Deep continual multitask severity assessment from changing clinical features

Pablo Ferri[1*], Carlos Sáez[1], Antonio Félix-De Castro[2], Purificación Sánchez-Cuesta[2], Juan M García-Gómez[1]

[1*]Biomedical Data Science Laboratory (BDSLab), Instituto de Aplicaciones de las Tecnologías de la Información y de las Comunicaciones Avanzadas (ITACA), Universitat Politècnica de València (UPV), Valencia, Spain.
[2]Conselleria de Sanitat Universal i Salut Pública, Generalitat Valenciana (GVA), Valencia, Spain.

*Corresponding author(s). E-mail(s): pabferb2@upv.es.com;
Contributing authors: carsaesi@upv.es; felix_antdec@gva.es; sanchez_pur@gva.es; juanmig@ibime.upv.es;

Abstract

When developing Machine Learning models to support emergency medical triage, it is important to consider how changes over time in the data distribution can negatively affect the models' performance. The objective of this study was to assess the effectiveness of various Continual Learning pipelines in keeping model performance stable when input features are subject to change over time, including the emergence of new features and the disappearance of existing ones. The model is designed to identify life-threatening situations, calculate its admissible response delay, and determine its institution jurisdiction. We analyzed a total of 1 414 575 events spanning from 2009 to 2019. Our findings demonstrate important performance improvements, up to 7.8% in life-threatening and 14.8% in response delay, in terms of F1-score, when employing deep continual approaches. We noticed that combining fine-tuning and dynamic feature domain updating strategies offers a practical and effective solution for addressing these distributional drifts in medical emergency data.

Keywords: Continual Learning, Deep Learning, Dataset Shifts, Emergency Medical Call Incidents, Emergency Medical Dispatch, Feature Domain Shift

1 Introduction

Out-of-hospital emergency medical triage presents a complex challenge, involving fast-paced decisions with considerable uncertainty, where errors can have fatal consequences. To aid dispatchers and reduce variability among professionals, emergency medical dispatch centers provide clinical guidelines, collectively known as clinical protocols [1]. These protocols include well-known systems such as the Manchester Triage System [2], the Canadian Triage Scale [3] or the Emergency Severity Index [4], which share a common structural arrangement as decision trees with clinical queries and branching pathways leading to a terminal node specifying the assigned priority level for 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 Health Services Department (HSD). Initially inspired by the Manchester Triage System, the protocol underwent iterative adaptations over time, drawing upon the insights and expertise of coordinator physicians. 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, the phenomenon of distributional drifts [5, 6] manifests over time. In the context of healthcare processes and medicine, these distributional variations are intrinsic [7–10], and out-of-hospital emergency medical triage processes in the Valencian Region are no exception. Focusing on this specific context, the occurrence of these shifts is attributed to a multitude of factors. Foremost, it is imperative to underscore the 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 and updates to clinical variables via the evolution of the in-house triage protocol, have collectively exerted their impact over time.

In the context of the Valencian Region, a deep multitask ensemble model named DeepEMC^2 was developed recently [11]. This model is constituted by four main deep neural networks, each one specialized on a specific data type: the *Context network*, the *Clinical network*, the *Text network* and the *Ensemble network*. DeepEMC^2, which is the global network resulting from the combination of the later four, showed promising results, providing a performance increase of 12.5%, 17.5%, and 5.1%, in terms of life-threatening, admissible response delay and emergency system jurisdiction respectively, considering the macro F1-score with respect to the current in-house triage protocol of the Valencian emergency medical dispatch service. However, despite its tested potential, due to data availability reasons, the data considered to train and evaluate the model spanned between 2009 and 2012.

As discussed in previous paragraphs, when developing a Machine Learning-based system for providing triage decision support, it is crucial to consider the dataset shift phenomenon that naturally appears over time [12, 13]. In the Valencian Region, the absence of measures or strategies to mitigate the negative effects of distributional drifts can lead to a gradual decline in system performance, as illustrated in Figure 1. This figure highlights the presence of dataset shifts, as evidenced by the declining performance of the *Clinical network* over time, particularly during the periods between 2012 and 2014 and between 2017 and 2018.

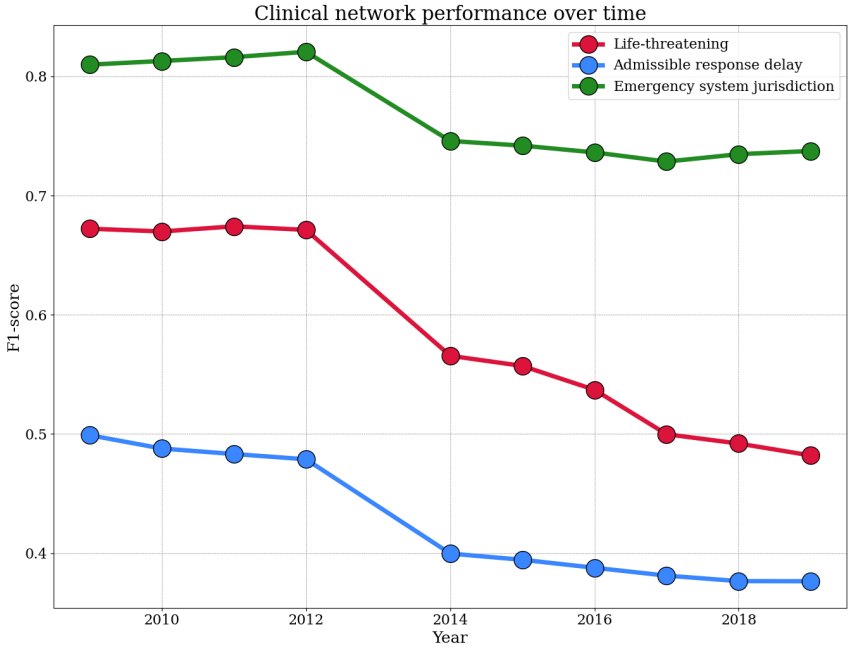


Fig. 1 Performance of the Clinical network of the DeepEMC^2 model over time, for each of the three severity labels, in terms of F1-score. This F1-score is referenced to the positive class in the life-threatening and emergency system jurisdiction label, while it is macro-averaged for the admissible response delay label.

Hence, upon observing the consequences of this phenomenon, 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 [14] 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 [15].

In this study, we have designed Deep Continual Learning pipelines centered on the evolution of model weights for preserving knowledge while allowing adaptability. Additionally, we have investigated the integration of mechanisms to handle the emergence of new clinical features and the obsolescence of existing ones over time. Finally, we have applied them to multiple temporal sets of out-of-hospital emergency medical data from the Valencian Region, to assess their effectiveness to mitigate the negative effects of dataset shifts over time.

2 Materials

2.1 Dataset

We considered a total of 1 414 575 independent out-of-hospital emergency medical incidents from the HSD of the Valencian Region, compiled from 2009 to 2019, excluding 2013—since the emergency information system changed during that year. Data usage was approved by the Institutional Review Board of the HSD. No information that may disclose the identity of the patient was kept for any of the analyses.

The 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 clinical tree variables and values associated with 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.

2.2 Framework

The implementation language of our experiments was Python [16], using the libraries Numpy [17] and Pandas [18] for data management. To implement and train the designed models, we considered PyTorch [19] and HuggingFace's Transformers [20]. Finally, we used Optuna [21] for hyperparameter tuning.

3 Methods

3.1 Data preparation

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. For example, the newly introduced feature "History of diabetes: yes" was equivalent to the previous feature "Patient history: diabetes"; hence, they should not be treated as distinct variables but rather as the same feature. This correspondence mapping process was supervised by experts from the HSD of the Valencian Region. With respect to the labels, they were encoded through one-hot encoding. As such, from the life-threatening label, two variables were derived. Likewise, four variables originated from the admissible response delay label and an additional two from the emergency system jurisdiction label.

