UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Deep continual multimodal multitask models for out-of-hospital emergency medical call incidents triage support in the presence of dataset shifts

February 2024

Author: Pablo Ferri Borredà

Directors: Prof. Juan Miguel García Gómez
Dr. Carlos Sáez Silvestre

All models are wrong, but some are useful.

— George E. P. Box

Acknowledgements

La realización de la presente tesis doctoral ha supuesto un claro punto de inflexión por lo que respecta a mi formación académica como investigador. No obstante, esta tesis ha traído consiguido también cambios a nivel de crecimiento personal, especialmente durante los últimos años de la misma. Todo ello en su conjunto ha supuesto un proceso de aprendizaje que, pese que ha resultado arduo en alguna de sus fases, considero de gran valor y provecho, yendo más allá de la dimensión técnica.

Es pertinente, por tanto, expresar mi sincero agradecimiento a mis directores de tesis, el Dr. Juan Miguel García Gómez y el Dr. Carlos Sáez Silvestre. Vosotros habéis sido los que me brindasteis la oportunidad de iniciar mi formación como investigador en el BSDLab hace ya muchos años. A lo largo de este tiempo, habéis contado siempre conmigo—demostrando una gran confianza en mi trabajo por la que estoy enormemente agradecido—aportando en todo momento la rigurosidad y honestidad necesarias para abordar de forma apropiada aquellas cuestiones que hacen posible el avance en ciencia. Si, a día de hoy, se me puede considerar un investigador con potencial para generar nuevo conocimiento, es gracias a vosotros.

Por otro lado, quisiera agradecer a Antonio Félix de Castro su gran involucración en el presente trabajo, sin el cual no hubiese sido posible. Pudiste traducir una voluntad de mejora en acciones—circunstancia que no es habitual—las cuales han y están derivando en mejoras en los procesos de gestión de emergencias extrahospitalarias de la Comunidad Valenciana. Y lo mismo puede decirse de Purificación Sánchez Cuesta, Antonio Gil, María y todos aquellos profesionales sanitarios de la Conselleria de Sanitat que han colaborado para hacer posible este trabajo.

Ringrazio anche Vincenzo Lomonaco e Lucia Passaro per i loro contributi nel campo dell'apprendimento continuo profondo e dell'elaborazione del linguaggio naturale. Inoltre, desidero ringraziare Andrea Cossu, Rudy Semola, Jacopo Massa, Antonio Boffa e tutti i membri del Dipartimento di Informatica dell'Università di Pisa che mi hanno fatto sentire a casa, nonostante fossi lontano dalla mia patria.

No podría pasar por esta sección de agradecimientos sin mencionar al Dr. Vicent Blanes, Marta Durà, Ángel Sánchez, el Dr. Javier Juan, el Dr. José Enrique Romero, el Dr. Elies Fuster, la Dra. Sabina Asensio, el Dr. José Vicente Manjón, la Dra. María del Mar Álvarez, Francisco Javier Gil-Terrón, Javier Salvador, Víctor Montosa, Carles López, Kevin García, María Gómez, Marina Ruiz y Sergio Morell. Empezaron siendo compañeros de trabajo, pero terminaron por convertirse en amigos míos, siempre dispuestos a ayudar cuando fuera necesario, muchas veces incluso más allá del ámbito laboral.

Emilio, Àngela, Patricia, Arantxa, muchas gracias por haberme escuchado a lo largo de estos años de transcurso de esta tesis. En momentos de mayor ofuscación en los que necesitaba hablar con alguien—aunque quizá no quisiera reconocerlo—siempre estuvisteis allí, ofreciendo vuestro apoyo. Asimismo, debo agradeceros también vuestra contribución en muchas de las ideas que dieron lugar a los desarrollos presentados aquí. A veces, compartir un problema con un amigo es la mejor forma de empezar a solucionarlo.

Finalment, m'agradaria agrair i dedicar aquesta tesi especialment als meus pares, Amparo i José. Des de que tinc memòria s'heu preocupat perquè no em faltara res, perquè tinguera al meu abast les opcions que, per circumstàncies de la vida, no vau poder tindre. A més, heu aconseguit transmetre'm la curiositat que es necessita per aprendre, per mera voluntat del saber, més enllà de qualsevol justificació utilitarista, i la creença de què poques dificultats resisteixen a una combinació de constància i esforç. Aquesta tesi no és sinó una conseqüència de l'aplicació d'eixos valors inculcats.

Abstract

Triage for out-of-hospital emergency incidents represents a tough challenge, primarily due to time constraints—requiring rapid priority assessment within a narrow time frame—and uncertainty—making decisions with limited available information. Furthermore, errors in this process can have severe consequences for patients, potentially leading to death. Therefore, any novel protocol, tool, or strategy that has been demonstrated to enhance these processes can offer substantial value in terms of patient care and overall management of out-of-hospital emergency medical incidents.

The fundamental hypothesis upon which this thesis is based is that Machine Learning, specifically Deep Learning, can significantly improve these processes by providing estimations of the severity of out-of-hospital emergency medical incidents, taking into account the information available to the dispatcher at the moment of incident prioritization during the emergency call. By analyzing millions of data derived from emergency calls from the Valencian Region (Spain) spanning from 2009 to 2019, we posited that Machine Learning models could extract patterns that may confer predictive capability to this task.

Hence, this thesis delves into designing and developing various Machine Learning models, specifically Deep Multitask Learning models that leverage multimodal out-of-hospital emergency medical data. Our primary objective was to predict three labels indicative of incident severity, thereby influencing its prioritization. These labels encompassed whether the incident posed a life-threatening situation, the admissible response delay (ranging from non-delayable to minutes, hours, or days), and whether it fell under the jurisdiction of the emergency system or primary care. Using data available from 2009 to 2012, the results obtained were promising. We observed substantial improvements in macro F1-scores, with gains of 12.5% for life-threatening classification, 17.5% for response delay, and 5.1% for jurisdiction classification, compared to the in-house triage protocol of the Valencian Region.

However, it is essential to note that systems, dispatch protocols, and operational practices naturally evolve over time. Models that exhibited excellent performance with the initial dataset from 2009 to 2012 did not demonstrate the same

efficacy when evaluated on data spanning from 2014 to 2019 (data from 2013 were not available). This later dataset had undergone significant modifications compared to the earlier one. These modifications led to dataset shifts, resulting in variations in probability distributions, which we have meticulously characterized and investigated in this thesis, focusing on their impact on model performance.

Continuing our research, we aimed to provide sustainable model performance over time or, at the very least, to mitigate the adverse effects of the inevitable distribution variations as effectively as possible. To address this challenge, we placed our focus on Deep Continual Learning. By incorporating the Continual Learning paradigm into our designs and developments, we could substantially mitigate the adverse performance effects and better understand how to manage model deployment over time in an emergency medical dispatch center. The results of our research indicate that when considering Deep Continual Learning, while it may not entirely eliminate performance fluctuations over time, it effectively maintains them within a manageable range. In particular, with respect to the F1-score, when distributional variations fall within the light to moderate range, the performance remains stable, not varying by more than 2.5%, as observed in our out-of-hospital medical incident data. Therefore, under these conditions, our models' performance is operationally acceptable.

Furthermore, our thesis demonstrates the feasibility of building auxiliary tools that enable dispatchers to interact with these complex deep models. Consequently, without disrupting professionals' workflow, it becomes possible to provide feedback through probability predictions for each severity label class and take appropriate actions based on these predictions.

Finally, the outcomes of this thesis hold direct implications for the management of out-of-hospital emergency medical incidents in the Valencian Region. The final model resulting from our research is slated for integration into the emergency medical dispatch centers of the Valencian Region. This model will utilize data provided by dispatchers to automatically compute severity predictions, which will then be compared with those generated by the in-house triage protocol. Any disparities between these predictions will trigger the referral of the incident to a physician coordinator, who will oversee its handling. Therefore, it is evident that our thesis, in addition to making significant contributions to the field of Biomedical Machine Learning Research, also carries substantial implications for enhancing the management of out-of-hospital emergencies in the context of the Valencian Region.

Resumen

El triaje de los incidentes de urgencias y emergencias extrahospitalarias representa un reto difícil, principalmente debido a las limitaciones temporales—que exigen una evaluación rápida de las prioridades en un estrecho margen de tiempo—y a la incertidumbre—tomar decisiones con la información disponible. Además, los errores en este proceso pueden tener graves consecuencias para los pacientes, con el consiguiente riesgo de muerte. Por lo tanto, cualquier protocolo, herramienta o estrategia novedosa que haya demostrado mejorar estos procesos puede ofrecer un valor sustancial en términos de atención al paciente y gestión global de los incidentes médicos de urgencias y emergencias extrahospitalarias.

La hipótesis fundamental en la que se basa esta tesis es que el Aprendizaje Automático, concretamente el Aprendizaje Profundo, puede mejorar significativamente estos procesos proporcionando estimaciones de la gravedad de los incidentes médicos de urgencia y emergencia extrahospitalaria, teniendo en cuenta la información de la que dispone el operador en el momento del triaje del incidente durante la llamada de emergencia. Mediante el análisis de millones de datos derivados de llamadas de emergencia de la Comunitat Valenciana (España) que abarcan desde 2009 hasta 2019, planteamos que los modelos de Machine Learning podrían extraer patrones que pueden conferir capacidad predictiva a esta tarea.

Por ello, esta tesis profundiza en el diseño y desarrollo de varios modelos de Aprendizaje Automático, concretamente modelos de Aprendizaje Profundo Multitarea que aprovechan los datos multimodales asociados a eventos de urgencias y emergencias extrahospitalarias. Nuestro objetivo principal era predecir tres etiquetas indicativas de la gravedad del incidente, influyendo así en su priorización. Estas etiquetas englobaban si el incidente suponía una situación de riesgo vital, la demora admisible de la respuesta (desde no demorable hasta minutos, horas o días) y si era competencia del sistema de emergencias o de atención primaria. Utilizando datos disponibles entre 2009 y 2012, los resultados obtenidos fueron prometedores. Se observaron mejoras sustanciales en las métricas macro F1, con ganancias del 12.5% para la clasificación de riesgo vital, del 17.5% para la demora en la respuesta y del 5.1%

para la clasificación por jurisdicción, en comparación con el protocolo interno de triaje de la Comunidad Valenciana.

Sin embargo, es esencial tener en cuenta que los sistemas, los protocolos de triaje y las prácticas operativas evolucionan de forma natural con el tiempo. Los modelos que mostraron un rendimiento excelente con el conjunto de datos inicial de 2009 a 2012 no demostraron la misma eficacia cuando se evaluaron con datos que abarcaban de 2014 a 2019 (los datos de 2013 no estaban disponibles). Este último conjunto de datos había sufrido modificaciones significativas en comparación con el anterior. Estas modificaciones provocaron cambios en el conjunto de datos, lo que dio lugar a variaciones en las distribuciones de probabilidad, que hemos caracterizado e investigado meticulosamente en esta tesis, centrándonos en su impacto en el rendimiento del modelo.

Continuando con nuestra investigación, nuestro objetivo era proporcionar un rendimiento sostenible del modelo a lo largo del tiempo o, como mínimo, mitigar los efectos adversos de las inevitables variaciones en la distribución de los datos de la forma más eficaz posible. Para hacer frente a este reto, nos centramos en el Aprendizaje Continuo Profundo. Al incorporar el paradigma del Aprendizaje Continuo a nuestros diseños y desarrollos, pudimos mitigar sustancialmente los efectos adversos sobre el rendimiento y comprender mejor cómo gestionar el despliegue de modelos a lo largo del tiempo en un centro de atención a la llamada de urgencias y emergencias médicas extrahospitalarias. Los resultados de nuestra investigación indican que, al considerar el Aprendizaje Continuo Profundo, si bien no elimina por completo las fluctuaciones de rendimiento a lo largo del tiempo, las mantiene efectivamente dentro de un rango manejable. En particular, con respecto a la métrica F1, cuando las variaciones distribucionales son ligeras o moderadas, el comportamiento se mantiene estable, sin variar más de un 2.5%, como se observa en nuestros datos de incidentes médicos extrahospitalarios. Por lo tanto, bajo estas condiciones, el rendimiento de nuestros modelos es operativamente aceptable.

Además, nuestra tesis demuestra la viabilidad de construir herramientas auxiliares que permitan a los operadores interactuar con estos complejos modelos. En consecuencia, sin interrumpir el flujo de trabajo de los profesionales, se hace posible proporcionar retroalimentación mediante predicciones de probabilidad para cada clase de etiqueta de gravedad y tomar las medidas adecuadas en función de estas predicciones.

Por último, los resultados de esta tesis tienen implicaciones directas en la gestión de las urgencias y emergencias extrahospitalarias en la Comunidad Valenciana. El modelo final resultante de nuestra investigación está previsto que se integre en los centros de atención de llamadas asociadas a urgencias y emergencias médicas de la Comunidad Valenciana. Este modelo utilizará los datos proporcionados por los operadores telefónicos para calcular automáticamente las predicciones de gravedad, que

luego se compararán con las generadas por el protocolo de triaje interno. Cualquier disparidad entre estas predicciones desencadenará la derivación del incidente a un coordinador médico, que supervisará su tratamiento. Por lo tanto, es evidente que nuestra tesis, además de realizar importantes contribuciones al campo de la Investigación en Aprendizaje Automático Biomédico, también conlleva implicaciones sustanciales para mejorar la gestión de las urgencias y emergencias extrahospitalarias en el contexto de la Comunidad Valenciana.

Resum

El triatge dels incidents d'urgències i emergències extrahospitalàries representa un repte difícil, principalment a causa de les limitacions temporals, que exigeixen una avaluació ràpida de les prioritats en un estret marge de temps, i de la incertesa, prendre decisions amb la escassa informació disponible. A més, els errors en aquest procés poden tindre greus conseqüències per als pacients, amb el consequent risc de mort. Per tant, qualsevol protocol, eina o estratègia innovadora que haja demostrat millorar aquests processos pot oferir un valor substancial en termes d'atenció al pacient i gestió global dels incidents mèdics d'urgències i emergències extrahospitalàries.

La hipòtesi fonamental en què es basa aquesta tesi és que l'Aprenentatge Automàtic, concretament l'Aprenentatge Profund, pot millorar significativament aquests processos proporcionant estimacions de la gravetat dels incidents mèdics d'urgència i emergència extrahospitalària, tenint en compte la informació de la qual disposa l'operador en el moment del triatge de l'incident durant la trucada d'emergència. Mitjançant l'anàlisi de milions de dades derivades de trucades d'emergència de la Comunitat Valenciana (Espanya) que abasten des de 2009 fins a 2019, plantegem que els models d'Aprenentatge Automàtic podrien extreure patrons que poden atorgar capacitat predictiva a aquesta tasca.

Per això, aquesta tesi aprofundeix en el disseny i desenvolupament de diversos models d'Aprenentatge Automàtic, concretament models d'Aprenentatge Profund Multitasca que aprofiten dades multimodals provinents d'incidents mèdics d'urgències i emergències extrahospitalàries. El nostre objectiu principal era predir tres etiquetes indicatives de la gravetat de l'incident, influint així en la seva prioritat. Aquestes etiquetes englobaven si l'incident suposava una situació de risc vital, la demora admissible de la resposta (des de no demorable fins a minuts, hores o dies) i si era competència del sistema d'emergències o d'atenció primària. Utilitzant dades disponibles entre 2009 i 2012, els resultats obtinguts van ser prometedors. Es van observar millores substancials en les mètriques macro F1, amb guanys del 12.5% per a la classificació de risc vital, del 17.5% per a la demora en la resposta i del 5.1% per a la classificació per jurisdicció, en comparació amb el protocol intern de triatge de la Comunitat Valenciana.

Tanmateix, és essencial tindre en compte que els sistemes, els protocols de triatge i les pràctiques operatives evolucionen de forma natural amb el temps. Els models que van mostrar un rendiment excel·lent amb el conjunt de dades inicial de 2009 a 2012 no van demostrar la mateixa eficàcia quan es van avaluar amb dades que abastaven de 2014 a 2019 (les dades de 2013 no estaven disponibles). Aquest últim conjunt de dades havia sofert modificacions significatives en comparació amb l'anterior. Aquestes modificacions van provocar canvis en el conjunt de dades, la qual cosa va donar lloc a variacions en les distribucions de probabilitat, que hem caracteritzat i investigat minuciosament en aquesta tesi, centrant-nos en el seu impacte en el rendiment del model.

Continuant amb la nostra investigació, el nostre objectiu era proporcionar un rendiment sostenible del model al llarg del temps o, com a mínim, mitigar els efectes adversos de les inevitables variacions de la distribució de les dades de la forma més eficaç possible. Per fer front a aquest repte, ens vam centrar en l'Aprenentatge Continu Profund. En incorporar el paradigma de l'Aprenentatge Continu als nostres dissenys i desenvolupaments, vam poder mitigar substancialment els efectes adversos sobre el rendiment i comprendre millor com gestionar el desplegament de models al llarg del temps en un centre d'atenció a la trucada d'urgències i emergències mèdiques extrahospitalàries. Els resultats de la nostra investigació indiquen que, quan es considera l'Aprenentatge Continu Profund, si bé no elimina completament les fluctuacions de rendiment al llarg del temps, les manté efectivament dins d'un rang manejable. En particular, respecte a la mètrica F1, quan les variacions distribucionals són lleugeres o moderades, el comportament es manté estable, sense variar més d'un 2.5%, com s'observa a les nostres dades d'incidents mèdics extrahospitalaris. Per tant, en aquestes circumstàncies, el rendiment dels nostres models és operativament acceptable.

A més, la nostra tesi demostra la viabilitat de construir eines auxiliars que permeten als operadors interactuar amb aquests models complexos. En conseqüència, sense interrompre el flux de treball dels professionals, es fa possible proporcionar retroalimentació mitjançant prediccions de probabilitat per a cada classe d'etiqueta de gravetat i prendre les mesures adequades en funció d'aquestes prediccions.

Finalment, els resultats d'aquesta tesi tenen implicacions directes en la gestió de les urgències i emergències extrahospitalàries a la Comunitat Valenciana. El model final resultant de la nostra investigació està previst que s'integre en els centres d'atenció de telefonades associades a urgències i emergències mèdiques de la Comunitat Valenciana. Aquest model utilitzarà les dades proporcionades pels operadors telefònics per calcular automàticament les prediccions de gravetat, que després es compararan amb les generades pel protocol de triatge intern. Qualsevol disparitat entre aquestes prediccions desencadenarà la derivació de l'incident a un coordinador mèdic, que supervisarà el seu tractament. Per tant, és evident que la nostra tesi, a més de realitzar importants contribucions al camp de la Investigació en Aprenentatge Automàtic

Biomèdic, també comporta implicacions substancials per a millorar la gestió de les urgències i emergències extrahospitalàries en el context de la Comunitat Valenciana.

Acronyms

ADAM Adaptive Moment Estimation

ANN Artificial Neural Network

AUC Area Under Curve

BHO Bayesian Hyperparameter Optimization

BERT Bidirectional Encoder Representations from Transformers

CDSS Clinical Decision Support System

DIL Domain Incremental Learning

EMCI Emergency Medical Call Incidents

GELU Gaussian Error Linear Unit

GRU Gated Recurrent Unit

GUI Graphical User Interface

HSD Health Services Department

LDA Latent Dirichlet Allocation

LSTM Long short-term memory

MLP Multi-layer perceptron

NLP Natural Language Processing

ReLU Rectifier Linear Unit

RNN Recurrent Neural Networks

RMSProp Root Mean Square Propagation

SGD Stochastic Gradient Descent

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

When dealing with Emergency Medical Call Incidents (EMCI), proper resource allocation within the shortest time frame possible is critical, since patient's life may be at risk. To adequately determine the most suitable resource to distribute, the priority of the incoming incidents needs to be assessed. Based on the severity determined by this triage process, corresponding actions will be taken. Hence, accurately carrying out emergency medical triage of these EMCIs is crucial for handling the incident appropriately.

However, performing out-of-hospital emergency medical triage is a tough challenge in a real setting. Uncertainty is high since the incident is handled remotely, and time constraints limit the data collection process. Hence, the information available for decision-making is often partial and incomplete, comprising sparse data integrated by different types of features.

During an emergency medical call, the dispatcher raises questions to the callers based on the provided information, the guidelines established by the coordination center, and the dispatcher's own experience. The information exchanged among participants is typically recorded digitally, yet, in most cases, it remains underutilized beyond fundamental retrospective analysis and straightforward quality controls.

In the context of the Valencian Region in Spain, the process of emergency medical triage adheres to an in-house clinical protocol. This protocol is manifested as a decision tree comprising multiple questions designed to encompass the intricate array of out-of-hospital emergencies in a structured manner while assigning priority to each situation. However, reality often proves exceedingly complex, and frequently,

additional information beyond this clinically structured data is acquired during the call, typically in the form of free text dispatcher observations.

In particular, free text dispatcher observations remain outside the scope of the in-house triage protocol, as they consist of unstructured data and are not incorporated into the decision tree. Similarly, additional contextual and demographic information is recorded during the call but falls outside the purview of the triage protocol. Consequently, alternative tools are imperative for extracting valuable insights from these intricate data sets, revealing potential latent informative data patterns—specifically, mathematical models rooted in statistics and computer science, such as Machine Learning models.

Furthermore, over time, the distributions of data about the information recorded by dispatchers change, a phenomenon referred to as dataset shifts. These shifts in data distribution can significantly impact the performance of any model. Hence, detecting and characterizing these alterations in data distribution becomes crucial. Subsequently, following a Continual Learning approach, a series of actions are essential to mitigate the adverse effects on model performance, primarily focusing on the design and training of the models.

Hence, the primary objective of this doctoral thesis is to enhance emergency medical dispatch procedures by providing decision-making support within the realm of emergency medical triage. This support entails furnishing estimations of incident severity based on available data. However, unlike the existing in-house triage protocol in the Valencian Region, our approach incorporates additional structured and unstructured data beyond the standard clinical variables associated with the decision tree. The central concept, therefore, revolves around extracting intricate mathematical relationships among this multifaceted data by utilizing Machine Learning models, specifically Deep Learning models. Furthermore, within the scope of this thesis, we aspire to deliver severity predictions in an environment where data distributions fluctuate over time. Consequently, we introduce mechanisms to mitigate the adverse effects of dataset shifts, striving to maintain performance stability to the greatest extent possible within the constraints of distributional variation.

1.2 Research questions

The main research questions posed in this thesis are:

RQ1 Are there latent informative patterns within the EMCI data that the Valencian Region's in-house triage protocol does not currently consider?

RQ2 Is it feasible to reveal these latent patterns of information using Machine Learning?

RQ3 Can traditional Machine Learning models enhance the performance of the existing in-house triage protocol of the Valencian Region when assessing incident severity?

RQ4 Can novel Machine Learning models, specifically Deep Learning, offer value as predictive models for assessing incident severity in out-of-hospital emergency medical triage?

RQ5 Can we detect and characterize the temporal dataset shifts that occur over time within the EMCI of the Valencian Region?

RQ6 How do these temporal dataset shifts impact the performance of a Deep Learning-based model trained with data from a specific period?

RQ7 If these temporal dataset shifts harm model performance, is it possible to design and implement Continual Learning pipelines to alleviate the adverse performance effects caused by distributional drifts over time?

RQ8 Is it possible to enable straightforward utilization of a Deep Learning model for incident severity assessment by an emergency dispatcher?

1.3 Objectives

Based on the research questions posed in the previous section, the next objectives are proposed:

O1 Review of the state-of-the-art in Machine Learning models designed to offer assistance in out-of-hospital emergency medical triage.

O2 Develop and evaluate Machine Learning models, with a particular emphasis on Deep Learning, for incident severity assessment in incoming EMCIs, comparing their performance with the in-house triage protocol of the Valencian Region.

O3 Discover and characterize latent statistical patterns hidden within unstructured data in the EMCI domain to shed light on new information predictive for EMCI classification.

O4 Study the presence of dataset shifts over time in the EMCI data of the Valencian Region, providing a comprehensive description and characterization of these shifts.

O5 Design, implement, and evaluate Continual Learning pipelines to deal with these dataset shifts, aiming to minimize model performance drops over time.

O6 Integrate the model into an auxiliary tool to facilitate dispatcher interaction and make it more user-friendly.

1.4 Thesis contributions

This section outlines the primary contributions of the thesis, encompassing the most significant ones as well as complementary contributions. Additionally, it presents the associated scientific publications and conferences related to the research.

1.4.1 Main contributions

C1 Development of DeepEMC^2, a Deep Ensemble Multitask Classifier for Emergency Medical Calls.

We carried out a comparative study of various Machine Learning approaches to classify EMCI by their severity levels. Deep Learning emerged as the most effective approach, yielding superior outcomes. Moreover, when comparing DeepEMC^2 with the in-house triage protocol of the Valencian Region, we observed a significant performance enhancement, achieving a macro F1-score improvement of 12.5%, 17.5%, 5.1% in life-threatening, response delay and jurisdiction classification, respectively.

C2 Discovery of key topics in emergency medical dispatch from free text dispatcher observations.

Using an unsupervised Bayesian Machine Learning approach based on Latent Dirichlet Allocation (LDA), we identified the existence of 15 significant latent topics within unstructured data (free text). The incorporation of these topics into structured clinical protocols holds the potential to enhance the value of the emergency medical triage process significantly.

C3 Study and characterization of dataset shifts over time affecting emergency medical incidents data.

In this thesis, we have extensively examined and characterized the presence of dataset shifts over time. Our investigation has focused on three distinct types of shifts: prior probability shifts, covariate shifts, and concept shifts. Our research findings provide compelling evidence for these shifts in our EMCI data spanning from 2009 to 2019. Notably, we have detected all three types of shifts across various feature modalities. Furthermore, we have quantified their impact on model performance over time.

C4 Development of Deep Continual Learning pipelines for emergency medical incidents classification.

We have carefully designed, implemented, and assessed Deep Continual Learning pipelines to mitigate the adverse consequences of dataset shifts. These pipelines have primarily focused on the dynamic update of parameters over time, effectively balancing knowledge retention and model adaptability, preserving pertinent information from the past while facilitating the integration of new information. Furthermore, by introducing architectural modifications in our models, we have addressed the challenge of dynamic feature domains, where some features emerge while others disappear over time. Overall, although it is challenging to eliminate the effects of dataset shifts entirely, our strategies have successfully curbed negative performance deterioration. As a result, model performance within our EMCI data has remained stable within a margin of approximately 2.5%, as long as the distributional shifts are not excessively severe.

C5 Development of a continual end-to-end version of DeepEMC^2.

We have designed, implemented, and evaluated a novel version of DeepEMC^2, which integrates the previously mentioned Continual Learning pipelines. Additionally, we have embraced a Multitask and Multimodal approach in an end-to-end fashion. Consequently, this approach has resulted in reduced memory requirements with respect to the previous version of DeepEMC^2. Simultaneously, incorporating novel architectural enhancements and Continual Learning strategies has proven beneficial in enhancing model performance concerning the assessment of EMCI severity.

C6 Development of an auxiliary tool to allow direct interaction with DeepECM^2.

We have created a prototype tool to enable external users, even those without expertise in Deep Learning, to interact with the model without effort. This tool replicates the input features that a Valencian Region dispatcher would typically input, invokes the model, and promptly retrieves a real-time response. It provides probability estimates for each of the severity labels. Consequently, the dispatcher, potentially the end user of the decision-support model, no longer needs to comprehend the intricate computations underlying the final predictions. Their workflow remains unchanged, as they can input and collect information within the system in the same familiar manner.

1.4.2 Complementary contributions

C7 In-depth study and development of techniques for extreme missing data imputation in Electronic Health Records.

We have conducted an in-depth investigation into the most appropriate techniques for data imputation, especially when dealing with a high rate of missing values. Specifically, we implemented and assessed a total of 30 preprocessing, imputation, and modeling pipelines. The imputation methods included missing mask, translation and encoding, mean imputation, k-nearest neighbors' imputation, Bayesian ridge regression imputation and generative adversarial imputation networks. The classifiers included k-nearest neighbors, logistic regression, random forest, gradient boosting and deep multilayer perceptron. Our findings indicate that in cases of high incompleteness, translating features and subsequently encoding the missing values represent a prudent choice more robust to noise than state-of-the-art standard imputation methods.

1.4.3 Main scientific publications

We proceed to present the main scientific publications associated with this thesis:

P1 Pablo Ferri, Carlos Sáez, Antonio Félix-De Castro, Javier Juan-Albarracín, Vicent Blanes-Selva, Purificación Sánchez-Cuesta, Juan M García-Gómez. *Deep ensemble multitask classification of emergency medical call incidents combining multimodal data improves emergency medical dispatch.* Artificial Intelligence in Medicine, 117, 102088. May 2021. (Ferri et al., 2021).

IF: 7.011 (JCR 2021): 21/98 Engineering, Biomedical (Q1); 32/145 Computer Science, Artificial Intelligence (Q1); 8/31 Medical Informatics (Q2).

P2 Pablo Ferri, Carlos Sáez, Antonio Félix-De Castro, Ángel Sánchez-García, Purificación Sánchez-Cuesta, Juan M García-Gomez. *An artificial intelligence tool to classify emergency medical incidents in real-time improves emergency medical dispatch.* European Emergency Number Association (EENA) Conference and Exhibition. Marseille, France. April 2022. (Ferri et al., 2022b).

P3 Pablo Ferri, Carlos Sáez, Antonio Félix-De Castro, Purificación Sánchez-Cuesta, Juan M García-Gómez. *Discovering key topics in emergency medical dispatch from free text dispatcher observations.* 32nd Medical Informatics Europe Conference (MIE). Nice, France: IOS Press. Studies in Health Technology and Informatics: 294. May 2022 (Ferri et al., 2022a).

P4 **Pablo Ferri**, Vincenzo Lomonaco, Lucia C.Passaro, Antonio Félix-De Castro, Purificación Sánchez-Cuesta, Carlos Sáez, Juan M García-Gómez. *Deep continual learning for emergency medical call incidents text classification under the presence of dataset shifts.* Addressing reviewer's comments at Computers in Biology and Medicine.

P5 Carlos Sáez, **Pablo Ferri**, Juan M García-Gómez. *Resilient artificial intelligence in health: a synthesis and research agenda towards next-generation trustworthy clinical decision support.* Under review at Journal of Medical Internet Research.

P6 **Pablo Ferri**, Carlos Sáez, Antonio Félix-De Castro, Purificación Sánchez-Cuesta, Juan M García-Gómez. *Deep continual out-of-sample multitask classification of emergency medical call incidents under dataset shifts affecting feature domain.* Submission in progress to npj Digital Medicine.

P7 **Pablo Ferri**, Carlos Sáez, Antonio Félix-De Castro, Purificación Sánchez-Cuesta, Juan M García-Gómez. *Deep continual multitask classification of emergency medical call incidents over time combining multimodal data.* Submission in progress to Artificial Intelligence in Medicine.

1.4.4 Complementary scientific publications

Next, we present the complementary scientific publications:

P8 **Pablo Ferri**, Nekane Romero-Garcia, Rafael Badenes, David Lora-Pablos, Teresa García Morales, Agustín Gómez de la Cámara, Juan M García-Gomez, Carlos Sáez. *Extremely missing numerical data in Electronic Health Records for machine learning can be managed through simple imputation methods considering informative missingness: A comparative of solutions in a COVID-19 mortality case study.* Computer Methods and Programs in Biomedicine, 107803. September 2023. (Ferri et al., 2023).

IF: 6.1 (JCR 2022): 15/111 Computer Science, Theory & Methods (Q1); 7/31 Medical Informatics (Q1); 25/110 Computer Science, Interdisciplinary Applications (Q1); 22/96 Engineering, Biomedical (Q1).

1.5 Projects and partners

The projects mainly related to the development of this thesis are listed as follows:

PJ1 **Desarrollo de un sistema experto de clasificación de la demanda sanitaria de urgencias, emergencias extrahospitalarias y demanda sani-**

taria 112. [Development of an expert system for classifying the health demand of out-of-hospital medical emergencies 112].

Objectives: The objective of this project is to develop Machine Learning-based solutions for the emergency medical triage process of the Valencian Region (Spain), comparing their performance with the current one of the in-house triage protocol.

Partners: Conselleria de Sanitat Universal I Salut Pública, Generalitat Valenciana (Valencia, Spain), Intelligent Data Analysis Laboratory, Universitat de València (Valencia, Spain) and Biomedical Data Science Lab, Universitat Politècnica de València (Valencia, Spain).

Funder: Agencia Valenciana de Seguridad y Respuesta a Emergencias.

Duration: July 2018 - May 2019.

PJ2 Módulo integrable de apoyo a la llamada sanitaria de urgencias y emergencias extrahospitalarias. [Integrable module to support out-of-hospital emergency medical calls].

Objectives: The objective of this contract was to integrate the Deep Learning model developed in this thesis into the information system of the 112 services of the Valencian Region (Spain). We collaborated with the multinational company Omda, the developer of the CoordCom platform, which is currently in use at the Valencian emergency dispatch center. The deep model has to be embedded in this platform, with the collaboration of professionals from the Conselleria de Sanitat.

Partners: Conselleria de Sanitat Universal I Salut Pública, Generalitat Valenciana (Valencia, Spain), Omda (Gothenburg, Sweden) and Biomedical Data Science Lab, Universitat Politècnica de València (Valencia, Spain).

Funder: Conselleria de Sanitat Universal I Salut Pública, Generalitat Valenciana.

Duration: June 2023 - June 2025.

Next, the projects in which the author was actively involved in parallel to the development of this thesis are presented:

PJ3 Severity Subgroup Discovery and Classification on COVID-19 Real World Data through Machine Learning and Data Quality assessment (SUBCOVERWD-19).

Objectives: The objective of this project is to uncover latent patterns present in COVID-19 data registered at the *Hospital 12 de Octubre* and *Hospital*

Clínic, disentangling patients according to their severity, in a context of data with quality problems that hinder this discovery process.

Partners: Hospital 12 de Octubre (Madrid, Spain), Hospital Clínic i Universitari (Valencia, Spain) and Biomedical Data Science Lab, Universitat Politècnica de València (Valencia, Spain).

Funder: CRUE - Santander Bank.

Duration: January 2021 - September 2021.

1.6 Grants

G1 Ayuda para contratos predoctorales para la formación de doctores dentro del programa propio de la Universitat Politècnica de València—Subprograma 1 (PAID-0-18). [Aid for pre-doctoral contracts for the training of doctors within the Universitat Politècnica de València's own program—Subprogram 1 (PAID-0-18)].

Project: Diseño de un sistema de guiado adaptativo para la gestión de llamadas y óptima asignación de prioridades en los servicios de urgencias y emergencias extrahospitalarias. [Design of an adaptive guidance system for call management and optimal prioritisation in out-of-hospital emergency services].

Funder: Universitat Politècnica de València.

Duration: March 2019 - September 2019.

G2 Ayuda predoctoral para la formación de profesorado universitario (FPU-2018). [Pre-doctoral aid for university teacher training (FPU-2018).]

Project: Desarrollo de una herramienta de apoyo a la llamada sanitaria de urgencias y emergencias extrahospitalarias.[Development of a tool to support out-of-hospital emergency medical calls].

Funder: Ministerio de Ciencia, Innovación y Universidades.

Duration: September 2019 - July 2023.

1.7 Research stays

RS1 Research stay at the Computational Intelligence and Machine Learning Group

Host institution: Computational Intelligence and Machine Learning Group (CIMLG), Università di Pisa (UniPi).

Person in charge at host institution: Professor Vincenzo Lomonaco.

Objective: The purpose of the research stay was to acquire knowledge regarding Continual Learning techniques and integrate them into the design of pipelines to alleviate performance degradation over time in the model developed for classifying EMCI.

Duration: from 03/05/2022 to 01/08/2022.

1.8 Thesis outline

This thesis is structured as follows. Chapter 1 has defined the thesis motivations, research questions, and main contributions. Chapter 2 describes the thesis rationale, exposing the complexity of the emergency medical call incidents triage challenge and presenting the theoretical-methodological background. Chapter 3 describes the development of DeepEMC^2, a deep ensemble multitask classification model of emergency medical call incidents able to combine multimodal data, highlighting its provided added value respect to the in-house triage protocol of the Valencian Region. Chapter 4 provides an unsupervised analysis of free text dispatcher observations, discovering key topics latent in those fields but highly relevant for emergency medical dispatch. Chapter 5 describes the study of dataset shifts over time in free text fields and the development and evaluation of multiple deep Continual Learning pipelines centered on handling the negative effects of these dataset shifts. Chapter 6 presents an analysis of dataset shifts over time focused on the clinical variables and the development and evaluation of different deep continual learning pipelines capable of facing the dataset shift challenge and the dynamic feature domain challenge. Chapter 7 presents a robust and end-to-end version of DeepEMC^2, aiming to deal with dataset shifts and dynamic feature domains. Chapter 8 describes how this model could be embedded in a tool usable by emergency medical dispatchers. Finally, Chapter 9 describes the main conclusions from this thesis.

Next, Figure 1.1 presents the outline of the thesis contributions showcasing the relationships among the chapters, contributions, publications, projects, and research stays.

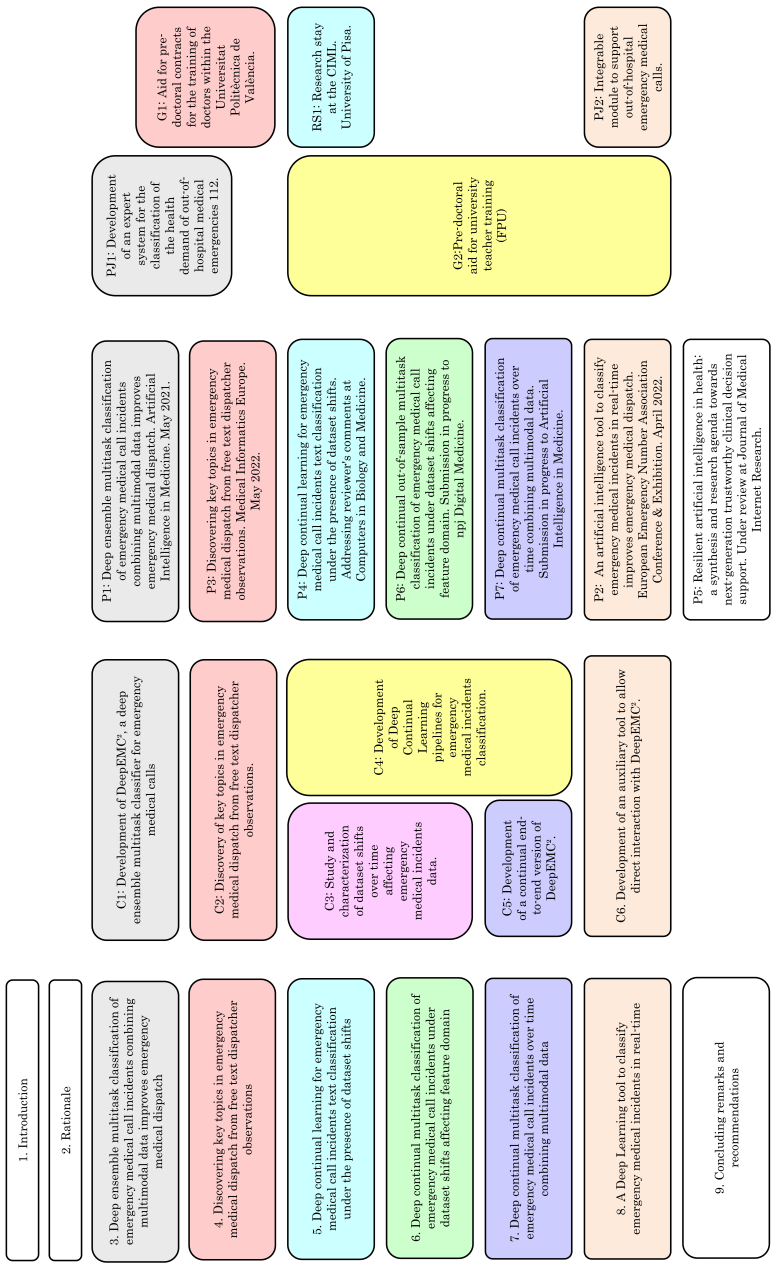

Figure 1.1: Thesis outline, including chapters, contributions, publications, projects, and research stays.

Chapter 2

Rationale

This chapter describes the Rationale of this thesis, including the technical background required to understand the challenges faced along this thesis, as well as the solutions provided. First, we introduce the basic concepts, objectives, and context of out-of-hospital emergency medical triage, particularizing in the processes followed at the Valencian Region (Spain). Next, we present the main technical foundations of this thesis. The first one is Machine Learning, and the second is Deep Learning. While Deep Learning is a subfield of Machine Learning, its significance in this thesis warrants a dedicated section. Finally, we discuss the current literature oriented on out-of-hospital medical emergency triage support based on Machine Learning-based models.

2.1 Out-of-hospital emergency medical triage

2.1.1 Background and definitions

An out-of-hospital emergency medical incident is a situation where an individual (or a group of individuals) requires non-delayable medical attention—due to an injury, illness, or other medical emergency—while occurring outside of a healthcare facility. Examples of out-of-hospital emergency medical incidents are cardiac arrest, severe trauma, stroke, respiratory distress, etc. At the same time, out-of-hospital emergency medical triage is defined as the process of categorizing patients involved in this type of event by their incoming severity and thus, according to their attendance necessities.

Emergency medical triage of out-of-hospital events is required for two main reasons. Firstly, patients involved in life-threatening situations, such as severe bleeding

or cardiac arrest, require immediate attention and should be treated first. Secondly, triage helps responders to ensure that all patients receive the appropriate level of care, even if resources are limited, prioritizing them based on the patient's needs.

However, out-of-hospital telephone emergency medical triage is a challenging procedure due to a variety of reasons. One of the most remarkable ones is the high level of uncertainty involved, which makes it difficult to assess the severity of the situation precisely. In addition, the limited time available for the triage process means that decisions must be made quickly, adding pressure on the dispatcher. Furthermore, dispatchers may also have limited medical knowledge or experience, making it hard to establish the appropriate level of care needed.

If that were not enough, errors in out-of-hospital emergency medical triage can imply severe consequences. Assigning the wrong severity level to a patient results in delays in treatment, which can worsen the patient's condition or even lead to death. On the other hand, over-triage, where a patient is assigned a higher level of severity than necessary, can also have negative consequences, such as using up limited resources or unnecessarily alarming the patient and their family. Hence, a proper assessment of patient severity through triage is crucial to ensure the appropriate emergency attention level is provided.

2.1.2 Emergency medical call incidents triage protocols

The origins of triage—word that comes from the French word *trier*, which means to sort or separate—can be traced back to the battlefield. The French surgeon Dominique Jean Larrey is credited with developing the first modern triage system during the Napoleonic Wars in the early 19th century (Blagg, 2004). Larrey observed that soldiers who received immediate medical attention had a higher chance of survival. Hence, given the large number of wounded soldiers and the limited number of resources, he developed a system of categorizing soldiers based on the severity of their injuries, with the most severely injured receiving immediate attention (Moskop & Iserson, 2007). Specifically, this system was grounded on a triage scale which assigned colors to each severity level: red for those soldiers presenting the most serious injuries, yellow for the one requiring an urgent response and green for the patients with less severe injuries who could wait some time before being attended.

During World War I, the triage system was further developed and refined by medical personnel on both sides of the conflict, becoming a fundamental component of military medicine (Pollock, 2008). Afterward, triage was later adapted for use in civilian emergency medicine, being applicable to wide spectrum of the population. In 1937, the first emergency call system, the 999, was launched in London, allowing a centralized and remote management of EMCI (Moss, 2018). With the introduction of

remote incident handling, the need of protocolized clinical guidelines aiming to deal with EMCI properly became evident.

Since those times, several clinical protocols have been developed to help dispatchers assess the severity of a patient's condition and allocate resources accordingly. Some of the most known protocols are the Emergency Severity Index (Gilboy et al., 2012), the Manchester System (Mackway-Jones et al., 2013), the Australasian Triage Scale (Considine et al., 2004) or the Canadian triage and acuity scale (Murray et al., 2004). However, despite presenting evident differences among them, they also have aspects in common: they are built following a tree structure, where each node is related to a specific question that admits closed answers—such as yes/no, higher/medium/lower, moderate pain/severe pain, etc.—and the final leaf node is associated to a severity value according to a priority scale.

While triage protocols and scales can be useful tools for dispatchers, they have limitations. One of the main challenges is that they may not capture the full complexity of a patient's condition, since the casuistry associated with the EMCI context is huge (Farand et al., 1995). For example, patients with multiple comorbidities or unusual symptoms may not fit neatly into the established categories, which can make it challenging to assign a level of severity accurately.

Additionally, triage protocols and scales may be difficult to evaluate and benchmark. Evaluating the effectiveness of a triage protocol requires a large sample size of patients, which can be difficult to obtain. Furthermore, the effectiveness of a triage protocol may depend on the resources available in a particular emergency medical dispatch system, which can vary widely between regions (FitzGerald et al., 2010; Lidal et al., 2013).

2.1.3 Emergency medical triage in the Valencian Region

Emergency Medical Call Incident Data

The Valencian Region accounts for 10% of the total population of Spain, with more than 5 million people (de Estadística, 2022). Consequently, the volume of daily emergency medical calls is substantial. Specifically, in 2022, there were 265 185 total calls registered, resulting in an average daily call volume of 727 emergency medical calls, according to the Health Services Department of the Valencian region (d'Emergències Sanitàries de la Comunitat Valenciana, 2022). This means that, on average, an emergency medical call requiring attention by an emergency medical dispatcher occurs approximately every two minutes.

A significant portion of the recent EMCI data was made available to develop this thesis. This data was generated within the context of two different information

systems: the CORDEX system and the CoordCom system, which replaced the former. Furthermore, there were variations in management policies and personnel during this period. In the following table (Table 2.1), we present the number of EMCI cases provided by the Health Services Department of the Valencian Region. This data is valuable as it includes information about the subsequent real incident severity, which is essential for the developments presented in this thesis.

Table 2.1: Number of Emergency Medical Incident data per year, suitable for training Machine Learning models, provided by the Health Services Department of the Valencian Region.

Year	Number data
2009	182 536
2010	178 458
2011	180 739
2012	180 537
2014	172 905
2015	197 987
2016	208 439
2017	254 008
2018	246 163
2019	252 922

It is worth noting that not all the data was initially available. Initially, we had access to data from 2009 to 2012 (inclusive), which was associated with the CORDEX information system. In 2013, significant changes occurred as the information system transitioned to a new one named CoordCom. These changes also led to variations in policies, protocols, and personnel handling incidents. After several years of working on this thesis, we were provided with the CoordCom data.

Additionally, it is important to mention that the use of this data was approved by the Institutional Review Board of the Health Services Department of the Valencian Region. No information revealing a patient's identity was retained for any of the analyses.

Design and deployment of the in-house emergency medical triage protocol

In the Valencian Region (Spain), out-of-hospital medical emergencies are handled through a telephone triage system. The telephone triage is carried out by dispatchers who follow an in-house triage protocol. This protocol has been built from the Manchester triage system and has evolved over the years to meet the needs of the Valencian Region—for example, as many pyrotechnic accidents happen in the Valencian Region during the *Fallas* traditional celebration, this casuistry is included in the

protocol. Hence, this in-house triage protocol relies on the collection of structured clinical variables in a sequential manner through questions raised to the caller during the call. This information is subsequently utilized to assess the urgency of patient care.

During the call, in addition to the recording of these structured clinical data based on the protocol, it is registered other relevant information, also provided by the caller. This additional information is usually entered as free text and hence, it can be considered as unstructured data. Since the in-house protocol is based on following a decision tree with a closed answer, it cannot automatically consider this unstructured data, which may provide crucial information to properly handling the incident, in terms of fine-grained detail about the patient's condition. Hence, this valuable resource cannot be used automatically during the triage process at the Valencian Region.

Evolution of out-of-hospital emergency medical triage over time

The in-house triage algorithm exposed in the previous section, dispatchers, additional training programs, and coordination of emergencies have not remained static over time at the Valencia Region. Quality control process and periodic revisions had incorporated multiple updates to the triage process to enhance patient's assistance as well as to use resources more efficiently. Examples of these updates could be the inclusion of novel questions—and hence, clinical variables associated with them—in the decision tree, specific dispatcher training programs—to properly identify uncommon incidents—etc.

Among the most relevant variations that have taken place in recent years, it stands out as the one that occurred in 2013. During this year, the information system used to handle incidents—named CORDEX—was changed to a new one, identified CoordCom. Furthermore, the institutions in charge of the out-of-hospital medical emergencies coordination changed, along with the in-house protocol—which suffered some modifications—and the dispatcher personnel.

2.2 Machine Learning

2.2.1 Background and definitions

Machine Learning can be defined as a discipline that focuses on empowering computer systems to acquire knowledge through experience without explicitly programming this knowledge by utilizing statistical-computational models (Jordan & Mitchell, 2015). Consequently, it involves the development of algorithms and sta-

tistical models that enable computers to learn from data and make predictions autonomously, without the need for human intervention.

In order for a problem to be suitable for Machine Learning, it necessitates the establishment of a well-defined problem with a quantitative measure indicating the algorithm's performance in the specific task at hand. Once this quantitative measure is defined, an iterative updating approach must be established to determine how model parameters should evolve across iterations, aiming to maximize performance.

Next, in Figure 2.1, we present the simplified general scheme of a Machine Learning process, summarizing what we have exposed previously:

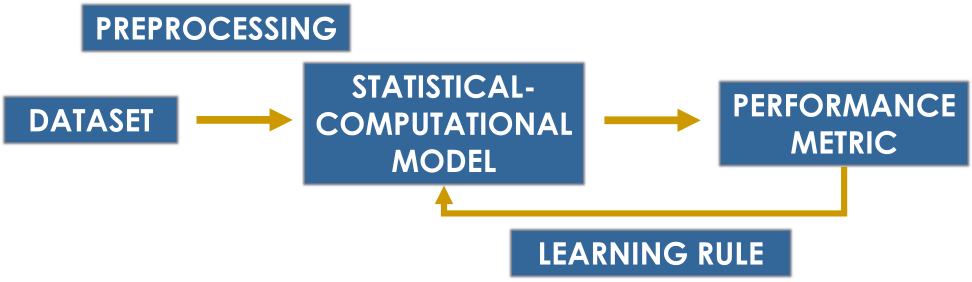

Figure 2.1: Machine Learning workflow.

Finally, it is important to remark on the differences between Machine Learning and Artificial Intelligence, since sometimes these concepts are used interchangeably, but they are not exactly the same. Artificial Intelligence is a broader field encompassing Machine Learning, but also includes other disciplines such as rule-based systems, expert systems, and heuristic search. The goal of Artificial Intelligence is to create machines that can perform tasks that normally require human intelligence, such as perception, reasoning, decision-making, and learning.

