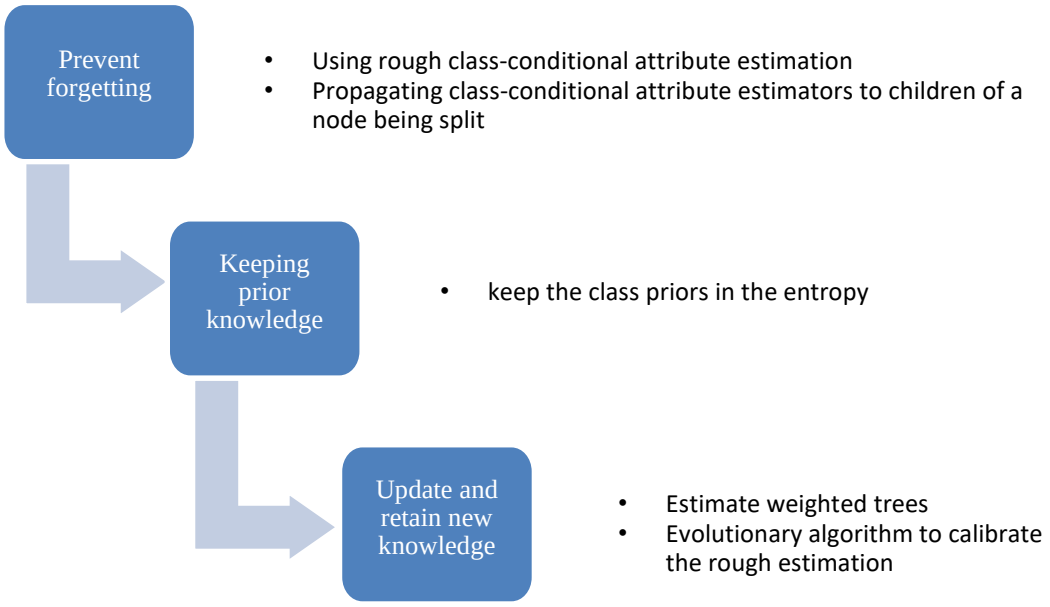

Figure 4: Lifelong learning approach of RF using GA

3.2.2 XGBoost-CBR Framework

Extreme Gradient Boosting (XGBoost) is a scalable ML technique for tree boosting used in regression and classification tasks [7]. The prediction is done in the form of an ensemble of weak prediction models. A weak prediction model can be referred to the randomized decision trees generated in a RF that leads to the final prediction as described in Section 2.2.1. XGBoost has been developed with two major improvements from the existing tree ensemble techniques: (i) optimisation of splitting the branches of a tree and (ii) scalability. The existing tree boosting mechanism supports a greedy algorithm that is computationally expensive as it iterates over all possible splits in the data. On the contrary, in XGBoost, an approximate algorithm is used to split the data reducing the requirements of high computation [8]. Moreover, it facilitates the scalability of the mechanism and can be implemented in distributed architecture. These two characteristics have made XGBoost preferable to numerous tasks across different application domains.

Case-based reasoning (CBR) is an artificial intelligence (AI) approach that utilizes previously gained experience to solve current problems. CBR is also known as lazy learning or instance-based learning since it does not reason a solution until it has to. In a CBR system, a new query instance is classified by analyzing similar samples while ignoring cases that are very different from the query. CBR methodology is analogous to human cognitive reasoning for solving problems based on previous experiences. Kolodner [9] described CBR as a reasoner that solves a new problem by remembering and using previous situations like the current one. The term ‘case’ represents an experience achieved from a previously solved problem. The term ‘based’ means cases or knowledge bases that are the source for reasoning. Finally, the term ‘reasoning’ implies the approach of problem-solving, i.e., to find a solution by drawing a conclusion using previously solved cases. Aamodt and Plaza [10] have described the CBR cycle, which contains four steps: Retrieve, Reuse, Revise and Retain, as shown in Figure 5.

The first step in the CBR cycle is the Retrieval step, where the system searches the case library for cases that resemble the new problem description. The system retrieves the most similar instances by calculating the similarity between the new case and stored cases. The top k number of similar cases

Co-funded by the European Union

are sent to the reuse step to develop a solution for the current case that needs to be solved. In CBR, the nearest neighbor (kNN) is one of the most common and used similarity measurement techniques, estimated by the Equation 1:

$$Similarity(T,S) = \sum_{i=1}^{n} w_i * f(T_i, S_i) \quad (1)$$

In Equation 1:

T is the target case,

S is the source case,

n is the number of features in each case,

i is the index of individual feature,

f is the similarity function

w is the weighting parameter of i-th feature.