Next, we proceeded to segment our data to capture temporal variations while assessing overfitting. We initially partitioned our dataset into distinct learning experiences [22]. The initial experience comprised data solely from the years spanning from 2009 to 2012 and, subsequently, each ensuing experience corresponded to a specific year (2014-2019). This partitioning strategy was selected due to forthcoming architectural and training variations within the *Clinical network* of the DeepECM[2]

model [11]. This network was trained collectively on the entire 2009-2012 data batch, rather than in a year-by-year manner. Thus, if we intend to evaluate the impact of architectural modifications, it is imperative to ensure the use of consistent data sets for performance metric comparisons.

Following this initial partition, another division was performed. For each experience, a training and test split was conducted, allocating 80% for training and 20% for testing. Subsequently, the training set was 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 2 represents how our dataset was divided according to the exposed procedure.

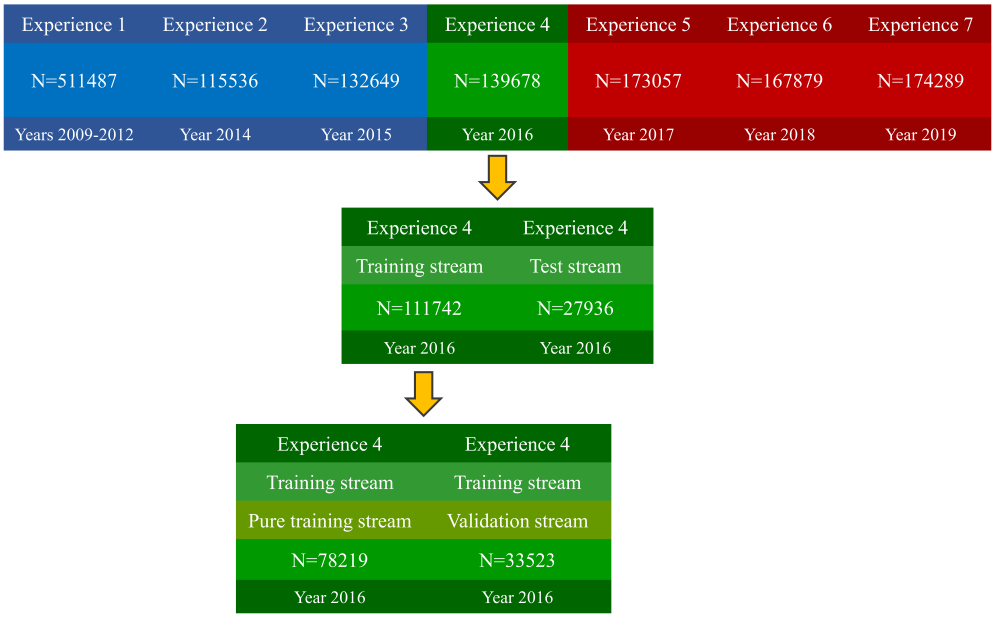


Fig. 2 The data organization process involves the division of data into distinct learning experiences. Concurrently, each experience is characterized by three distinct, non-overlapping data streams: a pure training stream, a validation stream, and a test stream. Here, N symbolizes the volume of data 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.

Subsequently, we transformed the string feature values into indexes. This conversion was necessary to enable the subsequent utilization of an Embedding Layer [23], 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.

3.2 Deep neural network design

Considering the outcomes detailed in [11], where Deep Learning models exhibited superior performance compared to other Machine Learning approaches, the focus in this work remains on models of a similar nature. However, the present study excludes the adoption of recurrent architectures or other sequential models like the Transformer [24]. This exclusion is due to the unavailability of information regarding the exact order in which each clinical variable was recorded during the call for the data spanning from 2014 to 2019. Consequently, our design had to center around an order-independent

model 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 next 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.

3.2.1 Clinical Invariant Network

The Clinical Invariant Network comprises a multitask [25] deep neural network, constituted by two main components: the Indexes Encoder and the Multitask Classifier. The Indexes 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 Indexes Encoder, we constructed it with an initial Embedding Layer [23]. 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. Following the Embedding Layer, an Adaptive Average Pooling block [26] 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 [27], Layer Normalization [28], a GELU activation function [29], and a Dropout Layer [30] 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 3, the main architecture of CliInvNet is visually represented.

3.3 Parameter tuning

Concerning the parameter tuning process, we used the AdamW [31] optimizer, a variant of the Adam [32] algorithm. The feeding paradigm followed was a mini-batch training approach [33], while the loss function considered was the soft F1-score [34]. 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 [35].

Likewise, layers featuring ReLU activation functions were initialized using Kaiming initialization [36], whereas layers incorporating the softmax activation function were initialized with Xavier's initialization [37].

3.4 Continual Learning

In this section, we present Continual Learning strategies that are specifically designed to support the model's adaptation across different experiences over time. It is important to emphasize a crucial distinction at this point, which is the differentiation between approaches that focus on dealing with a feature domain that undergoes temporal fluctuations and methodologies that revolve around the process of updating model weights.

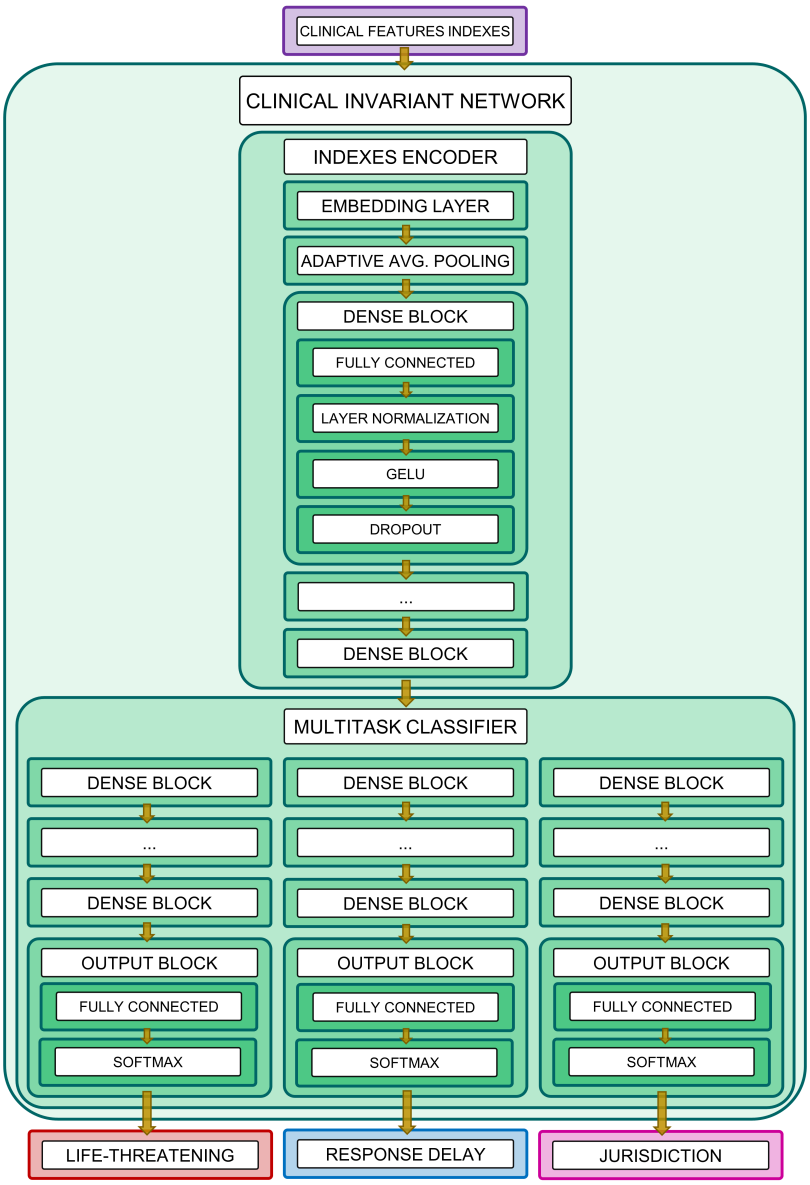


Fig. 3 Clinical Invariant Network architecture. It is a deep multitask neural network composed of two primary components: the Indexes Encoder and the Multitask Classifier. The Indexes Encoder transforms input clinical variables, encoded as indexes, into deep continuous embeddings. The Multitask Classifier then utilizes these embeddings to make predictions. It assesses the probability of a life-threatening event, categorizes the admissible response delay at various levels, and determines whether the incident falls under the jurisdiction of the emergency system.