2.2.2 Tasks

There are four main tasks of interest in the field of Machine Learning: Supervised Learning, Unsupervised Learning, Semi-supervised Learning, and Reinforcement Learning (James et al., 2013; R. S. Sutton & Barto, 2018). In the next sections, we present a brief description of those relevant to this thesis, i.e., Supervised and Unsupervised Learning:

Supervised Learning

Supervised Learning is a Machine Learning approach where the objective is to build a model that can predict the output given a set of inputs. The model is trained on a labeled dataset, where the input-output pairs are provided. In a Supervised Learning setting, we have a dataset consisting of input-output pairs. Let X be the input space and Y be the output space. Under this approach, we assume that there is an unknown but underlying Joint Probability Distribution $P(X, Y)$ over $X \times Y$.

We aim to learn a function $f : X \to Y$ that can accurately predict the output Y given an input X. To achieve this, we define a hypothesis space F, which represents the set of all possible functions we can choose from. In Machine Learning, F is often called the function class or the hypothesis class.

To measure the performance of a particular function f, belonging to F, in approximating the true relationship between X and Y, a loss function $L : Y \times Y \to R$ is introduced, which quantifies the dissimilarity between predicted and true outputs. The learning process in Machine Learning involves finding the best function f from the hypothesis space F based on the available data. This is done by minimizing the expected risk, which is the expected value of the loss function over the joint distribution $P(X, Y)$:

$$R(f) = E_{XY}[L(Y, f(X))] \tag{2.1}$$

The expected risk represents the average loss incurred by using the function f as the predictor. Our objective, thus, is to find a function f^* that minimizes the expected risk:

$$f^* = \underset{f \in F}{\operatorname{argmin}} R(f) \tag{2.2}$$

However, since the true distribution $P(X, Y)$ is unknown, we cannot directly minimize the expected risk. Instead, we use the empirical risk minimization principle. Given a training dataset $D = \{(x_i, y_i)\}_{i=1}^n$ consisting of n independent and identically distributed samples from $P(X, Y)$, the empirical risk is defined as:

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \tag{2.3}$$

The empirical risk estimates the average loss over the training data. The empirical risk minimization principle states that we should choose the function f that minimizes this empirical risk:

$$f^* = \underset{f \in F}{\operatorname{argmin}} \hat{R}_n(f) \tag{2.4}$$

This empirical risk minimization problem is the core optimization problem in Supervised Machine Learning. The goal is to find the function f^* that best fits the training data.

Unsupervised Learning

Unsupervised Learning is a Machine Learning approach where the goal is to find patterns or structure in the data, without any labeled output. It includes clustering, dimensionality reduction, and density estimation.

Let X be the input space, representing the set of possible input values. In Unsupervised Learning, we are given a training dataset $D = \{x_i\}_{i=1}^n$, consisting of n independent and identically distributed samples from an unknown probability distribution $P(X)$. Each sample x_i represents an input observation.

Unsupervised learning aims to find a suitable representation or transformation of the input data that captures relevant patterns or structures. This is typically done by defining a function or model that maps the original input space X to a transformed space Z.

The function or model that maps the original input space X to the transformed space Z is often denoted as $g : X \to Z$. The goal is to find an optimal function g that captures the relevant patterns or structures in the data.

To measure the quality of the transformation, we typically define a measure of discrepancy or dissimilarity between the observed input distribution and the induced distribution in the transformed space. This discrepancy measure can vary depending on the specific Unsupervised Learning task.

For example, if the Unsupervised Learning task is clustering, where the goal is to partition the input data into groups or clusters based on their inherent similarities, the objective is to find a function g^* that minimizes the discrepancy between the observed input distribution $P(X)$ and the induced distribution $P(Z)$ in the transformed space Z. This can be formulated as an optimization problem involving a discrepancy measure, such as:

$$g^* = \underset{g}{\operatorname{argmin}} D(P(X), P(Z)) \tag{2.5}$$

where D is a measure of discrepancy, and $P(Z)$ is the distribution induced by the function g on the transformed input data.

Another common Unsupervised Learning task is dimensionality reduction, where the goal is to find a lower-dimensional representation of the input data while preserving relevant information, in order to reduce the negative effects consequence of the curse of dimensionality. In this case, the objective is to find a function g^* that minimizes the discrepancy between the observed input distribution $P(X)$ and the induced distribution $P(Z)$ in the transformed space Z, while also satisfying certain constraints on the dimensionality of Z. This can be formulated as an optimization problem similar to clustering, with additional constraints on the dimensionality of Z.

2.2.3 Models

According to how the relation between the input features and the output of a Machine Learning model (such as classification labels, or cluster membership) is established, we can distinguish two main groups: discriminative models and generative models (Bishop & Nasrabadi, 2006; James et al., 2013).

A discriminative model is a model used to predict the value of the output given a set of input variables. In the context of a supervised classification problem, the goal of a discriminative model is to learn the boundary—or decision surface—that separates the different classes of the output variable in the input space. Examples of discriminative models in such a context include logistic regression, support vector machines, and neural networks. Mathematically, a classification discriminative model learns the conditional probability distribution of the output variable given the input variable, $P(Y|X)$.

A generative model, on the other hand, is a model that is used to learn the underlying probability distribution of the data. The goal of a generative model is to learn the joint probability distribution of the input variables and the output. This implies learning $P(X, Y)$ if we focus on a supervised classification domain or $P(X, Z)$ if we are centered in an unsupervised learning task, following the notation considered in the previous section. After learning this joint distribution, we can use it to generate new samples or to estimate the probability of a new input belonging to a certain class or group. Examples of generative models include Gaussian mixture models, hidden Markov models, and Generative Adversarial Networks.

Next, we include a brief presentation of the main generative and discriminative models considered in this thesis:

Generative

Naïve Bayes

A Naive Bayes model (Bayes & Price, 1763) is a probabilistic model based on Bayes' theorem and the assumption of independence among the features, which is used for classification tasks in Machine Learning. The name Naive comes from the assumption of independence among features, which is often unrealistic but simplifies the computation and makes the model easy to implement and understand.

The basic idea behind Naive Bayes is to use Bayes' theorem, which states that:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \tag{2.6}$$

Here, Y is the output variable (class) and X are the input variables (features). $P(Y|X)$ is the posterior probability of the class given the features, $P(X|Y)$ is the likelihood of the features given the class, $P(Y)$ is the prior probability of the class, and $P(X)$ is the marginal likelihood of the features.

In a Naive Bayes model, the likelihood $P(X|Y)$ is assumed to be a product of the individual likelihoods of the features given the class:

$$P(X|Y) = \prod_{k=1}^{K} P(X_k|Y) \tag{2.7}$$

This assumption of independence among the features allows us to simplify the computation and estimate the likelihood of each feature independently.

Finally, the classification is done by choosing the class that maximizes the posterior probability $P(Y|X)$, which can be computed by plugging the estimates of the prior probability $P(Y)$ and the likelihood $P(X|Y)$ into Bayes' theorem.

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model proposed in (Blei et al., 2003) that is used for discovering the latent topics in a corpus of text. It is based on the assumption that each document in a corpus is a mixture of latent topics and that each topic defines a probability distribution over a fixed vocabulary of words.

The model is based on three main assumptions:

1. The words in a document are generated by a mixture of topics.
2. Each topic is a probability distribution over words.
3. The topics for each document are drawn from a Dirichlet distribution.

In mathematical terms, the generative process of LDA is defined as follows:

For each document d in a corpus of D documents:

- Draw a topic mixture from a Dirichlet prior, $\theta_d \sim Dirichlet(\alpha)$
- For each word w_n in document d:
  - Draw a topic z_n from the multinomial topic mixture, $z_n \sim Multinomial(\theta_d)$
  - Draw a word w_n from the topic-specific word distribution, $w_n \sim Multinomial(\beta_{z_n})$

Here α is the hyperparameter of the Dirichlet prior, θ_d is the topic mixture for document d, and β is the matrix of the topic-specific word distributions.

The main goal of LDA is to estimate the latent topics and the topic-word distributions from the observed word counts in a corpus of documents. The estimation of the model parameters is typically done using a variation of the Expectation-Maximization (EM) algorithm called collapsed Gibbs sampling (Geman & Geman, 1984), which is a Markov Chain Monte Carlo method that allows for efficient sampling from the posterior distribution of the model parameters given the data.

Once the model parameters are estimated, LDA can be used for several tasks such as topic modeling, document classification, and information retrieval. The interpretation of the topics is a subjective task and it depends on the researcher's understanding of the corpus, but usually, the most common way to interpret the topics is by looking at the top words for each topic. These top words provide a rough idea of the main theme or concept that the topic represents. It is also possible to use external information such as labels or metadata to help interpret the topics.

The number of topics, K, is an hyperparameter of the model, the selection of the number of topics is a trade-off between interpretability and coherence of the topics, one common method for selecting the number of topics is using coherence measures such as the topic coherence (Röder et al., 2015) or the perplexity (Blei, 2012).

Discriminative

Logistic Regression

Logistic Regression (Nelder & Wedderburn, 1972) is a type of Generalized Linear Model that is used for binary and multinomial classification tasks. It is a discriminative model that models the probability of the output variable (class) given the input variables (features).

The basic idea behind Logistic Regression is to model the relationship between the input variables and the output variable using a logistic function, also known as the sigmoid function. The logistic function maps the input variable to a value between 0 and 1, which can be interpreted as the probability of the output variable being 1 (or belonging to a certain class). The logistic function is defined as:

$$P(Y|X) = \frac{1}{1 + e^{-(W^\top X + b)}} \tag{2.8}$$

Here Y is the output variable, X is the input variable, W is the weight vector, and b is the bias term.

The goal of Logistic Regression is to find the values of the model parameters W and b that maximize the likelihood of the data. The likelihood is a function of the model parameters and the data, and it measures how well the model fits the data. The maximum likelihood estimates of the model parameters can be found using optimization algorithms such as gradient descent, Newton's method, or fisher scoring algorithm.

Once the model parameters are estimated, Logistic Regression can be used for prediction by computing the probability of the output variable given the input variable, $P(Y|X)$, and choosing the class with the highest probability. Logistic regression can also be used for feature selection and interpretation. The magnitude and the sign of the model parameters indicate the importance and the direction of the effect of each feature on the output variable.

Decision Trees

A Decision Tree is a type of model used in Machine Learning for both classification and regression tasks (Breiman et al., 1984). It is a tree-like structure that represents a series of decisions based on the values of the input features, with each internal node of the tree representing a test of the feature value and each leaf node representing a predicted output value or class.

The basic idea behind decision trees is to recursively split the input space into smaller and smaller regions, each associated with a specific output value or class, in such a way that the samples within each region are as similar as possible with respect to the output variable. The process of creating the tree is called tree induction, and it starts by selecting the feature and the threshold that maximizes the reduction of impurity of the samples within the region.

The most common impurity measures are Gini impurity and information gain, which are used to quantify the homogeneity of the samples within a region. The Gini impurity is defined as:

$$Gini(p) = 1 - \sum_{c=1}^{C} p_c^2 \tag{2.9}$$

Here p is the vector of class probabilities.

On the other hand, information gain is defined as:

$$IG(D, A) = E(D) - \sum_{v=1}^{V} \frac{|D_v|}{|D|} E(D_v) \tag{2.10}$$

Here D is the set of samples, A is the feature, D_v is the subset of samples for which $A = v$, and E is the entropy function, which acts as impurity measure.

Once the tree is created, it can be used for prediction by traversing the tree from the root to a leaf node, following the decision path that corresponds to the input feature values.

Decision trees are widely used in many applications such as image recognition, natural language processing, and customer churn prediction. They are easy to interpret and understand, and they can handle both numerical and categorical data. They are also able to handle missing data and they are not sensitive to the scale of the features.

Random Forest

A Random Forest is an ensemble learning method, proposed in (Ho, 1995), that combines multiple decision trees to improve the performance and robustness of the model. It is an extension of the Decision Tree algorithm—presented in the previous subsection—that builds a collection of Decision Trees and averages their predictions to improve the accuracy and reduce the variance of the model.

The basic idea behind Random Forest is to generate multiple Decision Trees, each trained on a different random subset of the training data, and to average their predictions to improve the performance of the global model. This is done by randomly selecting a subset of the features at each split of the tree, in a process called feature bagging, which aids to decrease correlation among the trees and increases the diversity of the ensemble.

The model hyperparameters of a Random Forest are the same as those of a decision tree, but with the addition of a hyperparameter for the number of trees in the forest. This number controls the trade-off between bias and variance of the model.

Random Forest is used in many applications because they are easy to interpret and understand. Additionally, random forests are less prone to overfitting compared to single decision trees, which makes them a more robust model.

Gradient Boosting

Gradient Boosting is a Machine Learning technique, presented in (Friedman, 2001), for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically Decision Trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Formally, let $L(y, f(x))$ be a differentiable loss function, where y is the true label and $f(x)$ is the predicted label for a sample x. The basic idea of gradient boosting is to iteratively add weak models, $f_i(x)$, to the ensemble, in order to improve the overall prediction, $F(x) = f_0(x) + f_1(x) + ... + f_n(x)$, where $f_0(x)$ is an initial approximation of the solution and n is the number of iterations. At each iteration, the gradient of the loss function with respect to the current ensemble prediction is computed, and a new weak model is fit to the negative gradient, i.e., the direction of steepest decrease of the loss.

Model parameters are estimated using a two-step process:

1. Initialize the ensemble with a single weak model, e.g., a decision tree with a single split.

2. At each iteration, fit a new weak model to the negative gradient of the loss function with respect to the current ensemble prediction.

Gradient Boosting is used in many fields, including web search ranking, ecology and computer vision, due to its good performance and ability to handle diverse data types. Likewise, interpretation of the model is done by analyzing the individual weak models and their contributions to the final ensemble prediction.

Artificial Neural Networks

An Artificial Neural Network (ANN) is a Machine Learning model inspired by the structure and function of the human brain (Rosenblatt, 1958). It is a network of interconnected nodes, called artificial neurons, that are organized into layers. Each neuron receives input from other neurons, processes it through an activation function, and produces an output that is passed on to other neurons in the next layer.

Formally, an artificial neuron is a mathematical function that maps a set of input values X to an output value Y through a set of weights W and a bias term $b : Y = f(WX + b)$ where f is the activation function. The activation function is a non-linear function that introduces non-linearity into the model, allowing it to learn complex relationships between inputs and outputs.

The multiple layers of neurons in an ANN model allow learning hierarchical representation, where lower layers—the initial layers, closer to the input data—learn simple features and higher layers—the ones closer to the output layer—learn complex features. This can be represented by a computation graph, where the input is passed through multiple layers before producing the final output.

ANN are used in a wide range of applications, such as image recognition, natural language processing, and speech recognition. They are particularly useful for tasks that involve large amounts of data and complex relationships between inputs and outputs.

An ANN's model parameters (weights and biases) are typically estimated using stochastic gradient descent. It is an iterative algorithm that adjusts the parameters of the model to minimize the error between the predicted output and the true output.

Interpretation of an ANN model can be challenging, as the internal workings of the model are highly complex and non-linear. More information about this type of model can be found in the next Section Deep Learning.

2.2.4 Multimodal Learning

Definition

In the context of Machine Learning, "modality" refers to a specific type of data presented to a model. Thus, each modality signifies a different channel or form of input utilized by Machine Learning models. These modalities may encompass, among others, standard structured data, sequential structured data, free text, audio, or images. Consequently, Multimodal Learning is described as the methodologies aimed at

developing Machine Learning models capable of processing these heterogeneous data types cohesively (Baltrušaitis et al., 2018).

Approaches

Within the domain of Multimodal Learning, we can identify three primary approaches (Ramachandram & Taylor, 2017):

- **Early fusion:** this approach integrates various data modalities, sometimes highly disparate, into a single feature vector before using it as input for the Machine Learning model. However, early fusion of multimodal data might not completely leverage the complementary nature of the involved modalities and could result in overly large input vectors with potential redundancies.
- **Late fusion:** this approach involves aggregating decisions from various Machine Learning models, each trained on different modalities. A significant limitation of late fusion is that it may fail to capture the interactions between modalities.
- **Intermediate fusion:** this strategy combines information from different modalities at some point within the processing pipeline, rather than strictly at the beginning or end. It allows modalities to interact at multiple levels of abstraction prior to making a final prediction or decision. Intermediate fusion is especially beneficial for tasks where the modalities' relationship is complex and cannot be fully understood by processing them separately or combining them solely at the input or output stages, thus enabling a more profound integration of modalities.

2.2.5 Multitask Learning

Definition

According to the definition of (Caruana, 1997), Multitask Learning can be defined as *an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation*; what is learned for each task can help other tasks be learned better. In other words, Multitask Learning is a Machine Learning approach where a single model is trained to perform multiple tasks—which are assumed to be related—simultaneously. In traditional Machine Learning, different models are trained for each task separately. However, in Multitask Learning, a shared representation is learned that captures commonalities and differences between the tasks. By doing this, model performance can be improved while recurring to an inferior number of parameters—hence being more efficient—including regularization effects during training.

Mathematically, Multitask Learning can be defined this way, following the definition of (Ciliberto et al., 2015):

Let us assume to have T supervised scalar learning problems, each with a training set of input-output observations $S_t = \{(x_{it}, y_{it})\}_{i=1}^{n_t}$, with $x_{it} \in X$ input space and $y_{it} \in Y$ output space. Given a loss function $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ that measures the per-task prediction errors, we aim to solve the next joint regularized learning problem:

$$\min_{f \in H} \sum_{t=1}^{T} \frac{1}{n_t} \sum_{i=1}^{n_t} L(y_i^{(t)}, f_t(x_i^{(t)})) + \lambda \|f\|_H^2 \tag{2.11}$$

being H a Hilbert space of vector-valued functions $f : X \to Y^T$ with scalar components $f_t : X \to Y$, λ a scalar parameter controlling the regularization strength and $\|f\|_H^2$ is the regularization term, the norm of the model f in the Hilbert space H.

Approaches

Even though there are many approaches to develop Multitask Machine Learning models, we present next the two main ones (Ruder, 2017b), which are also related to the models developed in this thesis:

Hard parameter sharing

In Multitask Learning, hard parameter sharing refers to the approach where multiple tasks share the same set of parameters, i.e., weights, in a model. This means that the model has a single set of weights that are used for all tasks, rather than having separate sets of weights for each task. This can be useful when the tasks are closely related and have similar feature representations, as it allows the model to learn shared features that can be used to improve performance on all tasks.

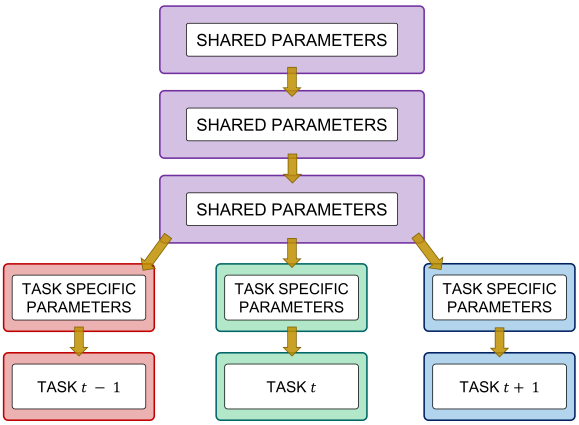

Figure 2.2: Schematic representation of the hard parameter sharing approach in Multitask Learning.

Soft parameter sharing

Alternatively, soft parameter sharing refers to the approach where multiple tasks have different sets of parameters, but the parameters are constrained to have some degree of similarity. This can be achieved by adding a regularization term to the overall objective function that encourages the parameters of different tasks to be similar.

In comparison to other multitask approaches, soft parameter sharing is generally considered when the tasks are less closely related but still have some shared features that can be used to improve performance. It is also useful when there is limited data for each task and the model can leverage the shared information among the tasks to improve generalization.

2.2.6 Meta-learning

Definition

Meta-learning is a subfield of Machine Learning that involves the development of algorithms that can learn how to learn. Specifically, it refers to the process of incorporating metadata from experiments to improve model performance in the next round of experiments (Vilalta & Drissi, 2002).

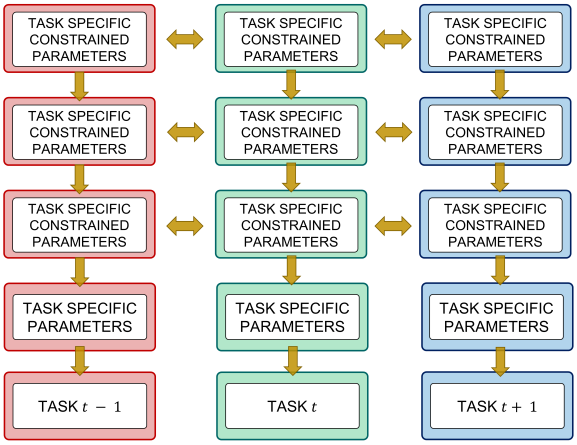

Figure 2.3: Schematic representation of the soft parameter sharing approach in Multitask Learning.

In the context of hyperparameter tuning, Meta-learning can be used to learn how to optimize hyperparameters more efficiently. Specifically, instead of performing a grid search or random search (Bergstra & Bengio, 2012) over a large set of hyperparameters, we can use Meta-learning to learn a model that can quickly adapt and suggest good hyperparameters.

Given that it has been used across all our studies involving model development, we present next Bayesian Hyperparameter Optimization (BHO) (Brochu et al., 2010), our Meta-learning approach chosen in this thesis.

Bayesian Hyperparameter Optimization

Bayesian Hyperparameter Optimization (BHO) is a Meta-learning method used to find the best set of hyperparameters of a Machine Learning model. The main idea behind this method is to model the Machine Learning model's performance as a function of its hyperparameters and use Bayesian statistics to infer the most likely set of hyperparameters that will lead to good performance.

BHO defines a prior probability distribution over the space of possible hyperparameters. This prior captures our initial belief about the likely values of the hyperparameters before any data is observed. As we gather more data, particularly after subsequent experiments, we refine our beliefs concerning the hyperparameters through the application of Bayes' rule. This updating process involves adopting a likelihood function that quantifies the likelihood of the observed data given a specific

set of hyperparameters. The posterior distribution is then defined as the product of this likelihood and the prior, normalized by the evidence.

Mathematically, the BHO can be defined as:

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} \tag{2.12}$$

where θ is the set of hyperparameters, D is the data, $p(\theta)$ is the prior, $p(D|\theta)$ is the likelihood function, and $p(D)$ is the evidence.

Once the posterior is obtained, it can be used to sample or optimize the hyperparameters. A common approach is to sample from the posterior considering Markov Chain Monte Carlo (MCMC) methods, such as the Metropolis-Hastings algorithm (Chib & Greenberg, 1995).

Finally, once the optimal hyperparameters are obtained, the model is retrained with these hyperparameters and the performance is tested to evaluate the model's performance.

2.2.7 Dataset shifts

Dataset shifts can be defined as changes in the data distribution between training and test data (Moreno-Torres et al., 2012; Quinonero-Candela et al., 2008). They can be caused by many factors, such as selection bias—the training dataset is not representative of the whole population—or variations in how the data is collected over time, or when testing on a new setting (such as a new location)—some variables disappear while new ones are created, for example.

Likewise, it must be considered that dataset shifts can appear suddenly, in the form of an abrupt drift, or conversely, happen in a more gradual manner. Furthermore, it is even possible that they manifest in following a recurrent pattern due to changes that present a seasonal component (Gama et al., 2014; Sáez et al., 2015).

Temporal variations in healthcare processes or protocols are inherent to the field of medicine. Such fluctuations can potentially give rise to dataset shifts, representing a data quality challenge when repurposing Electronic Health Records for secondary applications (Sáez et al., 2020).

Formally, in the context of a classification problem where we have some input features x, and an output variable y, a dataset shift occurs when training and test joint probability distributions differ:

$$P_{train}(x, y) \neq P_{test}(x, y) \tag{2.13}$$

This variation in the joint probability distribution can be related to multiple sources of drift. Although depending on the authors it is possible to find different definitions and nomenclature—notable contributions in this field are (Kull & Flach, 2014; Moreno-Torres et al., 2012; Quinonero-Candela et al., 2008; Storkey et al., 2009)—in this rationale and for the rest of the thesis, we consider three main sources of drift: prior probability shift, covariate shift and concept shifts. Next, we present a brief description of each type of shift as we understand them in this thesis, including the mathematical definition:

Prior probability shift

Prior probability shift happens when training and testing distributions of the output variable differ. It can be described this way:

$$P_{train}(y) \neq P_{test}(y) \tag{2.14}$$

Covariate shift

Covariate shift can be defined as the change in the input features distribution, that is:

$$P_{train}(x) \neq P_{test}(x) \tag{2.15}$$

Concept shift

Concept shift occurs when the conditional probability of the outcome with respect to input features suffers a variation between sets:

$$P_{train}(y|x) \neq P_{test}(y|x) \tag{2.16}$$

2.2.8 Continual learning

Continual learning

Continual Learning, also known as lifelong learning, refers to the ability of a Machine Learning model to learn new tasks or adapt to new data distributions without forgetting relevant learned knowledge (Parisi et al., 2019). This is a challenging problem, especially in the Deep Learning context, as ANN are known to suffer from *catastrophic forgetting* (McCloskey & Cohen, 1989), where the performance on previous experiences (or tasks) degrades when the model is trained on new experiences (or tasks) (Lomonaco, 2019).

We clarify here that, within the context of Continual Learning, an "experience" can be understood as a learning episode, constituted by a chunk of data. Within a Continual Learning problem, the model encounters multiple of such learning episodes, being the objective to retain knowledge from prior data chunks while acquiring knowledge from new ones. Similarly, a "task" refers to the specific learning objective associated with an experience, such as the classification of a subset of predefined classes.

Domain incremental learning

There are multiple Continual Learning scenarios, but given the topic of this thesis, we are going to focus on Domain Incremental Learning (Ven & Tolias, 2019):

DIL is a specific Continual Learning scenario where the model is trained to adapt to new domains, or environments, while preserving the knowledge acquired on previous ones. The main challenge of DIL is to learn representations that are robust to changes in the domain, while preserving the knowledge acquired on previous domains. This requires the model to learn domain-invariant features that are shared across domains, and domain-specific features (Ven & Tolias, 2019).

Hence, the DIL problem does not consist of learning new task over time. Instead, it is about to learn information from new experiences, carrying out the same task, preserving that information that is relevant.

A common approach to DIL is to use domain adaptation techniques, which aim to adapt a model trained on one domain to a new domain. These techniques typically involve adjusting the model's parameters to align the distributions of the source—environment where the model was designed and trained—and target domains—where the model is intended to be applied or deployed.

Strategies

There is a wide range of Continual Learning techniques to choose from. Next, we present the ones that are the most relevant in this thesis:

- **Fine-tuning:** at a particular experience e, the starting point from training is the model from the previous experience $e-1$. The weights are updated using data from the current experience, initialized with the values of the previous one.
- **Cumulative:** at a specific experience, the model is trained with data from all the previous experiences. The usual starting point from training is the model from the previous experience.
- **Replay:** at a particular experience, it trains the model with data from the previous experiences, in a similar manner than the Cumulative approach. However, instead of considering all the data, it just takes a sample of it to increase efficiency in terms of computation time and memory.
- **Synaptic intelligence:** it is a regularization technique that encourages a model to retain relevant parameter values for previously learned experiences. It involves computing an experience-specific importance measure for each weight. Then, a regularization term is added to the loss function that penalizes changes to important weights. Hence, it eases the updating of weights that are not important for previous experiences—offering plasticity—while keeping weight values that are relevant in previous experiences—showing resistance to catastrophic forgetting (Zenke et al., 2017). Formally, the loss function L to optimize at experience e presents the following structure:

$$L_e = H_e + c \sum_{k=1}^{K} \Omega_k^e (\tilde{\theta}_k - \theta_k)^2 \tag{2.17}$$

  Here, H_e represents the standard loss to minimize at experience e, c is a global dimensionless weighting parameter, Ω_k^e is the per-parameter (applied individually to each parameter of the model) regularization strength for parameter k and experience e, $\tilde{\theta}_k$ denotes the value of parameter k at the previous experience, and θ_k represents the value of parameter k at the current learning experience.

2.2.9 Machine Learning framework summary

Next, Figure 2.4 provides a summarized overview of the Machine Learning framework adopted in this thesis, building upon the topics discussed in previous sections. This is intended to offer a clearer, more holistic view of the methodological approach utilized. While each specific development in this work possesses unique characteristics, a common structural theme is observed. It is important to note, however, that despite these commonalities, each subproblem tackled in the different chapters of this thesis employs a distinct methodological approach. Although they share certain aspects with the framework outlined in Figure 2.4, in several instances, the methodologies are not entirely identical.

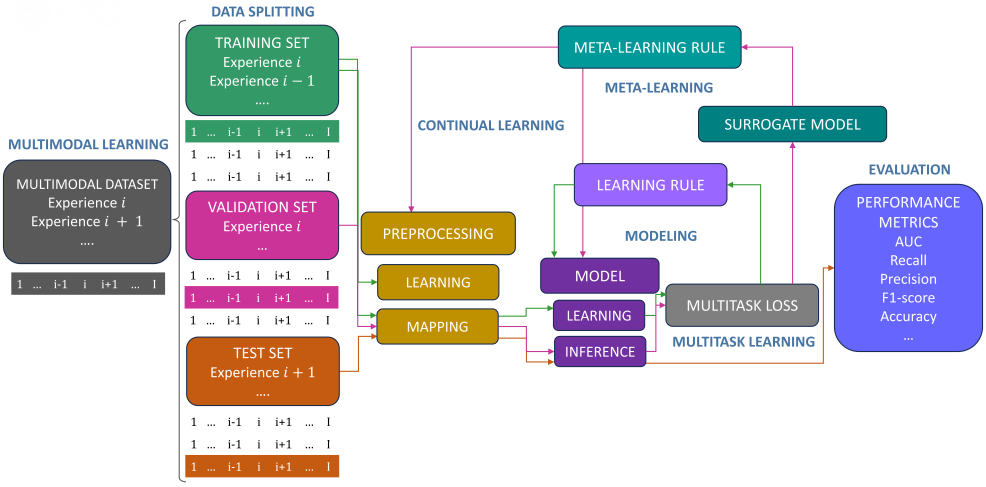

Figure 2.4: Machine Learning framework summary.

Initially, we work with a Multimodal dataset derived from merging various tables from the Valencian Region's emergency medical call incidents database. As evident, proper data handling is pivotal in our framework. This data is segmented into different learning experiences within the Continual Learning framework and further divided into training, validation, and test sets. In the modeling phase, we employ a Machine Learning model to predict incident severity, which, in our thesis, is structured on Deep Learning principles. Given the multidimensional aspect of severity, a Multitask approach is also adopted. Furthermore, considering the complexity of the models and strategies, Meta-learning algorithms are utilized, adhering to a Bayesian approach. Here, auxiliary models iteratively learn the relationship between the multitask loss function and the hyperparameters. These hyperparameters are relevant not only in the modeling stage but also in the preprocessing phase and within the

Continual Learning dimension. Finally, after thorough training and validation, the predictive model is tested on independent test sets using multiple performance metrics.

2.3 Deep Learning

2.3.1 Background and definitions

Definition

Deep Learning can be defined as a Machine Learning subfield focused on building ANN with multiple layers, called *deep* neural networks. These networks are designed to automatically learn representations of the input data, in an end-to-end fashion, that is, without requiring previously hand-crafted feature extraction (LeCun et al., 2015).

A Deep Learning model is typically composed of multiple layers of artificial neurons each with a set of weights that are learned during training. The layers closest to the network's input interact directly with the input features, and they are referred to as the *input layers*, whereas the layers near the network's output generate the model's predictions, and these are known as the *output layers*. Layers situated between them are referred to as *hidden layers*. The input to the model is passed through the layers, and each neuron applies a non-linear transformation to the input, based on its weights. The output of each neuron is then passed as input to the next layer (Goodfellow et al., 2016).

Activation functions

In the context of Deep Learning, an activation function is a non-linear function applied to the output of a neuron. Its purpose is to introduce non-linearity in the model and to allow the neural network to learn more complex functions. In formal terms, an activation function f takes an input z and produces an output a according to the equation:

$$a = f(z) \tag{2.18}$$

The choice of the activation function depends on the specific problem and the nature of the input data. There are several important activation functions used in Deep Learning. The most relevant ones with a notable presence in our developments are the Rectified Linear Unit (ReLU), the Leaky Rectified Linear Unit (Leaky ReLU),

the Gaussian Error Linear Unit (GELU), and the Softmax (Nwankpa et al., 2018), which we present next in more detail.

Rectified Linear Unit (ReLU)

The ReLU activation function is computationally efficient and computationally inexpensive to compute the derivative. It helps alleviate the vanishing gradient problem by allowing gradients to flow through for positive inputs. In addition, it introduces sparsity in the neural network, which can improve generalization.

Mathematically, it can be described as follows:

$$f(z) = max(0, z) \tag{2.19}$$

Next, a graph representing the ReLU activation function is displayed in Figure 2.5.

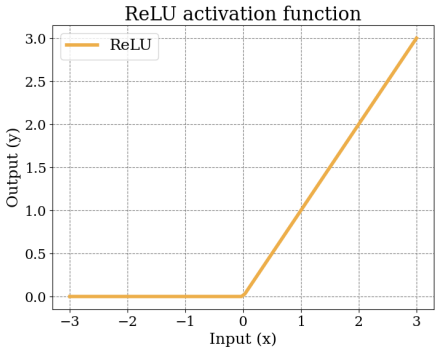

Figure 2.5: Rectified Linear Unit (ReLU) activation function.

Leaky Rectified Linear Unit (Leaky ReLU)

Leaky ReLU addresses the dying ReLU problem by allowing a small gradient for negative inputs, preventing neurons from becoming inactive. It maintains some non-linearity while preventing vanishing gradients.

The mathematical expression for it is:

$$f(z) = max(0, z) + \alpha \cdot min(0, z) \tag{2.20}$$

Here, α is a small positive scalar value controlling the angle of the negative slope, i.e., the leakiness of the function.

Next, a graph illustrating the Leaky ReLU activation function is shown in Figure 2.6.

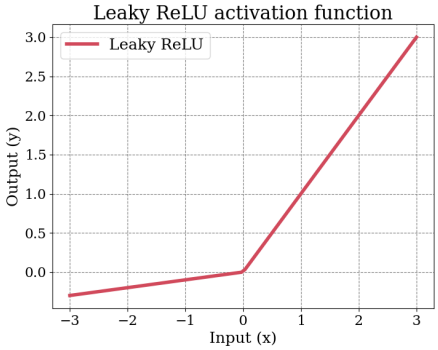

Figure 2.6: Leaky Rectified Linear Unit (ReLU) activation function.

Gaussian Error Linear Unit (GELU)

GELU is an activation function designed to capture a more Gaussian-like non-linearity, which can benefit certain types of data. It avoids the vanishing gradient problem, is computationally efficient, and has been found to perform well in deep neural networks.

Mathematically, it can be described as follows:

$$f(z) = z \cdot \Phi(z) \tag{2.21}$$

where $\Phi(z)$ is the Cumulative Distribution Function for Gaussian Distribution.

A graph depicting the GELU activation function is shown in Figure 2.7.

Softmax

The Softmax activation function is commonly used in the output layer of neural networks for multiclass classification problems. It transforms a vector of arbitrary values into a probability distribution over K classes, where K is the number of classes. The exponential function amplifies the differences between input values, highlighting the class with the highest score.

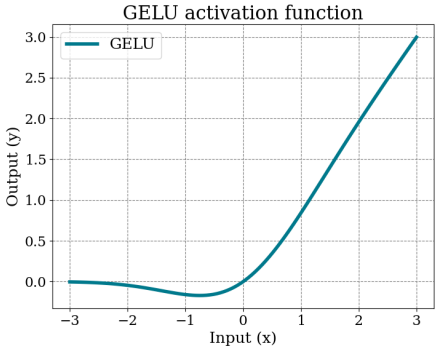

Figure 2.7: Gaussian Error Linear Unit (GELU) activation function.

The mathematical expression for it is:

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.22}$$

Loss functions

A loss function in Deep Learning is a function that measures the difference or dissimilarity between the predicted output of a model y_{pred} and the actual output y_{true}. The main purpose of a loss function is to measure how well the model is performing in relation to the task it is trying to learn (Janocha & Czarnecki, 2017).

The mathematical foundation of a loss function is based on the idea of minimizing the difference between the predicted output and the actual output. The loss function is a scalar value that quantifies this difference. The value of the loss function is used to update the parameters of the model during the training process, with the goal of minimizing the loss function over time.

Hence, a loss function is a mapping following this structure:

$$L(y_{true}, y_{pred}) \rightarrow \mathbb{R} \tag{2.23}$$

To avoid overfitting issues, some penalty terms—also named regularization terms—are usually added to the loss function. Although there is a huge variety of them, the most relevant ones are the L1 and L2 regularization.

L1 regularization

L1 regularization adds a penalty term to the loss function that is proportional to the absolute values of the model's weights. It encourages the model to have sparse weights by pushing some of them to be exactly zero. Mathematically, the L1 regularization term is defined as:

$$L_{L1} = \lambda \sum_{i=1}^{n} |w_i| \tag{2.24}$$

Where λ is the regularization strength, w_i represents the model's weight parameters and n is the total number of model parameters.

L2 regularization

L2 regularization adds a penalty term to the loss function that is proportional to the squared values of the model's weights. It encourages the model's weights to be small but doesn't force them to be exactly zero. Mathematically, the L2 regularization term is defined as:

$$L_{L2} = \lambda \sum_{i=1}^{n} \|w_i\|_2^2 \tag{2.25}$$

Where λ is the regularization strength, w_i represents the model's weight parameters and n is the total number of model parameters.

Dropout

A dropout layer (Hinton et al., 2012) in Deep Learning is a specialized type of layer that randomly deactivates, or sets to zero, a certain proportion of the input units during the forward pass of the neural network. The primary objective of a dropout layer is to counteract neuron co-adaptation, a phenomenon wherein neurons within the network become overly reliant on one another during training. When neurons co-adapt, they tend to depend on specific neighboring neurons to compensate for their individual weaknesses or to make accurate predictions.

Dropout induces diversity among the neurons in the network, encouraging them to learn a more robust and varied set of features. This, in turn, enhances the model's ability to generalize well during inference. Additionally, as noise is introduced into the input units, dropout contributes to preventing overfitting.

Normalization layers

Normalization layers are frequently used in the context of Deep Learning to reduce the internal covariate shift (Ioffe & Szegedy, 2015). Without normalization, the distribution of the inputs to a layer can change as the parameters of the network are updated, which can make training more difficult. By normalizing the inputs, the distribution remains more stable, which allows the network to learn faster and with better stability.

It's worth noting that normalization layers can also be beneficial for the final performance of the model, as it can make it more robust to different types of input and hence, acting as a regularizer.

The most common types of normalization layers are batch normalization and layer normalization. Batch normalization (Ioffe & Szegedy, 2015) normalizes the activations of a layer by subtracting the batch mean and dividing by the batch standard deviation, while layer normalization (Ba et al., 2016) normalizes the activations of a layer by subtracting the layer mean and dividing by the layer standard deviation.

Embedding layers

In the Deep Learning context, an embedding layer is a type of layer that is used to map discrete input data, such as words, characters, or categories, to a continuous vector space (Bengio et al., 2000). The main purpose of an embedding layer is to represent the input data in a way that can be easily computed by a neural network.

The mathematical foundation of an embedding layer is based on the idea of a vector space representation. In this representation, each discrete input data is

mapped to a unique vector, called an embedding vector. Embedding vectors are typically learned during the training process of the neural network.

The embedding vectors are typically represented by a matrix, where each row corresponds to an embedding vector for a specific input data. This matrix is often called an embedding matrix or a weight matrix.

For example, suppose you have a text dataset, and you want to use it to train a neural network for sentiment analysis. Asuming a simple tokenization step, each word in your dataset could be represented as a unique index, where each index corresponds to a word in your vocabulary. For example, in the sentence *I am passionate about Deep Learning*, "I" could be represented as 1, "am" as 2, "passionate" as 3, "about" as 4, "Deep" as 5 and "Learning" as 6.

In the embedding layer, you specify the size of the embedding vectors. For example, you might decide to use 50-dimensional vectors. The embedding layer initializes these vectors randomly and then learns to adjust them during training. Hence, each unique word index corresponds to a unique embedding vector: Index 1 (word "I") might correspond to the embedding vector [0.32, -0.23, 0.54, ...] in 50-dimensional space, index 2 (word "am") might correspond to the embedding vector [-0.47, 0.32, -0.21, ...] and so forth.

2.3.2 Parameter tuning

In a deep neural network, parameters, i.e., weights, are tuned, i.e., updated, following an iterative numerical algorithm based on gradient descent (Wright, 2006), where the values of the gradients are calculated following the backpropagation algorithm (Hecht-Nielsen, 1989). Next, we extend both concepts.

Gradient descent

Gradient descent is an optimization algorithm commonly used to train Machine Learning models, especially deep neural networks. The goal of the algorithm is to find the values of the model's parameters that minimize a given loss function, which measures the discrepancy between the model's predictions and the true values.

The basic idea behind gradient descent is to iteratively adjust the parameters in the direction of the negative gradient of the loss function with respect to the parameters. The negative gradient points in the direction of the steepest descent of the loss function, which is the direction that decreases the loss the most.

The most common form of gradient descent is called *batch gradient descent*, which uses the entire dataset to compute the gradient at each iteration. The update rule for the parameters at iteration t is given by:

$$\theta_{t+1} = \theta_t - \alpha \nabla L(\theta_t) \tag{2.26}$$

where θ_t is the value of the parameters at iteration t, $\nabla L(\theta_t)$ is the gradient of the loss function with respect to the parameters at iteration t, and α is the learning rate, which controls the step size of the update.

However, in practice, other variants of gradient descent are used, which consider a group of samples or mini-batch in each iteration (Bertsekas, 1994). Next, we present the most relevant numerical optimization procedures for Deep Learning, considered in this thesis.

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a variant of the gradient descent algorithm that is particularly useful for large-scale Machine Learning problems, where the dataset is too large to fit in memory or it takes too long to compute the gradient using the entire dataset (Bottou, 1998).

The basic idea behind SGD is to estimate the gradient of the loss function using a small, randomly selected subset of the data, called a *batch* or *mini-batch*, at each iteration. The update rule for the parameters at iteration t is given by:

$$\theta_{t+1} = \theta_t - \alpha \nabla L(\theta_t, x_i) \tag{2.27}$$

Here θ_t is the value of the parameters at iteration t, $\nabla L(\theta_t, x_i)$ is the gradient of the loss function with respect to the parameters at iteration t, computed using a random sample x_i from the dataset, and α is the learning rate, which controls the step size of the update.

One of the main advantages of SGD over batch gradient descent is that it can start making updates to the parameters right away, without having to wait for the entire dataset to be processed. This makes SGD well suited for online and streaming learning scenarios.

Another advantage of SGD is that it often results in a more robust optimization scheme, since the estimate of the gradient is less affected by specific samples in the dataset.

However, the main disadvantage is that the SGD updates are noisier since it is using a small subset of the data to update the parameters, and this can cause the algorithm to oscillate or converge to suboptimal solutions. To overcome this, people usually reduce the learning rate over time, or use techniques like momentum or adaptive learning rate optimization.

It is also worth noting that, in practice, the batches used in SGD are not truly random. They are usually selected in a cyclic manner, so that the algorithm *sees* all the examples multiple times. This is called *epoch training* where one epoch is one pass over the full dataset.

Adaptive Moment Estimation (Adam)

Adam (Adaptive Moment Estimation) (Kingma & Ba, 2017) is an optimization algorithm that is used to update the parameters of a neural network. It is a combination of two optimization techniques, SGD and Root Mean Square Propagation (RMSProp) (Sun et al., 2019).

The Adam algorithm updates the parameters using the following equations:

$$\begin{aligned} m_{t+1} &= \beta_1 m_t + (1-\beta_1)\nabla L(\theta_t, x_B) \\ \upsilon_{t+1} &= \beta_2 \upsilon_t + (1-\beta_2)\nabla L(\theta_t, x_B)^2 \\ \hat{m}_{t+1} &= \frac{m_{t+1}}{1-\beta_1^{(t)}} \\ \hat{\upsilon}_{t+1} &= \frac{\upsilon_{t+1}}{1-\beta_2^{(t)}} \\ \theta_{t+1} &= \theta_t - \alpha \frac{\hat{m}_{t+1}}{\sqrt{\hat{\upsilon}_{t+1}} + \epsilon} \end{aligned} \tag{2.28}$$

where m_{t+1} and υ_{t+1} are the first and second-moment estimates of the gradient, respectively, β_1 and β_2 are hyperparameters that control the decay rate of the moment estimates and ϵ is a small value added to the denominator to prevent division by zero. $\hat{m}_{t+1}$ and $\hat{\upsilon}_{t+1}$ are the unbiased estimates of the first and second moments. The parameter update step is similar to that of the standard SGD, with the addition of the scaling factor $\frac{\hat{m}_{t+1}}{\sqrt{\hat{\upsilon}_{t+1}}+\epsilon}$ which adapts the learning rate based on the historical gradient information.

Adam is generally considered to be a good optimization algorithm for many Deep Learning tasks, as it has been shown to converge quickly and perform well in practice (Sun et al., 2019).

Adaptive Moment Estimation with Weight Decay (AdamW)

AdamW (Loshchilov & Hutter, 2019) is an optimization algorithm for neural networks that combines the Adam algorithm with weight decay regularization (Krogh & Hertz, 1991). The AdamW algorithm introduces weight decay regularization to the Adam algorithm by adding a term to the gradients that penalizes large weight values. The regularization term is defined as the L2 norm of the weights multiplied by a weight decay factor. Hence, the overall update rule for the AdamW algorithm can be expressed this way:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_{t+1} + w\theta_t}{\sqrt{\hat{v}_{t+1}} + \epsilon} \tag{2.29}$$

where w is the weight decay factor.

The AdamW algorithm is similar to the Adam algorithm, but it is less prone to overfitting, as it encourages the weights to take smaller values.

Backpropagation

In the previous subsection we have explained how weights are iteratively updated with gradient descent-based numerical optimization algorithms. Here we briefly expose the backpropagation algorithm (Hecht-Nielsen, 1989), used to calculate the value of the gradients required to carry out the gradient descent updating step.

Backpropagation is an algorithm used to calculate gradients in neural networks. The algorithm works by propagating the error back through the network, starting with the output layer and working its way back to the input layer.

The mathematical notation for backpropagation can be quite complex, but the basic idea is to calculate the gradient of the loss function with respect to each weight in the network. This is done using the chain rule of calculus, which states that the derivative of a composite function can be computed by taking the derivative of each function in the composite and multiplying them together.

Assuming we have a loss function L that measures the discrepancy between the predicted outputs of the network and the true labels, backpropagation allows us

to compute the gradients of L with respect to the weights and biases of the network, as expressed in the following equation:

$$\begin{aligned}\frac{\partial L}{\partial w_{ij}^{[l]}} &= \frac{\partial L}{\partial h_j^{[l]}} \frac{\partial h_j^{[l]}}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial w_{ij}^{[l]}} \\ \frac{\partial L}{\partial b_j^{[l]}} &= \frac{\partial L}{\partial h_j^{[l]}} \frac{\partial h_j^{[l]}}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial b_j^{[l]}}\end{aligned} \tag{2.30}$$

where $w_{ij}^{[l]}$ denotes the weight that connects the neuron i from the previous layer $l-1$ with the neuron j from the current layer l, $b_j^{[l]}$ is the bias associated with neuron j from layer l, $h_j^{[l]}$ is the value of the activation of neuron j of layer l, and $z_j^{[l]}$ is the input to the activation function of neuron j of layer l.

2.3.3 Feed-forward neural networks

Feed-forward neural networks, commonly found in the form of Multilayer Perceptrons (MLP) (Rosenblatt, 1958), are a type of neural network that is widely used in Deep Learning. They are called *feed-forward* because the information flows through the network in one direction, from the input layer to the output layer, without any feedback loops.

Mathematically, a feed-forward neural network can be represented as a function that maps an input vector x to an output vector y, using a series of nonlinear transformations applied to the input. The input vector x is fed into the first layer of the network, i.e., the input layer. This input layer consists of a set of neurons, each of which corresponds to one element of the input vector.

Each neuron in the input layer is connected to one or more neurons in the next layer, called the hidden layer. The hidden layer consists of a set of neurons that apply a nonlinear transformation to the outputs of the neurons in the previous layer. The outputs of the neurons in the hidden layer are then fed into the next layer, and so on, until the output layer is reached.

We present next a more formal description of an MLP, the most common feed-forward neural network:

Let us consider an MLP with L layers, including the input layer, hidden layers, and the output layer. Each layer has a certain number of neurons. We denote the number of neurons in layer l as $N(l)$, where l ranges from 1 to L.

The activation of neuron i in layer l is denoted as $h_i(l)$, and the weighted sum of inputs to neuron i in layer l is denoted as $z_i(l)$. The weights connecting neuron i in layer $l-1$ to neuron j in layer l are denoted as $w_{ij}(l)$, and the bias term of neuron i in layer l is denoted as $b_i(l)$. Finally, the activation function used in each neuron is denoted as σ.

The equations describing the forward propagation of MLP at a given layer l are:

$$\begin{aligned} z_j^{[l]} &= \sum_{n=1}^{N(l)} w_{ij}^{[l]} h_i^{[l-1]} + b_j^{[l]} \\ h_j^{[l]} &= \sigma(z_j^{[l]}) \end{aligned} \tag{2.31}$$

Next, in Figure 2.8, we provide a schematic illustration depicting the structure of an ANN model.

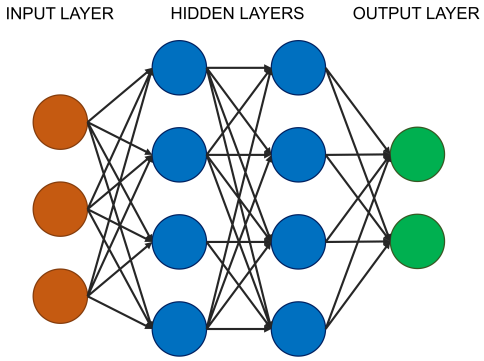

Figure 2.8: Schematic representation of a feed-forward neural network.

2.3.4 Recurrent neural networks (RNN)

Definition

Recurrent Neural Networks (RNNs) are a type of neural network designed to process sequential data, such as time series or natural language. They are called recurrent because they perform the same computation for every element in a sequence, with the output of one step being used as input for the next step.

The main architectures of RNNs are the simple RNN (Amari, 1972), the Long Short-Term Memory (LSTM) network (Hochreiter & Schmidhuber, 1997), and the Gated Recurrent Unit (GRU) network (Cho et al., 2014).

A simple RNN consists of a single layer of neurons that have a memory in the form of a hidden state h_t, which is passed from one step of the sequence to the next. The hidden state is updated at each step using the current input x_t and the previous hidden state h_{t-1}, according to the following equations:

$$\begin{aligned} h_t &= \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \\ y_t &= \sigma_y(W_y h_t + b_y) \end{aligned} \tag{2.32}$$

where h_t is the hidden state at time step t, x_t is the input at time step t, y_t is the output at time step t, σ_h is an activation function, σ_y is an output activation function, W_h, U_h, W_y, b_h, and b_y are the weights and biases of the network.