The weights associated with each feature give them a range of importance. Determining the weight for a feature value is a challenge, which may require domain knowledge. However, weight can also be learned by an adaptive learning process, i.e., learning or optimizing weights from the case library as an information source.

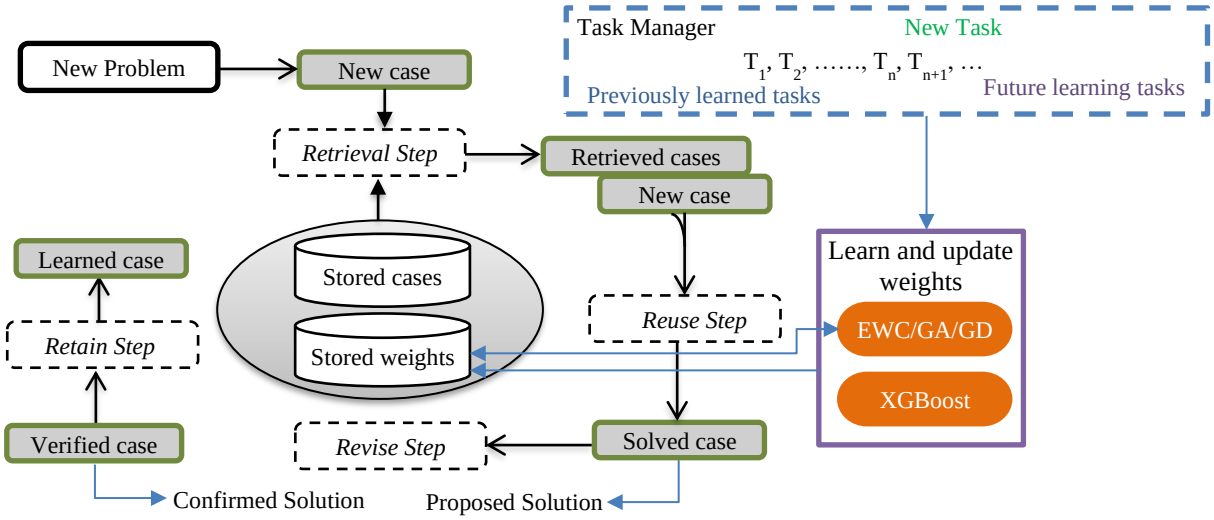

Figure 5: XGBoost-CBR framework for Lifelong learning and algorithmic transparency

Reuse is the next step that adapts solutions from the retrieved cases to acquire a solution for the new case. The solution can be proposed easily by returning the retrieved and unchanged solution. However, usually, the best matching case does not always provide a complete solution for a new problem case, and adaptation of the solution is often required to use it for a new case. This adaptation or change of the best matching cases is usually complex and requires domain knowledge. Adaptation can be achieved by calculating the difference between the retrieved and current cases. It is possible to apply

EUROPEAN PARTNERSHIP
Co-funded by the European Union

other algorithms and/or rules to combine two cases from the retrieved list of cases to obtain a solution. Also, an expert in the domain can suggest or determine if it is a reasonable solution for the current case, and if required, an expert can also modify the solution before approval. In the revise step, the proposed solution from the reuse step is evaluated to check success or failure. If success is received from the revision process, then the new solution and case are considered for future use [11]. The retain step is the learning process of the CBR cycle, where the verified solution with the new case is stored in the case library as new knowledge. Sometimes the case base is updated by the new case or modifying existing cases. In addition, if the solution is unreliable, additional information, e.g., changes made in the retrieved case, should be added so that it can be used carefully in future.