3.4.1 Variability in feature domain

We propose three different strategies to deal with varying feature domains: a static domain strategy, a dynamic domain strategy, and a predefined domain strategy.

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 *Clinical network* (CliNet) within DeepEMC^2. In this approach, this map remains unchanged after the initial experience and is not updated subsequently. When new variables emerge, if they maintain some correspondence with the 2009-2012 data batch, the map refers to a familiar clinical feature. Conversely, for features that emerge over time without any correspondence to prior variables,

they are linked to the index that designates unknown or infrequent nodes within the CliNet. Hence, 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. 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, 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. This is particularly important as certain subsequent parameter updating strategies involve combining data across multiple experiences.

Predefined domain

The predefined domain strategy employs a predefined embedding matrix derived from a large pretrained natural language processing model. Specifically, for this work, we selected the ALBERT model [38] pretrained on a Spanish corpus [39]. 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.

Under this feature domain approach, the 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.

3.4.2 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.

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 are 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

The fine-tuning strategy entails retraining the model exclusively with data from the most recent learning experience. For each new learning experience, the model begins with the adjusted weights from the previous training session's conclusion. This approach is adopted because it strikes an effective balance: it retains some past information since the model weights are not randomly initialized, yet it primarily facilitates the transfer of forward knowledge, thereby avoiding an over-reliance on past experiences.

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.

3.5 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 [40] 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 [41].

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.

3.6 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, 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.

To assess the real out-of-sample effect inherent to dataset shifts, we computed these metrics for each experience, but considering the training of the model up to the previous experience. This way we can understand how model performance diminishes when applied to novel incoming data, which may exhibit variations in data distributions.

Then, we 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 [42], 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 network* from DeepEMC[2] into this evaluation. This inclusion served as a baseline, allowing for comparing the performance of the pipelines examined in this study.

4 Results

Next, results relative to the evaluation of the Deep Continual Learning pipelines designed are presented:

4.1 Life-threatening

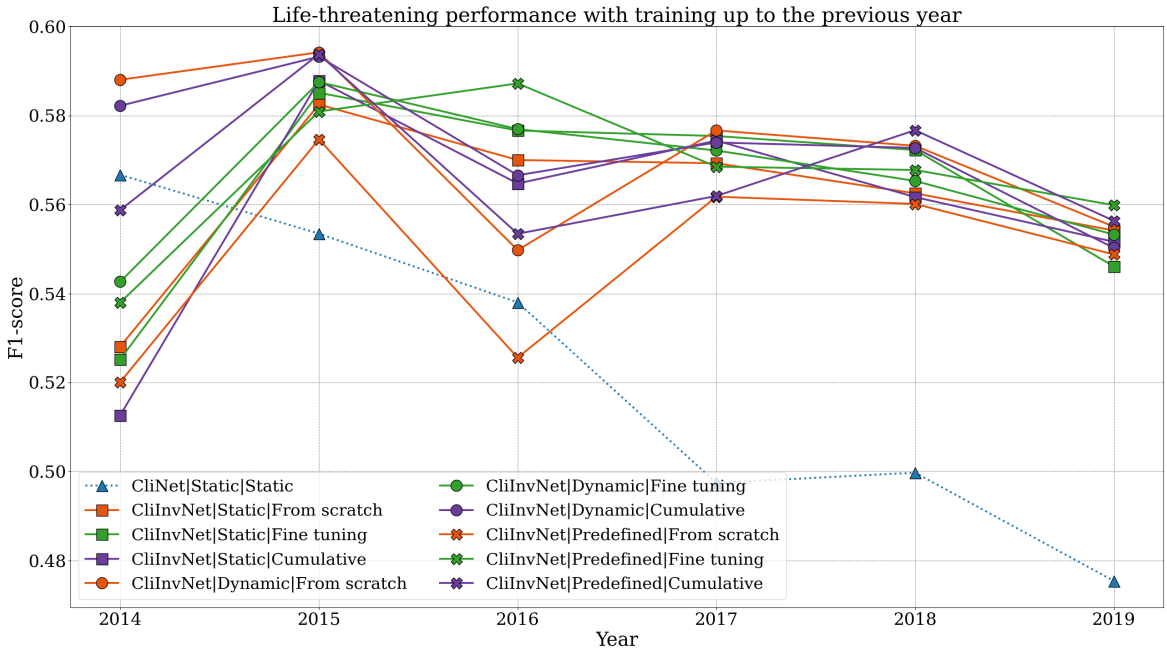


Fig. 4 Life-threatening performance over time with training up to the previous year for each pipeline.

Table 1 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			
	Static	Dynamic	Predefined	Mean
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 4 and Table 1, it becomes evident that the behavior for the subsequent year is moderately erratic, characterized by pronounced and abrupt transitions. In the context of the baseline model, the *Clinical network* 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, followed by a steady performance degradation.

Simultaneously, it is noteworthy to highlight that disparities emerge between the static, the dynamic, and the predefined feature domain approaches, with the dynamic approach yielding superior results. Regarding 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.

4.2 Admissible response delay

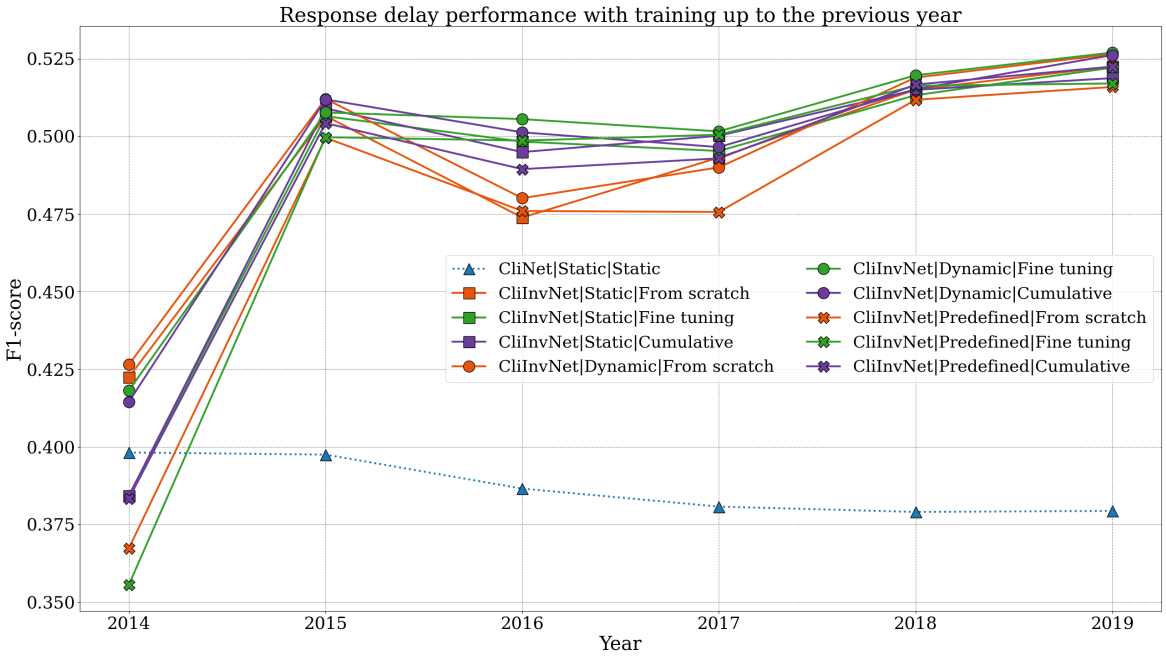


Fig. 5 Admissible response delay performance over time with training up to the previous year for each pipeline.