Next, a representation of the RNN model architecture is depicted in Figure 2.9:

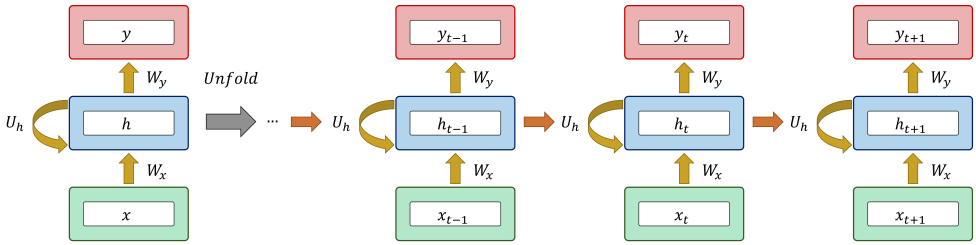

Figure 2.9: Recurrent Neural Network architecture.

LSTM and GRU are more complex variants of RNNs that are designed to overcome the problem of vanishing gradients ("Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies", 2009), which is a common problem in simple RNNs when processing long sequences. These architectures introduce additional states and gates that control the flow of information through the network, allowing it to retain or forget information from the past selectively. We emphasize here that within the LSTM context, a state represents a continuous vector containing specific information, whereas a gate serves as a mechanism for regulating the flow of information through the network by modifying the aforementioned state vectors.

The main reason to use RNNs is that they are able to process sequential data, which means that the input and output of the network are ordered. This allows RNNs to learn patterns in the data that depend on the order of the elements.

Long Short-Term Memory (LSTM) networks

Long Short-Term Memory (LSTM) networks are a type of RNN that are designed to overcome the problem of vanishing gradients, which is a common problem in simple RNNs when processing long sequences. Specifically, the problem of vanishing gradients is a challenge that occurs during training, specifically when backpropagating gradients through time. It refers to the issue where gradients become extremely small as they are propagated backward through the recurrent connections, making it difficult for the network to learn long-range dependencies in sequential data. This can result in the network struggling to capture and retain information from distant time steps, which is a critical limitation in tasks requiring such context.

LSTM networks introduce additional gates and states that control the flow of information through the network, allowing it to retain or forget information from the past selectively. An LSTM network consists of a series of LSTM cells, each containing three gates: an input gate, an output gate, and a forget gate. These gates are used to control the flow of information into and out of the cell's internal state, called the *memory* cell. The following equations define the gates:

$$\begin{aligned} i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\ f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \end{aligned} \tag{2.33}$$

where i_t, f_t and o_t are the input gate, forget gate, and output gate respectively, x_t is the input at time step t, h_{t-1} is the hidden state at time step $t-1$, W_i, U_i, W_f, U_f, W_o, U_o, b_i, b_f and b_o are the weights and biases of the network, and σ_g is the sigmoid activation function.

The main advantage of LSTM networks over standard RNNs is that they are able to selectively retain or forget information from the past, which allows them to effectively process long sequences without suffering from the problem of vanishing gradients. This makes LSTM networks well-suited to tasks such as language modeling, speech recognition, and time series forecasting.

2.3.5 Transformers

Definition

The Transformer is a Deep Learning model architecture introduced in (Vaswani et al., 2017). It is mainly considered for Natural Language Processing (NLP) tasks, such as language translation, text summarization and text generation.

The Transformer model architecture is based on the concept of self-attention, which allows the model to weigh the importance of different words in the input sequence when making a prediction. This contrasts with traditional RNN architectures, such as the LSTM or the GRU which use recurrence to weigh the importance of previous words in the input sequence.

The Transformer model consists of an encoder and a decoder, structures made up of multiple layers of self-attention and feed-forward neural networks. The encoder takes in a sequence of words, such as a sentence, and generates a set of hidden states that represent the meaning of the input sequence. The decoder then takes in the hidden states from the encoder and generates a new sequence of words, such as a translation of the input sentence. A representation of the encoder-decorder architecture of the Transformer model can be found in Figure 2.10:

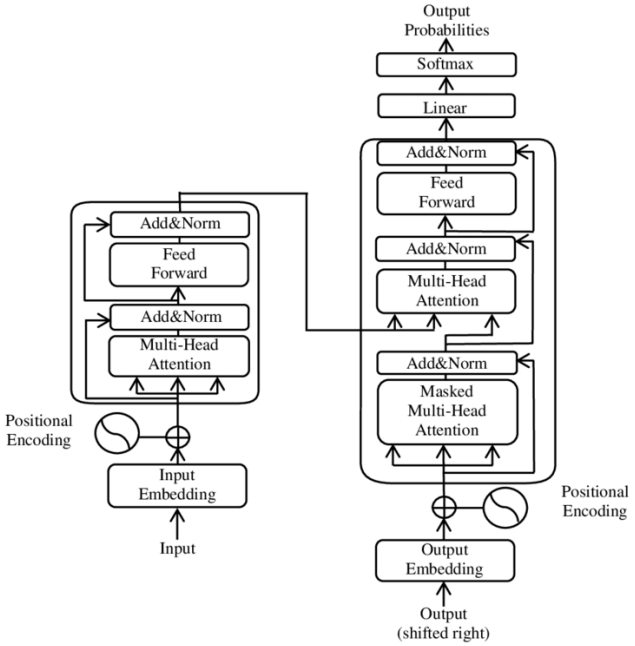

Figure 2.10: The Transformer architecture. Extracted from (Jia, 2019).

One of the main advantages of the Transformer architecture over RNN-based architectures is its ability to parallelize the computations. This allows the model to process an entire sequence at once, rather than processing it one word at a time as in RNNs. This results in faster training and inference times, which are particularly important for large-scale NLP tasks.

Another advantage of the transformer model is that it does not suffer from the vanishing gradient problem, which is a common issue with RNNs. The self-attention mechanism in the transformer allows the model to weigh the importance of words in the input sequence, regardless of their position, which makes it more robust to long-term dependencies.

Transformer architecture has implied a breakthrough in the field of Deep Learning. In fact, transformer-based models have gone beyond state-of-the-art results in various NLP tasks and have become the de facto standard in most NLP applications.

Next, we delve into a more detailed explanation of the self-attention mechanism, which forms the core of the Transformer architecture. Additionally, we explore the multi-head attention mechanism, which amplifies self-attention capabilities. Furthermore, we discuss positional encoding, a technique that enables computation parallelization while maintaining a sense of sequence ordering within the input.

Self-attention

Self-attention is a mechanism that enables a model to focus on various parts of the input sequence, assigning different levels of importance to different elements when making predictions. To clarify this concept, we introduce the query Q, key K, and value V matrices. These matrices are learnable parameters of the self-attention mechanism, and they encode various aspects of the input sequence:

Query (Q): The Query matrix is used to represent the elements in the sequence that we are currently trying to weight or attend to.

Key (K): The Key matrix is used to represent the elements in the sequence against which we want to compare the Query elements.

Value (V): The Value matrix is used to represent the content or information associated with each element in the sequence.

The self-attention mechanism can be mathematically represented as follows:

Given an input sequence of length T and a set of query, key and value matrices, Q, K, and V, respectively, the attention weights, A, are computed as the dot product of the query Q and the key K matrix, scaled by the square root of the dimension of the key K matrix and then passed through a softmax function.

The attention weights, A, for each position i in the input sequence are computed as:

$$A = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{2.34}$$

where d_k is the dimension of the key matrix.

The context vector, C, is then computed as the dot product of the attention weights A and the value matrix:

$$C = AV \tag{2.35}$$

Finally, the output of the self-attention layer is computed as the concatenation of the context vectors for all positions in the input sequence:

$$O = [C_1, ..., C_T] \tag{2.36}$$

Multi-head attention

Multi-head attention is an extension of the self-attention mechanism that allows the model to attend to different parts of the input sequence simultaneously. This is done by performing multiple self-attention operations, each with different weight matrices, and concatenating the resulting attention outputs.

The mathematical foundation of multi-head attention is similar to the self-attention, with the main difference being that multiple queries, keys, and values are created and multiple dot product attention are performed using different weight matrices. The output of each attention head is concatenated and passed through a final linear layer to obtain the final output.

The multi-head attention mechanism allows the model to attend to different parts of the input sequence simultaneously, which allows the model to learn more nuanced representations of the input. This is particularly useful in NLP tasks, where the meaning of a sentence can depend on multiple words and phrases.

Positional encoding

Positional encoding is a mechanism implemented in the Transformer architecture to provide the model with information about the relative position of the elements in the input sequence. This is necessary because the self-attention mechanism used in the Transformer architecture does not inherently consider the order of the elements in the input sequence.

In the Transformer architecture, each element in the input sequence is represented as a vector, and these vectors are passed through multiple layers of self-attention and feed-forward neural networks. However, since the self-attention mech-

anism only considers the relationships between the vectors, and not their order, the model is unable to distinguish between two sentences with the same elements but different order.

To overcome this limitation, positional encoding is used to add information about the relative position of the words in the input sequence to the vectors representing the words. This is done by adding a fixed vector, called the positional encoding vector, to each word vector. The value of the positional encoding vector is determined by the position of the word in the input sequence, and the mathematical function used to generate these vectors is based on transcendental functions, specifically on sine and cosine functions.

The mathematical function used to generate the positional encoding vectors, as presented in (Vaswani et al., 2017), is:

$$\begin{aligned} PE_{(pos,2i)} &= \sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \\ PE_{(pos,2i+1)} &= \cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \end{aligned} \tag{2.37}$$

where pos is the position of the word in the input sequence, i is the dimension of the word vector, and d_{model} is the dimension of the positional encoding vector.

The resulting word vectors with the added positional encoding are then passed through the multiple layers of the Transformer model.

We illustrate in Figure 2.11 how the position of each element in a sequence is encoded within the positional encoding paradigm.

Main transformer architectures

There is a wide variety of Transformer-based models currently used in NLP tasks. However, in this section we are going to present just those that are relevant to this thesis: the BERT model and the DistilBERT model.

Bidirectional Encoder Representations from Transformers (BERT)

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is a pre-trained Transformer-based neural network model for NLP tasks. It was designed to understand the meaning of a text by analyzing the context in which words appear. BERT is trained on a large corpus of text data, which allows it to learn the relationships between words and their meanings.

The BERT model is trained on a task of predicting missing words in a sentence, called the Masked Language Model (MLM) task. This task is used to train the model to understand the context of a word by looking at the words surrounding it, both to the left and the right. It is bidirectional in the sense that it considers the context from both directions, which allows it to understand the meaning of a word in the context of the entire sentence.

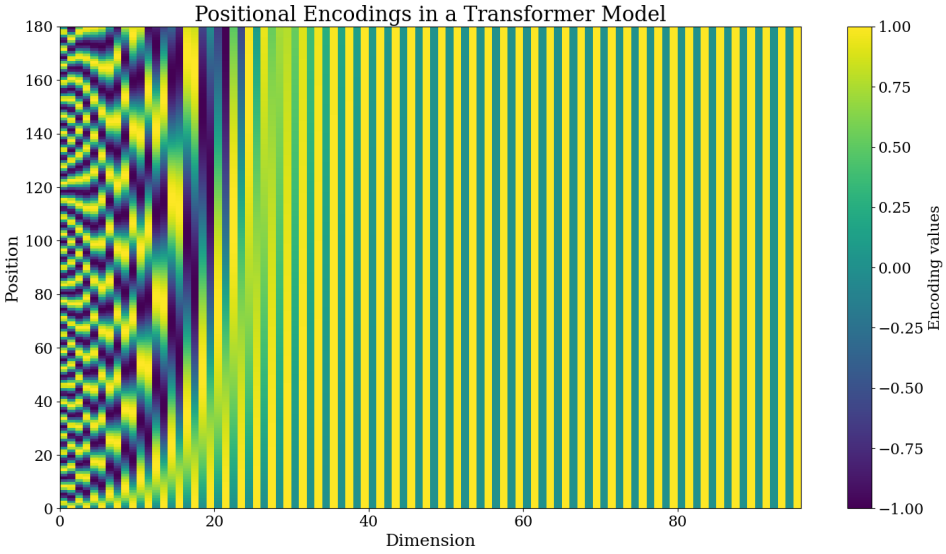

Figure 2.11: Positional encoding representation.

One of the main uses of BERT is for text classification tasks. To use BERT for text classification, the pre-trained model is fine-tuned on a labeled dataset for the specific task. Fine-tuning involves training the model on a new dataset while keeping the pre-trained weights fixed, and only updating the weights of the last layers of the model. The fine-tuned model can then be used to classify new text data into different categories.

BERT can be fine-tuned on various text classification tasks such as sentiment analysis, topic classification, and named entity recognition. The pre-trained BERT model can be fine-tuned on a specific dataset with a specific classification task, and it often gives state-of-the-art results on various text classification benchmarks.

Distilled Bidirectional Encoder Representations from Transformers (DistilBERT)

A distilled version of BERT (DistilBERT) (Sanh et al., 2020) is a smaller, faster, and cheaper version of the BERT model. DistilBERT is trained on the same data as BERT, but uses fewer parameters, which makes it faster and more efficient to run.

The model is distilled from a larger BERT model, meaning that it is trained to approximate the behavior of the larger model while using fewer parameters. This is achieved through knowledge distillation (Hinton et al., 2015), where the outputs from the larger model are used to guide the training of the smaller model. The result is a smaller model that maintains similar levels of performance as the larger model on a variety of NLP tasks.

DistilBERT has been shown to perform well on a wide range of natural language understanding tasks, such as question answering and sentiment analysis, while being smaller and faster than the original BERT model.

Overall, DistilBERT is a useful model for developers and researchers who want to use BERT's powerful language understanding capabilities in their applications but are limited by computational resources or want to reduce the size of the model for deployment.

2.4 Machine Learning models for Emergency Medical Call Incidents triage

Next, we expose the state-of-the-art of how Machine and Deep Learning techniques are being used in the emergency medical triage problem, considering research that has dealt with similar—or rather similar—problems than ours. We describe what they provide, as well as their limitations.

If we focus on Machine Learning-based solutions for out-of-hospital emergency medical triage, the literature we found is scarce, since 1) it is a very specific problem and 2) the field of Machine Learning—and Deep Learning—is relatively novel and hence, there is still some reluctancy in its usage by the emergency medical dispatch professionals. However, we present the most relevant works found in this field.

Finally, it has to be taken into account that at the time this thesis began, none of the studies we are about to present had been published.

Cardiac arrest recognition: (Blomberg et al., 2019) considered a Machine Learning approach to the detection of cardiac arrest from audio files of emergency calls. Their results show that sensitivity in this detection can be increased via

Machine Leaning while keeping a decent value of specificity. Specifically, they achieved sensitivity values of 84.1%, and specificity values of 97.3%.

Risk scores prediction: exposed in (Spangler et al., 2019), the authors developed Gradient Boosting models to predict the risk associated to each of the patients involved in a prehospital emergency medical event. Results from their study shown that Machine Learning-based scores outperformed rule-based triage algorithms and human prioritization decisions in this prehospital triage setting.

Conveyance needs for unconscious patients: exposed in (Tollinton et al., 2020), they developed Random Forest and Gradient Boosting models to determine whether free text dispatcher observations could improve the prediction of unconscious patients who require conveyance. Results from their study showed that considering this Machine Learning-based strategy improved the outcome in terms of predicting these conveyance needs from an AUC of 0.57 to 0.64.

Most probable clinical pathways: presented in (Veladas et al., 2021), they designed and implemented a text-based model to calculate the top-3 and the top-5 most probable clinical pathways associated to a telephonic emergency medical event. Results from their work revealed an accuracy of around 95% in the prediction of these clinical pathways.

Intelligent telephone triage: documented in (HAN, 2022), they developed a Random Forest model to determine the acuity of an out-hospital medical emergency case given the available structured information during the emergency medical call. Results reported in their work indicate a reduction in over-triage rates of around 15% whilst maintaining a similar level of under-triage rates.

Under-triage prediction: described in (Inokuchi et al., 2022), they evaluated different types of Machine Learning models, specifically Support Vector Machines, Lasso Regression, Random Forest, Gradient Boosting and Deep Learning models. These models can facilitate the early detection of under-triaged patients.

Chapter 3

Deep ensemble multitask classification of emergency medical call incidents combining multimodal data improves emergency medical dispatch

The objective of this work was to develop a predictive model to aid non-clinical dispatchers in classifying emergency medical call incidents by their life-threatening level (yes/no), admissible response delay (undelayable, minutes, hours, days), and emergency system jurisdiction (emergency system/primary care) in real-time. We used 1 244 624 independent incidents from the Valencian emergency medical dispatch service in Spain, compiled retrospectively from 2009 to 2012, including clinical features, demographics, circumstantial factors, and free text dispatcher observations. Based on them, we designed and developed DeepEMC^2, a deep ensemble multitask model integrating four subnetworks: three specialized to context, clinical, and text data, respectively, and another to ensemble the former. The four subnetworks are composed in turn by multi-layer perceptron modules, bidirectional long short-term memory units, and a bidirectional encoding representations from transformers module. DeepEMC^2 showed a macro F1-score of 0.759 in life-threatening classification, 0.592 in admissible response delay, and 0.757 in emergency system jurisdiction. These results show a substantial performance increase of 12.5%, 17.5%, and 5.1%, respectively, with respect to the current in-house triage protocol of the Valencian emergency medical dispatch service. Besides, DeepEMC^2 significantly outperformed a set of baseline machine learning models, including naive bayes, logistic regression, random

forest, and gradient boosting ($\alpha = 0.05$). Hence, DeepEMC^2 is able to: 1) capture information present in emergency medical calls not considered by the existing triage protocol, and 2) model complex data dependencies not feasible by the tested baseline models. Likewise, our results suggest that most of this unconsidered information is present in the free text dispatcher observations. To our knowledge, this study describes the first Deep Learning model undertaking emergency medical call incidents classification. Its adoption in medical dispatch centers would potentially improve emergency dispatch processes, resulting in a positive impact in patient well-being and health services sustainability.

The contents of this chapter were published in the journal publication (Ferri et al., 2021)—thesis contributions C1 and P1.

3.1 Introduction

EMD involves the reception and management of requests for medical assistance in an emergency medical services system (J. J. Clawson & Dernocoeur, 1988). It comprises two main dimensions: call-taking, where emergency medical calls are received and incidents are classified according to their priority—triaged—and controlling, where the best available resources are dispatched to handle the event (Blandford & William Wong, 2004).

The call-taking process is generally managed by emergency medical dispatchers (Stratton, 1992). These mediators are in many cases non-clinical staff, trained with the essential knowledge of medical emergencies for the proper and efficient management of the incident (Blandford & William Wong, 2004; J. J. Clawson & Dernocoeur, 1988; J. Clawson, 1981; Stratton, 1992). Dispatchers usually follow a clinical protocol, established in the medical dispatch center, and periodically verified by medical supervisors (Palumbo et al., 1996).

However, despite preparation and the existence of triage protocols, assigning priorities to EMCIs is a challenging and stressful task for dispatchers, requiring constant concentration (Ek et al., 2013; Forslund et al., 2004; Weibel et al., 2003). Additionally, there is always an inherent uncertainty on the real patient state, since the information of the event is gathered from telephonic interview processes. Furthermore, there are time constraints due to the incident priority or the need for tackling other incoming calls (Leprohon & Patel, 1995). A wrong priority assignment derives either in insufficient medical attention or unnecessary resource deployment (Hjälte et al., 2007; Sramek et al., 1994). In consequence, EMCIs triage protocols are continuously revised and enhanced.

Many triage algorithms, such as the Emergency severity index (Gilboy et al., 2012), the Manchester triage system (Mackway-Jones et al., 2013), the Canadian

triage and acuity scale (Murray et al., 2004) or the Australasian triage scale (Considine et al., 2004), have been widely studied and enriched (Christ et al., 2010; Seiger et al., 2011; Storm-Versloot et al., 2011; Zachariasse et al., 2017). However, they are difficult to benchmark, deriving in no international agreement about their use for EMD (FitzGerald et al., 2010). Likewise, these algorithms depend on structured clinical information which is not always available during the call (Farand et al., 1995). As such, improvements in EMD processes by redefining this sort of protocols are extremely costly and limited.

Hence, approaches for EMD improvement based on alternative paradigms such as Machine Learning, and particularly Deep Learning, are gaining momentum. Deep Learning is at the state of the art of Machine Learning in tasks involving complex types of data (LeCun et al., 2015), e.g., high dimensional, unstructured, sequential, multimodal (Hinton et al., 2012; Hirschberg & Manning, 2015; Russakovsky et al., 2015; Silver et al., 2016), such as those found in EMCI databases. Likewise, this and other Machine Learning tools have already been applied to tackle EMD challenges such as ambulance allocation (Chen & Lu, 2014; McLay & Mayorga, 2013), prediction of emergency calls volume (Channouf et al., 2007), automatic stress detection of the caller (Lefter et al., 2011), interpretable knowledge extraction (Barrientos & Sainz, 2012), performance monitoring (Klement & Snášel, 2011), cardiac arrest calls assistance (Blomberg et al., 2019) or triaging unconscious and fainting patients (Tollinton et al., 2020). Therefore, ML, and in particular Deep Learning, is a feasible and promising technology to improve EMD through EMCI classification. However, most current studies dealing with EMCI by means of ML—such as those previously exposed—tend to focus on specific disorders, developing high quality models but restricted to narrow domains, not being designed to handle the wide casuistry intrinsic to general EMCI classification.

In the Valencian Community (Spain), the triage of EMCI is currently supported by an in-house triage protocol, based on a clinical decision tree, grounded on heavily structured clinical variables, e.g., chest pain (yes or no), collected throughout the interview in a sequential manner. Therefore, free text dispatcher observations, with higher expressiveness than structured data, cannot be automatically processed by the protocol, limiting its generalization to situations beyond the established guidelines.

These limitations, along with the potential capability of Deep Learning to enhance general EMCI classification—through the provision of decision support to non-clinical dispatchers—was spotted by the Health Services Department of the Valencian region. Since no previous studies were known to have dealt with this problem before, a new study was required to assess the performance of Deep Learning in general EMCI classification.

In this chapter, we develop and evaluate a Deep Learning model to provide decision support to non-clinical dispatchers in EMCI triage from the medical dispatch

center of the Valencian region. Our model is designed to integrate the EMCI data collected during the call and carry out its classification. Despite of the existence of studies dealing with EMCI classification for specific disorders, to our knowledge, this is the first large-scale study undertaking a general EMCI classification trough Deep Learning.

3.2 Materials

3.2.1 Dataset

Overview

A total of 1 244 624 independent EMCI of the Health Services Department of the Valencian Community, were compiled in retrospective from 2009 to 2012. The Health Services Department board of the Valencian Community chose this time window due to its high data reliability—during this period, coordinating physicians supervised dispatcher' recordings—and the absence of critical changes in the emergency pathology of the population of the Valencian region over the last 15 years.

These EMCI data included during-call and after-call data. We categorized the data variables as structured—fixed fields—and unstructured—open fields—as well as stationary—with no implicit order—and sequential—with an implicit order (Figure 3.1).

During-call data

During-call data (Figure 3.1 top) are recorded during the emergency medical call. These data consist of demographics, circumstantial factors, clinical features—collected throughout the triage tree navigation—and free text dispatcher observations:

Demographics data—structured and stationary—include age, gender and risk group variables. Age is a numerical discrete feature, gender is a categorical binary variable (*male, female*) and risk group is a categorical multiclass variable—with multiple possible values, such as *asthmatic*, *allergic*, *cardiac*, *diabetic*, *neoplastic*, etc.

Circumstantial factors data—structured and stationary—include date and caller type variables. The latter consists on a categorical multiclass variable, keeping information about the person or institution which made the emergency medical call and taking values such as *police, red cross, the patient, a relative*, etc.

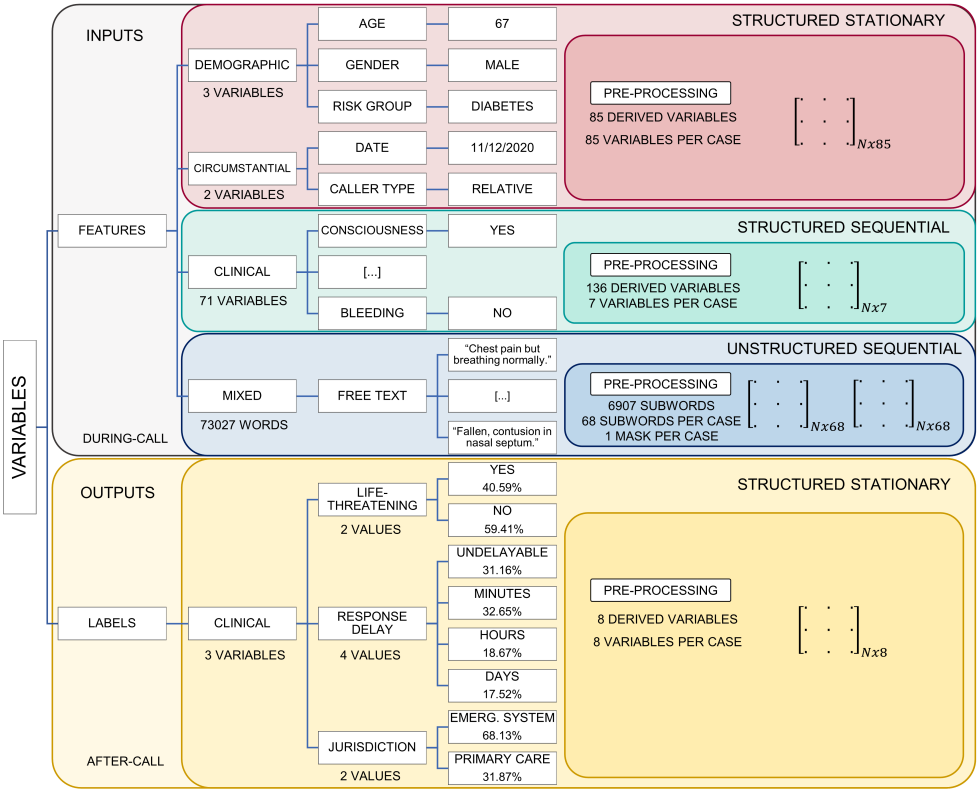

Figure 3.1: Dataset variables arranged by type. Names and cardinality, before and after pre-processing (derived variables), are presented, indicating how many variables—or subwords, when referring to text features—are available per case after pre-processing. Examples for their values are also included. Class frequencies for each output label are also reported. N is equal to the final 722 270 EMCI used in the study.

Clinical variables data—structured and sequential—include features providing relevant medical information. They are collected in a sequential manner during the call, registering a subset of them, from the total 71 variables available. A full list including all these variables is available in Table 3.1 and Table 3.2. These variables are categorical, presenting one possible value or multiple ones. An example of how four clinical variables and their values are registered during an emergency medical call could be: *previous trauma, yes; hemorrhage, yes; bleeding site, rectal bleeding; consequences of the clinic, severe blood loss.*

Table 3.1: Clinical variables with some of their example values. Certain variables have just one possible associated value, while others may exhibit multiple values. To ease presentation, example values are limited to three in this table.

Variable	Example values
Active arrhythmia	yes
Active suicide attempt	yes
Acute decompensation of mental illness	yes
Administration of medication	yes
Age	less than 1 year, over 70 years
Altered behavior	abnormal behavior, aggressiveness/agitation
Arterial vascular clinic	yes
Bleeding site	epistaxis, hematuria, melena
Blood glucose	abnormal
Blood or mucus in stool	no, yes
Breathing	absent, labored
Burn	yes
Causation of intake	autolysis attempt , medication error
Choking	yes
Clinic start	abrupt, progressive
Clinic triggers	upsetting
Clinical evolution	stable without worsening
Consequences of the clinic	mild blood loss, moderate blood loss, severe blood loss
Constipation	yes
Consumption of toxic substances	yes
Cyanosis	yes
Death	yes
Diarrhea	yes
Dizziness	yes
Drug intake	no, yes
Dyspnoea	no, yes
Dysuria and / or hematuria	yes
Eating / bilious vomiting	yes
Epidemiological criteria	contact with contaminated samples, contact with diagnosed cases
Epidemiological infectious disease	yes
Existence of neurological focality	yes
Fever	over 38, over 39
Flu syndrome	yes
Gastrointestinal symptoms	yes
Hemorrhage	no, yes
Hypertensive crisis	yes

Finally, free text dispatcher observations—unstructured and sequential—consist on short sentences, written during the call and providing additional relevant information which cannot be recorded in a structured manner. The language in which they are written is Spanish. Examples of two free text dispatcher observations bound each one to a different event are (translated into English): *according to the caller epileptic crisis, he has drunk and taken pills, he is half-conscious with half-closed eyes; patient bleeds abundantly from the head after falling at home, they have just found it in a pool of blood.*

Table 3.2: Clinical variables with some of their example values. Certain variables have just one possible associated value, while others may exhibit multiple values. To ease presentation, example values are limited to three in this table.

Variable	Example values
ICTUS code criteria	no
Impaired consciousness	yes
Impaired consciousness level	yes
Incident location	highway, inter-urban road, lakes or rivers and other inland waters
Injury severity	major, minor, moderate
Intake household product	yes
Intake of substance (medicine or toxic)	yes
Itchiness	yes
Medical history	cardiac pathology, copd, diabetes
Menstruation	yes
Nasal congestion	no, yes
Number of injured	from 1 to 3, over 3
Ongoing birth	yes
Pain	abdomen, generalized, head, lumbar area
Pregnant	no, yes
Previous trauma	no, yes
Prior care	no, yes
Recovered unconscious	yes
Regular medication	impossible to obtain, insulin, oral antidiabetics
Relationship and contact level	absent, present
Seizures	yes
Sickness	yes
Signs of severity	no, yes
Skin alteration type	edema/swelling
Skin disorders	yes
Symptoms of glottic edema	yes
Time of evolution	over 24 hours
Toxic substance	heroin
Treatment	prescribed treatment for the clinical picture, psychiatric medication
Type of accident	aggression, collision, drowning
Unconscious	no, yes
Vegetative picture	no, yes
Venous vascular clinic	yes
Vomiting	yes
Without further information	yes

After-call data

After-call data are recorded at a time after the call and used to derive EMCI classification labels, since they provide reliable up-to-date information about the real patient state. These data include: posterior physician diagnosis, standardized by International classification of diseases codes (I.C.D., 2021), such as *syncope (ICD 780.2)* or *acute myocardial infarction* (ICD 410); maneuvers and procedures indicating if the patient was *intubated, reanimated, sedated, received surgery*, etc.; and hospitaliza-

tions and urgency stays with information about the department where the patient was treated, the amount of time he stayed there and his discharge code.

Labels derivation

We transcribed the information contained in after-call data to three different and complementary EMCI classification labels (Figure 3.1 bottom): life-threatening level (yes/no), admissible response delay (undelayable, minutes, hours, days) and emergency system jurisdiction (emergency system/primary care). The mapping between after-call data and EMCI classification labels was established by a panel of 17 physicians from the Health Services Department of the Valencian Community, using a Delphi methodology (Dalkey, 1969).

Data quality assessment and inclusion criteria

To ensure the highest reliability of the model training data, we performed and reported a data quality analysis on the included data (Sáez et al., 2019). The analysis included the assessment of data quality dimensions of completeness and consistency, as well as temporal and multi-source variability (Sáez et al., 2015, 2016, 2017)—changes in the statistical distributions of data over time or among sources, respectively. The main findings included: approximately 30% of data with at least one missing label; and outlying distributions in some dispatchers, especially those with less than 100 calls.

According to these results, we considered, for the next stages of our work, those EMCI which after-call data were fully available, and which during-call data were registered by non-novice dispatchers—dispatchers with more than 100 calls managed. The final working dataset size comprised 722 270 EMCI.

3.2.2 Framework

The implementation language was Python 3.7.3 (G. van Rossum (Guido), 1995), making use of libraries Pandas (McKinney, 2010), NumPy (van der Walt et al., 2011), and Fuzzywuzzy (Cohen, 2011), for data pre-processing and Sklearn (Pedregosa et al., 2011), Pytorch (version 1.4.0) (Paszke et al., 2017), Hugginface transformers (Wolf et al., 2019) and Hyperopt (Bergstra et al., 2015) for modeling.

3.3 Methods

3.3.1 Data pre-processing

Depending on variable type, different pre-processing techniques were applied, mapping the original data to a matrix representation to be used for the Deep Learning model (Figure 3.1 right, highlighted pre-processing blocks):

Age, a structured stationary discrete ordinal variable, was mapped to a fuzzy (Zadeh, 1965) representation through piecewiselinear functions (Novák et al., 2012). These membership functions, represented in Figure 3.2, were validated by physicians of the Health Services Department of the Valencian Community. This smoothing transformation was carried out to avoid sharp transitions derived from grouping in a small set of categories discrete ordinal variables with high cardinality in their values.

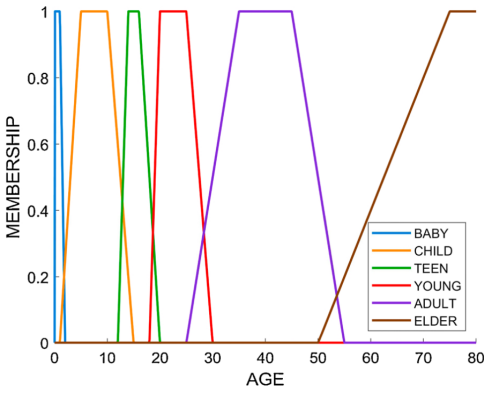

Figure 3.2: Piecewise linear functions representing age group membership.

Gender, risk group and caller type, structured stationary categorical variables, were one-hot encoded while several variables were derived from the date variable: weekday, month, if the day was or not a weekend day and if the day was or not was a bank holiday. These resulting variables, also structured stationary categorical variables, were one-hot encoded too.

Regarding the clinical variables, structured sequential variables, each variable-value pair was converted to an integer, conforming then, sequences of integers that were pre-padded afterwards, to ensure sequences of fixed length (Dwarampudi & Reddy, 2019). This length was equal to 7, since in more than 99% of the incidents reported, the number of clinical variables collected was equal or lower than 7.

Spelling correction processes by means of fuzzy string matching (Wagner & Fischer, 1974) were applied to the free text dispatcher observations, unstructured sequential variables, to reduce vocabulary dimensionality and noise. Besides, subword tokenization with WordPiece was carried out to reduce vocabulary size (Wu et al., 2016). To ensure sequences of fixed length while keeping information about the original sequences lengths, post-padding and attention mask generation were conducted. The padding length was set in 68, since in more than 99% of the incidents reported, the number of subwords written was equal or lower than 68.

Finally, labels, structured stationary categorical data, were one-hot encoded, deriving in a label matrix of 8 columns, each one associated with a specific label-class pair.

3.3.2 Data splitting and sampling

To evaluate model performance and tune hyperparameters without any bias, data were iteratively and randomly split into six subsets (Figure 3.3) (Kohavi, 1995).

First, data were randomly split into two disjoint *design* and *test sets*, with 80% and 20% proportions respectively. Next, the *design* set was randomly divided again into a *training* and a *validation set*, with 80% and 20% proportions. Finally, a sampling step was performed taking 100000 elements to define a *training* and a *validation sample*.

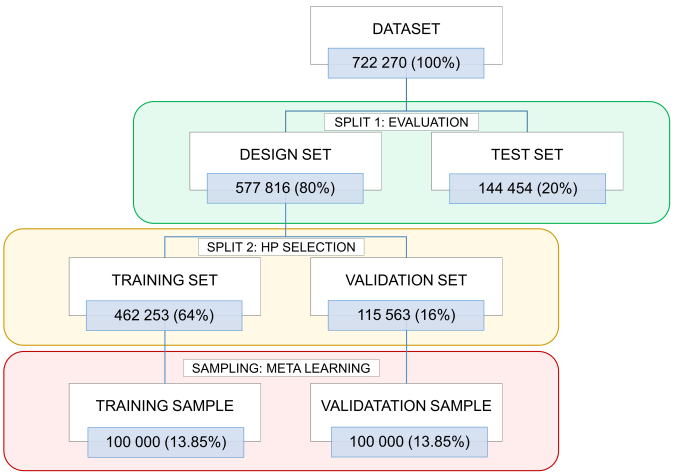

Figure 3.3: Data splitting and sampling. The number of data of each partition, along with its percentage respect the total number of data, are provided. Abbreviations: HP, hyperparameter.

3.3.3 Deep neural network design

The problem of classifying EMCI combining multimodal data was divided into four subproblems: three EMCI classification problems taking as inputs for each one EMCI data from the same type—structured stationary, structured sequential and unstructured sequential—and a last EMCI classification problem taking as inputs inner outputs obtained from the solution of the prior problems. To solve these four challenges, four Deep Learning subnetworks were developed: the *Context subnetwork* (ConNet), the *Clinical subnetwork* (CliNet), the *Text subnetwork* (TextNet) and the *Ensemble subnetwork* (EnsNet). Finally, once trained, they were combined in a single global modular neural network model (Kacprzyk & Pedrycz, 2015).

Likewise, as the life-threatening, response delay and jurisdiction labels provide different but related information, e.g., a life-threatening situation implies a low admissible response delay, a multitask learning (Caruana, 1997) paradigm was followed, to exploit these label dependences. To promote training efficiency and regularization while reducing the number of subnetworks parameters, a hard parameter sharing approach (Ruder, 2017b) was adopted. Hence, each of the four developed subnetworks presented a task-shared block—same set of parameters for all label prediction tasks—and a task-specific block—specific set of parameters for each label prediction task.

The ensemble of the four multitask subnetworks defined DeepEMC^2—Deep Ensemble Multitask Classifier for Emergency Medical Calls—the global and definitive Deep Learning model.

Next, we describe in detail each of the subnetworks integrated in DeepEMC^2, supported by Figure 3.4:

The *Context subnetwork* (Figure 3.4 left) deals with the demographics and circumstantial factors bound to an EMCI. It consists on a multi-layer perceptron (MLP) (Malsburg, 1986) due to its adequateness to model structured and stationary data, composed by dense and output blocks. A dense block integrates a fully connected layer (Goodfellow et al., 2016) a batch normalization layer (Ioffe & Szegedy, 2015) to manage internal covariate shift, a leaky ReLU (Maas et al., 2013) activation function to avoid vanishing and exploding gradients, while preventing dead neurons issues (Nwankpa et al., 2018) and a dropout layer (Hinton et al., 2012) to prevent neuron co-adaptation. An output block is composed by a fully connected layer and a softmax activation function, to dispose of a normalization score—between 0 and 1—for each class of each predicted label.

The *Clinical subnetwork* (Figure 3.4 center) deals with the clinical features collected during the call. It consists on a recurrent model, since clinical features are notified in a sequential manner, being their recording order potentially informative. It is composed by an embedding layer (Bengio et al., 2000), which compresses the sparse

input space into a smaller and dense one; a stack of multiple bidirectional long short-term memory (BLSTM) (Schuster & Paliwal, 1997) units, which capture long-term dependences far better than standard recurrent models; multiple skip connections (He et al., 2016) across the BLSTM units, to reduce the risk of losing relevant information during BLSTM propagation; a concatenation block—concatenates the outputs of these skip connections—and a MLP module, integrated by dense and output blocks, to act as an intermediary between the multiple BLSTM outputs and the final label predictions.

The *Text subnetwork* (Figure 3.4 right) deals with the free text dispatcher observations—unstructured and sequential—written during an EMCI. It is composed by a bidirectional encoding representations from transformers (BERT) (Devlin et al., 2019) block, since this model is at the state of the art in natural language processing tasks, including text classification, and a MLP module, to relate BERT outputs with label outputs. The BERT block is comprised in turn by an embedding block, an encoder block (Vaswani et al., 2017), and a pooler block, while the MLP component is constituted by dense and output blocks.

The *Ensemble subnetwork* (Figure 3.4 bottom) integrates inner outputs from the ConNet, the CliNet and the TextNet to generate the final outputs of DeepEMC^2. It consists of a concatenation block with a MLP component, composed by dense and output blocks. The inputs of the concatenation block are the outputs of the last layer of the dense block prior to the task-specific block of each one of the former subnetworks. It takes these inner outputs, and not the final output scores since these last values aggregate tons of information in just a small set of scalar values; hence, the modeling potential of the inner outputs is higher.

3.3.4 Parameter tuning

Subnetworks were trained in a constructive modularized manner (Kacprzyk & Pedrycz, 2015), so they were independently trained and assembled later as loosely coupled models. The optimizer selected for that was ADAM (Kingma & Ba, 2017), given its learning adaptability, noisy gradients management and learning process stability (Ruder, 2017a; Sun et al., 2019). A term of weight decay (Krogh & Hertz, 1991) was included in the parameters upgrading rule expression, to promote regularization. Likewise, it was followed a mini-batch upgrading approach (Bertsekas, 1994), computing gradients with backpropagation (Hecht-Nielsen, 1989) and backpropagation through time (Werbos, 1990). The objective function was a cross-entropy (Janocha & Czarnecki, 2017) loss (CEL). For each subnetwork, three CEL were calculated—one per label—averaged afterwards and finally backpropagated to carry out the parameter tuning process. Layers with leaky ReLU activation functions were initialized with Kaiming initialization (He et al., 2015), while softmax activation function layers were initialized with Xavier's initialization (Glorot & Bengio, 2010).

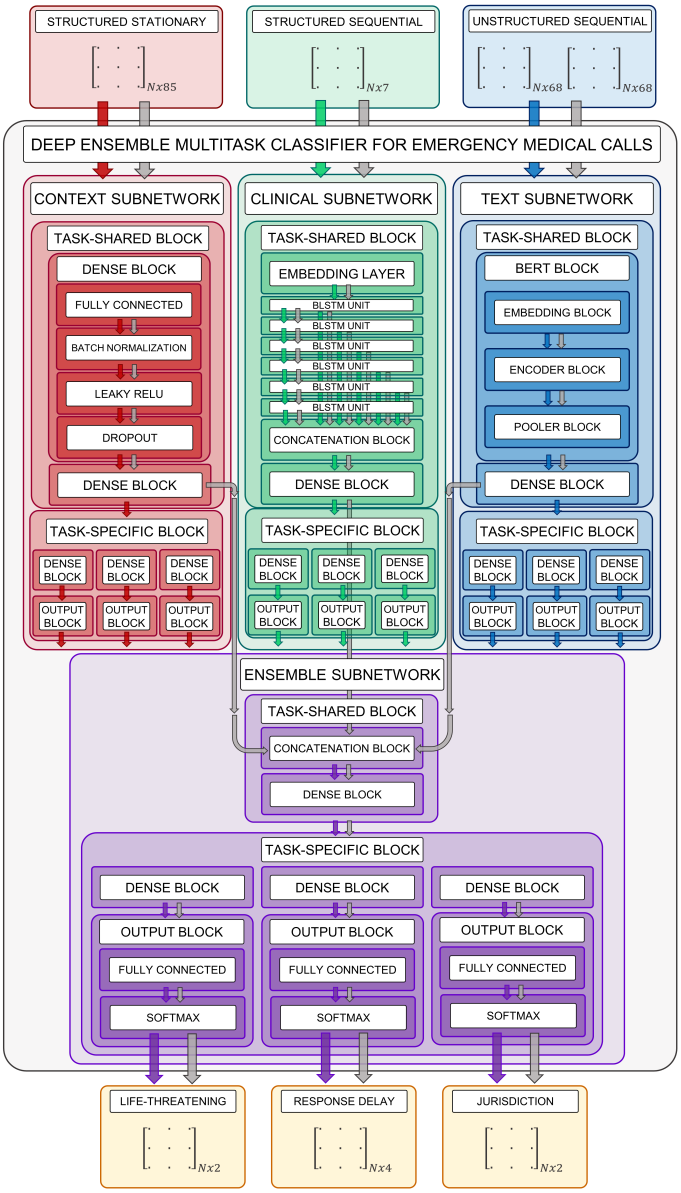

Figure 3.4: DeepEMC^2—Deep Ensemble Multitask Classifier for Emergency Medical Calls—architecture, including its constituting subnetworks—the *Context subnetwork*, the *Clinical subnetwork*, the *Text subnetwork* and the *Ensemble subnetwork*. Arrows indicate the forward propagation direction, for each subnetwork, as well as the global network (DeepEMC^2), colored according to the particular neural network they refer.

3.3.5 Hyperparameter tuning

The influence of hyperparameters over subnetworks performance was carefully considered in this work, in order to maximize the attainable outcomes. The hyperparameters studied were related with subnetworks architecture and optimizer settings.

Hyperparameters were tuned following a multi-step strategy (Figure 3.5).

The first step involved an automatic active learning (Settles, 2009) hyperparameter optimization process (Figure 3.5 top): four surrogate models—one per subnetwork—based on tree-structured parzen estimators (Bergstra et al., 2011), learned the conditional probability distribution of subnetworks hyperparameters given their associated CEL. Aiming to maximize the Expected Improvement (Jones, 2001) of the CEL, new hyperparamter configurations were iteratively sampled from the surrogate models, being upgraded after each training loop. Thereby, 280 different subnetworks—70 hyperparameter configurations times four subnetworks—were trained and evaluated in the training and validation samples, respectively.

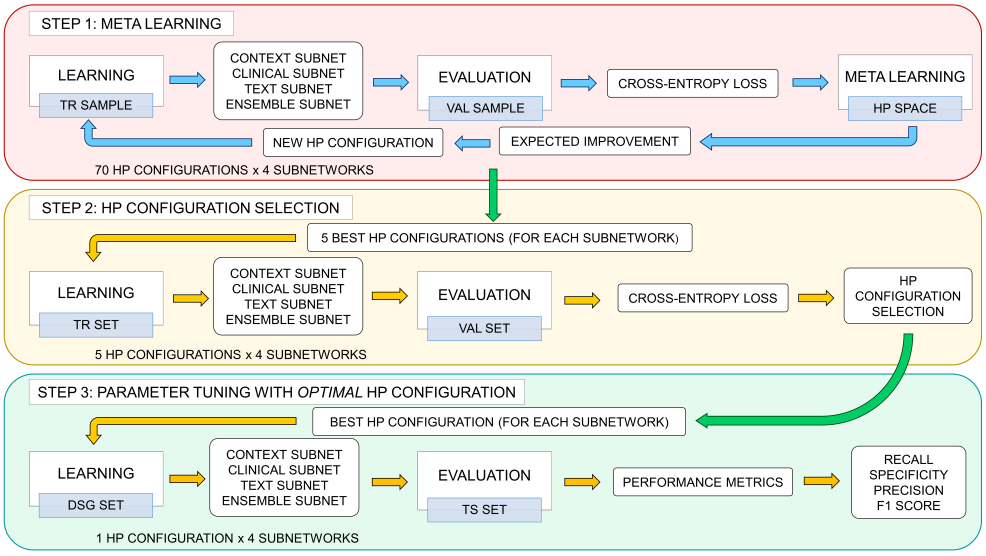

Figure 3.5: Multi-step hyperparameter tuning strategy. Yellow arrows imply unidirectionality, while blue arrows stand for a feedback loop, both inside a hyperparameter optimization step. Green arrows denote unidirectionality across hyperparameter optimization steps. Abbreviations: HP, hyperparameter; TR, training; VAL, validation; DSG, design TS, test.

Next, the best hyperparameter configurations proposed by the surrogate models were selected (Figure 3.5 middle). To prevent overfitting, the best five hyperparameter

configurations for each subnetwork were taken to retrain and validate the subnetworks, in the training and the validation set, respectively, obtaining a total of 20 models trained in this step. Then, the CEL was obtained for each of them and those hyperparameter configurations with the best value—lowest validation CEL—were considered as the optimal hyperparameter configuration.

Finally, the optimal hyperparameters were used to retrain the four subnetworks using the whole design set, to ensure a proper exploitation of the data (Figure 3.5 bottom). Once trained, its integration into a single architecture defined DeepEMC^2—the global network—evaluated later in the test set.

3.3.6 Evaluation

In-house triage protocol and baseline models

First, to assess if DeepEMC^2 provides an improvement in EMCI classification respect the existing clinical rules, performance metrics were obtained for the current in-house triage protocol of the Valencian emergency dispatch service.

Second, to compare the performance of the Deep Learning model respect well-known machine learning models in EMCI classification, we trained and evaluated the following baseline models:

1. Multinomial naive bayes (NB) (Bayes & Price, 1763): including a term of additive Laplace smoothing (Manning et al., 2008).

2. Logistic regression (LR) (Nelder & Wedderburn, 1972): including a penalty term for L2 regularization (Ng, 2004) and resorting to L-BFGS (Liu & Nocedal, 1989) as optimizer algorithm.

3. Random forest (RF) (Ho, 1995): considering Gini impurity as splitting criterion (Raileanu & Stoffel, 2004), while assembling a total of 300 tree estimators whose maximum depth was equal to 50, being these optimal values determined via hyperparameter tuning procedures.

4. Gradient boosting (GB) (Friedman, 2001): considering mean squared error with improvement score by Friedman (Friedman, 2001) as splitting criterion, with a total of 300 tree estimators whose maximum depth was set in 5, being these optimal values determined by hyperparameter tuning processes.

Notably, the input data for these baseline models had to be adapted to be processed by them. Clinical variables were one-hot encoded instead of being fed as sequences of integers. Regarding free text observations, once spelling correction

processes, subword tokenization and sentence truncation were carried out, subwords were one-hot encoded.

Metrics

Performance metrics were obtained in the *test set* (144 454 independent EMCI) for each label prediction task and each model trained—we recall here that EnsNet outputs are the same as DeepEMC^2. The evaluation metrics included accuracy, recall, precision and F1-score (Maimon & Rokach, 2010; Yang & Liu, 1999). For binary labels (life-threatening, jurisdiction), recall, precision and F1-score were referencing the interest class—life-threat and emergency system jurisdiction. Regarding the multiclass label (response delay), recall and precision were calculated for each class and then averaged following a macro approach. Likewise, for all labels, macro F1-score (Maimon & Rokach, 2010; Yang & Liu, 1999) was computed, to dispose of a balanced multiclass performance descriptor—not influenced by class frequencies. Finally, for all metrics, 95% confidence intervals were calculated by 1000 bootstrap samples (Efron & Tibshirani, 1994) extracted from the test set.

Metrics were calculated in the test set, for the protocol, the baseline models—naive bayes, logistic regression, random forest and gradient boosting—and the Deep Learning models developed—the ConNet, the CliNet, the TextNet and DeepEMC^2. We recall here that, although DeepEMC^2 is the definitive Deep Learning model which takes into account input data globally, results referring its constituting subnetworks, contrasted with baseline models trained with the same type of input data of each subnetwork, are also reported, to analyze the contribution of each set of inputs to the global model and where Deep Learning provides a substantial gaining over the other kind of models.

Likewise, percentage differences between DeepECM^2 and the protocol are also reported, as well as percentage differences between DeepECM^2 and the best baseline model—that baseline model with the best balanced multiclass performance—which has been measured in our work in terms of macro F1-score.

3.4 Results

Table 3.3, Table 3.4 and Table 3.5 show the classification performance results for the life-threatening level, admissible response delay and emergency system jurisdiction labels, respectively.