The XGBoost-CBR framework integrates lifelong learning and transparency together that can be summaries as follows:

- **Algorithmic Transparency**
    - Feature importance obtained from XGBoost model
    - Two level of similarity measures
        - Local similarity
        - Global similarity
    - Instance based comparison
- **Lifelong Learning**
    - Knowledge retainment
    - Knowledge transfer
    - Knowledge revision

CBR is an interpretable and transparent method since it relies on the similarity between cases. An explanation of a decision can easily be given by looking at the most similar instances that match the current problem. As mentioned earlier, one challenge in similarity calculation is the weights associated with each feature. In the proposed framework, weights are obtained from the XGBoost model, representing each feature's importance. Thus, like other explainable models such as LIME and SHAP, it is possible to show or even select a subset of essential features from the feature list. The Euclidian distance function is used in the proposed framework as the global similarity measure. Yet, based on domain knowledge, it will be possible to adapt different local similarity measure functions for each feature. This provides better transparency to the proposed framework.

The capability of retaining and revisioning a decision gives CBR further ability of knowledge retainment. In addition, knowledge transfer is achieved by adopting weights from the XGBoost model, as shown in Figure 5. Moreover, it is possible to use different methods such as EWC, genetic algorithm (GA), or gradient descent to update old weights with new tasks or datasets without changing too much but maintaining good prediction accuracy.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3.2.2.1 Experimental Setting

The experiment was run on sample data extracted from the dataset described in section 2.3 for the take-off time delay prediction. The summary of the dataset used in the experiment is presented in Table 4.

Table 4: A summary of the experiment dataset

Train Set	May – August ≈ 5903743 instances (approx. 78% of the clean dataset)
Test Set	First 5 days of September & October ≈ 158147 instances

The take-off time or delay prediction need to build a regression model that uses data $x \in X$, where instances are mapped by features $f_i \in F, f_i = 1, 2, \ldots\ldots, m$, training data is a subset of dataset $X_{train} \subset X$ and include prediction delays $y \in Y$ in minutes, and test dataset is a smaller subset of data entire dataset $X_{test} \subset X$, where $X_{train(i)} \neq X_{test(j)}$. The goal of the regression model $r(x_i)$ is to learn predictions $\hat{y}_i$ using the training dataset.

The performance of the prediction is measured using the metric mean absolute error (MAR), which is also the primary performance measure in [1]. MAE indicates the difference between two observations, i.e., the actual observation and a prediction from a model, as shown in Equation 2.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (2)$$

Where, n is the total observations in test dataset, y_i is the actual observation and $\hat{y}_i$ is the prediction from the model.

The XGBoost model was first trained with different parameter settings based on previous experience of the engineer and related works. In total, 288 combinations of hyperparameters, i.e., learning_rate, max_depth, min_child_weight, subsampl, colsample_bytree, and n_estimators are investigated. Based on the model's best performance, the final hyper parameters are chosen, which are listed in Table 5.

Table 5: Final hyperparameters list of the XGBoost model

learning_rate	0.1
max_depth	7
min_child_weight	1
colsample_bytree	0.5
n_estimators	500

As mentioned in Table 4, 158,147 instances as cases have been used to test the performance of the CBR model, where features weights from the XGBoost model are used to aggregate the similarity scores with weighted Euclidian distance. The k-NN algorithm is used for retrieving similar cases with the value of K = 3 closes neighbors. The results show an MAE of 6.56, which was found to be better than using the XGBoost (MAE = 10.11) alone.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3.2.3 Long Short-Term Memory Networks (LSTM)

LSTM networks are a special kind of Recurrent Neural Network (RNN) proposed by [12], which is designed to support the long-term dependency problem by introducing memory cell and two gates into the network. Many researchers have adopted the concept of LSTM and proposed slightly different versions of LSTM, such as [13, 14].

3.2.3.1 General Architecture of LSTM

LSTM is a chain of repeating modules of a neural network, the same as RNN, where the hidden layers give their output and feedback to themselves throughout time. Looking inside the LSTM network, as shown in Figure 6, we can find that it consists of three main elements: (i) the repeating module in LSTM, which has four neural networks layers consisting of three sigmoid layers and one tanh layer, (ii) a memory cell (cell state) which is like a data bus that contains all the network information and allows it to flow through time and (iii) three gates that control the information in the memory cell.