Table 2 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			
	Static	Dynamic	Predefined	Mean
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 5 and Table 2, 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 network* 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 and the cumulative approach yield the most favorable results, followed by the from-scratch strategy.

4.3 Emergency system jurisdiction

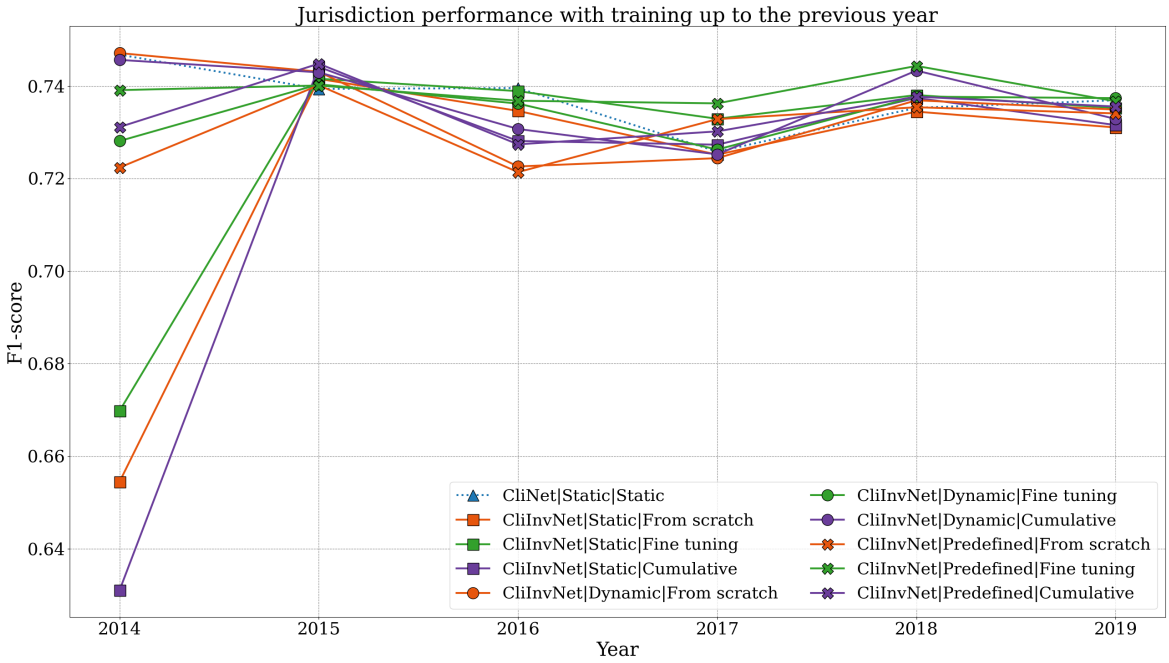


Fig. 6 Emergency system jurisdiction performance over time with training up to the previous year for each pipeline.

Table 3 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			
	Static	Dynamic	Predefined	Mean
From scratch	0.720 [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]**	0.730 [0.728, 0.732]

Upon examining both Figure 6 and Table 3, 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 network* of DeepEMC^2, it is evident that performance remains resilient and consistent over time, with minor fluctuations.

When analyzing 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.

5 Discussion

5.1 Relevance

From the analysis of the feature domain strategies considered, within the context of our out-of-hospital medical emergency data, the dynamic feature domain approach holds substantial value in forecasting for the upcoming year. Therefore, we find it appropriate to select this approach over the static and the predefined methods.

Regarding the parameter updating strategies, we conclude that fine-tuning stands as the optimal choice. Despite not always delivering optimal results, this approach stands out as the dominant strategy, demonstrating a balance of effectiveness and 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 network*, 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 the 2014 to 2019 data batch, which encompasses a higher proportion of primary care cases—not jurisdiction of the emergency system—than the data from 2009 to 2012. Consequently, including these incidents seems to have limited the model's performance in relation to this specific label.

Furthermore, it is important to highlight that the performance attained during the 2009-2012 period is not fully regained afterward, even after retraining. This holds true for both the life-threatening label and the emergency system jurisdiction label. This discrepancy could be attributed to a sample selection phenomenon, where the dataset shift that took place in 2013 led to an increase in the number of non-severe incidents being attended to.

Finally, while there are studies addressing the training and deployment of Machine Learning models in the context of medical data with temporal shifts [12, 13, 43–45], it is challenging to find similar studies for out-of-hospital emergencies. Furthermore, although the predefined feature domain strategy shares some similarities with the domain invariant feature approach proposed by [12] and the foundational model strategy described in [44], our approach in this work differs from previous solutions. We do not rely on raw feature aggregation [43], pre-shift patient weighting [13], or parsimonious models [45]. We instead integrate various strategies to address both the variability in the feature domain and the dynamic nature of parameter updating. This approach effectively evaluates their impact and offers an appropriate solution to the issue of changing feature domains.

5.2 Limitations

Our research has yielded important insights through designing, implementing, and evaluating numerous Deep Continual Pipelines. These pipelines focus on maintaining model performance stability over time, even with changes in the feature domain. However, we recognize that additional configurations remain unexplored. Exploring variants that incorporate strategies like Gradient Episodic Memory [46] or Synaptic Intelligence [47] could be beneficial. Nonetheless, due to the complexity of the problem faced, and the vast array of available options, we prioritized those most applicable to our problem, with a particular emphasis on efficiency and interpretability.

5.3 Future work

In future work, we envision incorporating innovative Continual Learning strategies to enhance the updating of model weights, such as Gradient Episodic Memory or Synaptic Intelligence. 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. Finally, we plan to incorporate the optimal pipelines obtained from this study into the system that will be deployed to support triage in a real scenario based on a Deep Learning approach in the Valencian Region.

6 Conclusions

Throughout time, data in healthcare and medicine naturally undergoes changes due to factors such as population evolution, the introduction of new health policies, and updates in information systems, leading to dataset shifts. These distributional changes, if unaddressed, can severely harm the performance of any Machine Learning model over time. In this work, our focus has been on the design, implementation, and analysis of multiple Continual Learning pipelines centered on providing estimates of the life-threatening level, the admissible response delay, and emergency system jurisdiction of an out-of-hospital emergency medical event, under the presence of these shifts, considering as input features a set of clinical variables which evolve over time. Results from our study reveal that, considering the pool of 1 414 575 out-of-hospital medical emergency events from the Valencian Region, a dynamic feature domain approach combined with a fine-tuning parameter updating strategy stands out as the best option. This approach provides improvements up to 7.8% in life-threatening and 14.8% in response delay, in terms of F1-score, when compared to the baseline *Clinical network*, while offering high efficiency. In the context of our data, when considering these strategies altogether, and excluding the abrupt changes that took place in the 2014 data, performance fluctuations in the subsequent years are overcome.