3.4.1 Life-threatening level

Table 3.3 shows that DeepEMC^2—the global Deep Learning model—highly outperforms the current protocol in the life-threatening prediction task with a 13.2% of accuracy improvement and a 12.5% of macro F1-score increment. This increment is statistically significant as reflected by the absence of overlapping in the 95% confidence intervals (CI). DeepEMC^2 captures more true life-threatening situations—higher recall—being much more precise—with less false positives.

In comparison to the baseline models, although DeepEMC^2 does not offer the best recall or precision, it achieves the best trade-off between them, as indicated by the best F1-score, being this metric statistically superior to those F1-scores attained by the baseline models. Likewise, referring to the best balanced two-class performance, DeepEMC^2 presents the best macro F1-score, with statistically significant difference respect to the baselines models.

Focusing on the subnetworks, the ConNet is the weakest Deep Learning model. The CliNet offers the better detection rate for true life-threatening situations but at the expense of a significant amount of false positives. Finally, the TextNet exhibits the overall better behavior although its capability to capture true life-threatening events is not the best among the subnetworks.

Regarding the comparative performance among the subnetworks and their respective baseline models, it stands out the performance similitude among the ConNet and some of their associated baseline models as well as the high outcomes resemblance among the CliNet and the baseline models using clinical variables. Finally, notably the TextNet presents greater differences respect its corresponding baseline models, being these differences notorious in the F1-score and macro F1-score.

Table 3.3: Performances of the in-house triage protocol, baseline models and Deep Learning models in life-threatening prediction (test set). Bootstrapped 95% confidence intervals are shown between brackets. Percentage differences between DeepEMC^2—the global Deep Learning model—and the protocol ΔP (%), along with percentage differences between DeepEMC^2 and the best baseline model ΔBM (%)—highest F1-score and F1-score^{MACRO}—are also reported. Abbreviations: MAC, macro; Ctx., context; Cli., clinical; Glo., global; NB, naive bayes; LR, logistic regression; RF, random forest; GB, gradient boosting; DL, Deep Learning; ΔP, DeepEMC^2 difference respect to the protocol; ΔBM, DeepEMC^2 difference respect to the best baseline model in life-threatening.

	Life-threatening level (yes/no)				
	Single-class metrics (yes)			Two-class metrics (yes/no)	
Model	Recall	Precision	F1-score	Accuracy	F1-score^{MAC}
Protocol	0.644[0.641,0.647]	0.547[0.544,0.551]	0.592[0.589,0.595]	0.639[0.637,0.641]	0.634[0.632,0.636]
Ctx. NB	0.407[0.404,0.410]	0.563[0.559,0.567]	0.472[0.469,0.475]	0.631[0.629,0.633]	0.594[0.592,0.596]
Ctx. LR	0.411[0.407,0.414]	0.577[0.573,0.581]	0.480[0.476,0.483]	0.638[0.636,0.640]	0.601[0.599,0.604]
Ctx. RF	0.465[0.462,0.469]	0.526[0.522,0.529]	0.494[0.491,0.497]	0.612[0.610,0.614]	0.590[0.588,0.592]
Ctx. GB	0.428[0.425,0.432]	0.588[0.584,0.592]	0.495[0.492,0.499]	0.646[0.644,0.648]	0.611[0.609,0.613]
Ctx. DL	0.440[0.436,0.443]	0.583[0.579,0.587]	0.501[0.498,0.504]	0.644[0.642,0.647]	0.613[0.610,0.615]
Cli. NB	0.732[0.729,0.735]	0.550[0.547,0.553]	0.628[0.625,0.630]	0.647[0.645,0.650]	0.646[0.644,0.649]
Cli. LR	0.752[0.750,0.755]	0.586[0.583,0.589]	0.659[0.656,0.661]	0.683[0.681,0.685]	0.682[0.680,0.684]
Cli. RF	0.764[0.761,0.767]	0.585[0.583,0.589]	0.663[0.661,0.665]	0.684[0.682,0.686]	0.683[0.681,0.685]
Cli. GB	0.763[0.760,0.766]	0.585[0.583,0.589]	0.663[0.660,0.665]	0.684[0.682,0.686]	0.683[0.681,0.685]
Cli. DL	0.790[0.787,0.793]	0.581[0.578,0.584]	0.669[0.667,0.672]	0.683[0.681,0.685]	0.682[0.681,0.685]
Text NB	0.681[0.678,0.685]	0.647[0.644,0.650]	0.664[0.661,0.666]	0.719[0.718,0.721]	0.711[0.710,0.714]
Text LR	0.629[0.626,0.633]	0.728[0.724,0.731]	0.675[0.672,0.678]	0.754[0.752,0.756]	0.738[0.736,0.740]
Text RF	0.514[0.511,0.517]	0.783[0.780,0.787]	0.621[0.618,0.624]	0.745[0.743,0.747]	0.714[0.712,0.716]
Text GB	0.578[0.575,0.581]	0.758[0.755,0.762]	0.656[0.653,0.659]	0.753[0.752,0.755]	0.732[0.730,0.734]
Text DL	0.638[0.635,0.642]	0.737[0.734,0.740]	0.684[0.681,0.687]	0.760[0.758,0.762]	0.745[0.744,0.747]
Glo. NB	0.729[0.726,0.732]	0.635[0.632,0.638]	0.679[0.676,0.681]	0.720[0.718,0.722]	0.715[0.713,0.717]
Glo. LR	0.652[0.649,0.656]	0.736[0.733,0.740]	0.692[0.689,0.695]	0.764[0.762,0.766]	0.750[0.748,0.752]
Glo. RF	0.585[0.582,0.589]	0.776[0.773,0.779]	0.667[0.665,0.670]	0.763[0.761,0.765]	0.742[0.740,0.744]
Glo. GB	0.616[0.613,0.620]	0.762[0.759,0.765]	0.681[0.679,0.684]	0.766[0.764,0.768]	0.748[0.746,0.750]
DeepEMC^2	0.671[0.668,0.675]	0.742[0.739,0.745]	0.705[0.702,0.707]	0.771[0.770,0.773]	0.759[0.757,0.761]
ΔP (%)	2.7[2.1,3.4]	19.5[18.8,20.1]	11.3[10.7,11.8]	13.2[12.9,13.6]	12.5[12.1,12.9]
ΔBM (%)	1.9[1.2,2.6]	0.6[-0.1,1.2]	1.3[0.7,1.8]	0.7[0.4,1.1]	0.9[0.5,1.3]

3.4.2 Admissible response delay

Table 3.4 shows that DeepEMC^2 outcomes are significantly superior to those achieved by the protocol in the response delay prediction task (CI 95%).

Overall detection of situations with a specific admissible response delay (undelayable, minutes, hours, days) is largely improved by DeepEMC^2—15.8% increment in macro recall—while remarkably enhancing overall precision—17.3% increment. Regarding the general performance in all classes, DeepEMC^2 significantly improves the protocol, with a 16.4% of accuracy improvement and a 17.5% of macro F1-score increment.

DeepEMC^2 does not offer the best overall precision compared to the baseline models. However, it improves the overall recall and the best—balanced multiclass performance, in terms of macro F1-score. Furthermore, this global performance is the best, in terms of statistically significance difference respect the baseline models, although the performance difference respect the global gradient boosting model—best baseline model in admissible response delay prediction—is at the limit, since 0 is the lower bound of the 95% confidence intervals for performance differences.

Focusing on DeepEMC^2 subnetworks for response delay prediction, the ConNet is at the bottom in performance terms, not being capable of outperforming the protocol. The CliNet is clearly over the ConNet and already beats the protocol, while the TextNet is the best DeepEMC^2 subnetwork in all metrics, with a substantial increase respect to the CliNet.

Regarding the comparative performance among the subnetworks and their respective baseline models, it can be appreciated the performance similitude among the ConNet and some of their associated baseline models as well as the high outcomes resemblance among the CliNet and the baseline models fed with the clinical variables. Finally, the TextNet presents greater differences respect its corresponding baseline models, being these differences significant in the macro F1-score metric.

Table 3.4: Performances of the in-house triage protocol, baseline models and Deep Learning models in response delay prediction (test set). Bootstrapped 95% confidence intervals are shown between brackets. Percentage differences between DeepEMC^2—the global Deep Learning model—and the protocol ΔP (%), along with percentage differences between DeepEMC^2 and the best baseline model ΔBM (%)—highest F1-score$^{\text{MACRO}}$—are also reported. Abbreviations: MAC, macro; Ctx., context; Cli., clinical; Glo., global; NB, naive bayes; LR, logistic regression; RF, random forest; GB, gradient boosting; DL, Deep Learning; ΔP, DeepEMC^2 difference respect to the protocol; ΔBM, DeepEMC^2 difference respect to the best baseline model in response delay prediction.

	Admissible response delay (undelayable, minutes, hours, days)			
Model	Recall$^{\text{MAC}}$	Precision$^{\text{MAC}}$	F1-score$^{\text{MAC}}$	Accuracy
Protocol	0.411[0.409,0.413]	0.416[0.414,0.419]	0.401[0.398,0.403]	0.428[0.426,0.430]
Ctx. NB	0.375[0.373,0.377]	0.382[0.379,0.385]	0.364[0.362,0.366]	0.396[0.394,0.399]
Ctx. LR	0.376[0.374,0.378]	0.396[0.393,0.398]	0.369[0.367,0.371]	0.406[0.403,0.408]
Ctx. RF	0.348[0.345,0.350]	0.357[0.354,0.359]	0.350[0.348,0.352]	0.371[0.369,0.373]
Ctx. GB	0.382[0.380,0.384]	0.414[0.411,0.417]	0.383[0.381,0.385]	0.415[0.413,0.417]
Ctx. DL	0.376[0.374,0.378]	0.415[0.412,0.418]	0.377[0.374,0.379]	0.413[0.411,0.415]
Cli. NB	0.458[0.456,0.460]	0.503[0.501,0.506]	0.460[0.458,0.462]	0.482[0.480,0.484]
Cli. LR	0.479[0.477,0.481]	0.522[0.520,0.525]	0.488[0.486,0.490]	0.505[0.503,0.507]
Cli. RF	0.477[0.475,0.479]	0.533[0.530,0.535]	0.485[0.483,0.488]	0.507[0.504,0.509]
Cli. GB	0.477[0.475,0.479]	0.532[0.530,0.535]	0.485[0.483,0.488]	0.507[0.504,0.509]
Cli. DL	0.477[0.475,0.479]	0.530[0.527,0.532]	0.485[0.483,0.487]	0.506[0.504,0.508]
Text NB	0.527[0.524,0.529]	0.517[0.515,0.519]	0.519[0.517,0.521]	0.533[0.531,0.535]
Text LR	0.544[0.542,0.546]	0.564[0.562,0.567]	0.550[0.548,0.553]	0.569[0.567,0.572]
Text RF	0.524[0.522,0.527]	0.583[0.581,0.586]	0.535[0.533,0.538]	0.563[0.561,0.566]
Text GB	0.545[0.543,0.547]	0.577[0.575,0.580]	0.554[0.552,0.556]	0.574[0.572,0.576]
Text DL	0.544[0.542,0.546]	0.583[0.580,0.585]	0.555[0.553,0.557]	0.576[0.574,0.578]
Glo. NB	0.537[0.534,0.539]	0.531[0.529,0.534]	0.533[0.531,0.535]	0.549[0.547,0.551]
Glo. LR	0.557[0.555,0.559]	0.579[0.577,0.581]	0.564[0.562,0.567]	0.582[0.580,0.585]
Glo. RF	0.547[0.545,0.549]	0.593[0.590,0.595]	0.557[0.555,0.560]	0.581[0.579,0.583]
Glo. GB	0.562[0.560,0.565]	0.593[0.591,0.596]	0.572[0.570,0.574]	0.589[0.587,0.592]
DeepEMC^2	0.569[0.567,0.571]	0.589[0.587,0.591]	0.576[0.574,0.579]	0.592[0.590,0.594]
ΔP (%)	15.8[15.4,16.2]	17.3[16.8,17.7]	17.5[17.1,18.1]	16.4[16,16.8]
ΔBM (%)	0.7[0.2,1.1]	-0.4[-0.9,0]	0.4[0,0.9]	0.3[-0.2,0.7]

3.4.3 Emergency system jurisdiction

Table 3.5 shows that DeepEMC^2 significantly outperforms the protocol in the jurisdiction prediction task (95% CI). It captures more situations which are jurisdiction of the emergency system—better recall—being more precise—with less false positives. Respect to the overall performance in both classes, DeepEMC^2 surpasses the protocol, with a 4.5% of accuracy improvement and a 5.1% of macro F1-score increment.

DeepEMC^2 does not offer the best recall or precision compared to the baseline models. However, it achieves, along with the gradient boosting model, the best trade-off between them, as indicated by their best F1-score, being this metric statistically superior to that attained by the logistic regression model—best baseline model in emergency system jurisdiction prediction. Likewise, referring to the best balanced two-class performance, DeepEMC^2 presents the best macro F1-score, with statistically significant differences respect the baselines models.

Focusing on DeepEMC^2 subnetworks, although the ConNet presents the highest recall values, its precision is not the best, with worse general results than the protocol in the jurisdiction prediction task. The CliNet provides a substantial improvement over the later subnetwork, with an overall performance above the protocol. As in life-threatening and response delay, the TextNet is the subnetwork attaining the best outcomes.

Regarding to the comparative performance among the subnetworks and their respective baseline models, notably the performance is similar among the ConNet and some of their associated baseline models as well as the high outcomes resemblance among the CliNet and the baseline models fed with the clinical variables. Finally, it has to be highlighted that the TextNet presents greater differences respect its corresponding baseline models, being these differences notorious in the F1-score and accuracy.

Table 3.5: Performances of the in-house triage protocol, baseline models and Deep Learning models in jurisdiction prediction (test set). Bootstrapped 95% confidence intervals are shown between brackets. Percentage differences between DeepEMC^2—the global Deep Learning model—and the protocol ΔP (%), along with percentage differences between DeepEMC^2 and the best baseline model ΔBM (%)—highest F1-score and F1-score$^{\text{MACRO}}$—are also reported. Abbreviations: MAC, macro; Ctx., context; Cli., clinical; Glo., global; NB, naive bayes; LR, logistic regression; RF, random forest; GB, gradient boosting; DL, Deep Learning; ΔP, DeepEMC^2 difference respect to the protocol; ΔBM, DeepEMC^2 difference respect to the best baseline model in jurisdiction prediction.

	Emergency system jurisdiction (yes/no)				
	Single-class metrics (yes)			Two-class metrics (yes/no)	
Model	Recall	Precision	F1-score	Accuracy	F1-score$^{\text{MAC}}$
Protocol	0.855[0.854,0.857]	0.800[0.798,0.802]	0.827[0.825,0.828]	0.756[0.754,0.757]	0.706[0.703,0.708]
Ctx. NB	0.892[0.891,0.894]	0.752[0.750,0.754]	0.816[0.815,0.818]	0.726[0.724,0.728]	0.638[0.636,0.640]
Ctx. LR	0.919[0.918,0.921]	0.746[0.744,0.748]	0.824[0.822,0.825]	0.731[0.729,0.733]	0.629[0.627,0.631]
Ctx. RF	0.850[0.848,0.852]	0.745[0.743,0.747]	0.794[0.793,0.796]	0.699[0.697,0.701]	0.618[0.615,0.620]
Ctx. GB	0.936[0.935,0.937]	0.744[0.742,0.746]	0.829[0.828,0.831]	0.737[0.735,0.739]	0.628[0.625,0.630]
Ctx. DL	0.945[0.943,0.946]	0.741[0.739,0.743]	0.830[0.829,0.832]	0.736[0.734,0.738]	0.620[0.618,0.622]
Cli. NB	0.897[0.896,0.899]	0.800[0.798,0.802]	0.846[0.844,0.847]	0.777[0.775,0.778]	0.720[0.718,0.723]
Cli. LR	0.906[0.904,0.908]	0.798[0.796,0.800]	0.848[0.847,0.850]	0.779[0.777,0.781]	0.721[0.718,0.723]
Cli. RF	0.901[0.899,0.902]	0.801[0.799,0.803]	0.848[0.847,0.850]	0.780[0.778,0.782]	0.724[0.722,0.726]
Cli. GB	0.916[0.914,0.917]	0.793[0.791,0.795]	0.850[0.849,0.851]	0.779[0.778,0.781]	0.717[0.714,0.719]
Cli. DL	0.900[0.899,0.902]	0.802[0.800,0.804]	0.848[0.847,0.849]	0.780[0.778,0.782]	0.724[0.722,0.726]
Text NB	0.793[0.791,0.795]	0.833[0.831,0.835]	0.812[0.811,0.814]	0.750[0.748,0.752]	0.719[0.717,0.721]
Text LR	0.896[0.895,0.898]	0.810[0.807,0.811]	0.851[0.849,0.852]	0.785[0.783,0.787]	0.734[0.732,0.736]
Text RF	0.936[0.934,0.937]	0.782[0.780,0.784]	0.852[0.851,0.853]	0.778[0.776,0.780]	0.704[0.702,0.707]
Text GB	0.906[0.905,0.907]	0.803[0.801,0.805]	0.851[0.850,0.853]	0.784[0.782,0.786]	0.728[0.726,0.730]
Text DL	0.917[0.916,0.919]	0.804[0.802,0.806]	0.857[0.856,0.858]	0.791[0.789,0.793]	0.734[0.732,0.736]
Glo. NB	0.818[0.817,0.820]	0.834[0.832,0.836]	0.826[0.825,0.828]	0.765[0.763,0.766]	0.731[0.729,0.733]
Glo. LR	0.902[0.901,0.904]	0.816[0.814,0.818]	0.857[0.855,0.858]	0.794[0.792,0.796]	0.745[0.743,0.747]
Glo. RF	0.925[0.924,0.926]	0.802[0.800,0.804]	0.859[0.858,0.860]	0.793[0.791,0.795]	0.734[0.732,0.737]
Glo. GB	0.914[0.913,0.916]	0.811[0.809,0.813]	0.860[0.858,0.861]	0.796[0.794,0.798]	0.743[0.741,0.745]
DeepEMC^2	0.895[0.894,0.897]	0.827[0.825,0.829]	0.860[0.858,0.861]	0.801[0.799,0.802]	0.757[0.755,0.759]
ΔP (%)	4[3.7,4.3]	2.7[2.3,3.1]	3.3[3,3.6]	4.5[4.2,4.8]	5.1[4.7,5.6]
ΔBM (%)	-0.7[-1,-0.4]	1.1[0.7,1.5]	0.3[0,0.6]	0.7[0.3,1]	1.2[0.8,1.6]

3.5 Discussion

3.5.1 Relevance

The superior performance of DeepEMC^2 and some of the baseline models, respect to the in-house triage protocol, suggests the existence of information provided during the emergency medical call not considered by the current protocol, but captured by the machine learning models. Likewise, the Deep Learning approach is preferable over the other families of models tested, since DeepEMC^2 outcomes are significantly above those attained by the baseline models.

In referring context and clinical variables, Deep Learning is not clearly at the top. However, regarding the free text dispatcher observations, the Deep Learning approach is, overall, remarkably superior. Likewise, as TextNet outcomes are far better than those attained by the ConNet and CliNet, the most valuable information provided during the emergency medical call would be present at these unstructured features. Since text fields are unbounded, they would embrace wider casuistry, allowing more precision in the EMCI description, lowering, consequently, its uncertainty.

Regarding the clinical variables, they stand as an excellent life-threatening detector features—about 80% of total cases. This could be due to the fact that dispatchers ask for them to reduce chances of missing situations where patient's life is at risk. Similarly, the outstanding emergency system jurisdiction recall of demographics and circumstantial factors—capturing about 95% of total cases—may be related with patient profiles highly susceptible from requiring emergency aid, e.g., elderly cardiac patient males.

Comparing classification scores across tasks, the hardest classification problem appeared to predict the admissible response delay, probably derived from the fact that it is a multiclass label, presenting twice possible outputs (undelayable, minutes, hours, days) than the other labels (life-threatening, jurisdiction), which are binary.

The modular approach followed in this work, assembling four specialized subnetworks into a single global network (DeepEMC^2), has shown that the potential of the aggregated network is superior to any of its individual components, balancing their respective weaknesses and strengths while properly integrating processed information within each one.

Unlike previous studies (Barrientos & Sainz, 2012; Blomberg et al., 2019; Channouf et al., 2007; Chen & Lu, 2014; Klement & Snášel, 2011; Lefter et al., 2011; Maxwell et al., 2009; McLay & Mayorga, 2013; Tollinton et al., 2020), we offer general EMCI classification, not being restricted to specific disorders. Hence, incidents can be classified as they come, without needing a prior routing step which may introduce fatal biases in posterior processes. Besides, this holistic approach eases the

embedding of DeepEMC^2 in a clinical decision support system platform, avoiding the implementation of filtering operations which may hinder its usability.

Finally, the results of this work imply that current emergency dispatch processes could be improved by means of Deep Learning, eventually deriving in a positive impact over patient wellbeing and health services sustainability.

3.5.2 Limitations

The main limitation of this work is the inherent uncertainty bound to the problem: in the studied dataset it was likely to find similar input combinations presenting completely different label values. In other words, the challenge faced in this work exhibits classes overlap, where different disorders may present the same clinical picture. For example, chest pain may imply a life-threatening situation, if the underlying unknown cause is a heart attack, or not, since it could be derived from a prior anxiety crisis. This non-discriminative variability sets bounds in terms of maximum performance attainable by any model—Bayes error (Fukunaga, 2013). As such, from a model applicability viewpoint, although DeepEMC^2 notably surpasses the in-house triage protocol, it is not error-free. However, its value lies in offering recommendations regarding to EMCI classification which tend to be more accurate than those offered by the protocol, leading to better decision support, which in turn derives in EMD improvement.

Besides, the data available to conduct this work lies between 2009 and 2012 years (both included). Even though the clinical framework of pathologies like heart failure or epileptic crisis could be fairly constant across time, an in-depth study of potential dataset shifts (Quinonero-Candela et al., 2008) and related abrupt or gradual changes regarding the statistical distributions of new data has to be carried out before implementing the model in emergency medical dispatch centers.

3.5.3 Future work

Next steps include the evaluation of DeepEMC^2 with prospective cases from the Valencia region—with more recent incidents, monitoring the aforementioned dataset shifts and acting in consequence. Passing this phase favorably will enable us to begin the integration of the model into a clinical Decision Support System (CDSS) in an emergency medical dispatch center. The DeepEMC^2 CDSS will incorporate a graphical user interface to allow fast and straightforward interactions between the dispatcher and the model during the call. Likewise, a prospective evaluation of the system performance and added value on routine settings through a randomized controlled trial for CDSS (Angus, 2020) will be carried out. Finally, once these steps have been conducted, and with the approval of emergency medical experts of the

Health Services Department, the resulting tool will be implemented in the emergency medical dispatch center of the Valencian Community.

3.6 Conclusions

A novel deep ensemble multitask model (DeepEMC^2) designed to aid non-clinical dispatchers during emergency medical calls to classify incidents by their life-threatening level, admissible response delay and emergency system jurisdiction, has been developed and successfully evaluated. To our knowledge, this is the first Deep Learning model implemented to face this challenge.

The performance achieved by the model is highly superior to that attained by the current in-house triage protocol of the emergency medical dispatch service of the Valencian Community, achieving a macro F1-score improvement of 12.5%, 17.5%, 5.1% in life-threatening, response delay and jurisdiction classification, respectively. Likewise, DeepEMC^2 outcomes are above those accomplished by the additional machine learning models tested, including naive bayes, logistic regression, random forest and gradient boosting. This increment was proved as statistically significant ($\alpha = 0.05$).

Remarkably, the network modular design with specialized subnetworks for the different data modalities has allowed discovering the potential benefit of the information contained in free text fields for the automatic classification of emergency medical call incidents. This information can be used to optimize current guidelines.

The implantation of this model in medical dispatch centers would have a remarkable impact in patient well-being and health services sustainability.

Chapter 4

Discovering key topics in emergency medical dispatch from free text dispatcher observations

The objective of this work was to discover key topics latent within free text dispatcher observations registered during emergency medical calls. We analyzed a total of 1 374 931 independent retrospective cases from the Valencian emergency medical dispatch service in Spain, spanning from 2014 to 2019. Text fields underwent preprocessing to reduce vocabulary size and filter out noise, including accent and punctuation mark removal, as well as the elimination of uninformative and infrequent words. Key topics were inferred from the multinomial probabilities over words conditioned on each topic from a Latent Dirichlet Allocation model, trained following an online mini-batch variational approach. The optimal number of topics was set by analyzing the values of a topic coherence measure based on the normalized pointwise mutual information, across multiple validation K-folds. Our results support the presence of 15 key topics latent in free text dispatcher observations, related with: ambulance request; chest pain and heart attack; respiratory distress; head falls and blows; fever, chills, vomiting, and diarrhea; heart failure; syncope; limb injuries; public service body request; thoracic and abdominal pain; stroke and blood pressure abnormalities; pill intake; diabetes; bleeding; consciousness. The discovery of these topics implies the automatic characterization of a huge volume of complex unstructured data containing relevant information linked to emergency medical call incidents. Hence, results from this work could lead to the update of structured emergency triage algorithms to directly include this latent information in the triage process, resulting in a positive impact in patient well-being and health services sustainability.

The contents of this chapter were published in the conference paper (Ferri et al., 2022a)—thesis contributions C2 and P3.

4.1 Introduction

Emergency medical dispatch entails the reception and management of demands for medical assistance in an emergency medical services system (J. J. Clawson & Dernocoeur, 1988). It involves emergency medical calls attendance and events triage according to their priority, process generally managed by emergency medical dispatchers. These mediators tend to follow a clinical protocol focused on a small set of structured clinical variables (Stratton, 1992).

In the Valencian Community (Spain), the triage of emergency medical call incidents (EMCI) is currently assisted by an in-house triage protocol, a clinical decision tree based on the collection of structured variables. The dispatcher raises questions to the caller until reaching a final tree node, which has a priority assigned to it, the incident priority.

However, information not covered by the decision tree is also registered during the call in an unstructured manner in free text fields. This information, complementary to that provided by the structured variables, cannot be taken into account automatically by the clinical protocols, and thus, it is left unused.

We have studied in Chapter 3 that considering these free text dispatcher observations notably improves EMCI triage. Specifically, we have developed DeepEMC^2, a Deep Learning model able to automatically deal with structured and unstructured information in real-time, providing performance increases of 12.5%, 17.5% and 5.1% in terms of macro F1-score in life-threatening, admissible response delay and emergency system jurisdiction prediction, respect to the current in-house triage protocol of the Valencian emergency medical dispatch service (Ferri et al., 2021).

In addition, prior studies have shown the potential of text mining techniques and, concretely, topic extraction methods, to infer high-level information from huge amounts of unstructured medical data (Cheng et al., 2020; Pérez et al., 2018).

Given the utmost relevance of free text dispatcher observations in EMCI triage and the availability of methods to explore them from a Machine Learning perspective, we present in this work an unsupervised analysis of these free text fields, with the aim of 1) discovering and understanding what information dispatchers report during emergency call incidents and 2) exploring how this latent information is distributed across incidents.

4.2 Materials and Methods

A total of 1 374 931 free text dispatcher observations linked to EMCI of the Health Services Department of the Valencian Community, were compiled in retrospective from 2014 to 2019. Given the data source, our available free text fields were written in Spanish.

A set of preprocessing operations were carried out in order to reduce dimensionality to enhance posterior topic extraction processes. Dispatcher observations were converted to lowercase and then, as text fields were written in Spanish, accents marks were deleted. Punctuation marks were also discarded along with stopwords. Words not appearing at least 50 times in the corpus were dropped, resulting in a vocabulary reduction from 74 914 to 4584—discarding 94% of terms—while keeping around 96% of the total word counts in the corpus. Finally, text fields were tokenized.

Data was split using a holdout (Kohavi, 1995) methodology, with proportions of 80% for training and then 20% for testing. Next, cross-validation (Kohavi, 1995) splits were conducted over the training set, taking $K = 4$, without allowing repetition.

Topics were inferred from the multinomial probabilities over words conditioned on each topic from a Latent Dirichlet Allocation (LDA) (Blei et al., 2003) model. We preferred LDA over Latent Semantic Analysis (Deerwester et al., 1990) because LDA offers a generative modeling approach, and LDA over Probabilistic Latent Semantic Analysis (Hofmann, 1999) (PLSA) because the number of parameters estimated in PLSA grows linearly with the number of training documents and generalization to new documents is easier with LDA. LDA is a hierarchical generative Bayesian model, which assumes the existence of K latent topics in a collection of text documents. Next we present the generative process of LDA, to generate a corpus D of M documents each one with N_d words:

For each document d in a corpus of D documents:

- Draw a topic mixture from a Dirichlet prior, $\theta_d \sim Dirichlet(\alpha)$
- For each word w_n in document d:
  - Draw a topic z_n from the multinomial topic mixture, $z_n \sim Multinomial(\theta_d)$
  - Draw a word w_n from the topic-specific word distribution, $w_n \sim Multinomial(\beta_{z_n})$

Here α is the hyperparameter of the Dirichlet prior, θ_d is the topic mixture for document d, and β is the matrix of the topic-specific word distributions.

The LDA model was trained following an online mini-batch variational inference approach (Hoffman et al., 2010). The optimal number of topics was set analyzing the values of a topic coherence measure, where word context vectors were created using the normalized pointwise mutual information (Röder et al., 2015). The distance among word context vectors was calculated with the cosine distance, obtaining the final coherence score as the arithmetic mean of all distances, following the procedures described in (Syed & Spruit, 2017).

We tested different number of topics, specifically 5, 10, 15, 20, 25, 30 and 35. For each combination, we trained four LDA models, one per training K-fold, and calculated the aforementioned topic coherence measure in their respective validation folds. That number of topics offering the best overall performance across the validation K-folds was considered as the optimal number of topics.

Finally, we retrained the model with all the training data using the optimal configuration. For each topic, the most probable words were extracted and studied to infer topic semantics and naming it. After that, we derived the topic distribution in the training and the test corpora, to understand which were the most frequent topics in dispatcher free text fields, as well as to evaluate potential overfitting issues.

4.3 Results

Figure 4.1 shows the value of the topic coherence performance metric across the different K-folds, for each number of topics combination:

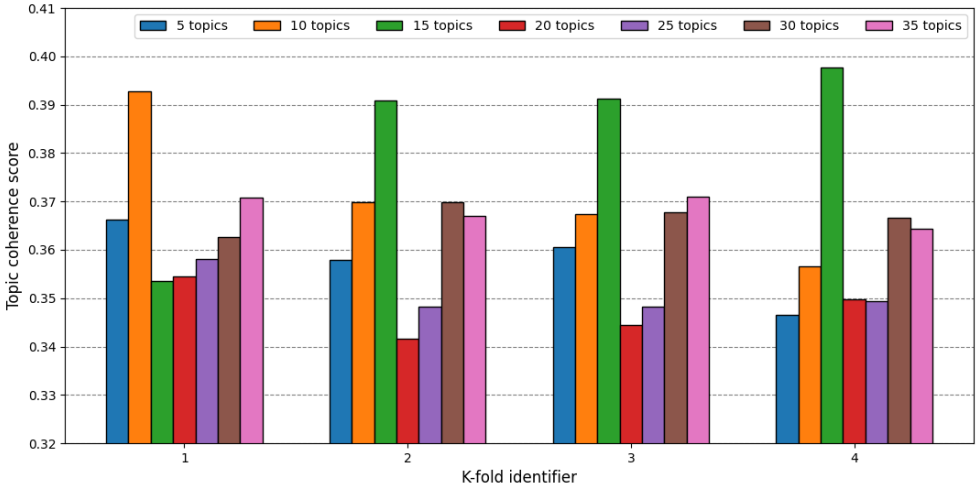

Figure 4.1: Number of topics selection. Topic coherence across K-folds over training set.

It can be appreciated that the optimal topic coherence value is reached at 15 topics.

Figure 4.2 displays the 8 words with higher associated probability respect to each topic. Each topic has been named according to the semantics its defining words represent.

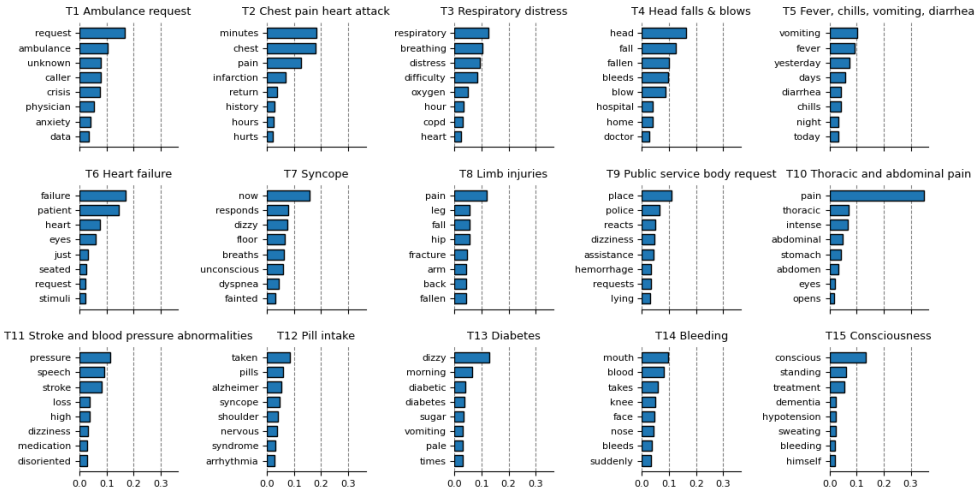

Figure 4.2: Topics discovered, described by their 8 words with highest probability conditioned on each topic. Word probabilities conditioned on each topic are represented in the x-axis.

It can be observed the presence of 13 clinical topics—T2, T3, T4, T5, T6, T7, T8, T10, T11, T12, T13, T14, T15—along with 2 resource dispatch topics—T1, T9. Likewise, most predominant semantics in the clinical topics are cardiovascular disorders—T2, T6, T11—and injuries T4, T8, T14.

Figure 4.3 presents the distribution of topics in the training and test corpora, sorted by its frequency of appearance in descending order.

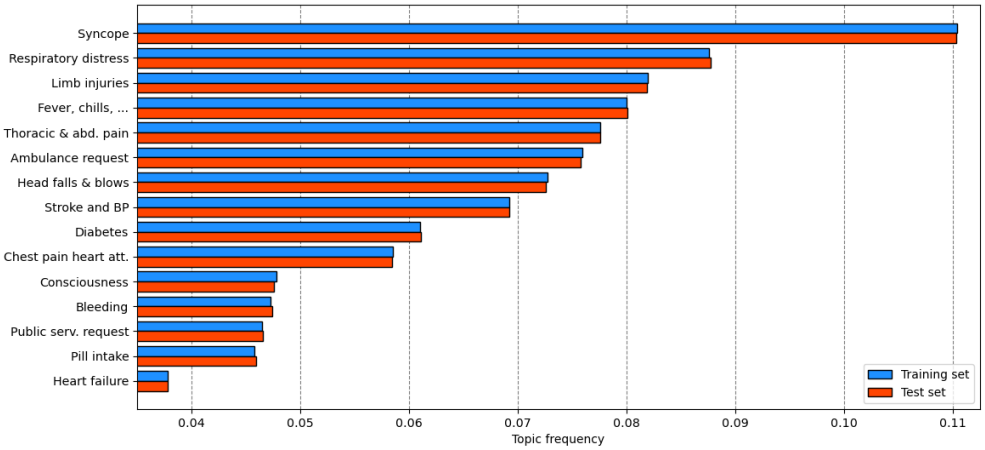

Figure 4.3: Topics distribution in the train and test corpus, sorted by frequency in descending order.

It can be inferred from this figure that there are no over-represented or under-represented topics. Likewise, there is a strong similarity between training and test topic distributions. Both are good signs indicating that overfitting does not seem to be present.

4.4 Discussion

The characterization of casuistry latent in complex unstructured data carried out in this chapter may lead emergency medical professionals to redefine structured decision tree algorithms in order to improve emergency medical dispatch processes.

Although the majority of topics are well-defined and delimited, some topics would require further study to evaluate the presence of topic mixtures and subtopics.

Future work will include studying relations among the topics found and potential clusters bound to the structured variables also registered during the incident. Finally, it is of interest to study why some words appear in different contexts, i.e., topics, despite having similar meanings, such as chest pain and thoracic pain.

4.5 Conclusions

This work has tackled the discovery of key topics in emergency medical dispatch from free text dispatcher observations. A pipeline comprising word filtering operations, number of topics selection and Latent Dirichlet Allocation model training, has been applied over 1 374 931 independent retrospective cases from the Valencian emergency medical dispatch service in Spain. Results support the existence of 15 latent topics, whose consideration could lead to the improvement of clinical triage protocols, deriving in turn, in a positive impact in patient well-being and health services sustainability.

Chapter 5

Deep continual learning for emergency medical call incidents text classification under the presence of dataset shifts

The aim of this chapter is to develop and evaluate a deep classifier that can effectively prioritize EMCI according to their life-threatening level under the presence of dataset shifts. We utilized a dataset consisting of 1 982 746 independent EMCI instances obtained from the Health Services Department of the Region of Valencia (Spain), with a time span from 2009 to 2019 (excluding 2013). The dataset includes free text dispatcher observations recorded during the call, as well as a binary variable indicating whether the event was life-threatening. To evaluate the presence of dataset shifts, we examined prior probability shifts, covariate shifts, and concept shifts. Subsequently, we designed and implemented four deep Continual Learning strategies—cumulative learning, continual fine-tuning, experience replay, and synaptic intelligence—alongside three Deep Continual Learning baselines—joint training, static approach, and single fine-tuning—based on DistilBERT models. Our results demonstrated evidence of prior probability shifts, covariate shifts, and concept shifts in the data. Applying Continual Learning techniques had a statistically significant ($\alpha = 0.05$) positive impact on both backward and forward knowledge transfer, as measured by the F1-score, compared to non-continual approaches. We can argue that the utilization of Continual Learning techniques in the context of EMCI is effective in adapting Deep Learning classifiers to changes in data distributions, thereby maintaining the stability of model performance over time. To our knowledge, this study represents the first exploration of a Continual Learning approach using real EMCI data.

The contents of this chapter are under review in the journal Computers in Biology and Medicine, specifically reviewer's comments are being addressed—thesis contributions C3, C4 and P4.

5.1 Introduction

EMD involves the reception and handling of requests for medical assistance in emergency situations (J. J. Clawson & Dernocoeur, 1988). It is a challenging task characterized by a high level of uncertainty, limited decision time, and scarce resources (FitzGerald et al., 2010). Given the potential severe consequences, including patient mortality and significant costs, associated with errors in this critical environment, there is a need for decision support tools to enhance call-taking situations.

The EMD process consists of two main components: triage, which assesses the priority of incidents, and resource allocation, which assigns the most appropriate resources to respond to each incident. In the context of triage, dispatchers typically follow predefined clinical guidelines in the form of decision trees (FitzGerald et al., 2010; Storm-Versloot et al., 2011). Examples of these triage protocols include the Emergency Severity Index (Wuerz et al., 2001) and the Manchester Triage System (Mackway-Jones et al., 2013). However, these clinical algorithms have two main limitations: firstly, they are based on archetypical cases, overlooking the vast number of incidents with complex characteristics, and secondly, they heavily rely on structured clinical information, which is not always available during emergency medical calls. As a result, these algorithms are unable to automatically handle unstructured data, such as free text.

During emergency medical calls, a significant amount of data is generated (Barroeta Urquiza & Boada Bravo, 2011). While these data are typically stored in health institution databases, they are often underutilized and only used for basic business intelligence analyses. Consequently, the latent information contained within these data, including hidden statistical patterns, is not considered to improve triage protocols. Moreover, a substantial portion of this data is in the form of unstructured information, which cannot be automatically processed by current triage protocols (Ferri et al., 2022a; Tollinton et al., 2020). Therefore, an alternative approach is needed to complement the limitations of existing triage protocols and enhance EMD processes.

Machine Learning stands out as one of the most promising approaches in the EMD environment. Numerous studies have demonstrated the value offered by Machine Learning tools in this domain. For instance, (Spangler et al., 2019) developed Machine Learning-based models to predict the risk associated with individual patients in prehospital emergency medical events. Their findings revealed that Machine Learning-based scores surpassed rule-based triage algorithms and human prioritization decisions in terms of performance. Similarly, (Blomberg et al., 2019) explored the

application of Machine Learning in detecting cardiac arrest from audio files of emergency calls. They demonstrated that Machine Learning techniques can increase sensitivity in cardiac arrest detection while maintaining a reasonable level of specificity. Furthermore, (Inokuchi et al., 2022) conducted an evaluation of different Machine Learning models and their impact on the early detection of under-triaged patients. Their study revealed that Machine Learning models can effectively aid in identifying under-triaged patients, leading to improved patient outcomes.

In the specific context of the emergency medical services of the Valencian Region, Spain, a project was undertaken with the objective of developing Machine Learning models utilizing historical EMD data for predicting incident priority and assessing its influence on the EMD process. As part of this project, a deep ensemble multitask Deep Learning model called DeepEMC^2 was created, as described in Chapter 3 and (Ferri et al., 2021). The model showed improvements in performance metrics compared to the existing in-house triage system. Specifically, it achieved better predictions in terms of life-threatening (+12.5%), admissible response delay (+17.5%), and emergency system jurisdiction (+5.1%). Notably, the model's success was attributed mainly to features extracted from free text, which proved to be more predictive than the clinical variables recorded during the call.

However, it should be noted that the data used to train the DeepEMC^2 model only covered the period from 2009 to 2012. As information systems, dispatchers, coordination centers, and demographics evolve over time, dataset shifts occur, leading to changes in the joint probability distribution of inputs and outputs between the training and testing stages (Moreno-Torres et al., 2012; Quinonero-Candela et al., 2008). In fact, the Valencian EMCI information system underwent significant changes in 2013, including updates to the in-house decision tree and dispatcher experience. Consequently, the DeepEMC^2 model developed in (Ferri et al., 2021) using data from 2009 to 2012 may require adjustments to mitigate potential performance degradation resulting from distributional shifts caused by these changes.

Therefore, it is reasonable to consider the incorporation of Continual Larning strategies to address the challenge of dataset shifts in EMCI. Continual Learning strategies facilitate the integration of new knowledge while avoiding catastrophic forgetting (McCloskey & Cohen, 1989; Parisi et al., 2019), enabling a sustainable learning process over time and providing adaptable decision support for call-takers. To implement this Continual Learninga approach, we exploit multiple learning experiences within our EMCI data, each associated with a different time period or batch of data (Lomonaco et al., 2021). Consequently, multiple data streams were derived from each batch, and the deep models learn from these streams according to the Continual Learning strategy defined.

Building upon the significance of text features highlighted in previous studies (Ferri et al., 2021, 2022a), our study has focused on developing Continual Learning

pipelines to ensure consistent decision support for the prediction of life-threatening levels using free text dispatcher observations.

The primary objective of this work is to investigate the extent to which various Continual Learning strategies enable lifelong adaptation of deep triage models over time. While acknowledging the inevitable negative impact on performance due to changes in data distributions in real-world scenarios, our framework aims to minimize such effects by leveraging Continual Learning techniques. To achieve this, we first assess the presence of dataset shifts and subsequently, explore and evaluate multiple Continual Learning pipelines designed to mitigate the adverse effects on model performance resulting from distributional drifts.

The findings of our study contribute to the advancement of decision support systems in emergency medical triage, with practical applications in real settings. These systems have the potential to positively impact patient well-being and enhance the sustainability of health services. Although we can find studies concerned with the development Machine Learning models focused on dealing with medical data in the presence of temporal distributional drifts (Guo et al., 2023), (Guo et al., 2022), (Lemmon et al., 2023), to the best of our knowledge, this is the first study to tackle real EMCI data using a Continual Learning approach, representing a significant contribution to the field and one of the earliest real-world applications of Continual Learning methods.

5.2 Materials

5.2.1 Dataset

A dataset comprising a total of 1 982 746 independent Emergency Medical Call Incidents (EMCI) was compiled from the Health Services Department (HSD) of the Valencian Region, covering the period from 2009 to 2019, with the exception of 2013 due to unavailability of data during the system update of the Valencian EMCI's information system.

The EMCI data consisted of both during-call and after-call information. During-call data were collected in real-time during the emergency medical call and included free text dispatcher observations written in the Spanish language. These observations were short sentences describing the incident, such as "stabbing chest pain with shortness of breath", "fever, general malaise, vomiting" or "traffic accident, profuse bleeding, unconscious". During inference, these observations were used as input for prediction.

After-call data were recorded at a later time, following the completion of the call. This data encompassed information such as physician diagnosis, hospitalizations,

urgent care visits, medical procedures, and treatments received by the patient. Importantly, these after-call data were not used during prediction but rather offline, for inferring whether the emergency event constituted a life-threatening situation, considering a mapping developed by expert physicians from the HSD of the Valencian Region. This binary variable served as the classification label in our work.

Next, we provide, in Table 5.1, some examples of the data considered to train and evaluate our deep triage models.

Table 5.1: Examples of free text notes belonging to the dataset. The Life-threat column indicates whether the situation is life-threatening (1) or not (0).

Text	Life-threat
83-year-old woman with respiratory and cardiac insufficiency. Neoplastic disease in progression.	1
14 year old male with fever of 39º, he has been like this for 1 hour and also general malaise.	0
85 year old woman with a lot of fatigue and cough since yesterday. Today saturation at 85, she is on oxygen at home.	0

5.2.2 Framework

Our experiments were implemented in Python (G. van Rossum (Guido), 1995), utilizing the libraries Numpy (Walt et al., 2011) and Pandas (McKinney, 2010) for data management, PyTorch (Paszke et al., 2017) and HuggingFace's Transformers (Wolf et al., 2019) for modeling, Avalanche (Lomonaco et al., 2021) for Continual Learning, and Optuna (Akiba et al., 2019) for hyperparameter tuning.

5.3 Methods

5.3.1 Data preparation

We utilized Natural Language Processing (NLP) techniques for both data preprocessing and inference with respect to free text variables. These techniques involved the utilization of language models, which are further described in Section 3.3. We applied pre-processing functions (e.g., lowercasing, removal of special characters, and accent marks) to enhance the languaje model encoding capability. Additionally, we employed sub-word tokenization using WordPiece (Wu et al., 2016) to reduce the size of the vocabulary. Subsequently, these sub-words were mapped to indexes, and padding and truncation operations were applied to ensure that all text records shared

the same sequence length, facilitating computation. Boolean attention masks were generated to exclude the impact of padding indexes.

The data were organized into ten learning experiences, each representing one year. Each learning experience consisted of a training stream and a test stream. Similarly, each training stream comprised a pure training stream and a validation stream, which were used for hyperparameter tuning operations without overfitting to the test stream. Next, the data arrangement process is presented in Table 5.2.

Table 5.2: Data arrangement process. The data is divided into experiences, with each experience corresponding to a different year. Furthermore, each experience consists of three distinct and non-overlapping data streams: a pure training stream, a validation stream, and a test stream. Abbreviations: Exp, Experience.

Exp	Year	Pure train	Validation	Test	Total
1	2009	101 669	43 710	36 864	182 243
2	2010	100 147	42 465	35 713	178 325
3	2011	101 245	43 459	35 930	180 634
4	2012	101 253	43 399	35 801	180 453
5	2014	92 396	39 860	33 134	165 390
6	2015	106 013	45 461	37 977	189 451
7	2016	110 883	47 302	39 591	197 776
8	2017	132 934	56 986	47 454	237 374
9	2018	129 963	55 692	46 111	231 766
10	2019	133 835	57 525	47 974	239 334

5.3.2 Dataset shifts assessment

As the phenomenon of dataset shift has already been introduced in the Rationale section, we do not provide an in-depth explanation in this section. Instead, we focus on elucidating how we assessed the three primary sources of drift in our work: prior probability shifts, covariate shifts, and concept shifts (Moreno-Torres et al., 2012).

Prior probability shift

To assess the presence of prior probability shifts in our study, we calculated the empirical probabilities of life-threatening events over time. These probabilities were plotted on a temporal graph, and a stationarity test was conducted. Specifically, we employed the Kwiatkowski–Phillips–Schmidt–Shin test (Kwiatkowski et al., 1992), which tests the null hypothesis of data distribution stationarity.

Covariate shift

To evaluate the presence of distributional changes in these covariates, we constructed multiple pairwise text classification models (refer to the Modeling section for more information about model structure and rationale) for each pair of years. Subsequently, we compared their performance in terms of the Area Under Curve (AUC) against the expected performance of a random model. This comparison allowed us to determine whether the model could predict the year from which the data originated. If the model demonstrated the ability to make accurate predictions, it would indicate the presence of a covariate shift, suggesting that the covariates varied across different years.

Concept shift

To assess the presence of concept shift, we trained a deep model (refer to the Modeling section for more details about the model architecture and rationale) for each year and evaluated its performance across all years. Any differences in model performance over the years could be interpreted as variations resulting from changes in the conditional distribution.

5.3.3 Modeling

The emergence of new Deep Learning architectures, such as Transformers (Vaswani et al., 2017), has significantly improved the performance of a wide array of NLP tasks. The attention mechanism in Transformers enables the model to access information from all elements of a text, allowing for contextual modeling of word and sentence meanings. Additionally, the Transformer architecture is well-suited for transfer learning in NLP, where knowledge gained from a more general task is utilized to specialize a model for new problems with limited data. This transfer of linguistic knowledge is achieved through pre-training and fine-tuning. Pre-training involves training language models using unsupervised learning tasks on extensive collections of text data, while fine-tuning involves further adapting the pre-trained model to specific supervised learning tasks, such as sequence classification. Transformer-based architectures have achieved state-of-the-art results across a wide range of NLP tasks. To ensure the effectiveness of the model in the presence of dataset shifts, we combine this paradigm with Continual Learning.

In the previous DeepEMC^2 model, deep models based on the BERT architecture (Devlin et al., 2019; Ferri et al., 2021) were employed, resulting in a significant improvement compared to non-Deep Learning approaches. However, considering that we are evaluating multiple Continual Learning strategies, the data volume is large, and our main objective is to study different Continual Learning pipelines, we opted

to use the DistilBERT (Sanh et al., 2020) model in this work. DistilBERT offers an excellent balance between performance and efficiency, as it has significantly fewer parameters than BERT with only a minor performance decrease due to knowledge distillation (Hinton et al., 2015). In addition, it can be used locally, eliminating the privacy risks that come into play when using APIs, which is particularly prudent given that we are handling sensitive data. It is important to note that we did not train our DistilBERT models from scratch; instead, we utilized the pretrained version available at (Wolf et al., 2019) and adopted a transfer learning approach by fine-tuning the model for the specific downstream task. Therefore, our model architecture consists of:

1. An embedding block, which includes a word embedding layer, positional encoding layer (Vaswani et al., 2017), layer normalization layer (Ba et al., 2016), and dropout layer (Hinton et al., 2012).

2. Multiple Transformer blocks, each composed of multi-head self-attention layers, layer normalization layers, and feed-forward layers.