- f_t is forget gate that controls how much of the old memory cell content we retain.
- i_t is the input gate that controls how much we take new data into account.
- o_t is the output gate that controls what data to pass as the output (hidden state).
- The equations that represent the LSTM are calculated as follows:

$$
\begin{aligned}
i_t &= \sigma(x_t w_{xi} + h_{t-1} w_{hi} + b_i) \\
f_t &= \sigma\left(x_t w_{xf} + h_{t-1} w_{hf} + b_f\right) \\
o_t &= \sigma(x_t w_{xo} + h_{t-1} w_{ho} + b_o) \\
\tilde{c}_t &= tanh(x_t w_{xc} + h_{t-1} w_{hc} + b_c) \\
c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t \\
h_t &= o_t * \tanh(c_t)
\end{aligned}
\qquad (3)
$$

Where, $b_i, b_f, b_o \; and \; b_c$ are biases, $w_{xi}, w_{hi}, w_{xf}, w_{hf}, w_{xo}, w_{ho}, w_{xc}, and \; w_{hc}$ are weight parameters, $\tilde{c}_t$ is a memory cell. $x_t, c_t \; and \; h_t$ are input, hidden state (output) and the memory cell of the current time step respectively, $c_{t-1} \; and \; h_{t-1}$ are the hidden state (output) and the memory cell of the previous time step.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

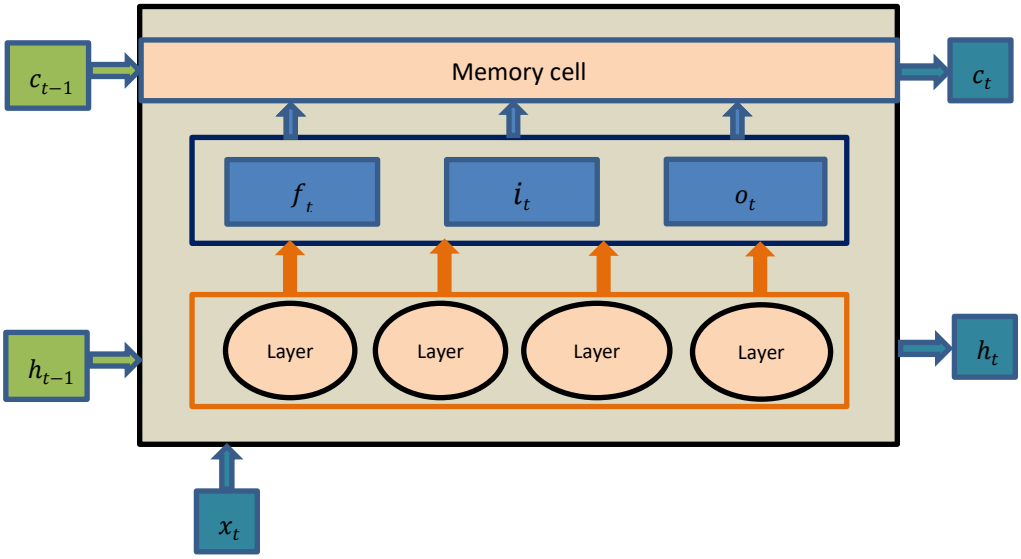

Figure 6: LSTM Architecture.

3.2.3.2 LSTM for Lifelong Learning

The advantages of LSTM for remembering information for long periods of time and solving the limitations of RNN (vanishing and exploding gradient problems) opened the door for researchers to take advantage of LSTM. In the last years, the LSTM has been applied to several issues and has proven its success and efficiency in real-world applications such as time series-based problems and text data, including language modelling and speech recognition.

In this project, to deal with a lifelong learning scenario, we utilized the EWC method, where some parameters in the model are essential to the previous tasks and only change the unimportant parameters. Given task A already been learned by the model, the loss function when learning a new task B is:

$$L(\theta) = L_B(\theta) + \sum_i \frac{\lambda}{2} F_i \left(\theta_i - \theta^*_{A,i}\right)^2 \quad (4)$$

Where, L_B is the loss function to be optimised for task B, λ is the hyperparameter controlling the tradeoff between accuracy on the old and new task, F_i is the Fisher Information Matrix to find how important this parameter, θ_i are the parameters to be learning and $\theta^*_{A,i}$ are the learned parameters from previous task A.

With LSTM, which deals very well with sequential data such as time series and adopting an Elastic Weight Consolidation (EWC) [3], the most popular lifelong learning strategy, the model in this work offers a unique combination of efficient problem solving and continual learning.

3.2.3.3 Lifelong for the Delay Prediction Task in ATM

Prediction problem scenario: Give the model several (features) of air traffic data at the prior time step as input, and the LSTM model will predict the delay at the current time (t).