Acknowledgments. This work has received support from the Ministry of Science, Innovation, and Universities of Spain through the FPU18/06441 program and the KINEMAI project (PID2022-138636OA-I00).

References

[1] Farand, L., Leprohon, J., Kalina, M., Champagne, F., Contandriopoulos, A.P., Preker, A.: The role of protocols and professional judgement in emergency medical dispatching. European Journal of Emergency Medicine **2**(3), 136–148

[2] Mackway-Jones, K., Marsden, J., Windle, J.: Emergency triage: Manchester triage group. John Wiley & Sons (2013)

[3] Murray, M., Bullard, M., Grafstein, E.: Revisions to the canadian emergency department triage and acuity scale implementation guidelines. CJEM **6**, 421–427

[4] Gilboy, N., Tanabe, P., Travers, D.A., Rosenau, A.M., Eitel, D.R. Emergency Severity Index, Version 4: Implementation Handbook. 95.

[5] Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset shift in machine learning. MIT Press (2008)

[6] Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1), 521–530 (2012) https://doi.org/10.1016/j.patcog.2011.06.019

[7] Sáez, C., García-Gómez, J.M.: 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 (2018)

[8] Sáez, C., Gutiérrez-Sacristán, A., Kohane, I., García-Gómez, J.M., Avillach, P.: Ehrtemporalvariability: delineating temporal data-set shifts in electronic health records. Gigascience **9**(8), 079 (2020)

[9] Guo, L.L., Pfohl, S.R., Fries, J., Posada, J., Fleming, S.L., Aftandilian, C., Shah, N., Sung, L.: Systematic review of approaches to preserve machine learning performance in the presence of temporal dataset shift in clinical medicine. Applied clinical informatics **12**(04), 808–815 (2021)

[10] Zhang, A., Xing, L., Zou, J., Wu, J.C.: Shifting machine learning for healthcare from development to deployment and from models to data. Nature Biomedical Engineering **6**(12), 1330–1345 (2022)

[11] Ferri, P., Sáez, C., Félix-De Castro, A., Juan-Albarracín, J., Blanes-Selva, V., Sánchez-Cuesta, P., García-Gómez, J.M.: Deep ensemble multitask classification of emergency medical call incidents combining multimodal data improves emergency medical dispatch. Artificial Intelligence in Medicine **117**, 102088 (2021)

[12] Guo, L.L., Pfohl, S.R., Fries, J., Johnson, A.E., Posada, J., Aftandilian, C., Shah, N., Sung, L.: Evaluation of domain generalization and adaptation on improving model robustness to temporal dataset shift in clinical medicine. Scientific reports **12**(1), 2726 (2022)

[13] Lee, S., Yin, C., Zhang, P.: Stable clinical risk prediction against distribution shift in electronic health records. Patterns **4**(9) (2023)

[14] Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113**, 54–71 (2019) https://doi.org/10.1016/j.neunet.2019.01.012

[15] Hadsell, R., Rao, D., Rusu, A.A., Pascanu, R.: Embracing change: Continual learning in deep neural networks. Trends in cognitive sciences **24**(12), 1028–1040 (2020)

[16] G. van Rossum (Guido): Python reference manual. CWI (1995). https://ir.cwi.nl/pub/5008 Accessed 2022-03-08

[17] Walt, S., Colbert, S.C., Varoquaux, G.: The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science Engineering **13**(2), 22–30 (2011) https://doi.org/10.1109/MCSE.2011.37 . Conference Name: Computing in Science Engineering

[18] McKinney, W.: Data Structures for Statistical Computing in Python, Austin, Texas, pp. 56–61 (2010). https://doi.org/10.25080/Majora-92bf1922-00a . https://conference.scipy.org/proceedings/scipy2010/mckinney.html Accessed 2022-01-03

[19] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch (2017). Accessed 2023-01-23

[20] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)

[21] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631. ACM, Anchorage AK USA (2019). https://doi.org/10.1145/3292500.3330701 . https://dl.acm.org/doi/10.1145/3292500.3330701 Accessed 2022-01-03

[22] Lomonaco, V., Pellegrini, L., Cossu, A., Carta, A., Graffieti, G., Hayes, T.L., Lange, M., Masana, M., Pomponi, J., Ven, G.M., Mundt, M., She, Q., Cooper, K., Forest, J., Belouadah, E., Calderara, S., Parisi, G.I., Cuzzolin, F., Tolias, A.S., Maltoni, D.: Avalanche: An end-to-end library for continual learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW, pp. 3595–3605 (2021). https://doi.org/10.1109/CVPRW53098.2021.00399

[23] Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. Advances in neural information processing systems **13** (2000)

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[25] Caruana, R.: Multitask learning. Machine learning **28**, 41–75 (1997)

[26] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)

[27] Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review **65**, 386–408 https://doi.org/10.1037/h0042519

[28] Ba, J.L., Kiros, J.R., Hinton, G.E. Layer Normalization. ArXiv:1607.06450 [Cs, Stat]. (2016). http://arxiv.org/abs/1607.06450

[29] Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)

[30] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580). arXiv. (2012). https://doi.org/10.48550/arXiv.1207.0580 . https://doi.org/10.48550/arXiv.1207.0580

[31] Loshchilov, I., Hutter, F. Decoupled Weight Decay Regularization (arXiv:1711.05101). arXiv. (2019). http://arxiv.org/abs/1711.05101

[32] Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. ArXiv:1412.6980 [Cs]. (2017). http://arxiv.org/abs/1412.6980

[33] Bertsekas, D.P.: Incremental least squares methods and the extended kalman filter. In: Proceedings of 1994 33rd IEEE Conference on Decision and Control, vol. 2, pp. 1211–1214. https://doi.org/10.1109/CDC.1994.411166

[34] Janocha, K., Czarnecki, W.M.: On Loss Functions for Deep Neural Networks in Classification. arXiv:1702.05659). arXiv. (2017). http://arxiv.org/abs/1702.05659

[35] Loshchilov, I., Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:1608.03983). arXiv. https://doi.org/10.48550/arXiv.1608.03983 . https://doi.org/10.48550/arXiv.1608.03983

[36] He, K., Zhang, X., Ren, S., Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. ArXiv:1502.01852 [Cs]. http://arxiv.org/abs/1502.01852

[37] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010). JMLR Workshop and Conference Proceedings

[38] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv:1909.11942). arXiv. https://doi.org/10.48550/arXiv.1909.11942 . https://doi.org/10.48550/arXiv.1909.11942

[39] Face., D.-b.-s.H. https://huggingface.co/dccuchile/albert-base-spanish

[40] Settles, B.: Active learning literature survey (2009)

[41] Bellman, R.: Dynamic programming and lagrange multipliers. Proceedings of the National Academy of Sciences of the United States of America **42**(10), 767–769