3. An output block consisting of feed-forward layers, with the last layer utilizing softmax as the activation function.

For parameter tuning, we utilized the AdamW (Loshchilov & Hutter, 2019) optimizer, a variant of the Adam (Kingma & Ba, 2017) algorithm, known for its suitability in training Transformer models (Loshchilov & Hutter, 2019). The model was trained using a mini-batch training approach, and the loss function employed was cross-entropy (Janocha & Czarnecki, 2017), weighted to address class imbalance:

$$L = \sum_{n=1}^{N} \frac{-\sum_{c=1}^{C} \frac{1}{v_c} log \frac{\exp^{x_{n,c}}}{\sum_{i=1}^{C} \exp^{x_{x,i}}} y_{n,c}}{N} \tag{5.1}$$

Here, N denotes the mini-batch size, C represents the number of classes, v indicates the class frequency in the dataset, x denotes the logits, and y refers to the true target value.

5.3.4 Continual learning baselines

To assess the added value of including Continual Learning strategies for model adaptation over time, we employed three baseline techniques: a static model, single fine-tuning, and joint training. These baselines allow us to evaluate the impact of different approaches on model performance over the course of learning experiences.

Static model

The static model represents the scenario in which the model is not retrained over time, providing insight into how performance may decline if no action is taken. In this approach, we fine-tuned our pretrained DistilBERT model using only the data from the first learning experience (i.e., the year 2009). This approach serves as a lower performance bound for forward transfer of knowledge (Lopez-Paz & Ranzato, 2017) since it does not update the model with instances from recent experiences.

Single fine-tuning

The single fine-tuning strategy involves retraining the original model, pretrained DistilBERT, using data exclusively from the current learning experience. For subsequent learning experiences, the model weights are not retained, and the model is reinitialized with the pretrained DistilBERT weights. This approach provides a lower performance bound for backward transfer of knowledge (Lopez-Paz & Ranzato, 2017) since it does not retain information from previous experiences, considering only the data from the current experience.

Joint training

To estimate the best performance achievable by any Continual Learning strategy, we employed the joint training approach. This strategy involves training our deep model using data from all learning experiences, incorporating data from all years. While this approach is not applicable in a real-world setting, as we do not have access to future data at a given year, implementing this approach allows us to establish an upper bound for performance in terms of both forward and backward transfer of knowledge.

5.3.5 Continual learning strategies

We evaluated the following Continual Learning strategies:

Cumulative

The cumulative strategy involves re-estimating model parameters using data from the current learning experience as well as all the data encountered in previous experiences. This approach utilizes all available information up to that point, but it can be computationally expensive and may not be applicable if data from certain time periods is not accessible due to privacy or regulatory concerns.

Continual fine-tuning

The continual fine-tuning strategy is based on an incremental fine-tuning process. At each learning experience (in our case, the year), the model weights are initialized with the weights from the previous experience. Training at the current experience only considers the data from that particular experience.

Experience replay

The experience replay strategy relies on an external memory, known as a replay buffer, with a predefined size B. This buffer stores data samples from previous learning experiences. At each experience, data samples are sampled from the replay buffer, allowing the model to retain information about previous data patterns. This approach does not require as much computational resources as the cumulative strategy.

Synaptic intelligence

Synaptic intelligence (Zenke et al., 2017) is a regularization-based strategy that mitigates catastrophic forgetting by incorporating a knowledge retention penalty into the loss function. Unlike the previous strategies, it does not rely on resampling or storing data from all previous experiences. The loss function to optimize at experience e follows the structure:

$$L_e = H_e + c \sum_{k=1}^{K} \Omega_k^e (\tilde{\theta}_k - \theta_k)^2 \tag{5.2}$$

Here, H_e represents the standard loss to minimize at experience e (in our case, the per-class weighted cross-entropy loss), c is a global dimensionless weighting parameter, Ω_k^e is the per-parameter regularization strength for parameter k and experience e, $\tilde{\theta}_k$ denotes the value of parameter k at the previous experience, and θ_k represents the value of parameter k at the current learning experience.

5.3.6 Evaluation

To evaluate and compare the advantages and disadvantages of each Continual Learning strategy, as well as to assess their performance in comparison to the baseline techniques, we calculated their backward and forward transfer (Lopez-Paz & Ranzato, 2017). Backward transfer refers to how learning from a particular experience affects prior knowledge, while forward transfer refers to how learning from a specific experience influences the acquisition of future knowledge.

For each Continual Learning strategy l and experience e, we computed the backward and forward transfer using the following formulas:

$$BWT_e^l = \frac{1}{e-1} \sum_{i=1}^{e-1} M_i^l \tag{5.3}$$

$$FWT_e^l = \frac{1}{E-e} \sum_{i=e+1}^{E} M_i^l \tag{5.4}$$

$$\tag{5.5}$$

Here, M_i^s represents the value of the performance metric of strategy l at experience i, and E denotes the total number of experiences (in our case, years).

Additionally, we calculated the global backward and forward transfer, which provides an average performance estimation across all experiences for a specific strategy:

$$BWT_{global}^l = \frac{1}{E-1} \sum_{j=1}^{E-1} BWT_j^l \tag{5.6}$$

$$FWT_{global}^l = \frac{1}{E-1} \sum_{j=1}^{E-1} FWT_j^l \tag{5.7}$$

$$\tag{5.8}$$

Furthermore, we utilized multiple evaluation metrics to assess the performance of each strategy. Specifically, we obtained the AUC, accuracy, recall, precision and F1-score.

Finally, 95% confidence intervals for the global backward and forward transfer were estimated for each strategy. To derive them, we followed the next expression:

$$CI_l^{95\%} = \bar{m} \pm 1.96 \frac{s_l}{\sqrt{E}} \tag{5.9}$$

Here $\bar{m}$ will correspond to the BWT_{global}^l or FWT_{global}^l, and s_l is the sample standard deviation computed with the series of BWT_e^l or FWT_e^l.

5.3.7 Hyperparameter tuning

To determine the optimal hyperparameters, an automatic active learning approach (Settles, 2009) was employed. For each Continual Learning strategy, a set of hyperparameters was defined, including parameters such as learning rate and batch size. Additionally, a range of values was proposed for each hyperparameter. For example, for the learning rate, values of 0.0001 and 0.00001 were considered, while for the batch size, values of 16 and 32 were explored. The sampling space for the hyperparameters was discrete to avoid overfitting issues due to the curse of dimensionality.

A Bayesian optimization strategy was then employed, where an auxiliary probabilistic generative model was iteratively trained. The purpose of this model was twofold: 1) to estimate the probability of the objective performance metric (in this case, the weighted cross-entropy) given a specific set of hyperparameters, and 2) to sample new hyperparameter values on each iteration in the hope of improving the performance metric.

Once the optimal hyperparameters were determined through these experiments on the pure training and validation sets, they were used for the final retraining stage of each strategy. The models were retrained using the full training set, and the performance metrics reported in this work were obtained from the test set.

5.4 Results

5.4.1 Dataset shifts assessment

Prior probability shift

The empirical probability of the life-threatening class over time is illustrated in Figure 5.1.

The plot reveals two distinct drops in the class probability: one occurring between the years 2012 and 2014, and another between 2016 and 2017. However, from 2009 to 2012, the life-threatening class probability showed a gradual increase. Furthermore, the empirical probability appears to stabilize qualitatively in the remaining time periods, namely between 2014 and 2016, and between 2017 and 2019.

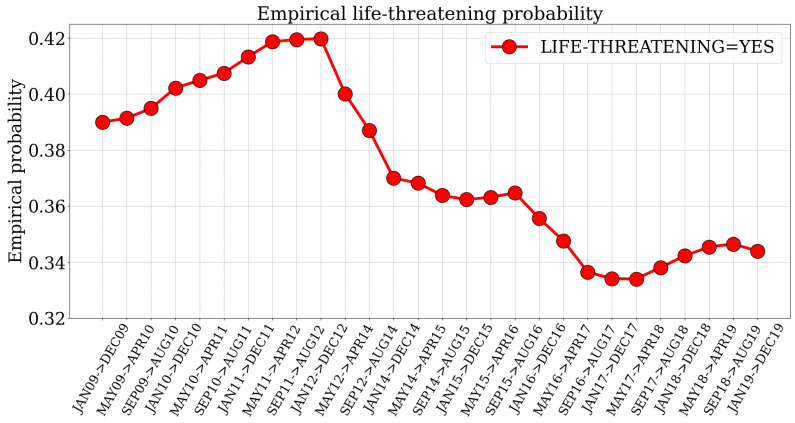

Figure 5.1: Empirical life-threatening probability over time. A significant drop in the class probability is observed over time.

In addition, the Kwiatkowski–Phillips–Schmidt–Shin test (Table 5.3) suggests rejecting the null hypothesis of stationarity.

Table 5.3: P-value of the Kwiatkowski–Phillips–Schmidt–Shin test. Assessment of the stationarity of the empirical life-threatening probability distribution over time.

Kwiatkowski–Phillips–Schmidt–Shin test
.018*

The findings from Figure 5.1 and Table 5.3 confirm the presence of a prior probability shift in our data.

Covariate shift

Figure 5.2 presents the performance, in terms of AUC, of DistilBERT text classification models trained to predict the year from which the data originated. The models utilized the free text dispatcher observations as input features.

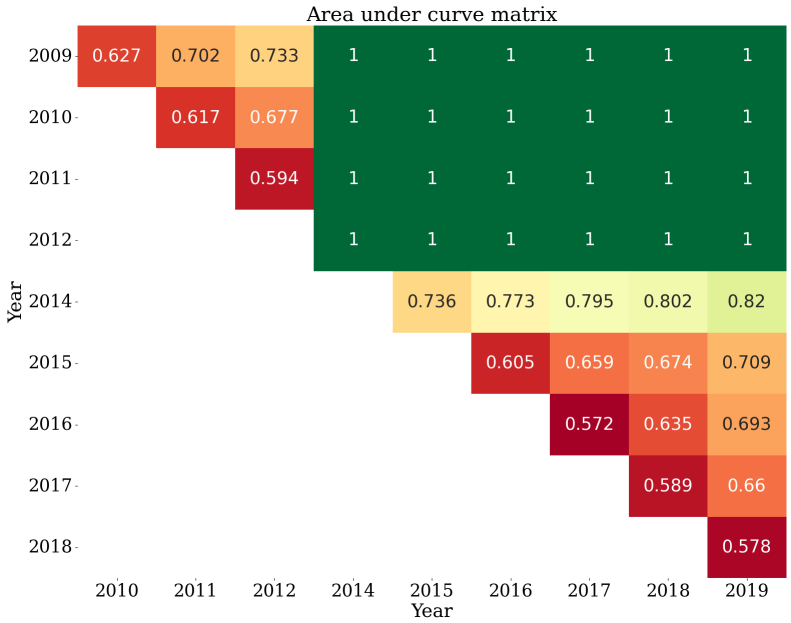

Figure 5.2: Area under the curve matrix of DistilBERT text classification models predicting the year from which the data originated. An abrupt covariate shift is observed between the 2009-2012 data batch and the 2014-2019 data batch.

As shown in Figure 5.2, the AUC values are consistently higher than those expected from a random model, which would have an AUC around 0.5. There is a clear distinction in the writing style of the free text fields between the 2009-2012 period and the 2014-2019 period, as indicated by the AUC of 1 on the test set. Moreover, within each time window, the AUC values gradually increase over time.

These observations confirm the presence of a distinct and abrupt covariate shift between the 2012 and 2014 periods, with smoother and gradual changes occurring within the 2009-2012 and 2014-2019 time windows.

Concept shift

The performance of the DistilBERT models trained to assess the presence of concept shift is depicted in Figure 5.3:

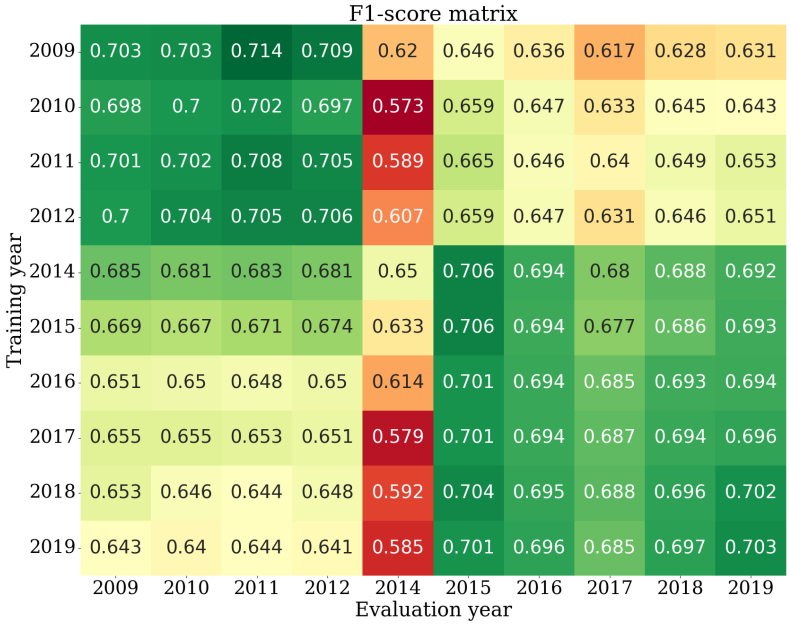

Figure 5.3: F1-score matrix of DistilBERT text classification models trained on data from one year (y-axis) and evaluated on the test set of all years (x-axis). A moderate performance drop is observed between the 2009-2012 batch and the 2015-2019 batch, with the year 2014 showing the lowest performance.

As illustrated in Figure 5.3, there is a significant performance drop in all models from 2009 to 2012, with the lowest F1-score observed in 2014. This confirms the presence of concept shift. Although there is a slight recovery in performance from 2015 to 2019, it still remains far from the values observed in the first period. Furthermore, the models trained on the 2015-2019 data show consistent performance within that time window, but experience a notable performance drop in the year 2014. This drop is less severe in the 2009-2012 models but is still noticeable. Thus, the existence of concept shifts is confirmed.

5.4.2 Continual learning

Backward transfer

The backward transfer, measured by the F1-score, for the Continual Learning strategies and baseline techniques is presented in Figure 5.4. The x-axis represents the model's performance in a specific year, while the y-axis indicates the average F1-score obtained when testing the model with data from previous years.

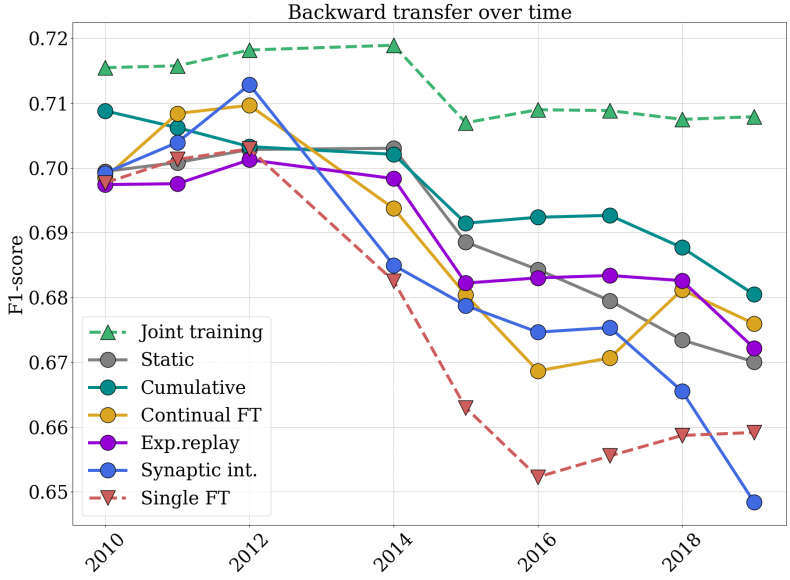

Figure 5.4: Backward transfer over time, spanning from 2010 to 2019 (excluding 2009 due to the lack of available data for 2008) computed using the F1-score. Continual Learning strategies enhance knowledge retention over time. Abbreviations: FT, fine-tuning; Exp, experience; int, intelligence.

As depicted in Figure 5.4, Continual Learning strategies prevent significant performance drops compared to not utilizing Continual Learning techniques. All Continual Learning strategies perform above the expected lower performance bound defined by single fine-tuning. Furthermore, there is a clear trend of decreasing average F1-scores over time, with a more pronounced drop in 2015 when 2014 is included in the backward transfer computation. However, the performance decrease is moderate rather than severe.

When comparing the different techniques, joint training stands out as the approach offering the best overall performance over time in terms of backward transfer measured by the F1-score. It serves as the upper baseline, as expected. The static baseline exhibits a performance decrease over time, although not as severe as some of the other approaches assessed. Among the Continual Learning strategies, the cumulative approach performs the best, while single fine-tuning represents the lower performance bound in terms of backward transfer.

Table 5.4 presents the global backward transfer, computed for the AUC, accuracy, recall, precision, and F1-score, along with their 95% confidence intervals.

Table 5.4: Global backward transfer for each reference metric, with 95% confidence intervals shown in brackets. Abbreviations: AUC, area under curve; JT, joint training; ST, static; CM, cumulative; CFT, continual fine-tuning; ER, experience replay; SI, synaptic intelligence; SFT, single fine-tuning.

Strategy	AUC	Accuracy	Recall	Precision	F1-score
JT	0.809[0.808,0.809]	0.766[0.764,0.767]	0.744[0.742,0.746]	0.685[0.684,0.686]	0.712[0.711,0.713]
ST	0.823[0.821,0.824]	0.737[0.733,0.74]	0.747[0.743,0.751]	0.647[0.64,0.654]	0.689[0.686,0.692]
CM	0.793[0.787,0.798]	0.755[0.754,0.757]	0.72[0.713,0.726]	0.677[0.675,0.68]	0.696[0.694,0.698]
CFT	0.8[0.795,0.805]	0.763[0.762,0.764]	0.672[0.663,0.681]	0.711[0.708,0.715]	0.687[0.684,0.691]
ER	0.753[0.747,0.758]	0.754[0.753,0.756]	0.698[0.692,0.703]	0.683[0.681,0.685]	0.689[0.686,0.691]
SI	0.809[0.805,0.813]	0.763[0.762,0.765]	0.657[0.645,0.67]	0.721[0.715,0.727]	0.683[0.678,0.687]
SFT	0.753[0.759,0.765]	0.76[0.759,0.761]	0.643[0.632,0.655]	0.72[0.715,0.724]	0.675[0.67,0.679]

Table 5.4 demonstrates statistically significant differences $\alpha = 0.05$ between the implemented Continual Learning pipelines and the lower baseline—since the 95% confidence intervals are not overlapping (Rosner, 2015)—indicating that Continual Learning techniques lead to performance improvements compared to not utilizing them. Among the Continual Learning strategies, the cumulative approach exhibits the best overall performance.

Forward transfer

Figure 5.5 illustrates the forward transfer, measured by the F1-score, for the Continual Learning strategies and baseline techniques. The x-axis represents a specific year, and the y-axis indicates the average F1-score obtained when testing the model with data from the incoming years.

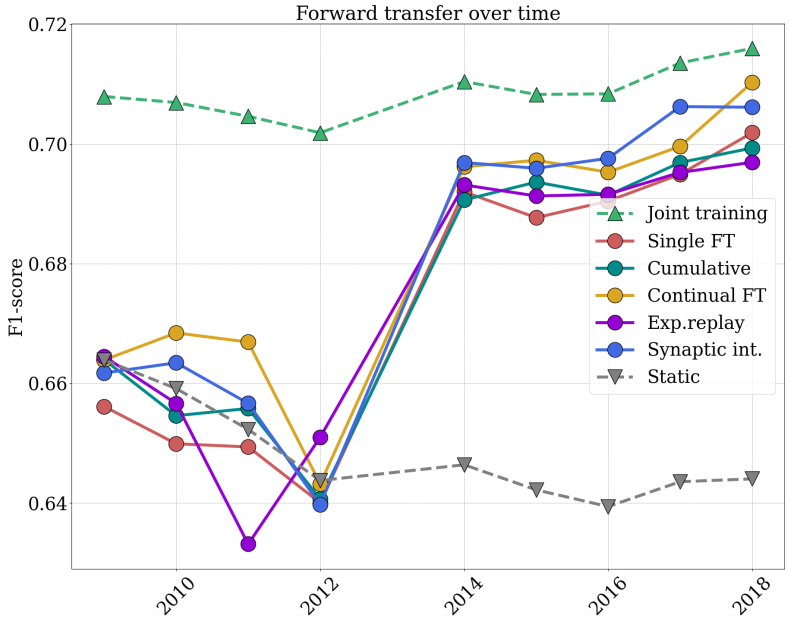

Figure 5.5: Forward transfer over time, spanning from 2009 to 2018 (excluding 2019 due to the lack of available data for 2020) computed using the F1-score. Continual Learning strategies are crucial for enabling forward knowledge transfer over time. Abbreviations: FT, fine-tuning; Exp, experience; int, intelligence.

As observed in Figure 5.5, Continual Learning strategies exhibit a distinct behavior compared to the baselines. The Continual Learning techniques show a common trend, with a notable increase in forward transfer in 2014. On the other hand, the baselines demonstrate the expected upper and lower bounds, with joint training serving as the upper bound and the static approach as the lower bound. Among the Continual Learning strategies, there is no clear winner as they interconnect over time, although continual fine-tuning and synaptic intelligence appear to perform better.

Table 5.5 presents the global forward transfer, computed for the AUC, accuracy, recall, precision, and F1-score, along with their 95% confidence intervals.

Table 5.5: Global forward transfer for each reference metric, with 95% confidence intervals shown in brackets. Abbreviations: AUC, area under curve; JT, joint training; SFT, single fine-tuning; CM, cumulative; CFT, continual fine-tuning; ER, experience replay; SI, synaptic intelligence; ST, static.

Strategy	AUC	Accuracy	Recall	Precision	F1-score
JT	0.818[0.817,0.819]	0.792[0.79,0.794]	0.724[0.723,0.725]	0.697[0.695,0.699]	0.709[0.708,0.71]
SFT	0.78[0.778,0.783]	0.755[0.746,0.765]	0.718[0.708,0.729]	0.644[0.631,0.657]	0.674[0.668,0.679]
CM	0.811[0.808,0.814]	0.749[0.739,0.758]	0.745[0.735,0.755]	0.628[0.616,0.641]	0.676[0.671,0.681]
CFT	0.814[0.811,0.817]	0.757[0.749,0.765]	0.743[0.737,0.75]	0.638[0.627,0.649]	0.682[0.677,0.687]
RP	0.789[0.784,0.795]	0.752[0.744,0.761]	0.732[0.721,0.743]	0.635[0.623,0.646]	0.675[0.67,0.68]
SI	0.823[0.819,0.826]	0.759[0.75,0.768]	0.73[0.722,0.737]	0.646[0.633,0.658]	0.68[0.675,0.686]
ST	0.811[0.811,0.812]	0.694[0.693,0.696]	0.808[0.805,0.811]	0.545[0.541,0.55]	0.648[0.646,0.65]

Table 5.5 indicates statistically significant differences ($\alpha = 0.05$) between the implemented Continual Learning pipelines and the lower baseline—since the 95% confidence intervals are not overlapping (Rosner, 2015)—implying that Continual Learning techniques lead to improvements compared to not utilizing them. Among the Continual Learning strategies, continual fine-tuning and synaptic intelligence stand out.

5.5 Discussion

5.5.1 Relevance

The findings of our study underscore the criticality of employing Continual Learning strategies for effective backward and forward knowledge transfer. To ensure the sustained performance of our EMCI classifier over time, the utilization of Continual Learning techniques becomes imperative. Importantly, our study represents the first investigation to incorporate Continual Learning within the learning pipelines of deep models designed for emergency triage support.

The identified dataset shifts, encompassing prior probability shifts, covariate shifts, and concept shifts, align closely with the changes implemented by the Health Services Department of the Valencian Community in 2014 regarding information system and coordination protocols. While these shifts may not be drastic, they are notable and should not be disregarded. Particularly, significant shifts in data distribution, pertaining to concepts and application-based data, can severely impede model performance. Consequently, our argument follows that the capacity to effectively handle these moderate yet significant data shifts may enhance model resilience when faced with more substantial changes in the future.

Figure 5.6 provides a comprehensive overview of the global backward and forward metrics, specifically focusing on F1-score, as discussed in the previous section. This visual representation clearly demonstrates the indispensability of Continual Learning strategies in mitigating catastrophic forgetting, as they facilitate the accumulation of knowledge over time, while also enabling effective knowledge forward transfer.

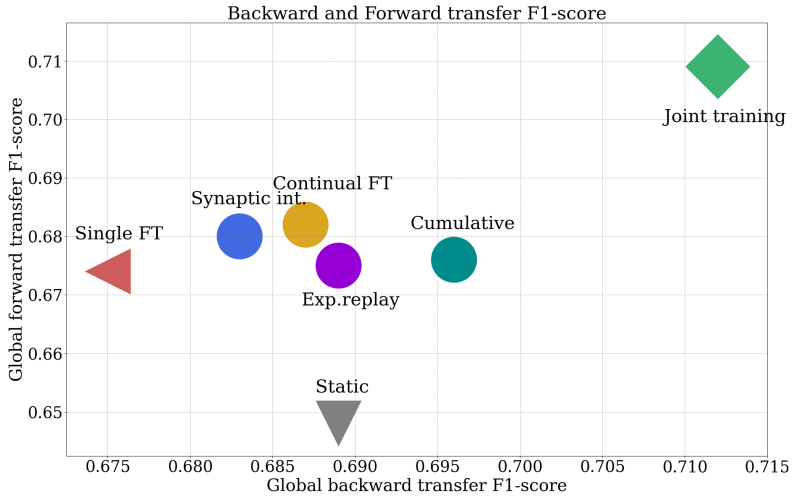

Figure 5.6: Global backward and forward transfer, computed with the F1-score. Continual Learning strategies play a vital role to enhance backward and forward knowledge transfer. Abbreviations: FT, fine-tuning; Exp, experience; int, intelligence.

Among the different Continual Learning strategies evaluated, both the cumulative and experience replay approaches exhibit similar behavior. The cumulative strategy can be viewed as an experience replay technique with unlimited memory. These two strategies outperform the others in terms of knowledge retention. However, in terms of knowledge transfer, the synaptic intelligence and continual fine-tuning approaches showcase superior performance, as they yield more positive impacts on predictive performance in subsequent years.

Considering the specific nature of our problem, where forward transfer holds greater significance than backward transfer, and taking into account the computational resources required for training time and memory, it would be reasonable to lean towards adopting a continual fine-tuning approach to address our problem. This choice offers several advantages, including easier integration into the retraining routine associated with the model, which can be seamlessly embedded into a deployed decision support system for emergency triage.

The continual fine-tuning approach allows us to capitalize on its superior ability to facilitate knowledge transfer and enhance predictive performance in subsequent years. Additionally, it offers practical benefits in terms of computational efficiency and resource utilization, which are valuable considerations when dealing with the constraints of our problem. By selecting this approach, we can effectively balance the importance of forward transfer, the available computational resources, and the ease of integration into our existing model retraining processes.

5.5.2 Limitations

The primary limitation of our work lies in the significant uncertainty associated with the phone triage process. Since data collection occurs remotely, within a time-critical context, the information gathered is often incomplete. Consequently, any model involved in providing decision support must rely on limited incoming data, which can introduce biases in certain cases. This inherent challenge imposes constraints on the achievable performance of any Machine Learning support model.

5.5.3 Future work

Regarding future endeavors, we identify two main directions for further exploration. Firstly, we propose a multitask Continual Learning approach to address the problem, incorporating considerations for admissible response delays and the jurisdiction labels of the emergency system. Secondly, we suggest the inclusion of additional input features, such as demographics, contextual information, or structured clinical features, thus forming a multimodal Continual Learning approach. These avenues of research hold promise for advancing the field and expanding the scope of our investigations.

5.6 Conclusions

In this work, we have conducted an extensive investigation into dataset shifts and Continual Learning strategies within the domain of EMCI triage. Our study provides compelling evidence of prior probability shifts, covariate shifts, and concept shifts within our data, which directly impact the performance of models over time. The utilization of Continual Learning strategies has been demonstrated as crucial in mitigating the adverse effects caused by distributional drifts, both in terms of backward and forward knowdledge transfer. Consequently, adopting a Continual Learning approach becomes highly valuable in maintaining the quality of clinical decision support within the context of EMCI triage. The implementation of the Continual Learning routines developed in this study for EMCI triage will have a significant and positive direct impact on patient well-being and the sustainability of health services.

Chapter 6

Deep continual multitask classification of emergency medical call incidents under dataset shifts affecting feature domain

The in-house triage protocol utilized by the emergency medical dispatch service of the Valencian Region, has undergone modifications since its inception. Additional branches have been integrated, while preceding questions have been removed in accordance with the specific needs and handling of distinct EMCI cases. Furthermore, as referenced in prior chapters, significant alterations occurred during the transition from the CORDEX to CoordCom information system in 2013, significantly impacting the structure of the clinical protocol. Dispatcher expertise and training have similarly progressed over time, shaping the utilization of this in-house triage protocol. Within this current chapter, we investigate the presence of dataset shifts, focusing on the clinical variables along with the three severity labels. Hence, this constitutes a multitask approach, spanning from 2009 to 2019, totaling 1 414 575 EMCI cases. Multiple deep continual pipelines were developed, considering two key aspects: the manner in which model parameters should be updated over the years—comprising static, cumulative, from-scratch, and fine-tuning approaches—and how variations in the feature domain impacted model performance over time. This evaluation entailed a static, a dynamic and a predefined approach. Our findings reveal that fluctuations in performance cannot be disregarded. Simultaneously, we posit that fine-tuning models in combination with the allocation of additional dimensions to incorporate new variables, present an effective and efficient solution for their gradual updates over time. Lastly, when taking into account this latter approach and excluding the year 2014, during which the information system underwent a change, it becomes apparent that performance

fluctuations in the upcoming years are constrained. Specifically, the variability in terms of F1-score performance across all three labels remained stable within 5% rate change.

The contents of this chapter are being submitted to the journal npj Digital Medicine—thesis contributions C3, C4 and P6.

6.1 Introduction

Out-of-hospital emergency medical triage is a complex challenge, demanding fast-paced decisions with great uncertainty, all within a context where errors can potentially lead to fatal outcomes. The professionals responsible for these critical decisions—emergency medical dispatchers—undergo specialized training programs designed to optimize their ability to handle incidents appropriately (Stratton, 1992). However, despite their knowledge, emergency medical dispatch centers provide clinical guidelines to dispatchers, outlining the prescribed procedures for conducting the triage process. This not only aids dispatchers in their task but also serves to minimize variability among professionals, ensuring a more equitable level of assistance (Farand et al., 1995).

The aforementioned set of guidelines employed is commonly referred to as clinical protocols. This category encompasses a wide array of protocols, with some of the most renowned ones being the Manchester Triage System (Mackway-Jones et al., 2013), the Canadian Triage Scale (Murray et al., 2004), the Emergency Severity Index (Gilboy et al., 2012) or the Australasian Triage Scale (Considine et al., 2004). Despite the diverse range of protocols and their individual distinctions, they exhibit significant shared characteristics. One of the most prominent commonalities is their structural arrangement in the form of decision trees, comprising clinical inquiries linked to various branches. Based on the responses offered by the caller, a distinct pathway is followed, culminating in a terminal node marked with the priority level to be assigned to the incident.

Within the domain of out-of-hospital emergency medical triage in the Valencian Region, an in-house triage protocol was conceived by experts within the HSD of the region. Initially inspired by the Manchester Triage System, the protocol underwent iterative adaptations over time, drawing upon the insights and expertise of coordinator physicians. Notably, certain unique features were integrated based on localized requirements. For instance, given the prevalence of pyrotechnic accidents in the region, these in-house triage protocol incorporates a dedicated branch to address such incidents—an aspect not found in the Manchester Triage System. Consequently, the protocol exhibits a hierarchical structure, featuring queries linked to distinct branches. The responses to these queries correspond to values attributed to structured clinical variables, culminating in final leaf nodes that are related to specific priority levels.

Nonetheless, as examined in the preceding chapter of this thesis, the phenomenon of distributional drifts (Moreno-Torres et al., 2012; Quinonero-Candela et al., 2008) manifests over time. In the context of healthcare processes and medicine, these distributional variations are intrinsic (Sáez & García-Gómez, 2018; Sáez et al., 2020), and out-of-hospital emergency medical triage processes in the Valencian Region are no exception. The occurrence of these shifts is attributed to a multitude of factors. Foremost, it is imperative to underscore the pivotal alteration in the information system during 2013, which engendered substantial shifts in protocols, personnel and emergency coordination. Furthermore, changes in telephone operators, targeted training initiatives, updates to clinical variables via the evolution of the in-house triage protocol—led by coordinator physicians—have collectively exerted their impact over time.

Hence, a meticulous examination of the implications of these transformations on data distributions becomes imperative, particularly when aiming to implement a Machine Learning-based system. Given that the outcomes of such a system could be impacted by disparities in data distributions across time, it is paramount to assess the extent of this variability. To achieve this, an initial quantitative analysis focusing on dataset shifts, encompassing evaluations of prior probability shifts, covariate shifts, and concept shifts, stands as an essential preliminary stage.

Upon completing the evaluation and exploration of potential distributional alterations, the incorporation of mechanisms to mitigate possible adverse consequences stemming from these shifts is imperative. The objective is to sustain model performance at a consistent level. Thus, the incorporation of Deep Continual Learning techniques (Parisi et al., 2019) is mandatory. These techniques not only retain valuable knowledge for subsequent experiences but also offer the necessary adaptability to promptly acclimate to new changes that usher in a paradigm shift.

In this chapter, we have developed multiple Deep Continual Learning pipelines, drawing from the most effective and efficient strategies identified in the previous chapter, considering its effectiveness and efficiency. Besides, we introduce the requirement of not solely considering the evolution of model weights but also recognizing the emergence of novel clinical features alongside the disappearance of existing ones over time.

6.2 Materials

6.2.1 Dataset

We considered a total of 1 414 575 independent EMCI from the Health Services Department of the Valencian Region, compiled from 2009 to 2019, excluding 2013—since the emergency information system changed during that year.

The EMCI data employed in these studies encompassed both during-call and after-call data. During-call data were recorded during the emergency medical call and included the clinical tree variables and values associated to the path followed by the dispatcher in the in-house decision tree. Some examples of possible clinical features sets associated each one to a different incident are: 1) "Previous trauma: no; Shortness of breath: yes; Nasal congestion: no" and 2) "Active arrhythmia: yes; History: cardiac pathology; Dizziness: yes; Incident location: public road/street". These data were used at inference time as input for the prediction. On the other hand, after-call data were recorded at a time after the call. They include physician diagnosis, hospitalizations, urgency stays, maneuvers and procedures the patient underwent. After-call data were used offline—i.e., not in prediction time—to infer the output variables of the predictive model: if the emergency event implied or not a life-threatening situation, which was the admissible response delay—undelayable, minutes, hours, days—and if the event was jurisdiction of the emergency system or primary care.

6.2.2 Framework

The implementation language of our experiments was Python (G. van Rossum (Guido), 1995), using the libraries Numpy (van der Walt et al., 2011) and Pandas (McKinney, 2010) for data management. To implement and train the designed models we considered PyTorch (Paszke et al., 2017) and HuggingFace's Transformers (Wolf et al., 2019). Finally, we used Optuna (Akiba et al., 2019) for hyperparameter tuning.

6.3 Methods

6.3.1 Data preparation

Prior to conducting any analyses involving dataset shifts or Deep Continual Learning pipelines, we prepared our data using the following procedures:

Concerning the clinical variables, our initial step involved harmonizing the newly introduced variables post-2013 and their counterparts from the previous period. This harmonization was pursued as many of these novel features retained identical meanings to those of existing variables, with the only distinction being changes in nomenclature. Recognizing the significance of maintaining accuracy and mitigating potential biases, this correspondence mapping process was meticulously supervised and validated by experts specializing in out-of-hospital medical emergencies. These experts were affiliated with the HSD of the Valencian Region, the very individuals responsible for curating and providing the crucial data that underpinned our study.

Labels categories were encoded through one-hot encoding. This encoding method facilitated the subsequent integration of these labels into the Deep Learning models that were developed. As such, from the life-threatening label, two distinct one-hot encoded variables were derived. Likewise, four such variables originated from the admissible response delay label, and an additional two stemmed from the emergency system jurisdiction label.

Next, we proceeded to segment our data in order to capture temporal variations while simultaneously mitigating concerns related to overfitting. To achieve this, we initially partitioned our dataset into distinct learning experiences (Lomonaco et al., 2021). The initial experience comprised data solely from the CORDEX system (2009-2012), and subsequently, each ensuing experience corresponded to a specific year under the CoordCom information system (2014-2019). This partitioning strategy was selected due to forthcoming architectural and training variations within the Clinical subnetwork of the DeepECM^2 model (Ferri et al., 2021). This subnetwork was trained collectively on the entire CORDEX data batch, rather than in a year-by-year manner. Thus, if we intend to evaluate the potential negative impacts of architectural modifications, it is imperative to ensure the use of consistent data sets for performance metric comparisons.

Following this initial partition, a subsequent iterative division was performed for each of these experiences. For each experience, an initial split was conducted for training and testing, allocating 80% for training and 20% for testing. This partition served to gauge the actual model performance. Subsequently, the training set was further divided into a pure training subset and a validation subset, with proportions 70% and 30%, respectively. The validation subset was exclusively employed for hyperparameter tuning, with no inclusion of cases from the test set. Figure 6.1 represents how our dataset has been divided according to the exposed procedure.

Subsequently, we transformed the string feature values into indexes. This conversion was necessary to enable the subsequent utilization of an Embedding Layer (Bengio et al., 2000), which will map every index to a dense vector in our models. Additionally, we undertook padding and truncation operations to ensure a consistent sequence length, thereby fastening training processes.

It is pertinent to note that this index conversion process exhibited variations depending on the Deep Continual Learning strategy followed, as well as if we were working with a training set—training, pure training—or an evaluation set—validation, test set. Details about the specific generation of the feature to index maps are exposed in posterior sections, where Continual Learning techniques are described. Here we comment that those relations between the feature string identifier and its corresponding index value were learned and updated just in the training sets, since these feature to index maps must be kept in the evaluation sets to estimate the overfitting

effect of this preprocessing operation—otherwise, posterior performance metrics will be higher but misleading.

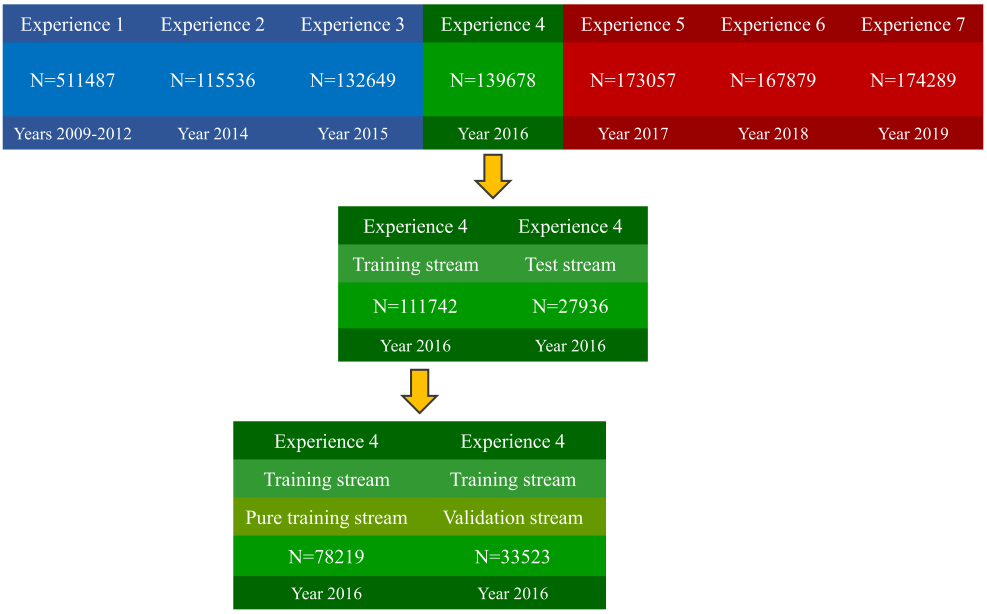

Figure 6.1: The data organization process involves the division of data into distinct learning experiences. Concurrently, each experience is characterized by three distinct, non-overlapping data streams: a pure training stream, a validation stream, and a test stream. Here, N symbolizes the volume of data within each experience and stream. The present learning experience is denoted in green, while future experiences—whose data remains unavailable—are depicted in red. In contrast, previous experiences—whose data has already been considered—are shaded in blue.

6.3.2 Dataset shifts assessment

As the phenomenon of dataset shift has already been introduced in the Rationale section, we do not provide an in-depth explanation in this section. Instead, we focus on elucidating how we assessed the three primary sources of drift in our work: prior probability shifts, covariate shifts, and concept shifts (Moreno-Torres et al., 2012).

Prior probability shifts

To assess the existence of prior probability shifts in our study, we computed the empirical class probabilities for each severity label—life-threatening, admissible response delay, and emergency system jurisdiction. Subsequently, we conducted an in-depth analysis to discern any trend in these probabilities across the temporal dimension.

Covariate shift

To assess the presence of distributional changes within these covariates over time, we computed the empirical probabilities for each individual input feature and scrutinized the temporal evolution of their respective trends.

Concept shift

To evaluate the existence of concept shifts, we aimed to compute the probabilities associated with each class for the three severity labels originating from the most frequent protocol pathways.By evaluating the extent of significant variations in these probabilities across time, one could infer the occurrence of a concept shift in the conditional probabilities. It is important to note, however, that a substantial disparity existed between the most frequent pathways in the CORDEX and CoordCom systems. Consequently, this significant discrepancy considerably diminished the available data for conducting this particular type of analysis. As a result, we were compelled to directly investigate the evolution of model performance across each of the predefined experiences.

6.3.3 Deep neural network design

Considering the outcomes detailed in Chapter 3, where Deep Learning models exhibited superior performance compared to other Machine Learning approaches, the focus in this chapter remains on models of a similar nature. However, in contrast to Chapter 3, the present chapter excludes the adoption of recurrent architectures or other sequential models like the Transformer (Vaswani et al., 2017). The primary rationale for this deviation lies in the unavailability of information regarding the order in which clinical variables were recorded during the call for CoordCom data. Consequently, our design had to center around a model that is order-independent when processing clinical variables. In essence, this model must generate consistent predictions even when presented with the same features in varying orders. Moreover, this model should adeptly handle the challenge posed by the emergence and disappearance of novel features over time.

In the subsequent section, we introduce the model that we developed to align with these specific requirements. Due to its inherent characteristics, we have named this model the *Clinical Invariant Network*, denoted as CliInvNet for brevity.

Clinical Invariant Network

The Clinical Invariant Network comprises a multitask (Caruana, 1997) deep neural network, constituted by two principal components: the Clinical Encoder and the Multitask Classifier. The Clinical Encoder, serving as the network's hard parameter sharing element, forms its core. Meanwhile, the Multitask Classifier contains distinct branches, each associated with a specific label. These branches are responsible for computing predicted probabilities for the various classes within each label.

Focusing on the Clinical Encoder, we constructed it with an initial Embedding Layer (Bengio et al., 2000). This layer facilitates the mapping of clinical variables, expressed as indexes, into dense vector representations, a significantly more efficient alternative to one-hot encodings. Moreover, this Embedding Layer enables the accommodation of novel features over time. We achieve this by pre-allocating a substantial number of entries within the corresponding lookup matrix without impacting subsequent architectural elements. Subsequent to the Embedding Layer, an Adaptive Average Pooling block (Szegedy et al., 2016) was employed. This component serves to aggregate the representations of all features within an observation into a singular representation. This functionality allows the network to accommodate varying numbers of features per entry. Additionally, the Adaptive Average Pooling Layer endows the network with order-invariant capabilities, preserving results even with altered feature orders. Following this, multiple dense blocks were introduced, each encompassing a Fully Connected Layer (Rosenblatt, 1958), Layer Normalization (Ba et al., 2016), a GELU activation function (Hendrycks & Gimpel, 2016), and a Dropout Layer (Hinton et al., 2012) to counteract neuron co-adaptation.

The Multitask Classifier, responsible for incorporating task-specific components into the architecture, consists of three branches. Each branch contains several dense blocks, culminating in an output block. These output blocks consist of a Fully Connected Layer followed by a Softmax activation function.

It is important to highlight that the selection of specific hyperparameter values, including the number of dense blocks and embedding dimensions, is discussed in a subsequent section focused exclusively on hyperparameter selection.

Illustrated in Figure 6.2, the main architecture of CliInvNet is visually represented:

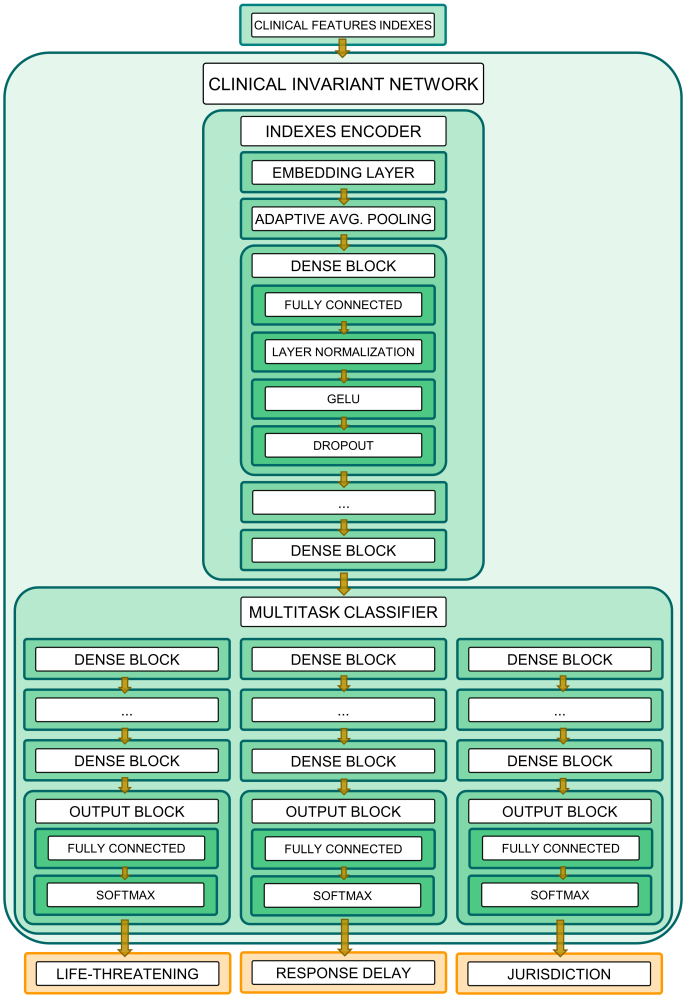

Figure 6.2: Clinical Invariant Network architecture.

6.3.4 Parameter tuning

Concerning the parameter tuning process, we used the AdamW (Loshchilov & Hutter, 2019) optimizer, a variant of the Adam (Kingma & Ba, 2017) algorithm. The feeding paradigm followed was a mini-batch training approach (Bertsekas, 1994), while the loss function considered was the soft F1-score (Janocha & Czarnecki, 2017). This choice was driven by its intrinsic suitability as a class-weighted metric, aligning

well with the argmax saturation procedures in the transition from output scores to the saturated predicted labels. An intriguing advantage here is that we need not fine-tune the threshold for each experience; instead, it remains constant, and the learning process inherently leverages class weighting through the loss. To further enhance our training, we incorporated a cosine annealing learning rate scheduler, which aligns aptly with deep transfer learning scenarios (Loshchilov & Hutter, 2017).

Likewise, it is pertinent to acknowledge that layers featuring ReLU activation functions were initialized using Kaiming initialization (He et al., 2015), whereas layers incorporating the softmax activation function were initialized with Xavier's initialization (Glorot & Bengio, 2010).

6.3.5 Continual Learning

Moving forward, we introduce the Continual Learning strategies designed to facilitate the model's adaptation across the distinct experiences over time. An essential distinction needs to be highlighted at this juncture—namely, the differentiation between approaches centered on scrutinizing the ramifications of coping with a feature domain that exhibits temporal fluctuations and methodologies geared towards the process of updating model weights.

Feature domain

We propose in this chapter three different strategies to deal with varying feature domains: a static domain strategy, a dynamic domain strategy, and a predefined domain strategy. We evaluated these three approaches, applying them across all the continual parameter updating strategies, which are presented in the posterior section. This systematic comparison enabled us to effectively discern the effects attributed to the weight updating process from those arising from the fluctuations in the feature domain.

Next, we present in detail each one of these feature domain strategies:

Static domain

The static domain strategy entails the utilization of the feature identifier-to-index conversion map from the CliNet within DeepEMC^2. In this approach, this map remains unchanged after the initial experience and is not updated subsequently. Nevertheless, it is essential to acknowledge that, as elucidated in the data preparation section, a mapping has been established between the CoordCom variables and CORDEX variables. Therefore, even when new variables emerge, if they maintain some correspondence with the CORDEX system, the map refers to a familiar clinical

feature. Conversely, for features that emerge over time without any correspondence to prior CORDEX variables, they are linked to the index that designates unknown or infrequent nodes within the CliNet. Thus, despite their novelty, we can still work with them.

Hence, under the static domain approach, the number of active entries of the Embedding Layer of the CliInvNet remains unchanged across all the learning experiences, although the value of those dense representations that are enabled varies over time as the model learns from new data.

Dynamic domain

The dynamic domain strategy is based on the recurrent update of the feature identifier-to-index map with each new experience—confined exclusively to the training sets, as previously elucidated. Consequently, we establish a frequency threshold, mirroring the one employed in the CliNet, to discern when a feature qualifies as infrequent. Such features are then either assigned to the unknown index or mapped to a distinct integer designated solely for that feature. Across the series of experiences, we monitor and revise the cumulative absolute frequency of each feature's occurrences. This iterative process facilitates the emancipation of features that were initially mapped to the unknown integer, permitting their adaptation in subsequent experiences and preventing them from becoming stagnant.

Hence, under the dynamic domain approach, the number of active entries of the Embedding Layer of the CliInvNet varies over the learning experiences, as long as the cumulative clinical variable frequency surpasses the required threshold. In addition, the value of those enabled dense representations vary over time as the model learns from new data.

Furthermore, it is imperative to underscore that while the index mapping fluctuates over time, the index used to represent infrequent features remains constant. This standardization ensures the avoidance of overlap and the introduction of noise. It is particularly important as certain subsequent parameter updating strategies involve amalgamating data across multiple experiences.

Predefined domain

The predefined domain strategy employs a distinctive approach to handle feature domain changes over time. At the core of this methodology is the use of a predefined embedding matrix derived from a large pretrained natural language processing model. Specifically, for this work, we selected the ALBERT model (Lan et al., 2020) pretrained on a Spanish corpus (Face., 2023). This choice is motivated by the

fact that our dataset contains clinical variables originally in Spanish. Additionally, the dimensionality of the ALBERT model's embeddings closely matches that of the embeddings used in the static and dynamic approaches, enabling effective comparisons across these strategies.

It is essential to note that, under this feature domain approach, structured clinical variables are transformed into an unstructured natural language processing representation. Subsequently, we apply subword tokenization to the unstructured clinical data, breaking down the text into smaller, meaningful subtokens. After subword tokenization, we utilize the embedding matrix derived from the pretrained ALBERT model. This matrix allows us to obtain stable numerical representations for each subtoken. By leveraging a pretrained NLP model like ALBERT, we harness its capacity to capture semantic and contextual information from the clinical features in text format, thus enhancing the quality of our embeddings.