3.2.3.3.1 Experimental Setting

In this part of the study, we aim to illustrate the data preparation, the evaluation measure used to evaluate the model and the setting of the specific parameters for model implementation. We

Co-funded by the European Union

conducted experiments on the real dataset of air traffic provided by [1] to validate the model's effectiveness.

For the data pre-processing, the LSTMs are sensitive to the scale of the input data, so we normalize the dataset to the range of (0,1) using a min-max scaler. Moreover, the LSTM network expects the input data (X) to be provided in a specific form; therefore, we reshape the dataset from (sample, features) to (sample, time step, features). For modelling, we implement many-to-one architecture of LSTM to produces one output (y(t)) value after receiving multiple input values (X(t), X(t+1), ...). The model is implemented using TensorFlow 2.6 with GPU, and all the experiments are conducted on Windows 10 with Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz 2.59 GHz and 16 GB of memory. Adam optimizer [15] is used with one-layer LSTM models.

3.2.3.3.2 Data preparation

The dataset is usually divided into two parts, a training set and a test set, a technique used to learn the model and validate the effectiveness. But in this experiment, to evaluate the models in the lifelong learning setting, we prepared the dataset into two tasks, A and B; for task A, data from 1-May-2019 to 31-June-2019 are used, and for task B, data from 1-July-2019 to 31-July-2019 are used.

3.2.3.3.3 Evaluation Metric

To measure the impact of catastrophic forgetting, first, we train the model on task A, then train task B on the trained model based on EWC; finally, the model is performed on task B and task A; the goal is to perform well on both tasks. In this study, we used prediction accuracy metrics, namely Mean Absolute Error (MAE), to measure the prediction as below.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (5)$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value and n is the number of observation/rows.

3.2.3.3.4 Parameters setting

The parameters tuning for LSTM and EWC used in this study are listed in Table 6. To select valid values of the parameters, we run a parameter sweep to find which values gave us the lowest MAE on the dataset.

Table 6: LSTM & EWC Parameters

Learning rate	0.001
Number of epochs	50
Batch size	32
Number of neurons	10
Time steps	5
λ	0.01

EUROPEAN PARTNERSHIP

Co-funded by the European Union

4 Integration of Causality Model

4.1 General overview of Causality Model

Structural Causal Model (SCM) was considered to infer and integrate causality within the ML models. The SCM involves a probability model with additional information not contained in traditional machine learning models. Like the probability theory, causal reasoning is the process of obtaining conclusions from a causal model about the outcomes of random experiments. The additional information of SCM allows analysing the effect of the intervention, i.e., changes in the data distribution. ML has purely empirical implications on observational data; on the other hand, SCM provides the necessary condition for including data under interventions. Figure 7 shows the structural difference between traditional machine learning and causal inference problems.

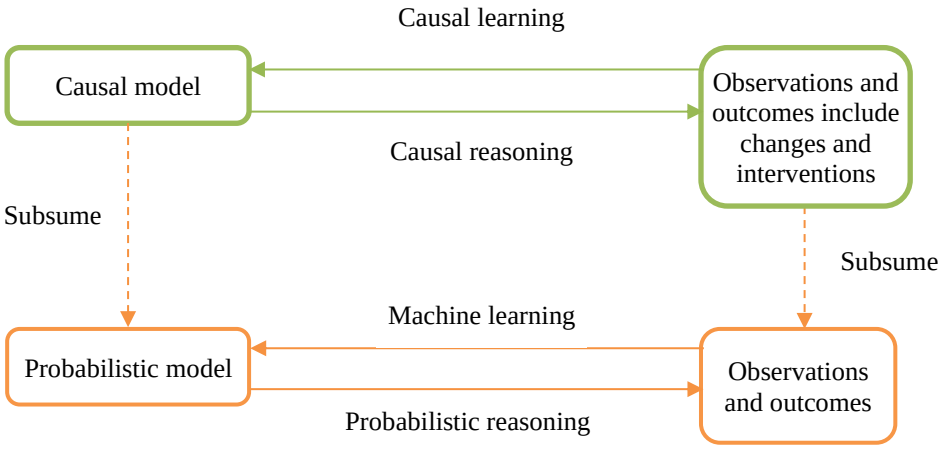

Figure 7: Structural difference between machine learning and Causal learning adapted from [16]

In general, SCMs are usually transparent. The SCM consists of three components, which Judea Pearl in his book [17] referred to as "The Ladder of Causation", (Figure 8). These three components are *association*, *intervention*, and *counterfactuals*. Machine learning models capture the first rung of the ladder (i.e., association) from the data or observations. Given observation $(x_1, y_1), \dots\dots, (x_n, y_n)$ where $x_i \in X$ are inputs variables and $y_i \in Y$ are outputs and each $(x_i, y_i) = 1, \dots\dots, n$ has been generated independently by the same unknown random experiment. The goal in ML is to find a function $f: X \to Y$ that can approximate the solution of f with minimum errors.