[42] Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. CRC Press

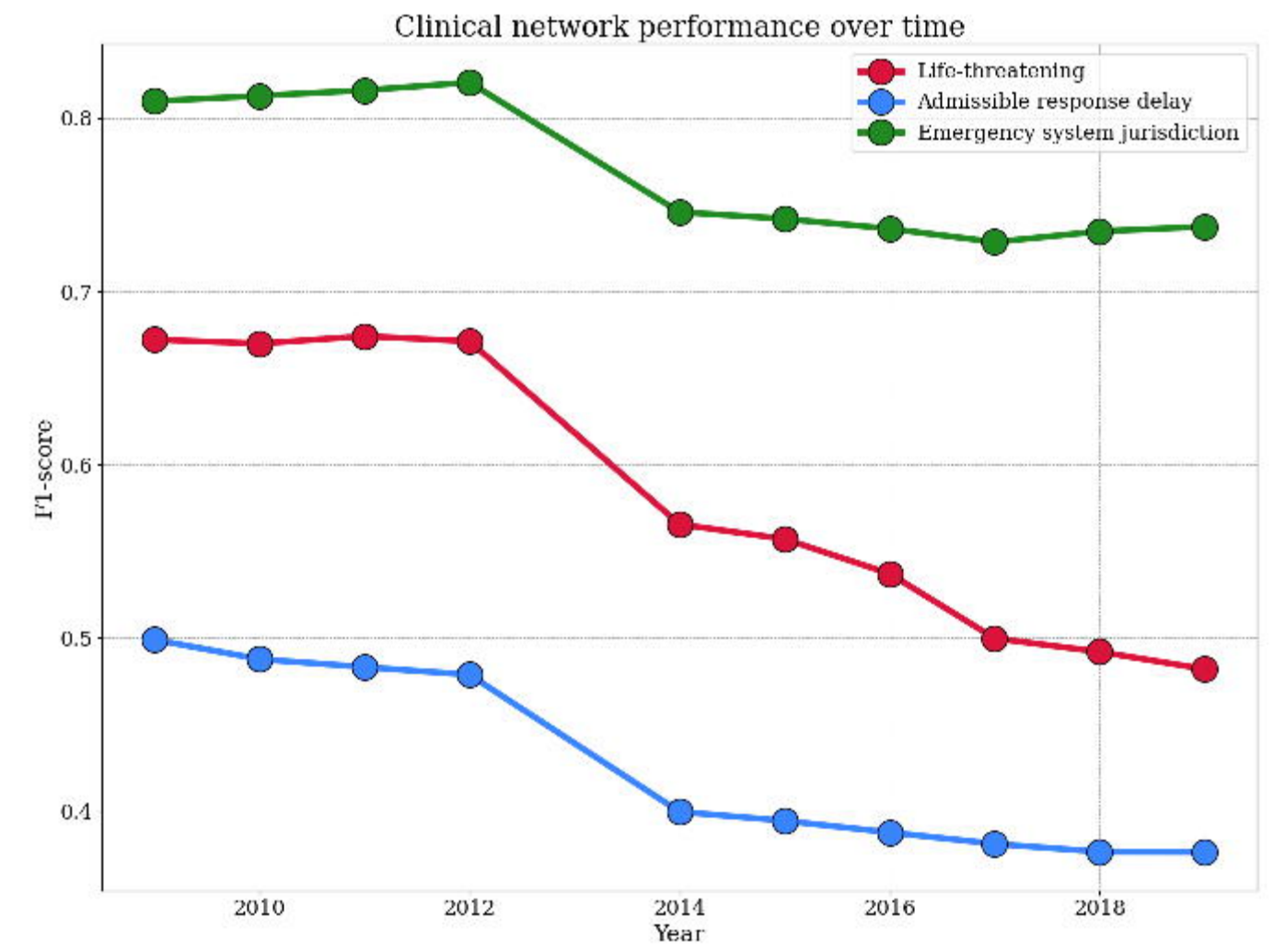
[43] Nestor, B., McDermott, M.B., Boag, W., Berner, G., Naumann, T., Hughes, M.C., Goldenberg, A., Ghassemi, M.: Feature robustness in non-stationary health records: caveats to deployable model performance in common clinical machine learning tasks. In: Machine Learning for Healthcare Conference, pp. 381–405 (2019). PMLR

[44] Guo, L.L., Steinberg, E., Fleming, S.L., Posada, J., Lemmon, J., Pfohl, S.R., Shah, N., Fries, J., Sung, L.: Ehr foundation models improve robustness in the presence of temporal distribution shift. Scientific Reports **13**(1), 3767 (2023)

[45] Lemmon, J., Guo, L.L., Posada, J., Pfohl, S.R., Fries, J., Fleming, S.L., Aftandilian, C., Shah, N., Sung, L.: 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 (2023)

[46] Lopez-Paz, D., Ranzato, M.A.: Gradient episodic memory for continual learning (2017)

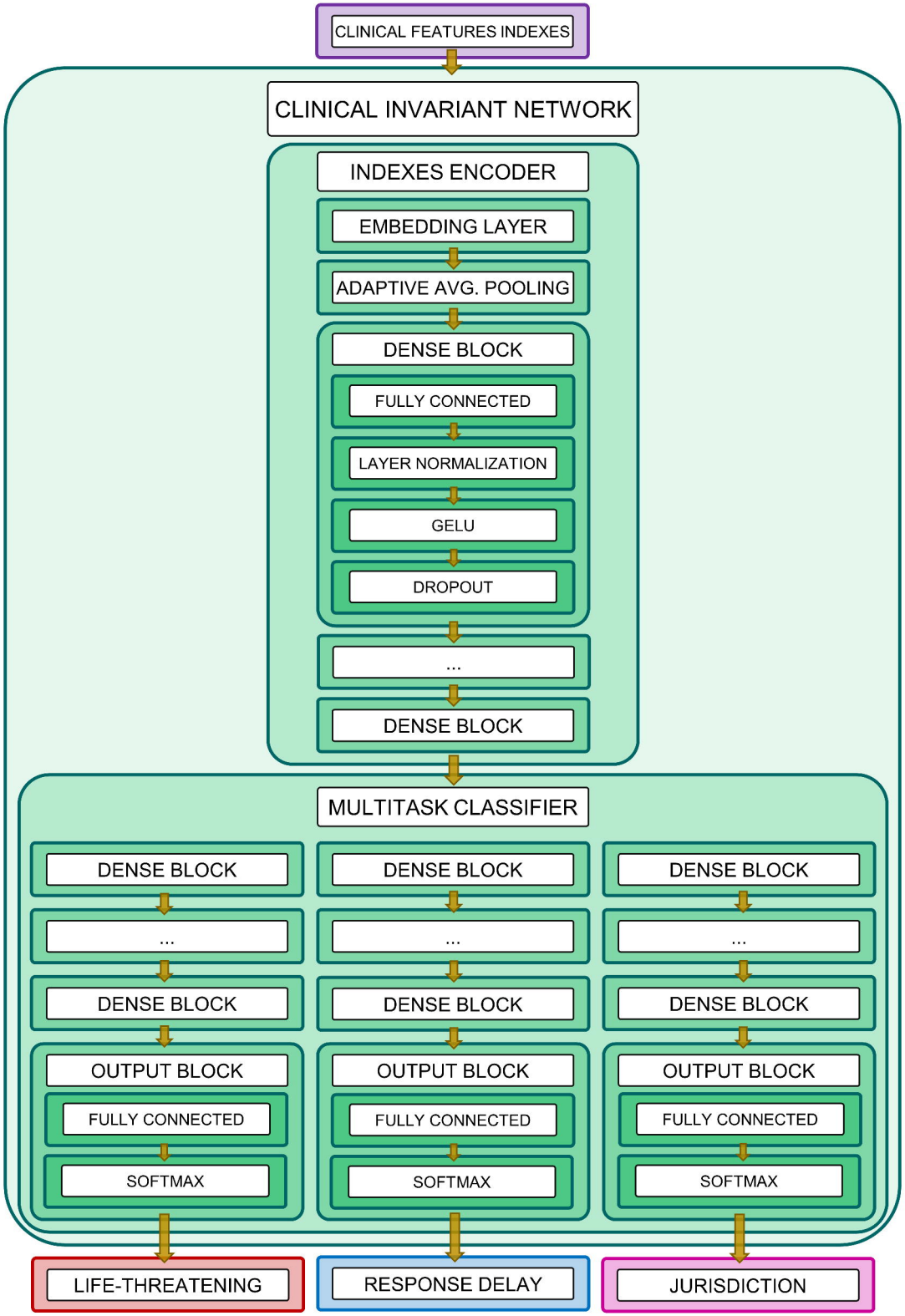
[47] Zenke, F., Poole, B., Ganguli, S.: Continual Learning Through Synaptic Intelligence. ArXiv:1703.04200 [Cs, q-Bio, Stat]. (2017). http://arxiv.org/abs/1703.04200