The predefined approach is intended to maintain stability through the consistent use of the ALBERT embedding matrix across all learning experiences. This ensures that the numerical representations for clinical variables remain robust and consistent, even as the model learns from new data.

Parameter updating over experiences

In the preceding subsection, we described the challenge posed by the variable feature domain across experiences. In this section, our focus shifts towards the dynamic adaptation of model weights across experiences. This adaptation aims to retain pertinent information for facilitating decision support in the forthcoming years, while simultaneously overwriting obsolete patterns and statistical associations. This plasticity effect introduces the capacity to assimilate novel knowledge.

In light of the findings from the previous chapter, which underscored similarities among certain Continual Learning strategies, our evaluation here will be directed primarily towards those strategies anticipated to exhibit distinct behaviors. Specifically, we will examine the cumulative strategy, prized for its capacity to accumulate knowledge, the from-scratch approach, esteemed for its resilience to past experience noise, and fine-tuning, identified for its adept balance between backward and forward knowledge transfer. Moreover, the interpretability of results from these techniques is notable, a crucial consideration as these techniques will be evaluated in conjunction with the feature domain approaches. Subsequently, we proceed to offer more detailed insights into each of these strategies:

From scratch

The from-scratch approach involves exclusively utilizing data from the current experience, necessitating the initialization of a new model and training it anew on each occasion. This approach may appear to be less advantageous, given that it incorporates a substantially smaller dataset compared to the majority of Continual Learning strategies. However, in scenarios where pronounced dataset shifts transpire over time, this approach may indeed be prudent. By eschewing the integration of noise from previous experiences into the current one, it emerges as a sensible option.

Fine-tuning

We consider this strategy given the results shown in the previous chapter. It strikes a balance by keeping some information about the past—as model weights are not initialized randomly—but favouring forward knowledge transfer, evading excessive anchoring to past experiences—a trait that could typify some other Continual Learning strategies.

Cumulative

In this strategy, data from the current experience is combined with data from all preceding experiences. Consequently, the volume of data employed for training expands with each new experience. This augmentation in data utilization brings about heightened computational demands and memory requirements. However, it offers the advantage of retaining a comprehensive record of previous data patterns. As a result, the model stands to benefit from a data accumulation standpoint—an advantageous attribute, given that Deep Learning models tend to exhibit enhanced performance with a larger pool of available data.

6.3.6 Hyperparameter tuning

Hyperparameter selection was carefully addressed in this study, as hyperparameters may have a substantial impact in the final performance.

An automatic active learning (Settles, 2009) approach was adopted. For each pipeline—comprising the feature domain approach coupled with the parameter updating strategy—an hyperparameter set was defined, e.g., learning rate, batch size. At the same time, for each hyperparameter, some range values were proposed, e.g., for learning rate we considered the values 0.0001 and 0.00001, and for batch size 32 and 64. Consequently, the sampling space allowed for the hyperparameters was

discrete, since a continuous one may derive in overfitting issues due to the curse of dimensionality (Bellman, 1956).

Subsequently, a Bayesian optimization strategy was followed, where an auxiliary probabilistic generative model was trained iteratively to 1) estimate the probability of the objective performance metric—in our case, the soft F1-score—given a set of hyperparameters and 2) sample new hyperparameter values on each iteration expecting to improve the performance metric.

Finally, it is crucial to emphasize that these *optimal* hyperparameters were derived from experiments conducted on the pure training and validation sets. Subsequent retraining was then carried out using the complete training set, with performance metrics calculated on the test set.

6.3.7 Evaluation

To evaluate the performance of each tested pipeline and determine the one best suited for consistent decision support over time, particularly in mitigating the adverse impact of performance drifts affecting feature domains, we calculated the F1-score associated with each severity label for every pipeline.

Specifically, we computed the F1-score for the positive class of the "life-threatening" label (i.e., the "life-threat" class) and the "jurisdiction" label (i.e., the "emergency system jurisdiction" class). For the "admissible response delay" label, we calculated the F1-score using macro-averaging, as we cannot designate a reference class among the four classes.

We computed these metrics for each experience, considering two approaches. First, we calculated them considering the training of the pipeline up to the current experience, allowing us to assess the absence of overfitting while estimating model performance in the current experience. Second, we computed them by training the model up to the previous experience. This approach helps us understand how model performance diminishes when applied to novel incoming data, which may exhibit variations in data distributions. Therefore, we obtain information about both in-sample and out-of-sample performance by considering these two assessments.

We then averaged and studied the performance for each feature domain and parameter updating strategy over the years to gain a better understanding of the effect of each approach. Additionally, we obtained non-parametric 95% confidence intervals using bootstrap resampling Efron and Tibshirani, 1994, with a total of 1000 resamples per pipeline.

It is worth noting that, even though it was not subjected to retraining, we incorporated the outcomes of the *Clinical subnetwork* from DeepEMC^2 into this eval-

uation. This inclusion served as a baseline, allowing for comparing the performance of the pipelines examined in this study.

6.4 Results

Next, the results of the analyses to determine the presence of dataset shifts are presented and described, along with the results relative to the evaluation of the Deep Continual Learning pipelines designed.

6.4.1 Dataset shifts assessment

Prior probability shifts

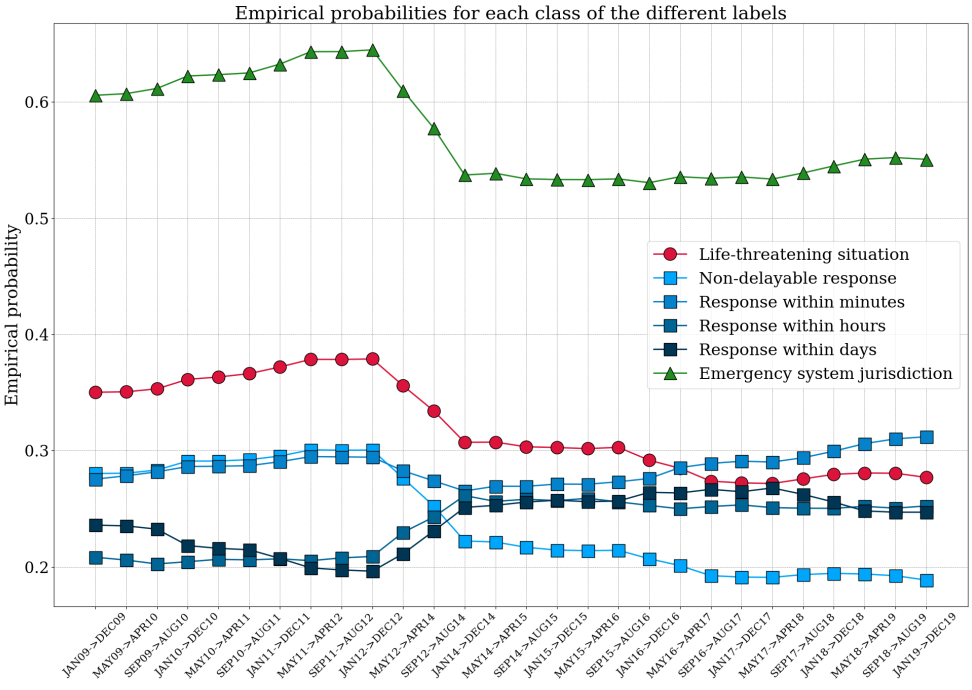

Figure 6.3: Prior probability shift assessment through empirical class probabilities.

In relation to the life-threatening label, a dataset shift is evident in terms of prior probability. Notably, a discernible decline in empirical probability is observed

between the CORDEX system and the CoordCom system, spanning the years 2012 to 2014. Furthermore, there is a noticeable shift in empirical probability between 2016 and 2017, albeit exhibiting a smoother transition. Consequently, a prior probability shift manifests within the life-threatening label.

Focusing on the empirical admissible response delay distributions, a marked and abrupt alteration in empirical probabilities becomes apparent between 2012 and 2014. Following this abrupt shift, a conspicuous and gradual drift is observable. During this drift, occurrences of undelayable events decrease while those associated with minutes continue to rise. Thus, a prior probability shift is also evident within the admissible response delay label.

Lastly, directing our attention to the emergency system jurisdiction label, a clear and abrupt transition can be inferred from the analysis of Figure 6.3. Subsequent to this transition, a discernible upward trend emerges, bearing semblance to the pattern observed in CORDEX, although operating within a distinct probability range. Based on this analysis, it is evident that a dataset shift concerning the emergency system jurisdiction label is present.

Covariate shift

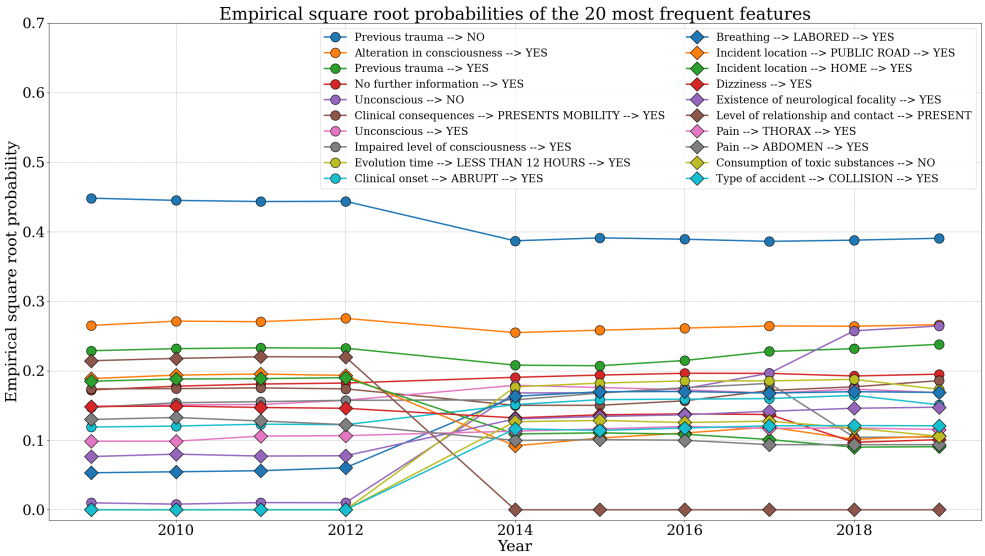

Figure 6.4: Square root of the empirical probabilities linked to the 20 most frequently occurring clinical features in our dataset.

Upon inspecting Figure 6.4 depicting covariate shifts, it becomes evident that a number of the most prevalent features present in CORDEX completely vanish during the transition to CoordCom. Conversely, it is noted that certain features that are highly documented in CoordCom were absent in the earlier CORDEX dataset. These scenarios collectively imply a significant covariate shift phenomenon. Furthermore, albeit of lesser magnitude, oscillations, and disruptions in trends are discernible, particularly during the CoordCom years. These irregularities become more pronounced in the transition from 2017 to 2018.

Concept shift

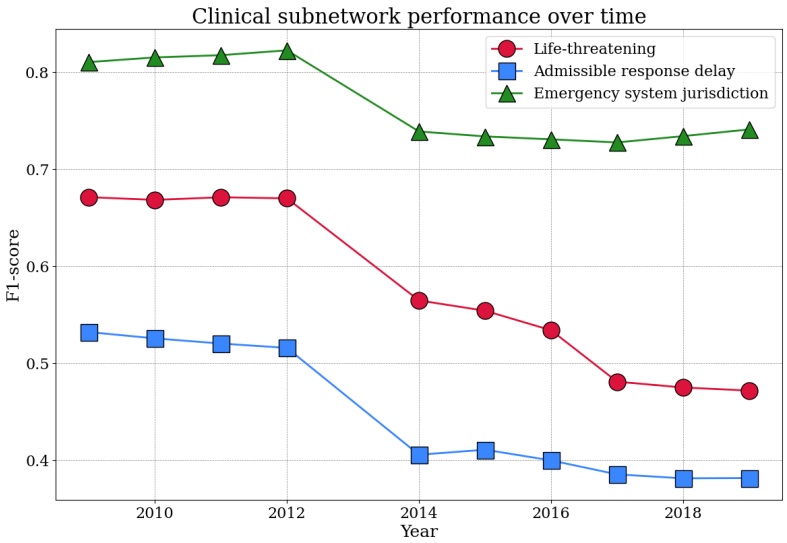

Figure 6.5: Performance of the Clinical subnetwork of the DeepEMC^2 model over time, for each of the three severity labels, in terms of F1-score. This F1-score is referenced to the positive class in the life-threatening and emergency system jurisdiction label, while macro-averaged for the admissible response delay label.

From the analysis of Figure 6.5, it is evident that dataset shifts are present. This is substantiated by the declining performance of the *Clinical subnetwork* over time, notably during the periods between 2012 and 2014, as well as between 2017 and 2018. However, the existence of a concept shift cannot be definitively established, as these performance variations are strongly correlated with the covariate shifts. Consequently, it is plausible that the performance fluctuations primarily stem from variations in input features and their scarcity in past years, rather than solely changes in conditional probabilities.

6.4.2 Continual learning

Next, we present the results achieved for the distinct Deep Continual Learning pipelines examined within this study. The differentiation is made between current and upcoming year performances, for each of the three severity labels.

Performance with training up to the current year

Life-threatening

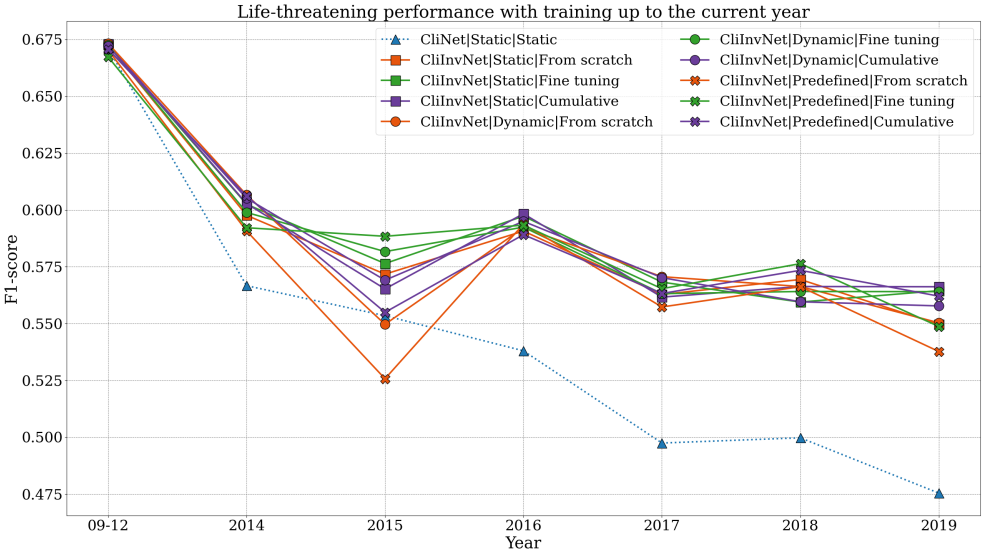

Figure 6.6: Life-threatening performance over time with training up to the current year for each pipeline.

Table 6.1: Average F1-score values for life-threatening performance with training up to the current year for each pipeline tested. Non-parametric 95% confidence intervals for each average value are provided between brackets.

Parameter updating	Feature domain			Mean
	Static	Dynamic	Predefined	
From scratch	0.588 [0.584, 0.591]	0.587 [0.584, 0.59]	0.577 [0.574, 0.58]	0.584 [0.581, 0.587]
Fine-tuning	0.592 [0.589, 0.595]	0.591 [0.588, 0.594]	0.59 [0.587, 0.593]	0.591 [0.588, 0.594]
Cumulative	0.591 [0.587, 0.594]	0.59 [0.587, 0.593]	0.589 [0.585, 0.592]	0.59 [0.587, 0.593]
Mean	0.59 [0.587, 0.593]	0.589 [0.586, 0.592]	0.585 [0.582, 0.588]	0.588 [0.585, 0.591]

Upon observing Figure 6.6, a discernible trend of declining performance becomes apparent, characterized by an extended continuity over time. Considering the *Clinical subnetwork* of DeepEMC^2, featuring a static domain and a static parameter updating strategy, it functions as an anticipated baseline, given its lack of retraining over time. When analyzing the effect of feature domains, distinguishing significant performance disparities between the static and dynamic approaches proves challenging, although the predefined domain method performs slightly worse. Shifting focus to the parameter updating strategies over time, marginal degradation in results is evident within the from scratch approach compared to the relatively analogous fine-tuning and cumulative strategies.

Admissible response delay

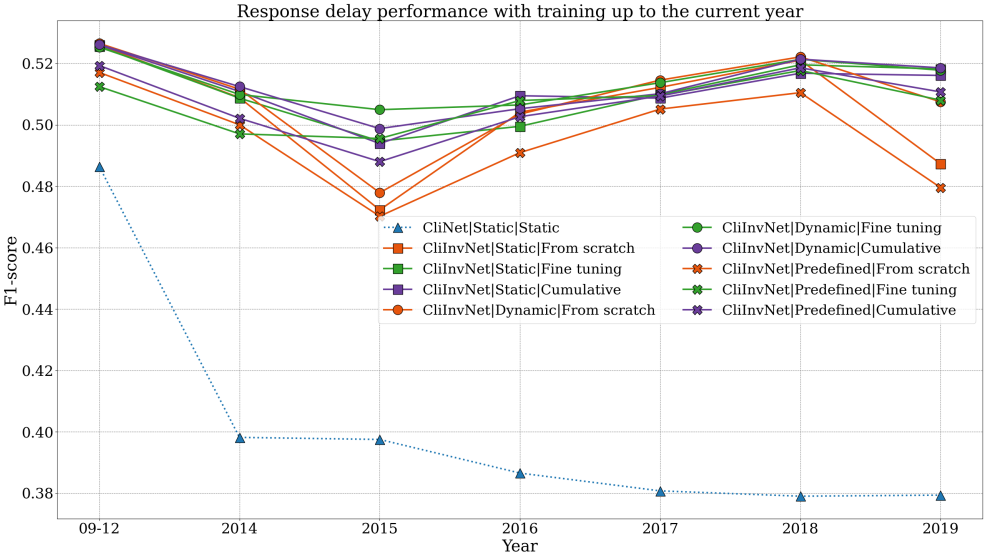

Figure 6.7: Admissible response delay performance over time with training up to the current year for each pipeline.

Table 6.2: Average macro F1-score values for admissible response delay performance with training up to the current year for each pipeline tested. Non-parametric 95% confidence intervals for each average value are provided between brackets.

Parameter updating	Feature domain			Mean
	Static	Dynamic	Predefined	
From scratch	0.505 [0.502, 0.507]	0.509 [0.507, 0.511]	0.496 [0.494, 0.498]	0.503 [0.501, 0.505]
Fine-tuning	0.511 [0.509, 0.513]	0.514 [0.512, 0.516]	0.507 [0.505, 0.509]	0.511 [0.508, 0.513]
Cumulative	0.512 [0.509, 0.514]	0.513 [0.511, 0.515]	0.507 [0.505, 0.509]	0.511 [0.509, 0.513]
Mean	0.509 [0.507, 0.511]	0.512 [0.51, 0.514]	0.503 [0.501, 0.505]	0.508 [0.506, 0.51]

Upon analyzing Figure 6.7, a minor performance dip in 2014 and 2015 is noticeable, subsequently recovering thereafter. Considering the *Clinical subnetwork*, it serves as the anticipated baseline, as it remains untrained. Its performance suffers significantly due to the domain shift in 2014, followed by a gradual yet smoother decline. Unlike the life-threatening label, where feature domain performances exhibited minimal disparity, discernible nuances emerge here, favoring the dynamic approach. Notably, average values for the dynamic feature domain consistently surpass those of the static approach as well as the predefined one, across all tested parameter updating strategies. Similar to the life-threatening label, parameter updating strategies mirror a similar pattern, with the from scratch approach yielding inferior results, overtaken by the fine-tuning and dynamic strategies. Notably, no pronounced discrepancies are discerned between the latter two.

Emergency system jurisdiction

Upon observing Figure 6.8, a general decline in performance becomes apparent, albeit within a functional range that preserves model utility. A parallel pattern to the life-threatening label manifests, albeit with a distinction: the emergency system jurisdiction label exhibits comparatively higher values, with a tendency to rebound slightly in the latter experiences. The *Clinical subnetwork* does not function as an anticipated baseline, surpassing other pipelines incorporating retraining. Correspondingly, equivalently to the life-threatening scenario, discernible discrepancies between the static and dynamic feature domain approaches remain absent, although, for the jurisdiction label, the predefined domain approach outperforms both the dynamic and static ones. Turning to the parameter updating strategies over time, their influence, though subtle, is evident. The fine-tuning strategy yields the best results, followed by the cumulative approach, and finally, the from-scratch approach.

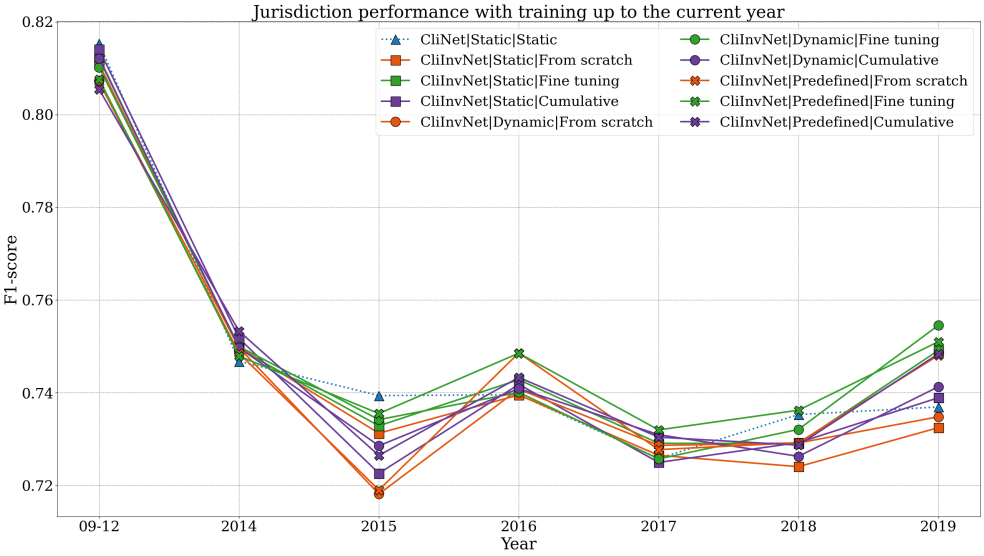

Figure 6.8: Emergency system jurisdiction performance over time with training up to the current year for each pipeline.

Table 6.3: Average F1-score values for emergency system jurisdiction performance with training up to the current year for each pipeline tested. Non-parametric 95% confidence intervals for each average value are provided between brackets.

Parameter updating	Feature domain			Mean
	Static	Dynamic	Predefined	
From scratch	0.745 [0.743, 0.747]	0.744 [0.742, 0.746]	0.747 [0.745, 0.749]	0.745 [0.743, 0.747]
Fine-tuning	0.749 [0.747, 0.751]	0.75 [0.748, 0.751]	0.751 [0.749, 0.753]	0.75 [0.748, 0.752]
Cumulative	0.746 [0.744, 0.748]	0.747 [0.745, 0.749]	0.748 [0.746, 0.75]	0.747 [0.745, 0.749]
Mean	0.747 [0.745, 0.749]	0.747 [0.745, 0.749]	0.749 [0.747, 0.751]	0.747 [0.746, 0.749]

Performance with training up to the previous year

Life-threatening

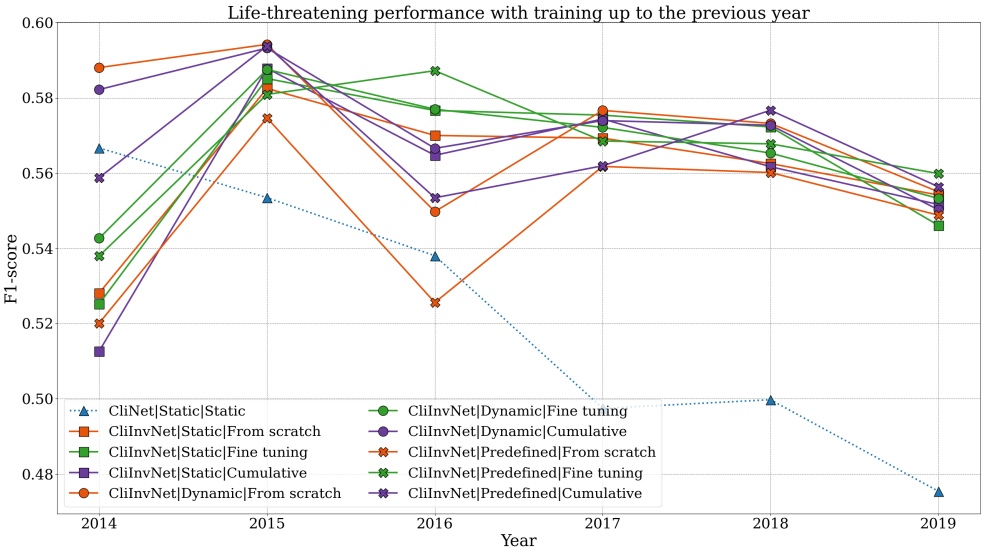

Figure 6.9: Life-threatening performance over time with training up to the previous year for each pipeline.

Table 6.4: Average F1-score values for life-threatening performance with training up to the previous year for each pipeline tested. Non-parametric 95% confidence intervals for each average value are provided between brackets.

Parameter updating	Feature domain			Mean
	Static	Dynamic	Predefined	
From scratch	0.561 [0.558, 0.565]	0.573 [0.569, 0.576]	0.549 [0.545, 0.552]	0.561 [0.557, 0.564]
Fine-tuning	0.563 [0.56, 0.567]	0.566 [0.563, 0.57]	0.567 [0.564, 0.57]	0.566 [0.562, 0.569]
Cumulative	0.559 [0.555, 0.562]	0.573 [0.57, 0.576]	0.567 [0.563, 0.57]	0.566 [0.563, 0.57]
Mean	0.561 [0.558, 0.565]	0.571 [0.567, 0.574]	0.561 [0.557, 0.564]	0.564 [0.561, 0.568]

Upon observing both Figure 6.9 and Table 6.4, and comparing them with their corresponding current year counterparts in Figure 6.6 and Table 6.1, it becomes evident that the behavior for the subsequent year is notably more erratic, characterized by pronounced and abrupt transitions. In the context of the baseline model, the *Clinical subnetwork* derived from DeepEMC^2 exhibits a gradual decline in performance

over time, marked by an initial drop in 2014 and a subsequent prominent dip in 2017. Contrasting with this, other pipelines showcase a conspicuous descent in 2014, which is subsequently compensated through retraining with data from the same year, followed by a steady performance degradation.

Simultaneously, it is noteworthy that discrepancies among pipelines are accentuated in the context of next year's performances, as opposed to their current year counterparts. Unlike the situation in the current year, distinct disparities emerge between the static, the dynamic, and the predefined feature domain approaches, with the dynamic approach yielding superior results. In terms of parameter updating strategies, the values tend to remain relatively similar between the fine-tuning and cumulative approaches, while the from scratch approach yields inferior results.

Admissible response delay

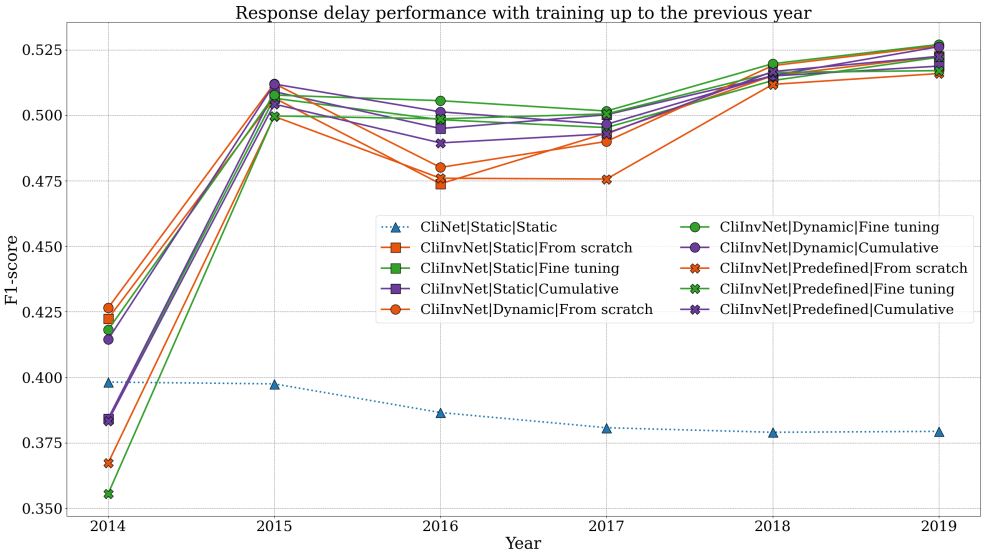

Figure 6.10: Admissible response delay performance over time with training up to the previous year for each pipeline.

Table 6.5: Average macro F1-score values for admissible response delay performance with training up to the previous year for each pipeline tested. Non-parametric 95% confidence intervals for each average value are provided between brackets.

Parameter updating	Feature domain			Mean
	Static	Dynamic	Predefined	
From scratch	0.489 [0.487, 0.491]	0.492 [0.49, 0.495]	0.474 [0.472, 0.477]	0.485 [0.483, 0.488]
Fine-tuning	0.487 [0.484, 0.489]	0.497 [0.494, 0.499]	0.481 [0.479, 0.484]	0.488 [0.486, 0.49]
Cumulative	0.487 [0.485, 0.489]	0.494 [0.492, 0.497]	0.485 [0.483, 0.487]	0.489 [0.486, 0.491]
Mean	0.488 [0.485, 0.49]	0.494 [0.492, 0.497]	0.48 [0.478, 0.482]	0.487 [0.485, 0.49]

From an examination of both Figure 6.10 and Table 6.5, it is evident that the behavior in the forthcoming year for the admissible response delay label exhibits a notably smoother trend when compared to that of the life-threatening label. Referring to the baseline model, the *Clinical subnetwork* originating from DeepEMC^2 demonstrates a gradual performance decline over time.

In relation to the pipelines incorporating retraining, a dip in performance is discernible in 2014, subsequently recuperating after retraining with data from the same year. Following this, performance exhibits a consistent upward trajectory. Additionally, it is worth noting that the dynamic feature domain approach presents a more favorable behavior than the static and predefined feature domain paradigms. Regarding parameter updating strategies, the comparison reveals that the fine-tuning strategy yields the most favorable results, followed by the cumulative approach, and lastly, the from-scratch strategy.

Emergency system jurisdiction

Upon examining both Figure 6.11 and Table 6.6, it becomes evident that certain pipelines experience a notable and abrupt decline in performance during 2014. However, this decline is effectively mitigated through retraining efforts. Subsequent to retraining, performance remains relatively stable. Turning to the baseline model, represented by the *Clinical subnetwork* of DeepEMC^2, it is evident that performance remains resilient and consistent over time, with minor fluctuations.

When scrutinizing other pipelines, fluctuations over time are worth noting. Nonetheless, performance remains within controlled ranges, without significant drops, except in 2014. Remarkably, the dynamic and predefined feature domain approaches consistently outperform the static approach across all tested pipelines, with similar performance between the dynamic and predefined methods. Similarly, when assessing different strategies for updating model weights over time, fine-tuning emerges as the best strategy, albeit with only marginal separation from the performance achieved by the from-scratch and cumulative strategies.

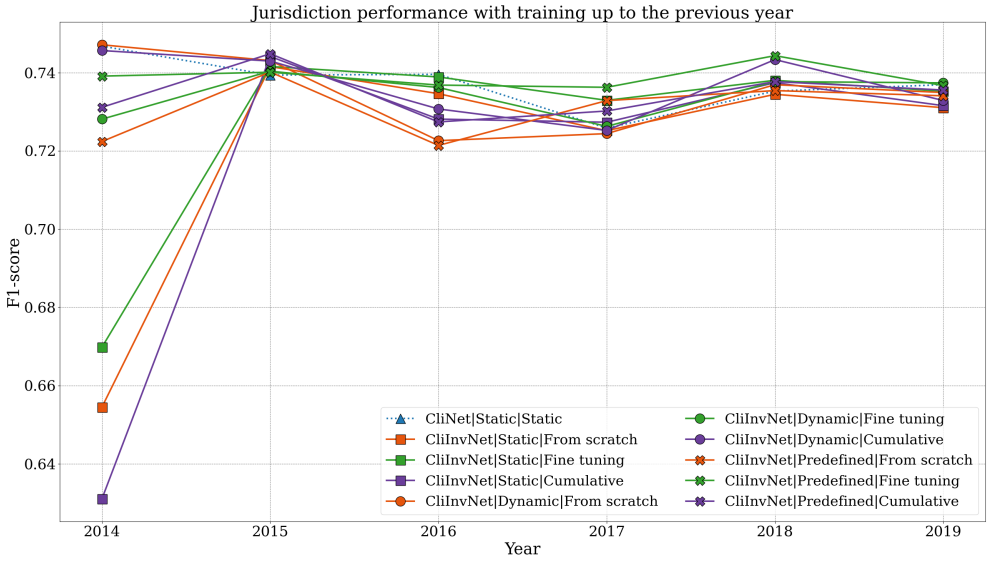

Figure 6.11: Emergency system jurisdiction performance over time with training up to the previous year for each pipeline.

Table 6.6: Average F1-score values for emergency system jurisdiction performance with training up to the previous year for each pipeline tested. Non-parametric 95% confidence intervals for each average value are provided between brackets.

Parameter updating	Feature domain			Mean
	Static	Dynamic	Predefined	
From scratch	0.72 [0.718, 0.722]	0.735 [0.733, 0.737]	0.731 [0.729, 0.733]	0.729 [0.726, 0.731]
Fine-tuning	0.726 [0.724, 0.728]	0.734 [0.732, 0.736]	0.739 [0.737, 0.741]	0.733 [0.731, 0.735]
Cumulative	0.717 [0.714, 0.719]	0.737 [0.735, 0.739]	0.734 [0.732, 0.737]	0.729 [0.727, 0.731]
Mean	0.721 [0.719, 0.723]	0.735 [0.733, 0.737]	0.735 [0.733, 0.737]	0.73 [0.728, 0.732]

Relative performance variation

Next, we present a bar plot in Figure 6.12 illustrating the relative performance variation. This variation is based on the optimal updating strategy, identified as fine-tuning, in conjunction with the optimal feature domain approach, which is the dynamic feature domain.

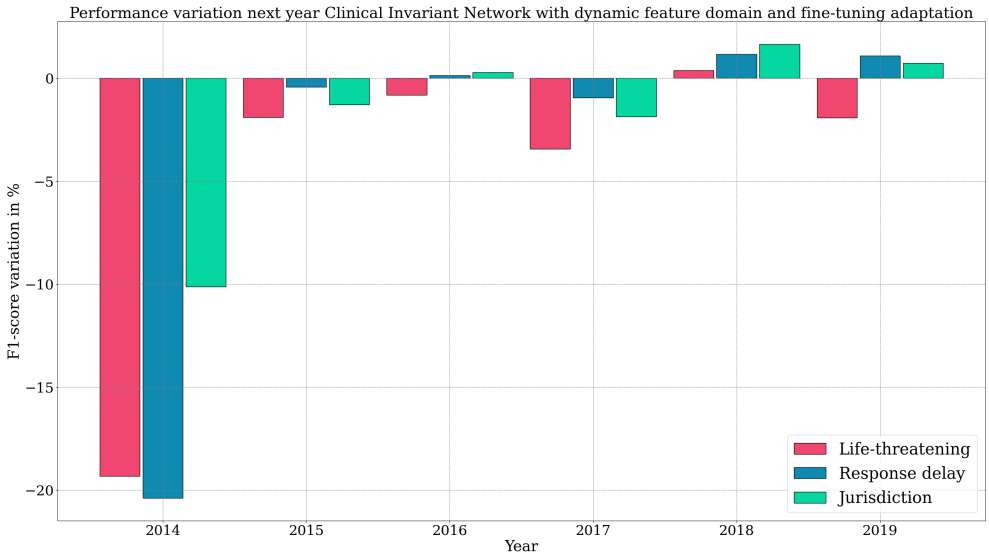

Figure 6.12: Performance with training up to the previous year variation over time—in percentage terms—for the pipeline achieving the most favorable outcomes—fine-tuning with dynamic feature domain—for each of the severity labels, namely: life-threatening, admissible response delay, and emergency system jurisdiction.

Upon observing this graph, a pattern emerges: the system change results in a significant performance deterioration from 2012 to 2014. This loss is particularly severe in the case of the life-threatening and admissible response delay labels. Subsequent to this period, performance exhibits oscillations, with certain years showing improvement while others witness declines. Notably, this variability remains within a 5% range, indicating a bounded and controlled fluctuation.

6.5 Discussion

6.5.1 Relevance

Through an analysis of the prior probability shift graphs for each severity label, it can be deduced that the severity of handled incidents was significantly reduced after the transition from CORDEX to CoordCom. This is reasonable, as new dispatchers and coordination protocols were introduced, which exhibited a more cautious and less specific approach compared to the previous operators. These new dispatchers dealt with events that were less urgent, unlike the prior operators who managed a higher proportion of genuine emergency cases.

Upon examining covariate shifts, it is evident that while most clinical features remained unchanged, some disappeared over the years while new ones were introduced. Additionally, the frequencies of certain existing features underwent variations. This phenomenon could be linked to the earlier mentioned increase in the number of non-severe cases that were attended to.

The study on concept shift was unable to definitively establish whether performance declines were due to actual shifts in conditional variations or were a consequence of the covariate shift itself. Nonetheless, the key takeaway is that without intervention, the performance of the *Clinical subnetwork* within the DeepEMC^2 model will gradually deteriorate over time.

Examining the evolution of the feature domain over the years in light of the results, it becomes evident that, while the performance with training up to the current year does not clearly reflect this, the dynamic feature domain approach holds substantial value in forecasting for the upcoming year—in terms of performance with training up to the previous year. Therefore, we find it appropriate to select this approach over the static and the predefined methods.

Shifting our focus to parameter updating strategies, we conclude that fine-tuning stands as the optimal choice. Although it may not consistently yield the best performance, it emerges as the prevailing strategy, showcasing effectiveness coupled with efficiency. This approach facilitates significant knowledge transfer at a reasonable computational cost, in contrast to the cumulative approach. Additionally, it retains partial information from past experiences during the initialization phase, a factor that provides this strategy with an advantage over the from-scratch approach.

Similarly, the observation that the cumulative approach, despite employing a larger pool of training data, does not consistently yield the optimal performance could potentially be attributed to the paradigm shift caused by distributional drifts. In this context, incorporating data from previous experiences might introduce noise rather than mitigate prediction errors. Therefore, we can assert that retraining with

historical data might impede the seamless transfer of knowledge and that discarding patterns from prior experiences could be more advantageous.

An additional noteworthy point for discussion is that the *Clinical subnetwork*, contrary to expectations, does not serve as a baseline when predicting emergency jurisdiction labels, neither in the present nor the subsequent year. This phenomenon may be attributed to the composition of CoordCom data, which encompasses a higher proportion of primary care cases compared to CORDEX. Consequently, the inclusion of these incidents seems to have limited the model's performance in relation to this specific label.

Finally, it is important to highlight that the performance attained during the CORDEX period is not fully regained in the CoordCom context, even after retraining. This holds true for both the life-threatening label and the emergency system jurisdiction label. This discrepancy could be attributed to the same sample selection phenomenon elucidated in previous sections, where the transition from CORDEX to CoordCom led to an increase in the number of non-severe incidents being attended to.

6.5.2 Limitations

While our study has provided valuable insights, and we have focused on evaluating a significant number of critical configurations, we acknowledge there are numerous additional combinations that remain unexplored but could also be relevant for implementation and testing. Exploring these scenarios may provide further insights into the temporal behavior of our models.

6.5.3 Future work

In future work, we envision incorporating innovative Continual Learning strategies to enhance the updating of model weights. These strategies would extend beyond the cumulative, from-scratch, and fine-tuning approaches currently explored. Additionally, we intend to broaden our assessment of feature domain methodologies. Furthermore, there is merit in extending the scope of our analysis to encompass diverse feature types, such as free text features and context data, particularly if a multimodal analysis approach is adopted.

6.6 Conclusions

In this chapter, our focus has been on investigating the existence of dataset shifts within the context of the clinical variables multitask prediction problem. We have also explored multiple pipelines designed to address these shifts and quantify the resultant anticipated performance declines. The outcomes of our analysis unveil significant alterations attributed to the transition from CORDEX to CoordCom, which occurred between 2012 and 2014, as well as more gradual variations witnessed between 2017 and 2018.

Moreover, we emphasize the necessity of dynamic feature domain updates on an annual basis to mitigate performance deterioration in subsequent years. In terms of parameter updates for our models, our findings suggest that retaining data from prior experiences for retraining purposes is not the optimal choice. Instead, we advocate for the adoption of fine-tuning approaches, as they offer a more effective and efficient solution. Therefore, our key recommendation from this chapter is to implement dynamic feature updating strategies in conjunction with fine-tuning mechanisms. When considering these recommendations and excluding the year 2014, during which the information system underwent a change, it becomes apparent that performance fluctuations in the subsequent years are limited. Specifically, the variability in terms of F1-score performance across all three labels remained stable within a 5% rate change.

Chapter 7

Deep continual multitask classification of emergency medical call incidents over time combining multimodal data

The development of the DeepEMC^2 model, as discussed in Chapter 3, has demonstrated the potential to enhance the triage process for out-of-hospital emergency calls using Deep Learning techniques. However, in subsequent chapters (Chapter 5 and Chapter 6), issues related to dataset shifts have been identified when analyzing our incidents data over time. This concern cannot be disregarded or bypassed, as it leads to detrimental performance effects on the DeepEMC^2 model, which is intended for deployment within emergency medical dispatch centers. Hence, in the preceding chapters, we have formulated and executed Continual Learning strategies aimed at tackling this challenge. Yet, these strategies were previously applied separately to the textual observations provided by dispatchers and the clinical features. However, in this particular chapter, we adopt a multimodal approach. This approach encompasses the necessary adaptations to the DeepECM^2 model, allowing us to collectively consider contextual, clinical and free text data for predicting severity labels—specifically, life-threatening conditions, admissible response delay, and emergency system jurisdiction. The overarching goal is to mitigate the adverse effects stemming from dataset shifts as much as possible, while also incorporating mechanisms to prepare the model for unanticipated changes.

The results from this chapter suggest that the model's predictive performance for the subsequent year remains within acceptable operational bounds. Consequently, if alterations are gradual and not excessively pronounced, the measures that have

been implemented can ensure a satisfactory level of performance for our decision-support model in the out-of-hospital medical triage context. Nevertheless, it remains crucial to diligently observe shifts in data distribution and performance metrics. This proactive monitoring is imperative to promptly address potential fluctuations that might be substantial, potentially resulting in adverse performance consequences with significant implications for deployment.

The contents of this chapter are being submitted to the journal Artificial Intelligence in Medicine—thesis contributions C4, C5 and P7.

7.1 Introduction

Developing an Artificial Intelligence model for the classification of out-of-hospital Emergency Medical Call Incidents (EMCI) holds significant potential in terms of improving patient well-being and sustaining health services (Ferri et al., 2021). Nevertheless, what may appear promising from a research standpoint might not translate as successfully into real-world deployment. This discrepancy often arises due to the presence of dataset shifts (Quinonero-Candela et al., 2008)—changes in data distributions that emerge post the model's development phase, inherent to medical domain (Sáez et al., 2020). Consequently, when striving to offer decision support for out-of-hospital medical emergencies, careful considerations are essential, and Artificial Intelligence models must be conceptualized with the anticipation that such shifts will inevitably materialize over time.

In prior chapters, we introduced the DeepEMC^2 model, an ensemble multitask multimodal approach that significantly enhances the in-house triage protocol of the Valencian Region. Nonetheless, the data employed encompassed the years between 2009 and 2012, corresponding to the CORDEX system dataset. Upon acquiring the CoordCom data—encompassing the years 2014 to 2019—we embarked on distinct analyses. Our focus was directed towards evaluating alterations in data distributions and feature domains across the two systems over various years. Additionally, we explored the integration of Continual Learning (Parisi et al., 2019) strategies to mitigate the adverse impacts of dataset shifts. Our analysis was concentrated on both clinical and textual features, underscoring the necessity of adopting a Continual Learning methodology to counterbalance the effects stemming from distributional shifts.

In this chapter, we adopt a comprehensive perspective, encompassing all accessible features and labels over the entire time span at our disposal. This undertaking presents a profound and multifaceted challenge, merging continual, multimodal, and multitask learning. Our input data is composed of contextual, clinical, and textual features, while our output data consists of severity labels—specifically, life-threatening, admissible response delay, and emergency system jurisdiction. The model is structured to undergo Continual Learning training across a series of experiences, with the

overarching objective of maximizing its future performance. Furthermore, measures are incorporated to augment the model's resilience towards missing data and dynamic feature domains.

7.2 Materials

7.2.1 Dataset

Overview

A comprehensive collection of 2 054 694 distinct Emergency Medical Call Incidents (EMCI) originating from the Health Services Department of the Valencian Region was taken into consideration. This dataset was compiled over the period spanning 2009 to 2019, excluding the year 2013 due to alterations in the emergency system during that time.

The EMCI dataset encompasses data obtained both during and subsequent to the call. The subsequent sections delineate a comprehensive breakdown of the elements constituting each of these categories:

During-call data

During-call data were recorded during the emergency medical call and were integrated by the contextual features—date, number of patients involved, age, sex—clinical features—clinical variables derived from the in-house decision tree—and free text features—unstructured data written by the emergency medical dispatcher capturing what it cannot be modeled by the prior variables. These data were used at inference time as input for the prediction.

After-call data

After-call data were recorded at a time after the call. They include physician diagnosis, hospitalizations, urgency stays, maneuvers and procedures the patient underwent. After-call data were used offline—i.e., not in prediction time—to infer if the emergency event implied or not a life-threatening situation, which was the admissible response delay—undelayable, minutes, hours, days—and if the event was jurisdiction of the emergency system or primary care. Hence, the after-call was used to derive the three severity labels to predict.

7.2.2 Framework

The implementation language of our experiments was Python (G. van Rossum (Guido), 1995), using the libraries Numpy (van der Walt et al., 2011) and Pandas (McKinney, 2010) for data management. To implement and train the designed models we considered PyTorch (Paszke et al., 2017) and HuggingFace's Transformers (Wolf et al., 2019). Finally, we used Optuna (Akiba et al., 2019) for hyperparameter tuning.

7.3 Methods

7.3.1 Data preprocessing

Distinct preprocessing techniques were employed based on the variable type, effectively transforming the initial data into a matrix format suitable for utilization in the Deep Learning models.

Given the occurrence of incidents involving multiple patients, the age variable necessitated categorization into distinct groups. This step facilitated the handling of incidents characterized by both single-patient and multi-patient scenarios. To achieve this categorization, we adopted fuzzy representations (Zadeh, 1965). Specifically, we chose sigmoid functions for their inherent smoothness, which helps avoid abrupt transitions between different age groups. Sigmoid functions provide a smoother behavior compared to the trapezoidal functions considered in Chapter 3. The membership functions employed are visually depicted in Figure 7.1.

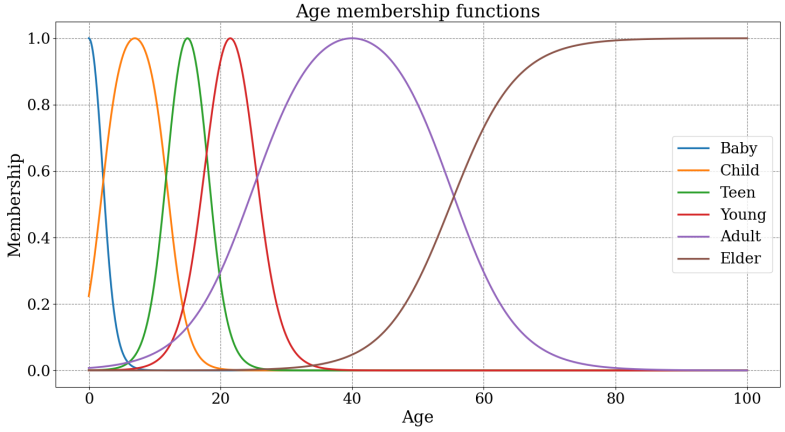

Figure 7.1: Sigmoid functions representing age group membership.

The representation of the sex variable involved employing ratios to account for various scenarios. This included the creation of a male ratio, a female ratio, and a missing ratio. This approach acknowledges the possibility of incidents with multiple patients, some of whose sex might be unknown. Similarly, the date variable underwent transformations to derive additional features. These features encompassed the weekday, month, a binary indicator for weekend days, and a binary indicator for labor days. The resultant non-binary features were subsequently mapped to indexes, with distinct mapping schemes for each temporal categorical variable. Furthermore, a number of patients feature was generated and subsequently normalized to fit within the interval [0, 1).

Concerning clinical variables, each variable-value pair underwent conversion to an index. This process generated sequences of integers that were subsequently subjected to post-padding and truncation, thereby ensuring uniform sequence lengths. For this particular case, the sequence length was set at 14, a value chosen because more than 99% of reported incidents included 14 or fewer clinical variables. It is crucial to emphasize that although these variables were organized as sequences, their inherent order is not of significance. This is primarily due to the absence of information regarding the order of clinical variables within the CoordCom system. Subsequently, in the modeling section, the methodology for achieving predictions that remain invariant to feature order will be elaborated upon.

As for text features, subword tokenization utilizing the WordPiece technique (Wu et al., 2016) was employed to minimize vocabulary size. To maintain sequences of consistent lengths while preserving information about the original sequence lengths, post-padding and attention mask generation were conducted. The padding length was set at 64, as over 99% of reported incidents featured subword sequences of 64 or fewer elements.

Lastly, labels, corresponding to structured categorical data, were encoded using a one-hot encoding scheme. This resulted in a label matrix with 8 columns, each associated with a specific label-class pair.

7.3.2 Data splitting

As we did in previous chapters, data was split into multiple experiences to capture the temporal variations while avoiding overfitting issues. First, we divided our dataset into multiple learning experiences (Lomonaco et al., 2021). We considered a first experience constituted by data solely derived from the CORDEX system. Subsequent experiences corresponded to individual years within the CoordCom information system. This partitioning strategy was chosen because, as detailed later, novel architectures, both for the DeepEMC^2 model (Ferri et al., 2021) and its constituent subnetworks, will be considered. Given that the model and its subnetworks were not

trained on a yearly basis but rather on the entire CORDEX data batch, employing the same data batches for performance metric comparison becomes imperative to assess the potential negative impacts introduced by architectural modifications.