- Three level of causal modelling
    - Association: observing one aspect and determining the probability of another aspect
    - Intervention: getting the specific causal relationship between events
    - Counterfactual: gain the top level of causal relationship that concerns the alternate versions of the past events

EUROPEAN PARTNERSHIP
Co-funded by the European Union

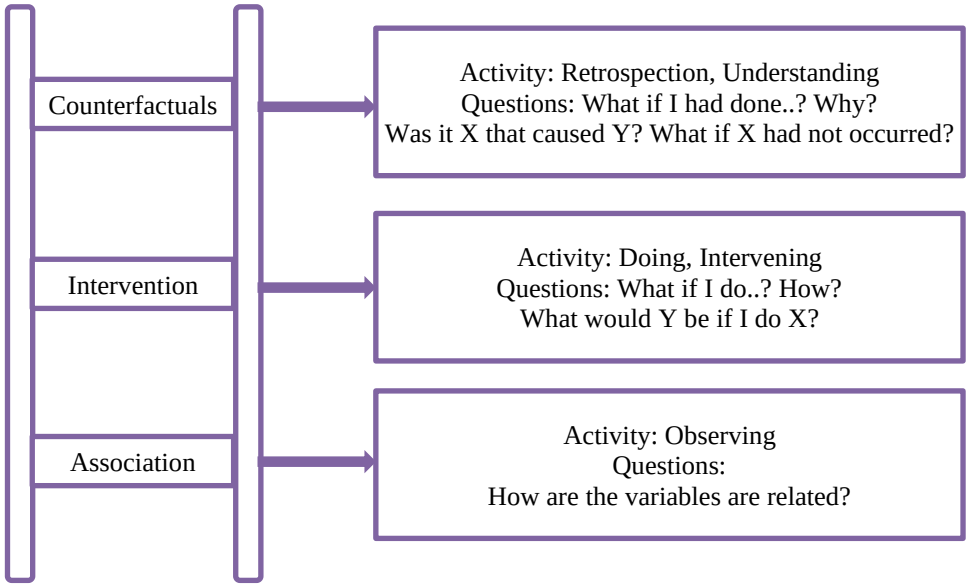

Figure 8: Causality Ladder

The second rung of the ladder of causal is the intervention that involves the association between input and output variables and can help identify how the decision can vary under perturbation. The top rung of the ladder of causation is the counterfactuals. Counterfactuals allow making possible modifications of an SCM by changing all its noise distributions. Thus, we can hypothesize what would have happened if variable X had been treated differently.

Counterfactual questions provide the answer to the questions, for instance, “was its X that caused Y?” and “What if X had not occurred.”

4.2 Structural Causal Model (SCM)

4.2.1 Basic Definition

Intuitively, SCM is a kind of Directed Acyclic Graph (DAG) that represents the flow of information. Mathematically, a SCM consist of a set of Exogenous and Endogenous variables connected through a set of Functions. In a system, Exogenous variables are the inputs and Endogenous variables are the outputs whose values are found from the set of Functions. In the DAG, the parent nodes, or the nodes without any incoming edge, are the graphical representation of the exogenous variables. On the other hand, the child nodes or the nodes that has incoming edges are the endogenous variables. And the edges represent the functions determining the outcome of the exogenous variables. A simple SCM is illustrated in Figure 9 to generate a clear idea on the functionality of SCM.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

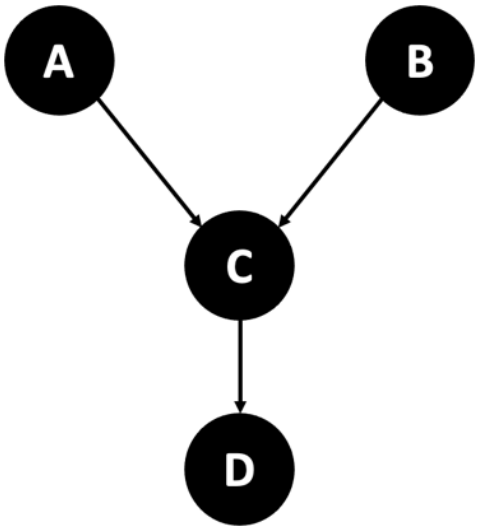

Figure 9: Graphical model of SCM.