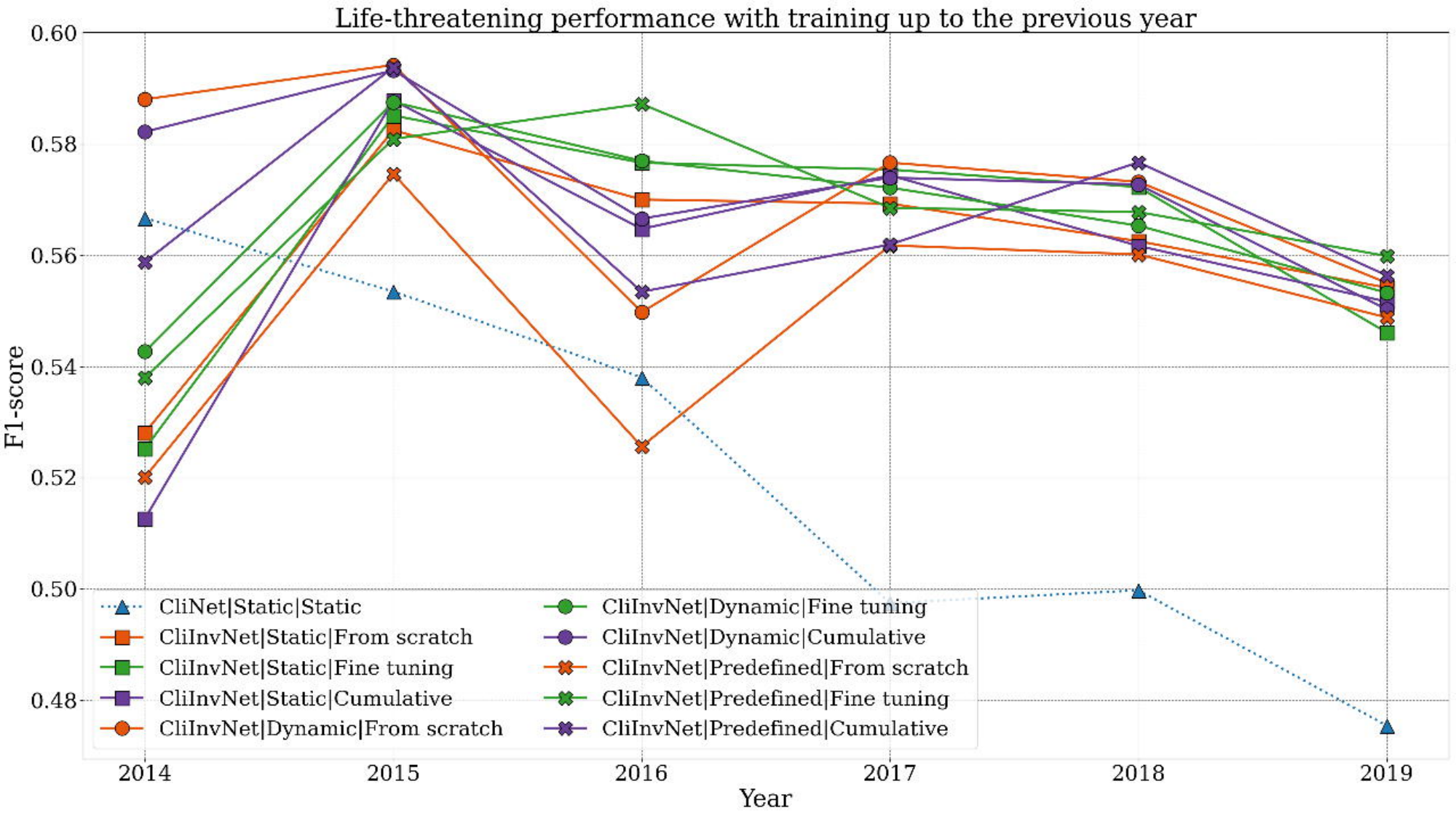

Clinical network performance over time
Life-threatening
Admissible response delay
Emergency system jurisdiction
0.8
0.7
0.6
0.5
0.4
F1-score
2010
2012
2014
2016
2018
Year

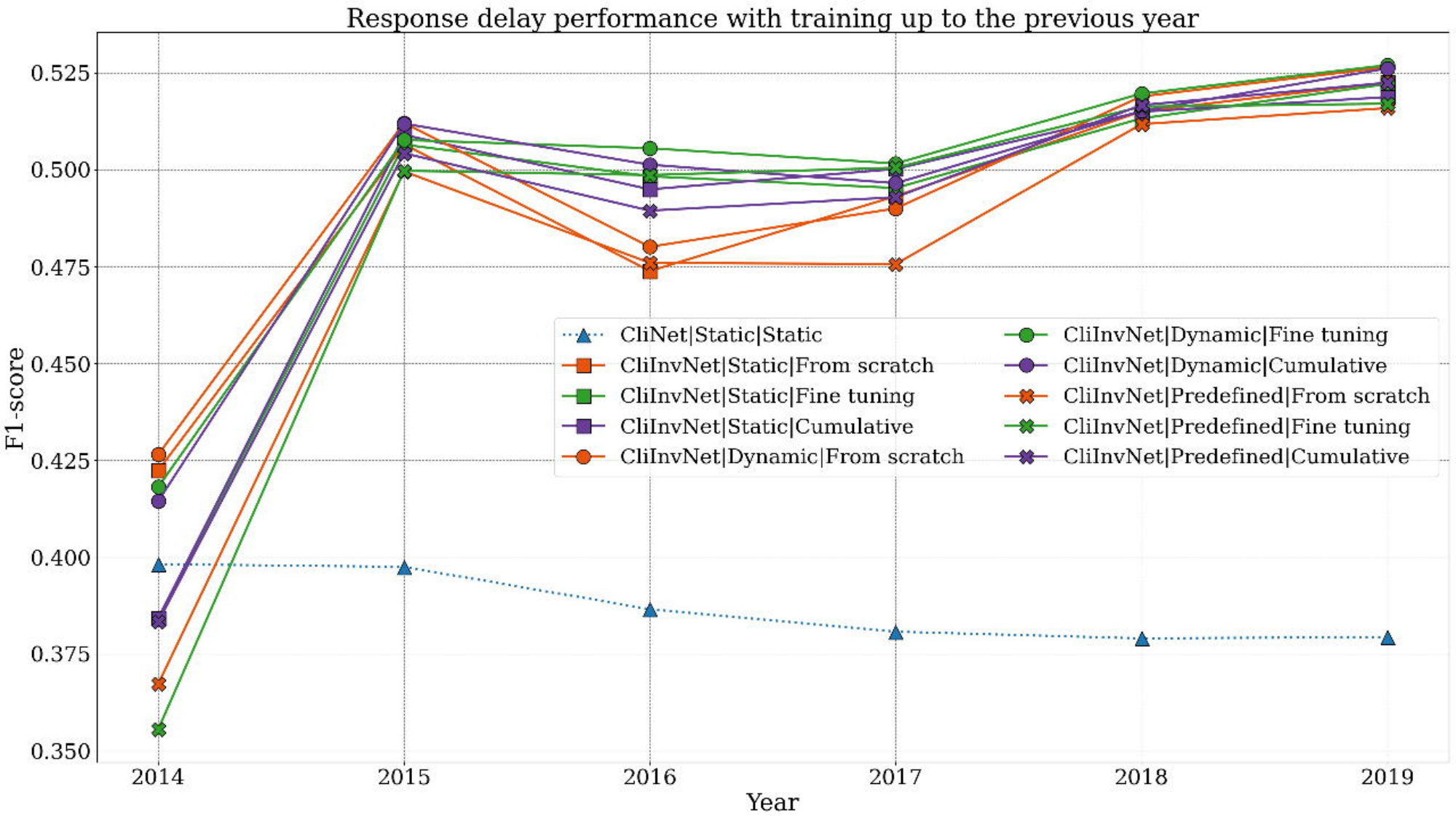
Experience 1	Experience 2	Experience 3	Experience 4	Experience 5	Experience 6	Experience 7
N=511487	N=115536	N=132649	N=139678	N=173057	N=167879	N=174289
Years 2009-2012	Year 2014	Year 2015	Year 2016	Year 2017	Year 2018	Year 2019