Following this initial partition, a second, iterative partitioning was implemented for each of the designated experiences. For every experience, an initial division into training and test sets was executed, with a sampling ratio of 80% for training and 20% for testing. This testing subset played a crucial role in estimating the real performance of the model. Subsequently, the previously mentioned training set underwent a further partition, resulting in distinct pure training and validation subsets. The allocation proportions for this secondary division were set at 70% for the training subset and 30% for the validation subset. It is noteworthy that the validation subset was exclusively employed for the fine-tuning of hyperparameters, without any data originating from the test subset being utilized in this process.

It is important to emphasize that we have deliberately chosen a specific time window for the defined experiences. While it may appear plausible to adopt a more granular time division, such as trimesters, it is crucial to take into account that retraining becomes feasible only when true labels are accessible. Hence, the selection of a one-year time interval presents a pragmatic approach. The acquisition of labels for every instance within shorter time spans, such as per month, is not practically feasible due to bureaucratic procedures. Moreover, it is pertinent to consider that the data exhibits distinct annual seasonality patterns. By employing a yearly division, we effectively mitigate the potential confounding influence introduced by this temporal variability. This strategy ensures that our analysis remains coherent and conclusive, especially when contrasted with a scenario where finer temporal divisions are utilized, such as monthly intervals.

7.3.3 Deep neural network design

As elaborated in Chapter 3, the task of classifying EMCI by combining multimodal data was deconstructed into four distinct subproblems. These subproblems encompassed three EMCI classification tasks, each focused on EMCI data of a specific nature, and a final EMCI classification task that employed the solutions from these subproblems to carry out a global multimodal EMCI classification. To tackle these four challenges, four distinct Deep Learning networks were formulated: the Context Network (ConNet), the Clinical Network (CliNet), the Text Network (TextNet), and the Global Network (GloNet).

Given that the labels for life-threatening incidents, response delay, and jurisdiction provide distinct yet interconnected information, a multitask learning approach (Caruana, 1997) was pursued to exploit these label dependencies. In order to enhance training efficiency, introduce regularization, and minimize the overall number of net-

work parameters, a hard parameter sharing strategy (Ruder, 2017b) was adopted. Consequently, each of the four developed subnetworks incorporated a task-shared block—employing the same parameter set across all label prediction tasks—and a task-specific block—utilizing distinct parameter sets for each label prediction task.

It has to be underscored that the Global Network presented in this chapter can be regarded as an evolution of the original DeepEMC^2 model (Ferri et al., 2021). However, as elucidated in subsequent sections, it integrates mechanisms capable of accommodating variations in distributions over time, including alterations in feature domains. In addition, unlike its predecessor, this version is trained end-to-end, negating the need for a preliminary training phase involving the subnetworks, followed by the network responsible for aggregating inner representations.

The forthcoming subsections provide a more comprehensive breakdown of the architectural details pertaining to each of the networks developed within this chapter.

Context Network (ConNet)

The Context Network (ConNet), illustrated in Figure 7.2, is designed to handle the contextual data associated with an EMCI. This includes information such as age, sex, the number of patients involved, whether it is a labor day, and features related to the weekday and month. ConNet is structured as a multitask (Caruana, 1997) deep neural network, composed of two primary components: a Context Encoder and a Multitask Classifier.

The Context Encoder constitutes the segment of the model responsible for hard parameter sharing. Within this encoder, we find two Embedding Layers (Bengio et al., 2000). The first Embedding Layer, denoted as W, maps the indexes representing weekdays to dense numerical representations. The second Embedding Layer, denoted as M, performs a similar function for the indexes referencing months. Following these embedding layers, a Concatenation Block merges the encoded representations from the previous embedding layers with the remaining features that already possess appropriate numerical representations. These features include age membership data, sex-related features, the number of patients, and the labor day indicator. Subsequent to this Concatenation Block, a sequence of dense blocks processes these encoded representations. A Dense Block comprises a Fully Connected Layer (Rosenblatt, 1958), Layer Normalization (Ba et al., 2016), a GELU activation function (Hendrycks & Gimpel, 2016), and a Dropout Layer (Hinton et al., 2012).

The outputs emerging from the final Dense Block are then transmitted to the Multitask Classifier. This component is responsible for introducing task-specific elements into the architecture and comprises three label branches. Each branch consists of multiple dense blocks culminating in an output block. These output blocks encompass a Fully Connected Layer, followed by a Softmax activation function.

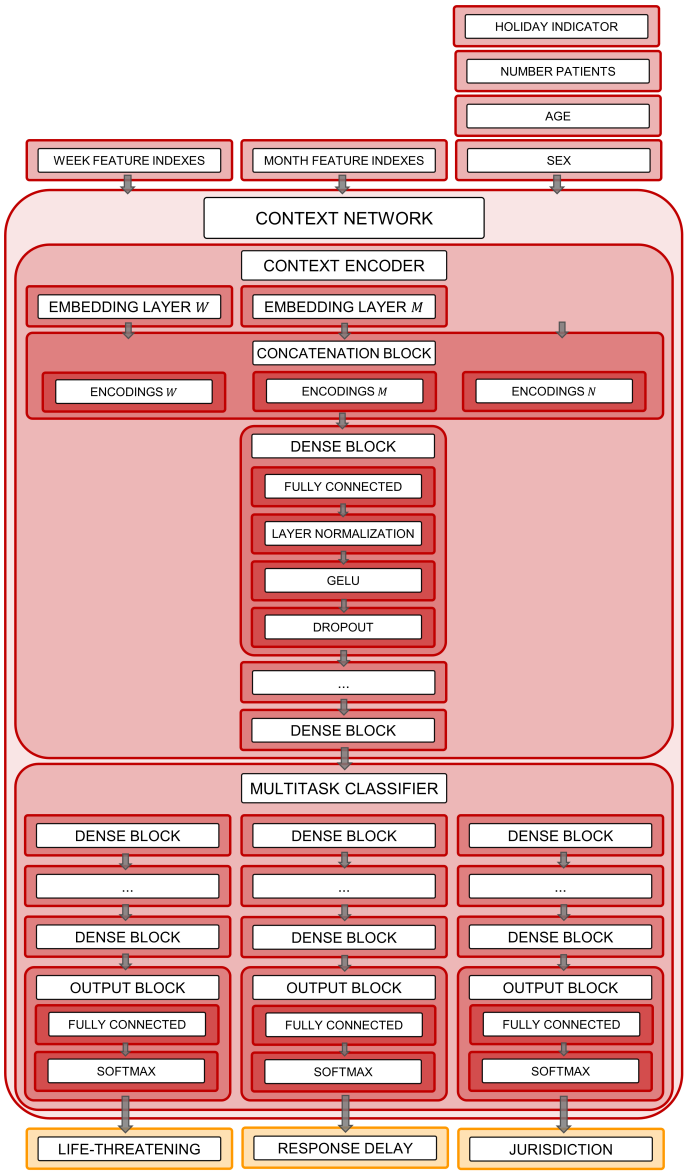

Figure 7.2: Context Network (ConNet) architecture.

Clinical Network (CliNet)

The Clinical Network (CliNet), as depicted in Figure 7.3, is engineered to process the clinical variables associated with an EMCI. It adheres to the identical architecture as the Clinical Invariant Network, previously presented in Chapter 6. CliNet is formulated as a multitask deep neural network, comprising two primary constituents: the Clinical Encoder and the Multitask Classifier. The Clinical Encoder, serving as the bedrock of the network's hard parameter sharing mechanism, constitutes its core. Meanwhile, the Multitask Classifier accommodates separate branches, each dedicated to a specific label. These branches are tasked with generating predicted probabilities for the various classes within their respective labels.

Zooming in on the Clinical Encoder, its construction initiates with an initial Embedding Layer. This layer facilitates the transformation of clinical variables, expressed as indexes, into dense vector representations—an approach far more efficient than one-hot encodings. Importantly, this Embedding Layer enables the network to gracefully adapt to novel features over time. This adaptability is achieved by preallocating an extensive number of entries within the associated lookup matrix, all without impacting subsequent architectural elements. Subsequent to the Embedding Layer, an Adaptive Average Pooling block (Szegedy et al., 2016) is deployed. This component serves to aggregate the representations of all features within an observation into a single representation. This functionality allows the network to accommodate varying numbers of features per entry. Moreover, the Adaptive Average Pooling Layer confers order-invariant capabilities to the network, ensuring consistent results regardless of alterations in feature order. Following this, a series of dense blocks is introduced, with each block incorporating a Fully Connected Layer, Layer Normalization, a GELU activation function, and a Dropout Layer.

The structure and functionality of the Multitask Classifier align with that of the Context Network, as presented in the previous subsection, following a similar architecture and methodology.

Text Network (TextNet)

The Text Network (TextNet), illustrated in Figure 7.4, is specialized in handling the free text dispatcher observations linked to an EMCI. It operates as a multitask deep neural network, composed of two primary components: the Text Encoder and the Multitask Classifier. The Text Encoder, serving as the cornerstone of the network's hard parameter sharing, constitutes its nucleus. Meanwhile, the Multitask Classifier comprises separate branches, each dedicated to a specific label. These branches compute the predicted probabilities for the various classes within their respective labels.

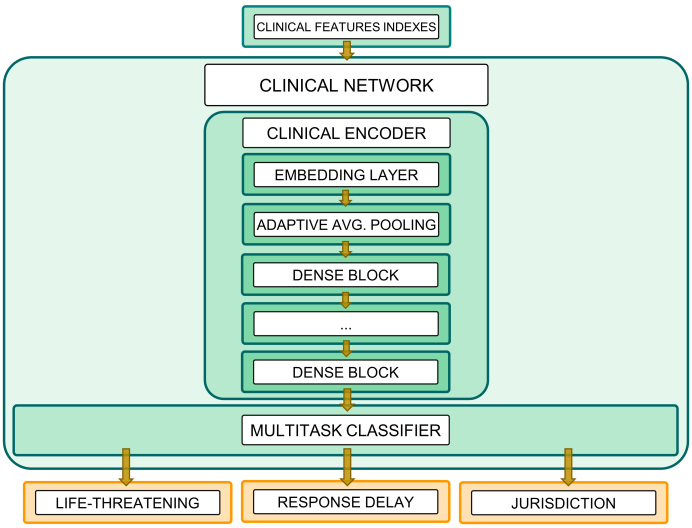

Figure 7.3: Clinical Network (CliNet) Architecture.

Directing our focus towards the Text Encoder, it assumes the form of a BERT (Devlin et al., 2019) module, albeit with certain distinctions. This variant of the Transformer (Vaswani et al., 2017) model encompasses an Embedding Block as its initial component. This Embedding Block comprises a Token Embedding Layer, a Position Embedding Layer, a Token Type Embedding Layer, Layer Normalization, and Dropout. Subsequently, a total of twelve attention blocks process the embeddings generated earlier, culminating in the Pooler. The Pooler is tasked with consolidating the inner representations within encodings while significantly reducing dimensionality.

The Multitask Classifier adheres to the same structure and functionality as presented in the previous subsection for the Context and Clinical networks.

It is crucial to underscore that the model is not trained from scratch. Instead, a pretrained multilingual version of the BERT model was leveraged to constitute the Text Encoder. This pretrained model is available at (Town, 2023). Furthermore, most of the model parameters are frozen to capitalize on pretraining, mitigate computational expenses, and enhance performance.

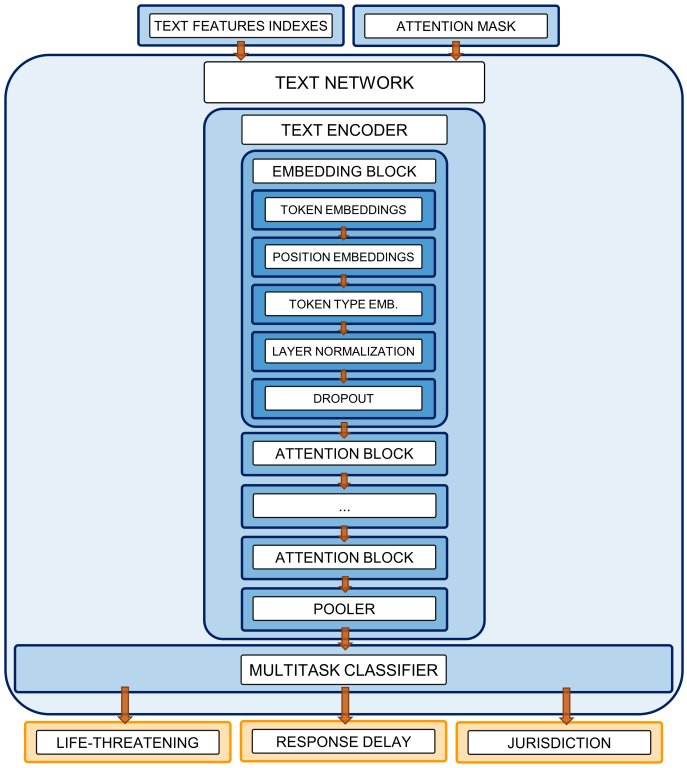

Figure 7.4: Text Network (TextNet) architecture.

End-to-end DeepEMC² - Global Network (GloNet)

The End-to-end Deep Ensemble Multitask Classifier for Emergency Medical Calls, denoted as End-to-end DeepEMC^2 or Global Network (GloNet), as illustrated in Figure 7.5, operates at the intersection of context data, clinical data, and text data simultaneously, functioning as a multimodal model. It is a multitask deep neural network composed of two primary components: an Encoder and a Multitask Classifier. The Encoder, base of the network's hard parameter sharing mechanism, is a composite structure comprising the previously introduced Context Encoder, Clinical Encoder, and Text Encoder. The outputs generated by each of these individual encoders are amalgamated within a Concatenation Block before being relayed to the Multitask Classifier.

The Multitask Classifier mirrors the structure detailed for the preceding networks, with separate branches designated for each label. These branches compute

the predicted probabilities for the various classes encompassed within each label. It adheres to the same architecture and principles outlined in the prior subsections.

It is important to emphasize a notable distinction in this novel iteration of the DeepEMC^2 model. Apart from the incorporation of mechanisms to address changes in data distributions and feature domain variability (as elucidated in forthcoming sections), a pivotal shift pertains to its training methodology. Unlike the original DeepEMC^2 model, which involved training individual encoders followed by the Multitask Classifier, the End-to-end DeepEMC^2 model is specifically designed to be trained end-to-end. Nevertheless, it is crucial to note that a significant proportion of the parameters within the Text Encoder are preserved as non-trainable, as previously justified in the preceding subsection.

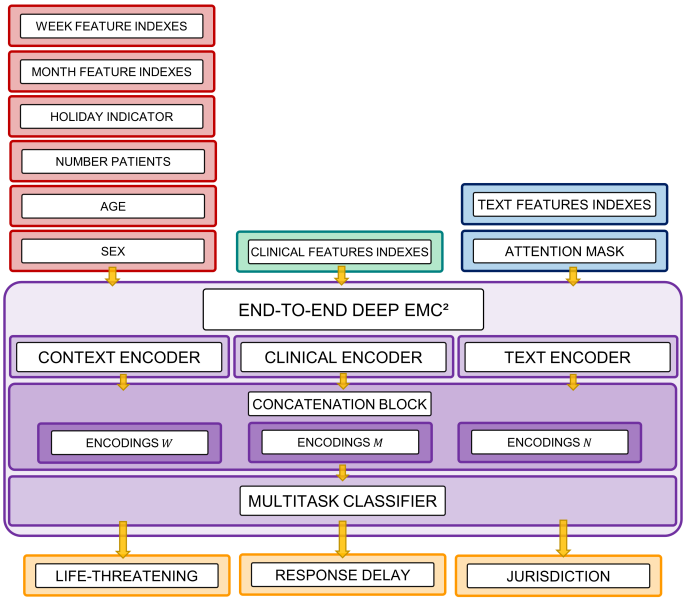

Figure 7.5: End-to-end Deep Ensemble Classifier for Emergency Medical Calls (DeepEMC^2) architecture. This model is also identified as the Global Network (GloNet).

7.3.4 Parameter tuning

Regarding the parameter tuning process, we adopted the AdamW optimizer (Loshchilov & Hutter, 2019), which is a variation of the Adam algorithm (Kingma & Ba, 2017), chosen for its efficacy in training transformer models (Loshchilov & Hutter, 2019). Our training process adhered to a mini-batch approach (Bertsekas, 1994) for data feeding. The selected loss function was the soft F1-score (Janocha & Czarnecki, 2017), since is advantageous for incorporating argmax saturation procedures during the transition from output scores to the saturated predicted labels. Importantly, this obviates the necessity to adjust the threshold for each experience, as the constant threshold remains intact, and the learning process naturally attunes itself. In addition, a learning rate scheduler was integrated, specifically a cosine annealing learning rate scheduler, which is well-aligned with the demands of deep transfer learning (Loshchilov & Hutter, 2017).

Furthermore, it's worth noting that layers featuring GELU activation functions were initialized using the Kaiming initialization technique (He et al., 2015), while layers employing the softmax activation function were initialized using Xavier's initialization (Glorot & Bengio, 2010).

7.3.5 Continual Learning

Scalability of feature support

To accommodate the gradual integration of novel clinical features over time due to updates in the in-house decision tree, we implemented a dynamic feature domain updating strategy, firstly presented in Chapter 6.

Hence, for clinical features, we updated the mapping between feature identifiers and their corresponding indices for every new experience, exclusively within the training sets and not the evaluation sets. This entailed establishing a frequency threshold that determined when a feature was considered infrequent and thus designated to be mapped either to the unknown index or to an index exclusive to that particular feature. Across different experiences, we tracked and updated the cumulative absolute frequency of occurrences for each feature. Consequently, features that were initially mapped to the unknown index due to their rarity could potentially be *unblocked* in subsequent experiences, preventing them from being indefinitely restricted. It is imperative to underscore that while the index mapping evolves over time, the index designated for representing infrequent features remains constant. This strategic decision safeguards against index overlap and the introduction of extraneous noise.

Parameter updating over experiences

As demonstrated in chapters 5 and 6, fine-tuning, despite its relative simplicity from a technical standpoint, yields the most favorable outcomes in terms of effective knowledge forward transfer. Notably, this approach proves to be not only effective but also efficient, as it obviates the necessity to retain historical data. This efficiency presents a compelling trade-off compared to alternative strategies, such as cumulative learning. Moreover, fine-tuning aligns seamlessly with considerations of data privacy, as information pertaining to previous data is encoded within the model's weights. Consequently, there is no requirement to access past data, which might only be available for a limited duration. Furthermore, this approach harmonizes with contemporary trends in the realm of Deep Learning. Transfer learning methodologies, particularly prominent in the context of Natural Language Processing, are recognized as state-of-the-art practices. The consistency of the fine-tuning approach with these modern practices underscores its relevance and potential.

7.3.6 Hyperparameter tuning

In this study, meticulous attention was devoted to the selection of hyperparameters, recognizing their influence on the ultimate performance outcomes.

An automated active learning approach (Settles, 2009) was adopted for this purpose. For each pipeline—context, clinical, text and end-to-end DeepEMC^2—a distinct set of hyperparameters was established. This encompassed parameters like learning rate and batch size. Furthermore, a range of values was proposed for each hyperparameter. For instance, in the case of the learning rate, values such as 0.0001 and 0.00001 were considered, while for batch size, options included values such as 64 or 128, among others. However, the chosen hyperparameter sampling space remained discrete, given that a continuous space could potentially lead to overfitting issues owing to the curse of dimensionality (Bellman, 1956).

Subsequently, a Bayesian optimization methodology was employed. This iterative process involved training an auxiliary probabilistic generative model, which fulfilled two primary objectives: 1) estimating the probability of achieving the objective performance metric—in this instance, the soft F1-score—given a specific set of hyperparameters, and 2) generating new hyperparameter values during each iteration with the expectation of enhancing the performance metric.

It is imperative to emphasize that these hyperparameters deemed as *optimal* were established through experiments conducted on the pure training and validation sets. Following this determination, retraining procedures were executed using the complete training set, and performance metrics were subsequently computed on the test set. This comprehensive methodology ensured that the chosen hyperparameters were robustly validated avoiding overfitting to the test set.

7.3.7 Evaluation

To asses the performance of each of the individual networks and the end-to-end deep model developed over time, we calculated the F1-score for each of the severity labels associated to each pipeline. This F1-score was relative to the positive class for the life-threatening label—life-threat class—as well as for jurisdiction label—emergency system jurisdiction—while it was macro averaged for the admissible response delay label—as we cannot set a reference class among the four categories. In addition to this metric calculation, we included non-parametric 95% confidence intervals, using the bootstrap technique (Efron & Tibshirani, 1994), by considering a total of 1000 resamples.

We computed these metrics for each experience, considering two approaches. First, we calculated them by training the model up to the current experience, allowing us to assess the absence of overfitting while estimating model performance in the current experience. Second, we computed them by training the model up to the previous experience. This approach helps us understand how model performance diminishes when applied to novel incoming data, which may exhibit variations in data distributions. Therefore, we obtain information about both in-sample and out-of-sample performance by considering these two assessments.

Furthermore, it is essential to mention that we also calculated baseline performance on the CORDEX dataset. This enabled us to gauge the behavior of the implemented models in situations where they were comparable to the DeepEMC^2 model, covering the time span from 2009 to 2012. This step served to verify that despite the architectural changes introduced to provide scalability and robustness over time, the modeling performance was not severely affected and in line with the original DeepEMC^2 model.

7.4 Results

7.4.1 Baseline performance in CORDEX

Next, we present in Table 7.1 the performance, in terms of F1-score obtained for each of the severity labels, considering the models from Chapter 3 and the ones from this chapter (Chapter 7).

Table 7.1: Baseline performance comparison, i.e., model performances in the CORDEX dataset, which integrates our first learning experience, comprising the years from 2009 to 2012. The F1-scores for each of the severity labels are reported, including the 95% non-parametric confidence intervals between brackets.

		Labels		
Inputs	Methodology	Life-threatening	Response delay	Jurisdiction
Context	Chapter 3	**0.501 [0.498, 0.504]**	**0.377 [0.374, 0.379]**	**0.830 [0.829, 0.832]**
	Chapter 7	0.483 [0.479, 0.486]	0.316 [0.313, 0.318]	0.739 [0.738, 0.742]
Clinical	Chapter 3	0.669 [0.667, 0.672]	0.485 [0.483, 0.487]	**0.848 [0.847, 0.849]**
	Chapter 7	**0.672 [0.669, 0.675]**	**0.491 [0.488, 0.494]**	0.837 [0.835, 0.839]
Text	Chapter 3	0.684 [0.681, 0.687]	0.555 [0.553, 0.557]	**0.857 [0.856, 0.858]**
	Chapter 7	**0.706 [0.703, 0.709]**	**0.567 [0.564, 0.569]**	0.846 [0.844, 0.847]
Global	Chapter 3	0.705 [0.702, 0.707]	0.576 [0.574, 0.579]	**0.860 [0.858, 0.861]**
	Chapter 7	**0.712 [0.709, 0.716]**	**0.577 [0.574, 0.579]**	0.851 [0.85, 0.853]

From the analysis of Table 7.1, it is evident that within the dataset established by the CORDEX data, the metrics achieved by both individual networks and the global model align with those documented in Chapter 7. However, some important observations should be noted.

Firstly, the exclusion of caller and risk-group information has resulted in a noticeable loss in performance, particularly for the emergency system jurisdiction label, as seen in the Context Network.

Excluding the Context Network, the modifications introduced in this chapter have proven beneficial, resulting in improved performance compared to what was previously achieved, except for the jurisdiction label. In the case of the jurisdiction label, performance has been detrimentally affected, possibly due to the use of a more class-balanced loss function, such as the soft F1-score, compared to the previously used cross-entropy loss. As the emergency system jurisdiction class is the most prevalent, the F1-score for this class has decreased.

When we compare the performance of individual networks, we observe a trend similar to what was noted in Chapter 3. The Context Network (ConNet) consistently exhibits the lowest F1-score across the three severity labels, followed by the Clinical Network (CliNet), the Text Network (TextNet), and, ultimately, the Global Network (GloNet). Notably, TextNet's performance closely approaches that of GloNet, albeit slightly inferior.

Focusing on the networks developed in this chapter, we find the most favorable outcomes for the emergency jurisdiction label, with all F1-scores surpassing 0.73. The life-threatening label follows in terms of performance, with F1-score values ranging from 0.48 to 0.71. Lastly, the admissible response delay label aligns with expectations,

exhibiting the least favorable performance, with values ranging from around 0.31 to 0.57.

These findings reinforce the notion that free text input provides the most substantial predictive utility. It is worth noting that the collective integration of the three distinct input types only marginally outperforms the predictive capability of standalone free text input.

7.4.2 Performance with training up to the current year

Life-threatening

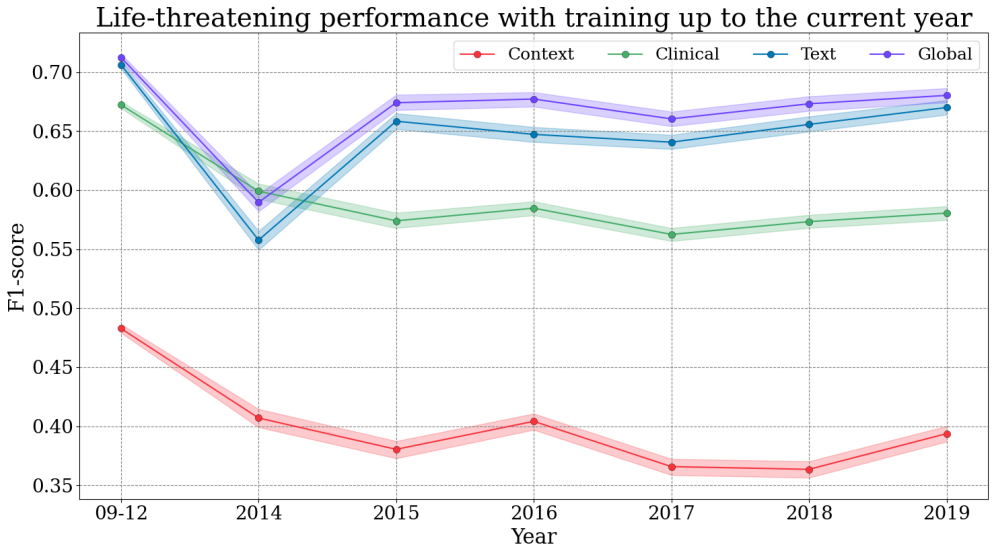

Figure 7.6: Life-threatening performance over time with training up to the current year for each model. Non-parametric 95% confidence intervals are displayed with shaded areas.

Upon careful examination of Figure 7.6 performance graph for the life-threatening label, a notable observation surfaces: the transition between the CORDEX and CoordCom information systems, which occurred in 2013, engendered a significant decline in performance across all networks.

Following the pivotal year of 2014, a recovery in performance is evident in the Text and Global networks. Nonetheless, it is worth noting that these networks fail to fully recover the initial CORDEX performance levels. Conversely, the Clinical and

Context networks exhibit a continuous decline in performance until 2017 and 2018, respectively. Although a partial recovery is observed thereafter, it remains slight.

Lastly, it is imperative to emphasize that the GloNet consistently emerges as the model attaining the most favorable outcomes throughout the examined time frame. However, a notable exception occurs in 2014 when the CliNet momentarily surpasses it. The second-strongest performer is the TextNet, followed by the CliNet. In contrast, the ConNet lags significantly behind in terms of performance, registering notably lower metrics compared to its counterparts.

Admissible response delay

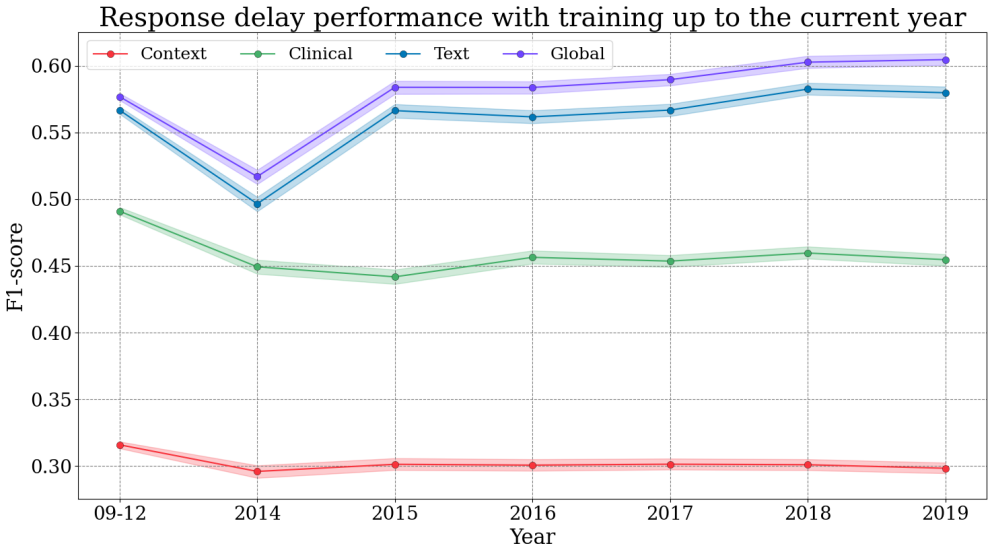

Figure 7.7: Admissible response delay performance over time with training up to the current year for each model. Non-parametric 95% confidence intervals are displayed with shaded areas.

From the analysis of Figure 7.7 performance graph for the admissible response delay label, a distinct pattern emerges: the transition from the CORDEX to the CoordCom information system in 2013 precipitated a significant and notable drop in performance across all subnetworks.

After this performance dip, both the ConNet and CliNet subnetworks exhibit a degree of stability, with minor fluctuations oscillating around a central trend. In

contrast, the Text and Global networks exhibit a capacity for performance recovery, eventually surpassing the macro F1-score achieved within the CORDEX system.

It is crucial to underline that among the networks, the GloNet consistently emerges as the frontrunner in terms of performance. Following suit, the TextNet takes the second position, while the CliNet lags notably behind in a distant third place. The ConNet experiences even more pronounced performance challenges, ranking last in terms of performance pertaining to this particular severity label and metric.

Emergency system jurisdiction

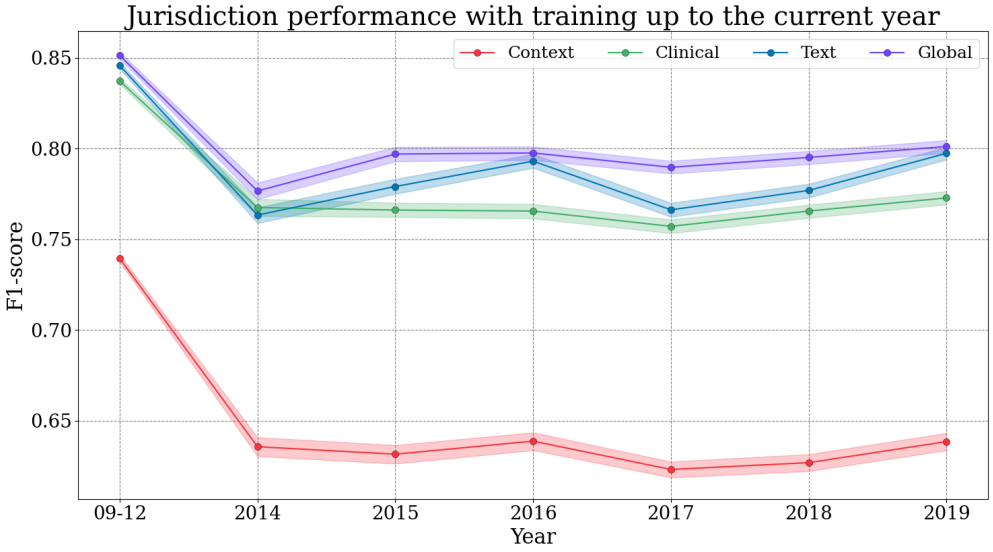

Figure 7.8: Emergency system jurisdiction performance over time with training up to the current year for each model. Non-parametric 95% confidence intervals are displayed with shaded areas.

Upon careful observation of Figure 7.8 performance graph for the emergency system jurisdiction label, a discernible pattern emerges: the transition between the CORDEX and CoordCom information systems in 2013 yielded a marked and substantial decline in performance across all networks.

Subsequent to this performance downturn, the networks' performance exhibits fluctuations around a central trend, albeit at varying levels. Importantly, it is notable that any of the implemented networks has the potential to rebound to CORDEX performance levels.

Lastly, it is crucial to underscore that once again, the GloNet consistently emerges as the network achieving the most favorable outcomes over the course of time. Following suit, the TextNet takes the second position, with the CliNet closely trailing, capable of momentarily surpassing it in 2014. Notably trailing behind is the ConNet, positioned at a significant distance in terms of performance metrics.

7.4.3 Performance with training up to the previous year

Life-threatening

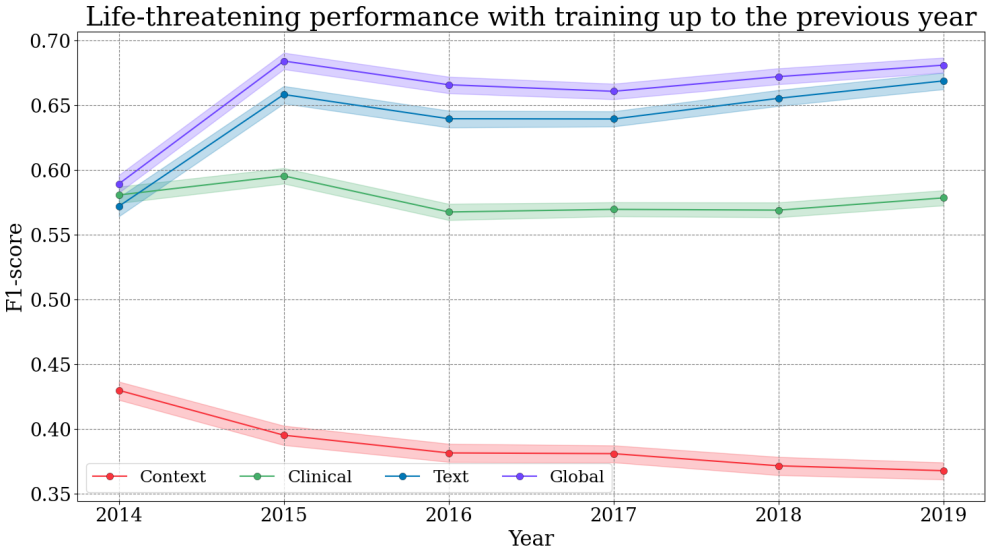

Figure 7.9: Life-threatening performance over time with training up to the previous year for each model. Non-parametric 95% confidence intervals are displayed with shaded areas.

By examining Figure 7.9 performance graph for the life-threatening label, a clear deduction can be made: all networks experience a decline in performance in 2014. Notably, the ConNet exhibits a continuous decrease in performance across subsequent years. The CliNet showcases oscillations around a central trend, without displaying indications of recuperating to previous levels. Conversely, both the TextNet and GloNet, after the dip in 2014, manage to restore their performance levels in the subsequent years. Furthermore, the ranking order of networks, from the best to the least performing, remains consistent with the findings from the current year results.

Admissible response delay

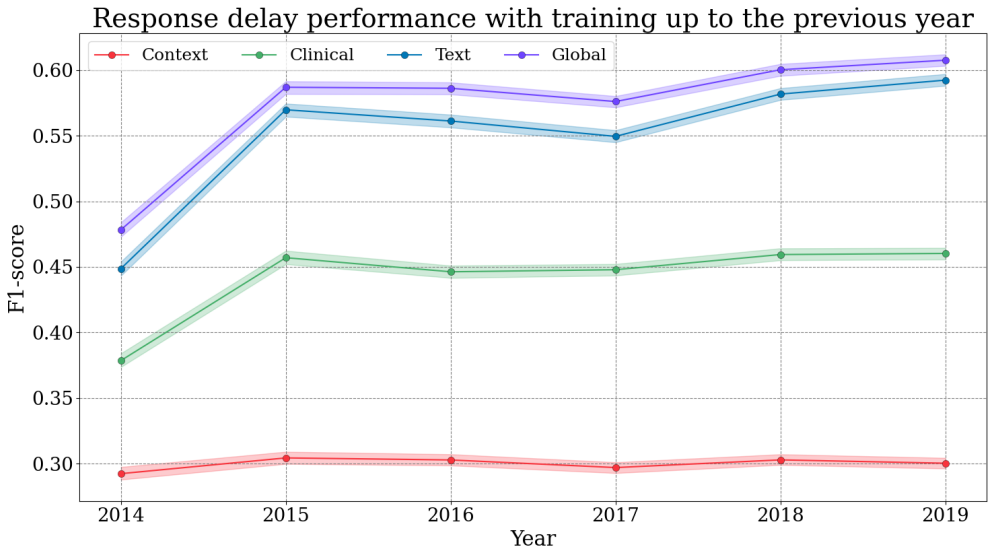

Figure 7.10: Admissible response delay performance over time with training up to the previous year for each model. Non-parametric 95% confidence intervals are displayed with shaded areas.

Analyzing Figure 7.10 performance graph for the admissible response delay label, a notable observation is evident: all networks encounter a performance decline in 2014. Subsequent to this, from 2015 onwards, the subsequent year's performance trends exhibit stability for the CliNet and ConNext. In contrast, both the TextNet and GloNet display an augmentation in performance, commencing in 2017. Furthermore, it remains noteworthy that the GloNet consistently ranks as the top-performing network, followed by the TextNet, the CliNet, and ultimately, the ConNet.

Emergency system jurisdiction

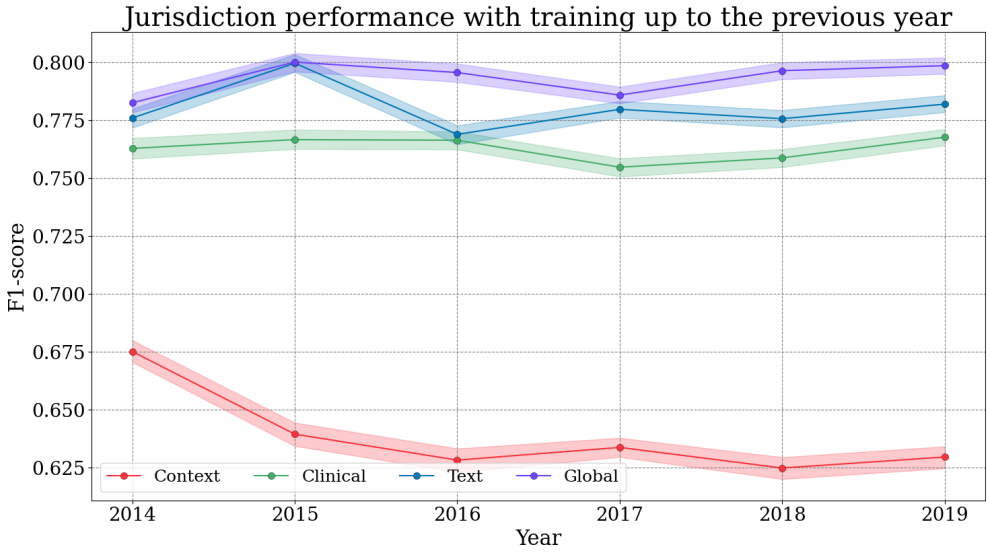

Figure 7.11: Emergency system jurisdiction performance over time with training up to the previous year for each model. Non-parametric 95% confidence intervals are displayed with shaded areas.

Based on the analysis presented in Figure 7.11, it is deducible that all networks experience a decline in performance during 2014. The ConNet fails to restore its performance, resulting in a gradual decrease over time. Similarly, the CliNet is unable to rebound; however, it manages to maintain a consistent level of performance over time. The TextNet and the GloNet exhibit some recovery in 2015, but beyond that point, they exhibit oscillations without displaying a distinct upward or downward trend. It is noteworthy that the GloNet achieves the most favorable outcomes, followed by the TextNet, the CliNet, and the ConNet. Nevertheless, it should be highlighted that the performance of the ConNet significantly lags behind the F1-score achieved by the other networks.

7.4.4 Relative performance variation

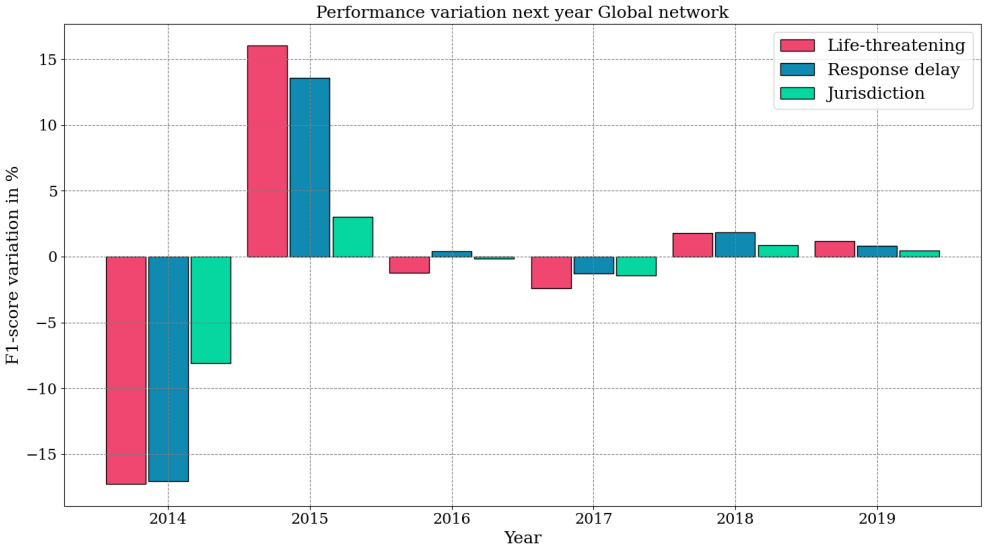

Figure 7.12: Performance with training up to the previous year variation over time—in percentage terms— for the end-to-end DeepEMC^2 model (Global Network). This relative variation is shown for each of the severity labels: life-threatening, admissible response delay and emergency system jurisdiction.

Upon examining Figure 7.12, it becomes evident that even the most high-performing model experiences a substantial decline in performance relative to the alteration of the information system from CORDEX to CoordCom. Furthermore, it is noticeable that the emergency system jurisdiction label undergoes a discernible reduction of approximately 7%, which, although significant, is notably less severe than the performance decreases of around 17% for the life-threatening and response delay labels. Following the fine-tuning of the model in 2014, the performance for the subsequent year shows a marked improvement compared to the performance in 2014 itself. This improvement suggests that the data from 2015 might contain less noise than the data from 2014. Upon refining the model with data from 2015, performance demonstrates fluctuations in the subsequent years. Some years exhibit decreases, while others show improvements. However, these year-to-year variations are minimal in contrast to the variations experienced during the system transition. Furthermore, starting from 2015, these fluctuations remain consistent across all the labels.

7.5 Discussion

7.5.1 Relevance

Based on the analysis of the results presented in the preceding section, it can be deduced that text emerges as the data modality that offers superior predictive capability. This holds true not only within the CORDEX dataset, as previously evidenced by earlier experiments, but also across all subsequent years in the CoordCom dataset. Furthermore, text exhibits the least pronounced decline in performance over time, rendering it more resilient to shifts in the dataset. This resilience can be attributed to the inherently unstructured nature of textual data. Dispatcher observations, even if slightly disparate between systems, are unlikely to display substantial variations, in contrast with structured data that can manifest in various representations.

An additional noteworthy observation pertains to the similarity in performance between the prior iteration of the DeepEMC^2 model and the current one—End-to-end DeepEMC^2 (i.e., the Global Network) introduced in this chapter—within the foundational CORDEX setting. However, it is crucial to emphasize that the model developed in this chapter incorporates mechanisms that impart tolerance to missing data, dataset shifts, and dynamic feature domains. Regrettably, these mechanisms were absent in the initial version of DeepEMC^2.

Similarly, a decline in model performance can be observed when transitioning from one dataset to another. Despite consistent retraining efforts, the performance metrics within CoordCom do not reach the same heights as those achieved in CORDEX, except for the admissible response delay label. As elaborated in the previous chapter, this discrepancy could stem from a sample selection bias. Notably, the presence of over-triage and the greater frequency of non-emergency events handled by CoordCom dispatchers may have introduced noise into the data.

Although the Global Network outperforms other models across all experimental tests, it does not provide a huge difference respect to the performance trajectory defined by the Text Network. This phenomenon holds true for both the CORDEX and CoordCom experiments and reinforces the relevance of free text dispatcher observations.

Ultimately, the outcomes derived from this chapter substantiate the feasibility of deploying the model in practical scenarios. Notably, the fine-tuning strategy demonstrates remarkable efficacy in maintaining the model's performance similar to that achieved during the training year, even in the presence of minor changes. We contend that as long as this fine-tuning approach is applied annually, barring substantial alterations, the expected performance will remain stable within acceptable variability limits of no more than 2.5%. This assertion gains further credence from the consistent performance of the subsequent year since 2015.

7.5.2 Limitations

While we have implemented measures to address dataset shifts and thereby alleviate the decline in performance over time, it is important to acknowledge that in the presence of unforeseen shifts in data distribution, the performance of subnetworks could potentially encounter challenges. Therefore, it becomes imperative to rigorously monitor both performance metrics and distributional changes. This proactive monitoring will enable swift responses to sudden alterations, ensuring timely adaptations to the model to maintain its effectiveness.

7.5.3 Future work

In terms of future research directions, we recommend an exploration of various combinations of input features. For instance, investigating the fusion of clinical encoder with the text encoder could yield insightful results. Similarly, there exists a compelling opportunity to assess alternative methods of combining the inner representations, moving beyond concatenation. Techniques such as pooling these representations or employing cross-attention mechanisms among different modalities are avenues that warrant investigation and could potentially contribute to enhancing the model's performance and understanding of the underlying data.

7.6 Conclusions

This chapter has focused on the comprehensive examination of the DeepEMC^2 model's updated version, encompassing design, implementation, and evaluation aspects. This revision encompasses alterations in architecture, training procedures, and preprocessing techniques, aimed at addressing challenges associated with the adverse impacts of dataset shifts and facilitate automatic adaptation to evolving feature domains. Importantly, this adaptation can be achieved without the necessity of completely overhauling the model each time a new feature emerges or an existing one is removed.

The findings of this study indicate that the model's performance for predicting outcomes in the following year remains within operational bounds. Consequently, if changes are gradual and not overly drastic, the implemented measures can safeguard a reasonable level of performance for our decision-support model in the context of out-of-hospital medical triage. Nevertheless, it remains vital to diligently monitor alterations in data distribution and performance metrics. This proactive monitoring is essential in order to rapidly respond to potential variations that could be substantial, potentially leading to detrimental performance consequences with significant deployment implications.

Chapter 8

A Deep Learning tool to classify out-of-hospital emergency medical incidents in real time

Previous chapters have encountered the challenge of devising a Deep Learning-based model to facilitate out-of-hospital medical emergency triage. However, should we aspire to deploy this model practically, apart from the foundational backend advancements, there arises a necessity for corresponding front-end enhancements. In other words, we must establish a means through which the dispatcher, a potential end-user of DeepECM^2, can engage with the model without necessitating explicit comprehension of its underlying mechanisms. Consequently, within this chapter, we present the conceptualization and assessment, with some archetypical cases, of a prototypical tool integrated by a basic Graphical User Interface (GUI) and the DeepECM^2, including the necessary preprocessing operations before feed the data to the deep model. This tool serves the purpose of enabling interaction between a user—potentially an out-of-hospital emergency medical dispatcher—and DeepEMC^2, the Deep Learning model conceived in preceding chapters. The tool is meticulously structured to retain identical inputs as those employed by an emergency medical dispatcher in the Valencian Region's genuine operational context. Moreover, the computational efficiency of our models during inference ensures real-time responsiveness, aligning aptly with the exigencies of online problem-solving. The creation of this tool signifies a substantial stride forward, facilitating users' engagement with DeepEMC^2 in an intuitive and accessible manner. Lastly, it is noteworthy that ongoing collaboration with Omda, the multinational entity overseeing the CoordCom system, is underway. The objective is to formulate and integrate a Clinical Decision Support System (CDSS) for real-time interaction with DeepEMC^2, within the emergency medical dispatch center of the Valencian Region.

The contents of this chapter were presented at the conference (Ferri et al., 2022b)—thesis contributions C6 and P2. In addition, the GUI is available online at ("Deep multitask ensemble classification of emergency medical call incidents combining multimodal data", 2023).

8.1 Introduction

Tools designed to facilitate interaction between users and complex Deep Learning models play a crucial role in simplifying the adoption of advanced Machine Learning technologies. These tools serve as a bridge, enabling users to harness the capabilities of Deep Learning models without requiring an in-depth understanding of the underlying algorithms. This user-friendly approach ensures that individuals and organizations can seamlessly engage with Deep Learning models, even without delving into the intricacies of neural networks, optimization techniques, or feature engineering. Instead, users can focus on obtaining real-time predictions and insights that enhance their decision-making processes (Berner, 2007).

The value of these interaction tools is particularly pronounced in various domains where Deep Learning models are applied. They abstract away technical complexities, allowing users to make informed decisions based on AI-driven recommendations. Whether it involves tasks like image recognition, natural language understanding, or predictive analytics, these tools empower users to integrate Deep Learning capabilities into their workflows effortlessly. For instance, such a tool could enable users to input data and receive predictions or classifications generated by the model's outputs, all without requiring them to possess an extensive grasp of the intricate mathematical details and algorithms that power the model (Milde et al., 2018; Yaacoub et al., 2022).

Furthermore, these interaction tools assume a pivotal role in enabling real-time engagement with Deep Learning models, a critical aspect in applications like autonomous vehicles or emergency response systems, mirroring the context of our thesis work. In scenarios where swift and accurate decisions are imperative, a straightforward interaction tool acts as a conduit for users to receive real-time predictions promptly. This capability empowers users to make informed decisions based on the model's outputs, thereby enhancing operational efficiency and safety in time-sensitive contexts (R. T. Sutton et al., 2020).

In preceding chapters, our primary focus has revolved around the development of a model aimed at achieving optimal predictive performance for severity labels, all while effectively navigating the challenges posed by evolving dataset distributions over time. However, the current chapter signifies a pivotal shift in our narrative. Here, our attention turns toward constructing and demonstrating a practical tool

that facilitates the model's utilization by emergency medical dispatchers. This tool seamlessly combines a basic GUI with the capabilities of DeepEMC^2.

While this tool currently exists as a prototype designed for demonstration purposes, it represents a foundational step. It lays the groundwork for the integration of our model into the information system that governs the emergency medical dispatch service within the Valencian Region. Future advancements in this endeavor will encompass the seamless incorporation of DeepEMC^2 into Omda's information system, more specifically, the CoordCom information system that is currently in active use. This progressive integration will establish a direct and immediate interaction channel between dispatchers and the Deep Learning model, thereby enhancing the provision of triage support in real-time scenarios.

8.2 Materials and methods

8.2.1 Deep Learning tool design and implementation

Overview

The DeepEMC^2 interaction tool, accessible via ("Deep multitask ensemble classification of emergency medical call incidents combining multimodal data", 2023), embodies a web-based graphical user interface meticulously devised to facilitate user interaction with the Deep Learning model DeepEMC^2. Its primary purpose revolves around supporting the out-of-hospital emergency medical triage procedure by effectuating predictions of three critical labels: the gravity level pertaining to potentially life-threatening situations, permissible response timeframes, and jurisdiction allocation within the emergency system.