From the illustration in Figure 9, the details of underlying SCM are easily understandable –

- In the subgraph with nodes A, B and C, the exogenous variables are A and B as they have no incoming edge. On the other hand, C is the endogenous variable.
    - C has two direct causes A and B which is illustrated with the edges. In other words, the value of C depends on a function of A and B.
- Similarly in the subgraph containing C and D, they are exogenous and endogenous variables respectively.
    - The value of D depends on a function of C.

In addition to the above observations, each edge of a SCM corresponds to the following causal assumptions –

- If a variable Y is the child of a variable X, it can be said that "X is the direct cause of Y" or "Y is caused by X".
- If a variable Y is the descendant (child of a child and so on) of a variable X, it can be said that "X is the potential cause of Y" or "Y is potentially caused by X".

Based on these assumptions and the graphical model, the following decomposition can be written for the whole system without needing to explicitly know about the functions underlying each variable. The product decomposition for the SCM illustrated in Figure 9 –

$$P(A, B, C, D) = P(A)\, P(B)\, P(C|A, B)\, P(D|C)$$

The product decomposition with the conditional probabilities can further analysed to assess the effects of changing the values of different variables on the target variable.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

4.2.2 SCM for Delay Prediction

Considering the basics of causal modelling, a SCM is developed for Delay Prediction where the target variable is “DELAY_TAKEOFF”. The SCM is illustrated in Figure 10. The SCM is quite complex as it accommodates 42 different variables representing various parameters collected from flight plan, Enhanced Tactical Flow Management System (ETFMS) messages etc. To present the idea in more understandable way, some parts of the model are enlarged.

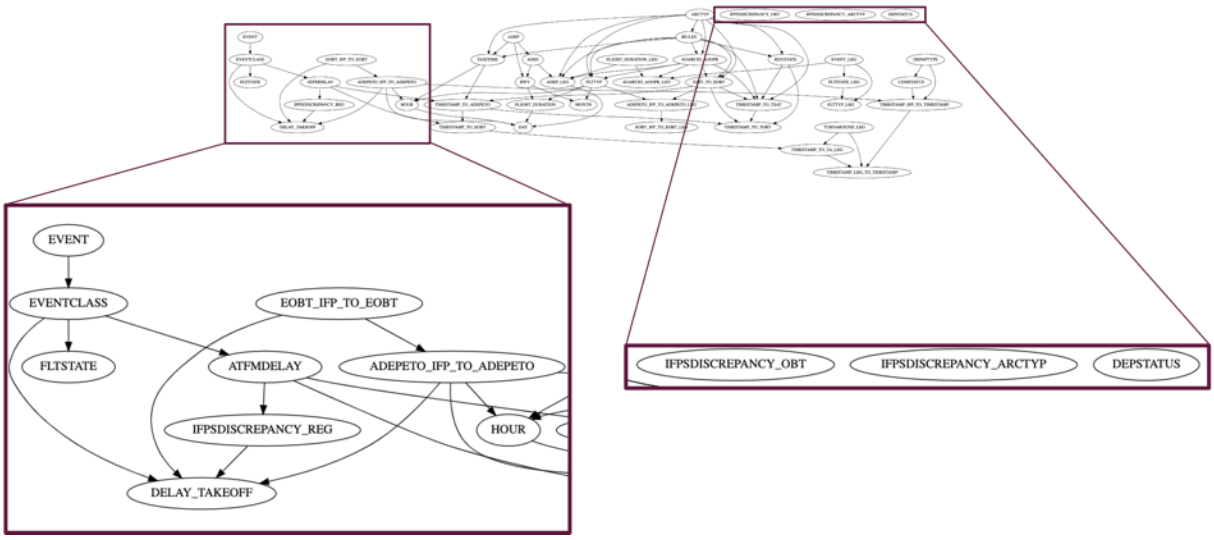

Figure 10: SCM for Delay Prediction.