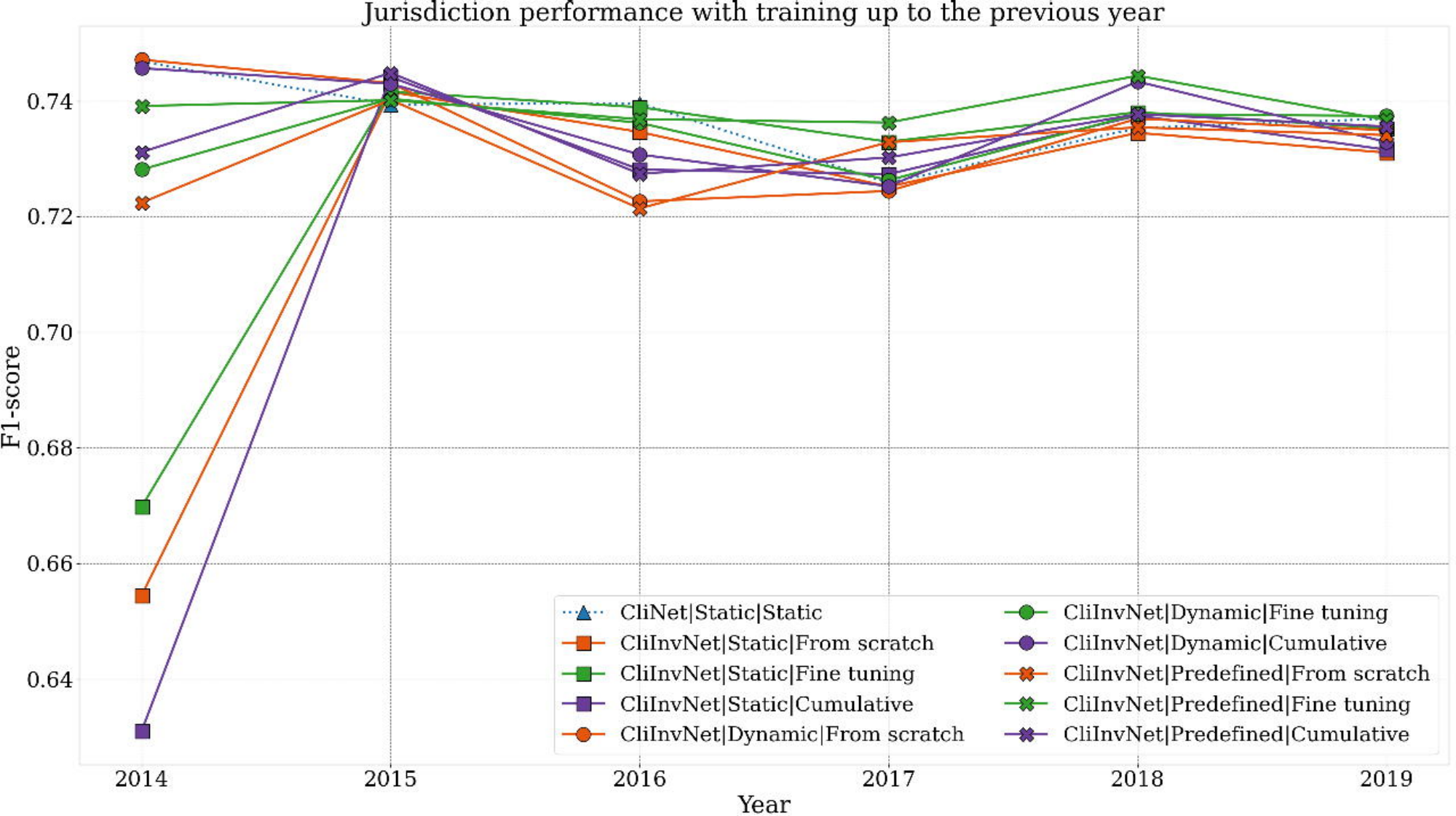
Experience 4	Experience 4
Training stream	Test stream
N=111742	N=27936
Year 2016	Year 2016

Experience 4	Experience 4
Training stream	Training stream
Pure training stream	Validation stream
N=78219	N=33523
Year 2016	Year 2016


CLINICAL FEATURES INDEXES
CLINICAL INVARIANT NETWORK
INDEXES ENCODER
EMBEDDING LAYER
ADAPTIVE AVG. POOLING
DENSE BLOCK
FULLY CONNECTED
LAYER NORMALIZATION
GELU
DROPOUT
...
DENSE BLOCK
MULTITASK CLASSIFIER
DENSE BLOCK
...
DENSE BLOCK
OUTPUT BLOCK
FULLY CONNECTED
SOFTMAX
DENSE BLOCK
...
DENSE BLOCK
OUTPUT BLOCK
FULLY CONNECTED
SOFTMAX
DENSE BLOCK
...
DENSE BLOCK
OUTPUT BLOCK
FULLY CONNECTED
SOFTMAX
LIFE-THREATENING
RESPONSE DELAY
JURISDICTION


Life-threatening performance with training up to the previous year
F1-score
0.60
0.58
0.56
0.54
0.52
0.50
0.48
2014
2015
2016
2017
2018
2019
Year
CliNet|Static|Static
CliInvNet|Static|From scratch
CliInvNet|Static|Fine tuning
CliInvNet|Static|Cumulative
CliInvNet|Dynamic|From scratch
CliInvNet|Dynamic|Fine tuning
CliInvNet|Dynamic|Cumulative
CliInvNet|Predefined|From scratch
CliInvNet|Predefined|Fine tuning
CliInvNet|Predefined|Cumulative


Response delay performance with training up to the previous year
F1-score
0.525
0.500
0.475
0.450
0.425
0.400
0.375
0.350
2014
2015
2016
2017
2018
2019
Year
CliNet|Static|Static
CliInvNet|Static|From scratch
CliInvNet|Static|Fine tuning
CliInvNet|Static|Cumulative
CliInvNet|Dynamic|From scratch
CliInvNet|Dynamic|Fine tuning
CliInvNet|Dynamic|Cumulative
CliInvNet|Predefined|From scratch
CliInvNet|Predefined|Fine tuning
CliInvNet|Predefined|Cumulative


Jurisdiction performance with training up to the previous year
F1-score
0.74
0.72
0.70
0.68
0.66
0.64
2014
2015
2016
2017
2018
2019
Year
CliNet|Static|Static
CliInvNet|Static|From scratch
CliInvNet|Static|Fine tuning
CliInvNet|Static|Cumulative
CliInvNet|Dynamic|From scratch
CliInvNet|Dynamic|Fine tuning
CliInvNet|Dynamic|Cumulative
CliInvNet|Predefined|From scratch
CliInvNet|Predefined|Fine tuning
CliInvNet|Predefined|Cumulative