Furthermore, owing to the Spanish linguistic nature of the training data and the targeted user base encompassing dispatchers of the emergency medical dispatch service within the Valencian Region, the tool has been configured to operate exclusively in Spanish.

Framework

The tool interface was developed using the Django framework (Forcier et al., 2008) in Python (G. van Rossum (Guido), 1995), integrating the Pytorch (Paszke et al., 2017) implementation of the DeepEMC^2 model.

Input data

The input data within the DeepEMC^2 tool interface is categorized across distinct input sections, delineated as follows:

- **Contextual data**: encompassing age and sex variables.
- **Clinical variables**: these are the pertinent clinical variables aligned with each question within the in-house triage protocol.
- **Free text dispatcher's observations**: this entails the provision of unstructured textual information, to be inputted within a dedicated text box.

It is noteworthy that date-derived variables are not manually entered as they undergo automatic generation. Similarly, the calculation of the number of involved patients is automated. Importantly, it is worth emphasizing that input fields need not be uniformly populated to initiate the model, as DeepEMC^2 has been engineered to accommodate the lack of data. However, it is prudent to acknowledge that the omission of information can potentially influence prediction quality. Consequently, it is advisable to supply the model with as much available data as possible to enhance prediction performance.

Output data

The tool furnishes the user with an array of prediction data linked to each of the three severity labels: namely, life-threatening, admissible response delay, and emergency system jurisdiction. Simultaneously, it warrants emphasis that the probabilities corresponding to each class within every label are visually presented through color-coded bars. This presentation mechanism serves the purpose of imparting to the user a sense of the prediction's inherent uncertainty.

Data privacy

The data inputted by the user within the interface tool and the resulting predicted labels are intentionally not retained, a measure driven by considerations surrounding privacy and memory constraints. This particular functionality is slated for incorporation in forthcoming iterations of the tool, an initiative undertaken in partnership with Omda. This integration process will culminate in the seamless assimilation of the tool into the CoordCom emergency medical dispatch information system.

8.2.2 Basic functionality assessment

We evaluated the basic functionality of our tool, ensuring its capability to receive user input information, preprocess it, input it to the model, generate predictions, and retrieve them to the user in real-time and in an understandable manner. However, aspects such as user satisfaction and usability are beyond the scope of our work and will be addressed in later development phases during collaboration with Omda. This collaboration will occur when the final decision support tool for use in the Valencian dispatch center is being developed. Therefore, we emphasize that the tool presented in this chapter serves as an example of how an end-user might interact with DeepEMC^2 without knowledge of its implementation details and should be considered a prototype.

To execute this functionality evaluation, we curated a repertoire of exemplary scenarios mirroring real-world instances encountered within out-of-hospital emergency medical contexts. These incidents presented different severity, as well as diverse input data availability. Furthermore, it is pertinent to clarify that although these scenarios are featured in Table 8.1 in the English language, they were originally written in Spanish for input into the model. This choice is in alignment with the training data language of DeepEMC^2, which is rooted in Spanish.

Table 8.1: Example cases evaluated with the Graphical User Interface. Although these scenarios are presented in English, they were originally written in Spanish for input into the model. This choice aligns with the language of the training data used for DeepEMC^2, which is Spanish.

Case ID	Age	Sex	Tree	Text
1	22	Female	No previous trauma, fever over 38, flu syndrome	Fever 38, sore throat with general malaise, no history of illness
2	67	Male	No previous trauma, with chest pain, abrupt onset of symptoms, dyspnea	Possible heart attack cold sweat heart pain
3	Unknown	Unknown	Unknown	Possible heart attack cold sweat pain in the heart
4	Unknown	Unknown	Unknown	Cold sweat and pain in heart but rules out heart attack, possible anxiety crisis

8.3 Results

8.3.1 Deep Learning tool design and implementation

We present the web-based GUI implementation, as it is shown on the web in Figure 8.1.

Figure 8.1: User interface of the tool to interact with DeepEMC^2.

Next, we have split the presentation into different parts, demarcating the diverse input categories as well as the resultant outputs:

Contextual data section

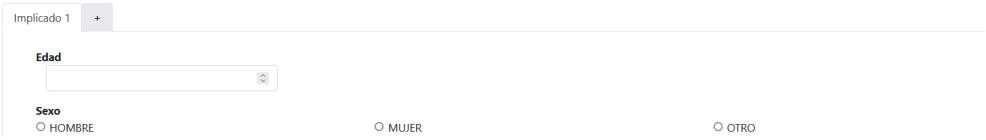

Figure 8.2: Contextual data section of the user interface.

From the observation of Figure 8.2 we can appreciate that the user can input their age via the designated field, or alternatively, through the utilization of arrow-based selection. Sex is implemented with buttons that only allow an exclusive selection—either one or other but not both. Likewise, information about multiple patients can be introduced, wherein users can left the cells unfilled. Complementary context variables as the number of patients involved and date-derived variables are calculated internally, without requiring the user to input them.

Clinical variables section

Figure 8.3: Clinical variables section of the user interface.

The section pertaining to clinical variables, as depicted in Figure 8.3, adopts an iterative scheme, mirroring the structure of the in-house triage protocol, which adheres to a tree-like arrangement. Upon selecting a particular variable, the screen transitions to exhibit the array of subsequent variables accessible along the designated tree path. To accommodate for potential errors, users are equipped with a dedicated button to navigate backward, if necessary, allowing them to re-enter another node that might be better suited for case definition. Analogous to the context features, this section is also open to being left unfilled.

Free text observations section

Figure 8.4: Free text box of the user interface.

The inclusion of dispatcher observations in the form of free text is facilitated through a designated text box positioned at the bottom of the GUI, as illustrated in Figure 8.4. This feature is characterized by its intuitive application, as it merely involves entering unstructured data into the provided field. Much akin to the handling of context and clinical data, this free text box is also capable of being left unfilled.

Prediction results

Upon the user's initiation by clicking the *Predict* button within the GUI, the window presenting prediction results is activated. Each distinct label occupies an individual row within this window: the foremost row corresponds to the life-threatening label, followed by the admissible response delay label, and ultimately, the emergency system jurisdiction label. Concurrently, for every label, the probabilities of each respective class are exhibited through horizontal bar graphs, wherein distinct colors are assigned to each class.

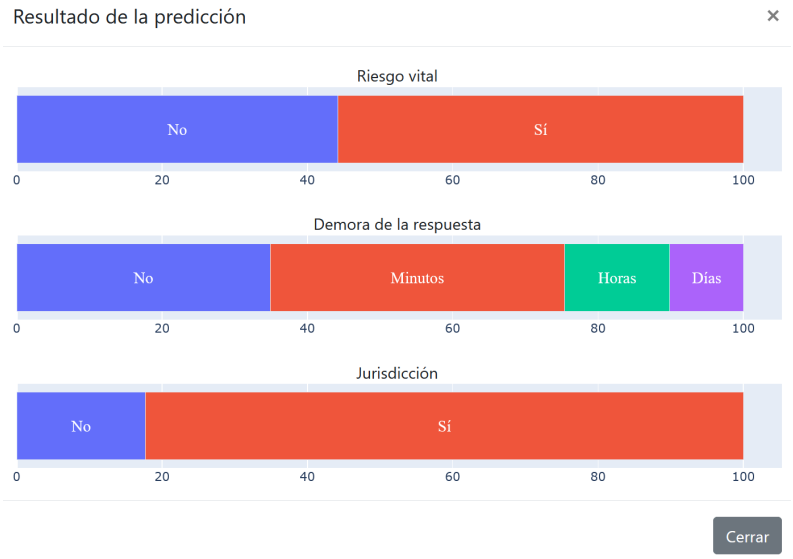

Figure 8.5: Prediction outcomes of tool to interact with DeepEMC^2. Within the figure, *Riesgo vital* stands out for Life-threatening, *Demora en la respuesta* for Admissible response delay, *Jurisdicción* for Emergency system jurisdiction, *Sí* for Yes, *No* for No, *Minutos* for Minutes, *Horas* for Hours and *Días* for Days.

8.3.2 Basic functionality assessment

Next, we proceed to showcase the outcomes corresponding to each of the exemplar cases selected for the evaluation of the implemented tool's functionality:

Case 1

After observing the model outputs as depicted in Figure 8.6, a clear observation emerges. The model is able to discern that the presented case does not pertain to a severe scenario, and further, does not warrant engagement with the emergency medical dispatch system. Instead, it implies a situation that could be suitably addressed within primary care facilities.

Figure 8.6: Predicted outcomes for Case 1 by the model across all severity labels. Distinct colors correspond to each class within the labels. Within the figure, *Riesgo vital* stands out for Life-threatening, *Demora en la respuesta* for Admissible response delay, *Jurisdicción* for Emergency system jurisdiction, *Sí* for Yes, *No* for No, *Minutos* for Minutes, *Horas* for Hours and *Días* for Days.

Case 2

The observations drawn from Figure 8.7 distinctly indicate that the model confers a substantial degree of priority to the facets of life-threatening severity, admissible response delay, and emergency system jurisdiction in a scenario wherein an urgent intervention is imperative, as evident in the case of a heart attack.

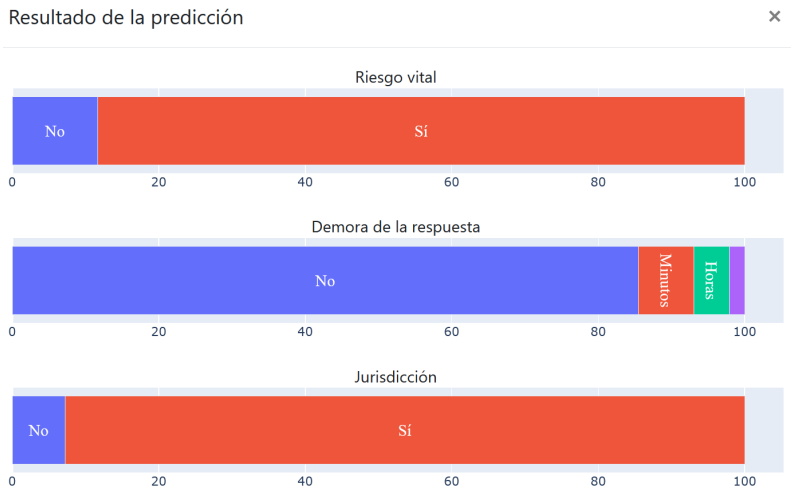

Figure 8.7: Predicted outcomes for Case 2 by the model across all severity labels. Distinct colors correspond to each class within the labels. Within the figure, *Riesgo vital* stands out for Life-threatening, *Demora en la respuesta* for Admissible response delay, *Jurisdicción* for Emergency system jurisdiction, *Sí* for Yes, *No* for No, *Minutos* for Minutes, *Horas* for Hours and *Días* for Days.

Case 3

Through an examination of Figure 8.8, it becomes evident that the model adeptly manages incomplete input data, making accurate predictions solely reliant on the dispatcher's free text observations. Furthermore, it's noteworthy that the model effectively determines the severity, albeit with a slight reduced level of certainty. This outcome aligns with anticipated and desired behavior, as less comprehensive information has been furnished, thus warranting a corresponding decrease in the anticipated level of certainty.

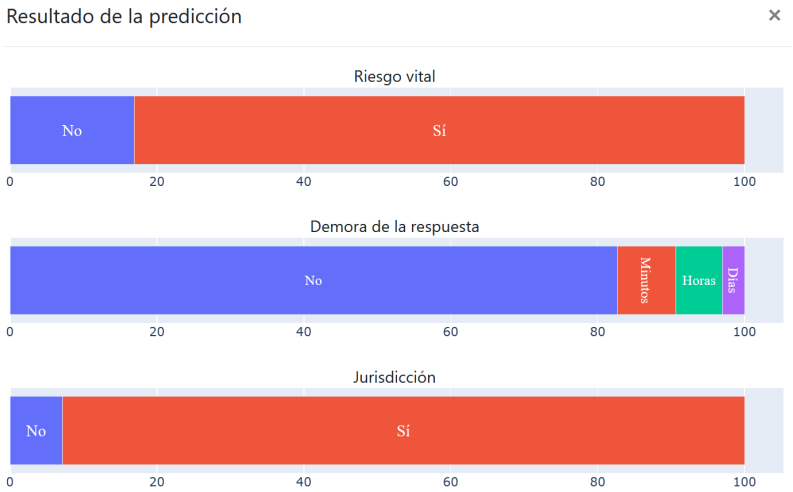

Figure 8.8: Predicted outcomes for Case 3 by the model across all severity labels. Distinct colors correspond to each class within the labels. Within the figure, *Riesgo vital* stands out for Life-threatening, *Demora en la respuesta* for Admissible response delay, *Jurisdicción* for Emergency system jurisdiction, *Sí* for Yes, *No* for No, *Minutos* for Minutes, *Horas* for Hours and *Días* for Days.

Case 4

Upon comparing Figure 8.8 and Figure 8.9, it is observed that the model exhibits natural language comprehension, accurately deciphering contextual nuances. This observation substantiates the fact that the model doesn't rely on keyword-based mechanisms; instead, it demonstrates the ability to perform intricate high-level interpretive tasks. This capacity is particularly pivotal in the context of out-of-hospital emergency medical triage situations.

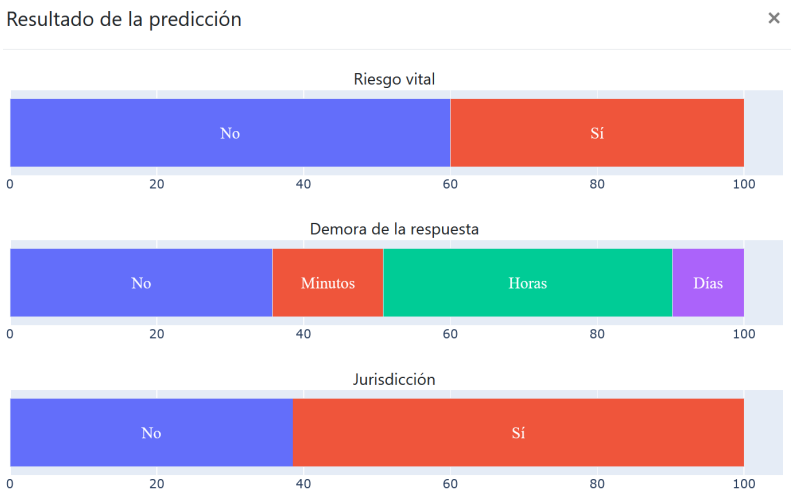

Figure 8.9: Predicted outcomes for Case 4 by the model across all severity labels. Distinct colors correspond to each class within the labels. Within the figure, *Riesgo vital* stands out for Life-threatening, *Demora en la respuesta* for Admissible response delay, *Jurisdicción* for Emergency system jurisdiction, *Sí* for Yes, *No* for No, *Minutos* for Minutes, *Horas* for Hours and *Días* for Days.

8.4 Discussion

8.4.1 Relevance

The outcomes in the preceding section demonstrate the feasibility of developing a tool that bridges the gap between users, potentially including emergency medical dispatchers, and DeepEMC^2. In this paradigm, users are not obligated to possess knowledge or comprehension of the inner workings of the deep model; they can effectively wield it to acquire real-time recommendations concerning life-threatening situations, admissible response delays, and emergency system jurisdiction. This seamless interaction between the user and the model through the GUI embodies a straightforward process.

The examination of various cases has substantiated that the tool responds effectively, aligned with the expected outcomes. This validation serves to underscore the functional efficacy of the ongoing developments. Despite being in the prototype phase, these developments represent a foundational stride towards the eventual integration of the deep model within the operational routine of out-of-hospital emergency medical triage procedures.

8.4.2 Limitations

While we have tested the fundamental functionality of the tool using a few example cases, we have not explored dimensions such as aesthetics and usability through interactions with potential users or incorporated their feedback. This decision aligns with our initial goal, as mentioned at the beginning of this chapter, which was to create a prototypical tool to demonstrate how an end user can interact with the complex deep model. It is essential to note that the final system for integration into the Valencian emergency dispatch center will be developed in collaboration with Omda.

8.4.3 Future work

In terms of future endeavors, we are actively engaged in collaboration with the multinational entity Omda, the architect of the CoordCom system—an information system presently operational within the medical dispatch services of the Valencian Region. The objective of this collaboration lies in the seamless incorporation of DeepEMC^2 into the prevailing system utilized by emergency dispatchers within the Valencian Region. This integration necessitates the requisite adaptations to align the deep model with the existing framework. As a consequence of these modifications, the operator's workflow will remain unaltered; however, the deep model will operate inconspicuously in the background, delivering insights to the emergency medical coordinators stationed within the center.

Subsequent to the embedding of the model within the CoordCom system, our roadmap entails the implementation of additional evaluation and monitoring measures. The objective of this phase is to gauge the contributions of the decision-support model within an authentic operational environment. This evaluative endeavor will take into consideration the dimensions that were not extensively covered within the prototype, thus furnishing a comprehensive assessment of the model's impact and efficacy.

8.5 Conclusions

In the course of this chapter, we have developed a prototype tool aimed at facilitating user interaction—potentially by emergency medical dispatchers—with the DeepEMC^2 model. The tool interface design adheres to principles of simplicity and transparency, granting users the capacity to engage with the model effortlessly. It empowers users to input the same types of information they would utilize in actual emergency scenarios. Although the tool stands as a prototype, it offers an instantaneous response, instantaneously conveying recommendations from the Deep Learning model to the user. This information can be seamlessly integrated into the decision-

making process for out-of-hospital emergency medical triage. Crucially, it is worth underscoring that as of the current juncture in drafting this thesis manuscript, our collaboration with Omda is underway. The objective is to seamlessly integrate the model into the CoordCom system, subsequently evaluating its performance within a real-world context. This phase will be complemented by the necessary adjustments aimed at optimizing the model's value within the workflow of out-of-hospital medical emergencies.

Chapter 9

Concluding remarks and recommendations

This chapter describes the primary concluding remarks and recommendations. In addition to this summary, it offers valuable insights for furthering scientific research and development based on the findings presented.

9.1 Concluding remarks

Technological advancements have been pivotal in enhancing the management of out-of-hospital medical emergencies throughout history. Commencing with the invention of the ambulance and the telephone and evolving to include innovations such as the defibrillator and GPS, these technologies, once groundbreaking, are now deemed essential components of any emergency medical dispatch service. In fact, any service lacking access to these tools would be considered outdated, ill-equipped, and ill-prepared to address out-of-hospital emergencies effectively.

This thesis has been founded on the premise that Artificial Intelligence and Machine Learning tools represent the contemporary innovative technologies that will become significant in the near future within the realm of emergency medical dispatch. Through our research, we have demonstrated that Deep Continual Multimodal Multitask models provide a substantial value when determining incoming incident severity, compared to traditional approaches. This implies that out-of-hospital emergency medical emergency triage has wide room for improvement with the integration of these technologies within emergency medical dispatch workflow. Hence, this thesis has proven the potential of a Deep Continual Multimodal Multitask framework for

triage support while justifying its integration into emergency medical dispatch centers, which could greatly benefit patient outcomes.

In fact, in light of the results from this thesis, various stakeholders are adopting policies to include decision support models of this nature to enhance the triage process. This is evident in the collaboration with the Health Services Department of the Valencian Region and the Omda company (Project PJ2). The objective is to integrate the advanced model developed in this thesis into the Valencian Region's emergency medical dispatch center. In the foreseeable future, Artificial Intelligence and Machine Learning tools will be indispensable components of every emergency medical dispatch center.

Furthermore, beyond the immediate practical impact of the research presented in this thesis on the Valencian Emergency Medical Dispatch Service, our work has made significant contributions to the state-of-the-art in the fields of Machine Learning, Deep Learning, and Biomedical Data Science. The Deep Continual Multimodal Multitask framework developed in this thesis, along with the end-to-end DeepEMC2 architecture and its training procedures, can serve as templates for other researchers working on similar problems, extending beyond the emergency medical context. The complexity of the problem addressed in this thesis has led us to design novel models and strategies to tackle it. These methodologies could be adapted to other domains facing problems with similar structures. The scientific publications in top-ranked journals and presentations at international conferences resulting from this thesis validate the quality of the research conducted in the domains of Biomedical Data Science, Machine Learning, and Deep Learning.

The specific concluding remarks of this thesis are listed as follows:

CR1 Machine Learning, and particularly Deep Learning, has demonstrated significant potential for enhancing the out-of-hospital emergency medical call triage process. In particular, Deep Learning models have exhibited the capability to outperform the existing in-house triage protocol of the Valencian emergency medical dispatch service by a substantial margin. Specifically, regarding macro F1-score, the performance enhancement was of 12.5%, 17.5%, and 5.1% in life-threatening, admissible response delay, and jurisdiction determination, respectively. This improvement suggests that integrating AI-based tools into the emergency medical triage workflow should be seriously considered, given the potential advantages they could offer regarding patient well-being and efficient resource allocation. It is worth noting that while our models do not directly recommend specific resources, their decision-making is strongly influenced by the incident's assigned priority, making them an invaluable asset in optimizing the allocation of resources.

CR2 Free text dispatcher observations represent a valuable source of information that can be effectively harnessed by Deep Learning models and characterized using unsupervised Machine Learning techniques. This realization carries several important implications. Firstly, it suggests that the most pertinent information for assessing the severity of an incident is not necessarily encoded within structured data, such as the clinical variables found in the in-house triage protocol or other triage protocols such as the Manchester triage system or the Emergency severity index. Instead, it appears that the extensive and intricate array of scenarios inherent to emergency medical call incidents demands a more nuanced description, which is furnished by the unstructured nature of free text data. Moreover, Deep Learning has witnessed substantial advancements in recent years. Pretrained natural language processing models, trained on vast datasets, are readily accessible online, often available for free, and can be downloaded and fine-tuned for specific tasks. These models exhibit robust capabilities, including natural language understanding, enabling them to effectively capture intricate patterns within the free text dispatcher observations.

CR3 Dataset shifts are a phenomenon inherently associated with Machine Learning and the medical domain; the context of out-of-hospital emergency medical triage aided by Deep Learning-based models is no exception. Data distributions naturally differ across domains and evolve over time for various reasons that are beyond our control. These shifts in data, if unattended, can lead to significant performance declines that may severely impact the effectiveness of any model deployed to assist in the emergency medical triage process. Given the critical consequences of errors in the emergency medical triage process, it becomes imperative to actively monitor these dataset shifts, promptly detect them, and subsequently take necessary corrective actions to address them.

CR4 Continual Learning techniques are essential for maintaining the Machine Learning model's performance over time, allowing the model to adapt to abrupt and gradual distributional shifts. In the context of the EMCIs in the Valencian Region, we have discovered that fine-tuning techniques exhibit an excellent balance between effectiveness and efficiency. These techniques outperform others considering higher amounts of data during the training phase, such as cumulative learning. This suggests that, for our dataset, prioritizing forward knowledge transfer over backward knowledge transfer is a prudent approach to maximize the performance of our Deep Learning models in the coming years. Additionally, the continual feature domain techniques developed in this thesis have demonstrated their utility in enhancing out-of-sample predictions for the following year.

CR5 Employing a Deep Continual Multimodal Multitask learning approach, combining contextual information, clinical structured information from the in-house protocol, and free text dispatcher observations, while including mechanisms to

mitigate temporal distributional drifts, is a prudent choice, notwithstanding the inherent technical complexity in its design and implementation. This approach leverages information from multiple modalities, enhancing its overall value, and can be implemented as an end-to-end solution, reducing training time and memory requirements while increasing robustness. Moreover, the multitask approach permits a reduction in the number of parameters necessary to capture the intricate patterns present in out-of-hospital emergencies, leveraging the fact that life-threatening, response delay, and jurisdiction labels are closely related. This reduction is particularly significant, as a Continual Learning approach is mandated, and expediting the training process by diminishing model dimensionality is paramount. Additionally, it results in reduced inference times and decreased storage requirements for allocating the model.

CR6 It is feasible to interact with a Deep Continual Multimodal Multitask model, similar to the ones developed in this thesis, by means of an interface tool. This enables users unfamiliar with the intricate computational processes involved, to interact with the model effortlessly. In this setup, the dispatcher provides the input features in a straightforward manner, and the model computes the predicted probabilities for each of the different severity labels, in real-time and in accordance with data privacy concerns. Importantly, data does not need to be stored on the server once predictions for a specific case have been calculated. The initial interface tool prototype presented in this thesis represents the initial step toward integrating the model into the Valencian emergency medical dispatch service. This integration project is currently underway as of the writing of this thesis and is being conducted in collaboration with the company Omda.

9.2 Recommendations

Despite the significant value offered by the models developed in this thesis, Emergency Medical Triage remains a complex challenge, owing to the time and uncertainty constraints involved, coupled with the potential consequences of errors within this context. It necessitates ongoing scrutiny, analysis, and refinement to remain in a state of constant progress and evolution, ultimately providing more effective prioritization of out-of-hospital emergency medical incidents.

The research and developments outlined in this thesis serve as a foundational framework for future research branches and technological advancements. With the overarching goal of perpetually enhancing out-of-hospital emergency medical triage using Machine Learning and Deep Learning tools, we propose the following recommendations:

R1 The most relevant information required for effective out-of-hospital emergency medical triage is often found within unstructured text fields. Consequently,

processes should be geared toward integrating innovative models and tools specialized in handling such unstructured data, thereby enhancing the automated natural language understanding capabilities. Furthermore, the significance of text-based information opens up avenues for new research endeavors, where automatic audio transcription technologies can be combined with natural language understanding techniques. While including free text dispatcher observations already provides substantial value compared to in-house triage protocols, encompassing a comprehensive incident depiction through the transcription of the entire real-time conversation into text could yield even more accurate predictive outcomes.

R2 Monitoring model performance over time is essential because excellent performance metrics at a specific time point do not necessarily guarantee the safe deployment of a model indefinitely. Dataset shifts are inevitable due to natural causes, and their early detection is of paramount importance. It is of utmost importance to characterize these shifts, identify their underlying causes, and assess their effects on both marginal and joint distributions. Understanding whether changes are gradual, abrupt, or recurrent is also relevant as it effectively informs the appropriate strategies for managing and mitigating these shifts.

R3 Continual Learning techniques are indispensable for maintaining and recuperating model performance over time, and we strongly advocate for their adoption in the context of deep out-of-hospital emergency medical incident classification. Furthermore, it is worth emphasizing that fine-tuning the deep model periodically with current out-of-hospital emergency medical data represents a straightforward yet highly effective and efficient strategy for retaining knowledge about past patterns while incorporating information about new ones. To evaluate whether novel Continual Learning approaches provide additional value, it is advisable to compare these approaches against the fine-tuning method, which can serve as a baseline for assessment.

R4 Certain features may emerge and disappear over time, leading to a specific type of dataset shift that presents significant challenges. It is imperative to address these challenges proactively before they manifest. Any model designed to contend with this phenomenon should incorporate mechanisms to handle this issue effectively. In this thesis, we have proposed a straightforward yet effective method for addressing this challenge, which has proven to be beneficial in the context of our out-of-hospital emergency medical call data. When confronting similar issues in problems like the one presented in this thesis, we encourage considering this dynamic feature domain adaptation approach. Its utility and ease of implementation make it a viable choice, serving as a preliminary step before delving into more intricate and resource-intensive strategies, just as we recommended earlier to prioritize fine-tuning before exploring more complex and costly Continual Learning strategies.

R5 The emergency medical dispatcher does not need to possess detailed knowledge or understanding of the inner workings of a deep triage support model running in the background. Their primary requirements are that their workflow remains unaltered, the system interaction is intuitive, and the feedback provided is comprehensible, ultimately aiding them in performing their profession more effectively. Consequently, any decision-support tool designed for implementation in an emergency medical dispatch service should prioritize these considerations. Moreover, with a focus on enhancing professionals' trust in these complex systems, new avenues of research could emerge, centered on the interpretability and explainability of deep models. Providing insights into what aspects of an incident the model is paying attention to (such as specific words, clinical features, or contextual features) could contribute to user acceptance, complementing rigorously calculated evaluation metrics. This transparency in model behavior can help build confidence among users and improve their overall experience with the system.

Bibliography

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631.

Amari, S.-I. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers*, *C–21(11*, 1197–1206.

Angus, D. (2020). Randomized clinical trials of artificial intelligence. *JAMA*, *323*(11), 1043–1045.

Ba, J., Kiros, J., & Hinton, G. (2016). http://arxiv.org/abs/1607.06450

Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, *41*(2), 423–443.

Barrientos, F., & Sainz, G. (2012). Interpretable knowledge extraction from emergency call data based on fuzzy unsupervised decision tree. *Knowledge-based systems*, *25*(1), 77–87.

Barroeta Urquiza, J., & Boada Bravo, N. (2011). Los servicios de emergencia y urgencias médicas extrahospitalarias en españa. *Mensor*.

Bayes, T., & Price, n. (1763). Lii. an essay towards solving a problem in the doctrine of chances. by the late rev (F. b. M. P. Mr. Bayes & A. John Canton, Eds.). *Philosophical Transactions of the Royal Society of London*, *53*, 370–418.

Bellman, R. (1956). Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, *42*(10), 767–769.

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, *13*.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, *24*.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2).

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. (2015). Hyperopt: A Python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, *8*(1), 014008.

Berner, E. S. (2007). *Clinical decision support systems* (Vol. 233). Springer.

Bertsekas, D. (1994). Incremental least squares methods and the extended kalman filter. *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, *2*, 1211–1214.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Blagg, C. R. (2004). Triage: Napoleon to the present day. *Journal of nephrology*, *17*(4), 629–632.

Blandford, A., & William Wong, B. (2004). Situation awareness in emergency medical dispatch. *International Journal of Human-Computer Studies*, *61*(4), 421–452.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Blomberg, S. N., Folke, F., Ersbøll, A. K., Christensen, H. C., Torp-Pedersen, C., Sayre, M. R., Counts, C. R., & Lippert, F. K. (2019). Machine learning as a supportive tool to recognize cardiac arrest in emergency calls. *Resuscitation*, *138*, 322–329.

Bottou, L. (1998). Online algorithms and stochastic approximations. *Online learning in neural networks*.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. wadsworth int. *Group*, *37*(15), 237–251.

Brochu, E., Cora, V., & Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning [arXiv:1012.2599). arXiv.].

Caruana, R. (1997). Multitask learning. *Machine learning*, *28*, 41–75.

Channouf, N., L'Ecuyer, P., Ingolfsson, A., & Avramidis, A. (2007). The application of forecasting techniques to modeling emergency medical system calls in calgary, alberta. *Health Care Management Science*, *10*(1), 25–45.

Chen, A., & Lu, T.-Y. (2014). A gis-based demand forecast using machine learning for emergency medical services [1634–1641.].

Cheng, X., Cao, Q., & Liao, S. (2020). An overview of literature on covid-19, mers and sars: Using text mining and latent dirichlet allocation. *Journal of Information Science*, *0165551520954674*.

Chib, S., & Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The American Statistician*, *49*(4), 327–335.

Cho, K., Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014).

Christ, M., Grossmann, F., Winter, D., Bingisser, R., & Platz, E. (2010). Modern triage in the emergency department. *Deutsches Ärzteblatt International*, *107*(50), 892–898.

Ciliberto, C., Mroueh, Y., Poggio, T., & Rosasco, L. (2015). Convex learning of multiple tasks and their structure. *International Conference on Machine Learning*, 1548–1557.

Clawson, J. J., & Dernocoeur, K. B. (1988). Principles of emergency medical dispatch. *(No Title)*.

Clawson, J. (1981). Dispatch priority training: Strengthening the weak link. *JEMS*, *6*(2), 32–35.

Cohen, A. (2011). Fuzzywuzzy: Fuzzy string matching in python. *ChairNerd Blog*, *22*, 51.

Considine, J., LeVasseur, S., & Villanueva, E. (2004). The australasian triage scale: Examining emergency department nurses' performance using computer and paper scenarios. *Annals of Emergency Medicine*, *44*, 516–523.

Dalkey, N. (1969). An experimental study of group opinion. *Futures*, *1*(5), 408–426.

Deep multitask ensemble classification of emergency medical call incidents combining multimodal data [Accessed on 1st September 2023]. (2023). Biomedical Data Science Lab. http://112inteligenciaartificial.upv.es/

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

de Estadística, I. N. (2022). *Avance de la estadística del padrón continuo a 1 de enero de 2022* [2023-10-02]. https://www.ine.es/prensa/pad_2022_p.pdf

d'Emergències Sanitàries de la Comunitat Valenciana, S. (2022). *Sescv en cifras* [2023-10-02]. https://ses.san.gva.es/es/ses-en-cifras

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding [arXiv:1810.04805). arXiv.]. http://arxiv.org/abs/1810.04805

Dwarampudi, M., & Reddy, N. (2019). Effects of padding on lstms and cnns [arXiv:1903.07288). arXiv.]. http://arxiv.org/abs/1903.07288

Efron, B., & Tibshirani, R. (1994). *An introduction to the bootstrap*. CRC Press.

Ek, B., Edström, P., Toutin, A., & Svedlund, M. (2013). Reliability of a swedish pre-hospital dispatch system in prioritizing patients. *International emergency nursing*, *21*(2), 143–149.

Face., D.-b.-s. ·. H. (2023, October 6). https://huggingface.co/dccuchile/albert-base-spanish

Farand, L., Leprohon, J., Kalina, M., Champagne, F., Contandriopoulos, A., & Preker, A. (1995). The role of protocols and professional judgement in emergency medical dispatching. *European Journal of Emergency Medicine*, *2*(3), 136–148.

Ferri, P., Romero-Garcia, N., Badenes, R., Lora-Pablos, D., García Morales, T., de la Cámara, G. A., García-Gómez, J., & Sáez, C. (2023). Extremely missing numerical data in electronic health records for machine learning can be managed through simple imputation methods considering informative missingness: A comparative of solutions in a covid-19 mortality case study. *Computer Methods and Programs in Biomedicine*, 107803.

Ferri, P., Sáez, C., Félix-De Castro, A., Juan-Albarracín, J., Blanes-Selva, V., Sánchez-Cuesta, P., & García-Gómez, J. (2021). Deep ensemble multitask classification of emergency medical call incidents combining multimodal data improves emergency medical dispatch. *Artificial Intelligence in Medicine*, *117*, 102088.

Ferri, P., Sáez, C., Félix-De Castro, A., Sánchez-Cuesta, P., & García-Gómez, J. (2022a). Discovering key topics in emergency medical dispatch from free text dispatcher observations. *Studies in Health Technology and Informatics*, *294*, 859–863.

Ferri, P., Sáez, C., Félix-De Castro, A., Sánchez-García, A., Sánchez-Cuesta, P., & García-Gómez, J. (2022b). An artificial intelligence tool to classify emergency medical incidents in real time improves emergency medical dispatch. *EENA Conference and Exhibition*.

FitzGerald, G., Jelinek, G., Scott, D., & Gerdtz, M. (2010). Emergency department triage revisited. *Emergency Medicine Journal*, *27*(2), 86–92.

Forcier, J., Bissex, P., & Chun, W. J. (2008, October). *Python Web Development with Django* [Google-Books-ID: M2D5nnYlmZoC]. Addison-Wesley Professional.

Forslund, K., Kihlgren, A., & Kihlgren, M. (2004). Operators' experiences of emergency calls. *Journal of Telemedicine and Telecare*, *10*(5), 290–297.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. Elsevier.

G. van Rossum (Guido). (1995, January). Python reference manual. Retrieved March 8, 2022, from https://ir.cwi.nl/pub/5008

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, *46*(4), 1–44 37.

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6(6*, 721–741.

Gilboy, N., Tanabe, P., Travers, D., Rosenau, A., & Eitel, D. (2012).

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gradient flow in recurrent nets: The difficulty of learning longterm dependencies. (2009). In J. Kolen & S. Kremer (Eds.), *A field guide to dynamical recurrent networks. ieee*.

Guo, L. L., Pfohl, S. R., Fries, J., Johnson, A. E., Posada, J., Aftandilian, C., Shah, N., & Sung, L. (2022). Evaluation of domain generalization and adaptation on improving model robustness to temporal dataset shift in clinical medicine. *Scientific reports*, *12*(1), 2726.

Guo, L. L., Steinberg, E., Fleming, S. L., Posada, J., Lemmon, J., Pfohl, S. R., Shah, N., Fries, J., & Sung, L. (2023). Ehr foundation models improve robustness in the presence of temporal distribution shift. *Scientific Reports*, *13*(1), 3767.

HAN, W. (2022). Intelligent telephone triage in pre-hospital emergency care.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification [ArXiv:1502.01852 [Cs].]. http://arxiv.org/abs/1502.01852

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 770–778.

Hecht-Nielsen. (1989). Theory of the backpropagation neural network. *International 1989 Joint Conference on Neural Networks*, 593–605 1.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network [ArXiv:1503.02531 [Cs, Stat].]. http://arxiv.org/abs/1503.02531

Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors [arXiv:1207.0580). arXiv.].

Hirschberg, J., & Manning, C. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266.

Hjälte, L., Suserud, B.-O., Herlitz, J., & Karlberg, I. (2007). Why are people without medical needs transported by ambulance? a study of indications for pre-hospital care. *European Journal of Emergency Medicine*, *14*(3), 151–156.

Ho, T. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282 1.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hoffman, M., Bach, F., & Blei, D. (2010). Online learning for latent dirichlet allocation. *advances in neural information processing systems*, *23*.

Hofmann, T. (1999). Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 50–57.

I.C.D. (2021). Icd-9-cm - international classification of diseases. *Ninth Revision, Clinical Modification*. https://www.cdc.gov/nchs/icd/icd9cm.htm

Inokuchi, R., Iwagami, M., Sun, Y., Sakamoto, A., & Tamiya, N. (2022). Machine learning models predicting undertriage in telephone triage. *Annals of Medicine*, *54*(1), 2989–2996.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448–456.

James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Janocha, K., & Czarnecki, W. (2017). On loss functions for deep neural networks in classification [arXiv:1702.05659). arXiv.]. http://arxiv.org/abs/1702.05659

Jia, Y. (2019). Attention mechanism in machine translation. *Journal of physics: conference series*, *1314*(1), 012186.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, *21*, 345–383.

Jordan, M., & Mitchell, T. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260.

Kacprzyk, J., & Pedrycz, W. (Eds.). (2015). *Springer handbook of computational intelligence*. Springer.

Kingma, D., & Ba, J. (2017). Adam: A method for stochastic optimization [ArXiv:1412.6980 [Cs].]. http://arxiv.org/abs/1412.6980

Klement, P., & Snášel, V. (2011). Using som in the performance monitoring of the emergency call-taking system. *Simulation Modelling Practice and Theory*, *19*(1), 98–109.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, *2*, 1137–1143.

Krogh, A., & Hertz, J. (1991). A simple weight decay can improve generalization. *Advances in neural information processing systems*, *4*.

Kull, M., & Flach, P. (2014). Patterns of dataset shift. *First international workshop on learning over multiple contexts (LMCE) at ECML-PKDD*, *5*.

Kwiatkowski, D., Phillips, P., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, *54*(1), 159–178.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations [arXiv:1909.11942). arXiv.].

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 7553.

Lefter, I., Rothkrantz, L., Leeuwen, D., & Wiggers, P. (2011). Automatic stress detection in emergency (telephone) calls. *International Journal of Intelligent Defence Support Systems*, *4*(2), 148.

Lemmon, J., Guo, L. L., Posada, J., Pfohl, S. R., Fries, J., Fleming, S. L., Aftandilian, C., Shah, N., & Sung, L. (2023). Evaluation of feature selection methods for preserving machine learning performance in the presence of temporal dataset shift in clinical medicine. *Methods of Information in Medicine*, *62*(01/02), 060–070.

Leprohon, J., & Patel, V. (1995). Decision-making strategies for telephone triage in emergency medical services. *Medical Decision Making*, *15*(3), 240–253.

Lidal, I. B., Holte, H. H., & Vist, G. E. (2013). Triage systems for pre-hospital emergency medical services-a systematic review. *Scandinavian journal of trauma, resuscitation and emergency medicine*, *21*(1), 1–6.

Liu, D., & Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, *45*(1–3), 503–528.

Lomonaco, V., Pellegrini, L., Cossu, A., Carta, A., Graffieti, G., Hayes, T., Lange, M., Masana, M., Pomponi, J., Ven, G., Mundt, M., She, Q., Cooper, K., Forest, J., Belouadah, E., Calderara, S., Parisi, G., Cuzzolin, F., Tolias, A., & Maltoni, D. (2021). Avalanche: An end-to-end library for continual learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW*, 3595–3605.

Lomonaco, V. (2019). Continual learning with deep architectures.

Lopez-Paz, D., & Ranzato, M. (2017). Gradient episodic memory for continual learning.

Loshchilov, I., & Hutter, F. (2017). Sgdr: Stochastic gradient descent with warm restarts [arXiv:1608.03983). arXiv.].

Loshchilov, I., & Hutter, F. (2019). http://arxiv.org/abs/1711.05101

Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. (2013). Rectifier nonlinearities improve neural network acoustic models. *Proc. icml*, *30*(1), 3.

Mackway-Jones, K., Marsden, J., & Windle, J. (2013). *Emergency triage: Manchester triage group*. John Wiley & Sons.

Maimon, O., & Rokach, L. (Eds.). (2010). *Data mining and knowledge discovery handbook*. Springer US.

Malsburg, C. (1986). Frank rosenblatt: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. In G. Palm & A. Aertsen (Eds.), *Brain theory* (pp. 245–248). Springer.

Manning, C., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*, *506*.

Maxwell, M., Henderson, S., & Topaloglu, H. (2009). Ambulance redeployment: An approximate dynamic programming approach. *Proceedings of the 2009 Winter Simulation Conference (WSC*, 1850–1860.

McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In G. Bower (Ed.), *Psychology of learning and motivation* (pp. 109–165, Vol. 24). Academic Press.

McKinney, W. (2010). Data Structures for Statistical Computing in Python, 56–61.

McLay, L., & Mayorga, M. (2013). A model for optimally dispatching ambulances to emergency calls with classification errors in patient priorities. *IIE Transactions*, *45*(1), 1–24.

Milde, S., Liebgott, A., Wu, Z., Feng, W., Yang, J., Mauch, L., Martirosian, P., Bamberg, F., Nikolaou, K., Gatidis, S., Schick, F., Yang, B., & Küstner, T. (2018). Graphical user interface for medical deep learning—application to magnetic resonance imaging. *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC*, 838–847.

Moreno-Torres, J., Raeder, T., Alaiz-Rodríguez, R., Chawla, N., & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, *45*(1), 521–530.

Moskop, J. C., & Iserson, K. V. (2007). Triage in medicine, part ii: Underlying values and principles. *Annals of emergency medicine*, *49*(3), 282–287.

Moss, E. (2018). "dial 999 for help!" the three-digit emergency number and the transnational politics of welfare activism, 1937–1979. *Journal of Social History*, *52*(2), 468–500.

Murray, M., Bullard, M., & Grafstein, E. (2004). Revisions to the canadian emergency department triage and acuity scale implementation guidelines. *CJEM*, *6*, 421–427.

Nelder, J., & Wedderburn, R. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General*, *135*(3), 370–384.

Ng, A. Y. (2004). Feature selection, l 1 vs. l 2 regularization, and rotational invariance. *Proceedings of the twenty-first international conference on Machine learning*, 78.

Novák, V., Perfilieva, I., & Mockor, J. (2012). *Mathematical principles of fuzzy logic* (Vol. 517). Springer Science & Business Media.

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning [arXiv:1811.03378). arXiv.]. http://arxiv.org/abs/1811.03378

Palumbo, L., Kubincanek, J., Emerman, C., Jouriles, N., Cydulka, R., & Shade, B. (1996). Performance of a system to determine ems dispatch priorities. *The American Journal of Emergency Medicine*, *14*(4), 388–390.

Parisi, G., Kemker, R., Part, J., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, *113*, 54–71.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in PyTorch. Retrieved January 23, 2023, from https://openreview.net/forum?id=BJJsrmfCZ

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, *12*, 2825–2830.

Pérez, J., Pérez, A., Casillas, A., & Gojenola, K. (2018). Cardiology record multi-label classification using latent dirichlet allocation. *Computer methods and programs in biomedicine*, *164*, 111–119.

Pollock, R. A. (2008). Triage and management of the injured in world war i: The diuturnity of antoine de page and a belgian colleague. *Craniomaxillofacial Trauma & Reconstruction*, *1*(1), 63.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. (2008). *Dataset shift in machine learning*. MIT Press.

Raileanu, L., & Stoffel, K. (2004). Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, *41*(1), 77–93.

Ramachandram, D., & Taylor, G. W. (2017). Deep multimodal learning: A survey on recent advances and trends. *IEEE signal processing magazine*, *34*(6), 96–108.

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 399–408.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408.

Rosner, B. (2015). *Fundamentals of biostatistics*. Cengage learning.

Ruder, S. (2017a). An overview of gradient descent optimization algorithms [ArXiv:1609.04747 [Cs].]. http://arxiv.org/abs/1609.04747

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

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Sáez, C., & García-Gómez, J. M. (2018). Kinematics of big biomedical data to characterize temporal variability and seasonality of data repositories: Functional data analysis of data temporal evolution over non-parametric statistical manifolds. *International journal of medical informatics*, *119*, 109–124.

Sáez, C., Gutiérrez-Sacristán, A., Kohane, I., García-Gómez, J. M., & Avillach, P. (2020). Ehrtemporalvariability: Delineating temporal data-set shifts in electronic health records. *Gigascience*, *9*(8), giaa079.

Sáez, C., Liaw, S.-T., Kimura, E., Coorevits, P., & Garcia-Gomez, J. M. (2019). Guest editorial: Special issue in biomedical data quality assessment methods.

Sáez, C., Robles, M., & García-Gómez, J. M. (2017). Stability metrics for multi-source biomedical data based on simplicial projections from probability distribution distances. *Statistical methods in medical research*, *26*(1), 312–336.

Sáez, C., Rodrigues, P. P., Gama, J., Robles, M., & García-Gómez, J. M. (2015). Probabilistic change detection and visualization methods for the assessment of temporal stability in biomedical data quality. *Data Mining and Knowledge Discovery*, *29*, 950–975.

Sáez, C., Zurriaga, O., Pérez-Panadés, J., Melchor, I., Robles, M., & García-Gómez, J. M. (2016). Applying probabilistic temporal and multisite data quality control methods to a public health mortality registry in spain: A systematic approach to quality control of repositories. *Journal of the American Medical Informatics Association*, *23*(6), 1085–1095.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). Distilbert, a distilled version of bert (f. Smaller, cheaper, & lighter, Eds.) [arXiv:1910.01108). arXiv.]. http://arxiv.org/abs/1910.01108

Schuster, M., & Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Seiger, N., van Veen, M., Steyerberg, E., Ruige, M., Van Meurs, A., & Moll, H. (2011). Undertriage in the manchester triage system: An assessment of severity and options for improvement. *Archives of disease in childhood*, *96*(7), 653–657.

Settles, B. (2009). Active learning literature survey.

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 7587.

Spangler, D., Hermansson, T., Smekal, D., & Blomberg, H. (2019). A validation of machine learning-based risk scores in the prehospital setting. *PLoS One*, *14*(12), e0226518.

Sramek, M., Post, W., & Koster, R. W. (1994). Telephone triage of cardiac emergency calls by dispatchers: A prospective study of 1386 emergency calls. *Heart*, *71*(5), 440–445.

Storkey, A., et al. (2009). When training and test sets are different: Characterizing learning transfer. *Dataset shift in machine learning*, *30*(3-28), 6.

Storm-Versloot, M., Ubbink, D., Kappelhof, J., & Luitse, J. (2011). Comparison of an informally structured triage system, the emergency severity index, and the manchester triage system to distinguish patient priority in the emergency department. *Academic Emergency Medicine*, *18*(8), 822–829.

Stratton, S. (1992). Triage by emergency medical dispatchers. *Prehospital and Disaster Medicine*, *7*(3), 263–269.

Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A survey of optimization methods from a machine learning perspective [ArXiv:1906.06821 [Cs, Math, Stat].]. http://arxiv.org/abs/1906.06821

Sutton, R. T., Pincock, D., Baumgart, D. C., Sadowski, D. C., Fedorak, R. N., & Kroeker, K. I. (2020). An overview of clinical decision support systems: Benefits, risks, and strategies for success. *NPJ digital medicine*, *3*(1), 17.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Syed, S., & Spruit, M. (2017). Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA*, 165–174.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tollinton, L., Metcalf, A. M., & Velupillai, S. (2020). Enhancing predictions of patient conveyance using emergency call handler free text notes for unconscious and fainting incidents reported to the london ambulance service. *International Journal of Medical Informatics*, *141*, 104179.

Town, N. (2023). Bert-base-multilingual-uncased-sentiment [Hugging Face Model Hub].

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation [Conference Name: Computing in Science Engineering]. *Computing in Science Engineering*, *13*(2), 22–30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Veladas, R., Yang, H., Quaresma, P., Gonçalves, T., Vieira, R., Sousa Pinto, C., Martins, J., Oliveira, J., & Cortes Ferreira, M. (2021). Aiding clinical triage with text classification. In G. Marreiros, F. Melo, N. Lau, H. Cardoso, & L. Reis (Eds.), *Progress in artificial intelligence* (pp. 83–96). Springer International Publishing.

Ven, G., & Tolias, A. (2019). Three scenarios for continual learning [ArXiv:1904.07734 [Cs, Stat].]. http://arxiv.org/abs/1904.07734

Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial Intelligence Review*, *18*(2), 77–95.

Wagner, R., & Fischer, M. (1974). The string-to-string correction problem. *Journal of the ACM*, *21*(1), 168–173.

Walt, S., Colbert, S., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, *13*(2), 22–30.

Weibel, L., Gabrion, I., Aussedat, M., & Kreutz, G. (2003). Work-related stress in an emergency medical dispatch center. *Annals of emergency medicine*, *41*(4), 500–506.

Werbos, P. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wright, J. (2006). Numerical optimization. springer publication. http://103.62.146.201:8081/xmlui/handle/1/9595

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wuerz, R., Travers, D., Gilboy, N., Eitel, D., Rosenau, A., & Yazhari, R. (2001). Implementation and refinement of the emergency severity index. *Academic Emergency Medicine: Official Journal of the Society for Academic Emergency Medicine*, *8*(2), 170–176.

Yaacoub, J., Gleave, J., Gentile, F., Stern, A., & Cherkasov, A. (2022). Dd-gui: A graphical user interface for deep learning-accelerated virtual screening of large chemical libraries (deep docking.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR*, *99*, 42–49.

Zachariasse, J., Seiger, N., Rood, P., Alves, C., Freitas, P., Smit, F., Roukema, G., & Moll, H. (2017). Validity of the manchester triage system in emergency care: A prospective observational study. *PLOS ONE*, *12*(2), 0170811.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, *8*(3), 338–353.

Zenke, F., Poole, B., & Ganguli, S. (2017). Continual learning through synaptic intelligence [ArXiv:1703.04200 [Cs, q-Bio, Stat].]. http://arxiv.org/abs/1703.04200