Two important aspects are emphasized in the enlarged part of the SCM in Figure 10. The enlarged part on the left shows the conditional dependencies among the important features that contribute to the “DELAY_TAKEOFF” based on the available data. Again, on the right, three features are pointed out which do not have any edge to other features, which expresses that they might not contribute substantially to the take-off delay. The findings from the SCM can either be used to generate more robust prediction model or utilized to generate more human understandable explanations.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

5 Conclusions

The main objectives of this report were to investigate the integration of a lifelong Machine Learning framework with causality and draw attention to this integration for supporting ATM functions. Experimental results in a real dataset (ETFMS) for three algorithms, Random Forest, XGBoost -CBR and LSTM, to investigate lifelong ML showed different MAE values for delay prediction, 10.11, 6.56 and 7.23, respectively. These outcomes showed that XGboost-CBR performs better i.e., MAE: 6.56, among the other algorithms, i.e., RF and LSTM. In this report, the fundamental idea behind Structural Causal Model (SCM) is to adopt three levels of causal modelling within the ML models. Here, the findings from the SCM can either be used to generate a more robust prediction model or utilized to generate more human-understandable explanations. From the algorithm development point of view, EWC is also found to be applicable to the XGBoost-CBR framework. The XGBoost-CBR framework also showed better results in predicting takeoff delay time. Also, by definition, CBR is a transparent and lifelong learning method; hence possible research interest involves investigating CBR for XAI with other methods such as LIME and SHAP. Further, instead of depending on feature importance provided by LIME or SHAP methods, it can be possible to develop an XAI system using CBR. However, it is necessary to include domain knowledge and experts while developing a CBR system. Though the proposed methods are evaluated for delay prediction tasks, we believe these methods can also be used for other prediction and classification tasks in the ATM domain, such as delay propagation prediction.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

6 References

1. R. Dalmau, F. Ballerini, H. Naessens, S. Belkoura, S. Wangnick, An explainable machine learning approach to improve take-off time predictions, Journal of Air Transport Management, Volume 95, 2021, 102090, ISSN 0969-6997Z. Chen and B. Liu, Lifelong Machine Learning, Second Edition. 2018.
2. J. Kirkpatrick et al., 'Overcoming catastrophic forgetting in neural networks', Proc. Natl. Acad. Sci., vol. 114, no. 13, p. 3521, Mar. 2017, doi: 10.1073/pnas.1611835114.
3. Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32.
4. Breiman, L. (1996). Bagging Predictors. Machine Learning, 24, 123-140.
5. Korycki, Ł., & Krawczyk, B. (2021, september). Streaming Decision Trees for Lifelong Learning. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 502-518). Springer, Cham.
6. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794. 2016.
7. Li, Ping, Qiang Wu, and Christopher Burges. "Mcrank: Learning to rank using multiple classification and gradient boosting." In Advances in Neural Information Processing Systems, pp. 897-904. 20. 2007.
8. Kolodner, J. (1992). An introduction to case-based reasoning. Artificial Intelligence Review, 6, 3-34.
9. Aamodt, A., & Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun., 7, 39-59.
10. Lopez De Mantaras, R., Mcsherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. L., Cox, M. T., Forbus, K., Keane, M., Aamodt, A. & Watson, I. (2005) Retrieval, reuse, revision, and retention in case-based reasoning. The Knowledge Engineering Review, 20(3), 215-240.
11. Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997; 9 (8): 1735–1780.
12. Yao, Kaisheng, Trevor Cohn, Katerina Vylomova, Kevin Duh, and Chris Dyer. "Depth-gated recurrent neural networks." arXiv preprint arXiv:1508.03790 9 (2015): 98.
13. F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, 2000, pp. 189-194 vol.3, doi: 10.1109/IJCNN.2000.861302.
14. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
15. Peters, J., Janzing, D., & Schölkopf, B. (2017). "Elements of causal inference: foundations and learning algorithms". (p. 288). The MIT Press.
16. Pearl, J., & Mackenzie, D. (2018). "The book of why: the new science of cause and effect. Basic books.
17. A. Degas et M. R. Islam, « D3.2 Report on transparent AI models with explainability » https://www.artimation.eu/wp-content/uploads/2022/03/D3.2-Report-on-the-Development-plan-based-on-ATM-tasks-with-supporting-AI_v00.02.00-1.pdf.

Co-funded by the European Union