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

Pablo Ferri[1], Carlos Sáez[1], Antonio Félix-De Castro[2], Javier Juan-Albarracín[1], Vicent Blanes-Selva[1], Purificación Sánchez-Cuesta[2] and 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: pabferb2@upv.es

ABSTRACT

Objective: To develop a predictive model to aid non-clinical dispatchers to classify 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.

Materials: A total of 1 244 624 independent retrospective incidents from the Valencian emergency medical dispatch service in Spain from 2009 to 2012, comprising clinical features, demographics, circumstantial factors and free text dispatcher observations.

Methods: A deep multitask ensemble model integrating four subnetworks, composed in turn by multi-layer perceptron modules, bidirectional long short-term memory units and a bidirectional encoding representations from transformers module.

Results: The model showed a micro F1 score of 0.771 in life-threatening classification, 0.592 in response delay and 0.801 in jurisdiction, obtaining a performance increase of 13.2%, 16.4% and 4.5%, respectively, with regard to the current in-house triage protocol of the Valencian emergency medical dispatch service.

Discussion: The model captures information present in emergency medical calls not considered by the existing in-house triage protocol, but relevant to carry out incident classification. Besides, the results suggest that most of this information is present in the free text dispatcher observations.

Conclusion: To our knowledge, this study presents the development of 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 wellbeing and health services sustainability.

Keywords: medical emergencies, emergency medical calls**,** emergency medical dispatch, deep learning, multitask learning, ensemble learning.

BACKGROUND AND SIGNIFICANCE

Emergency medical dispatch (EMD) involves the reception and management of requests for medical assistance in an emergency medical services system.[1] It comprises two main dimensions: call-taking, where emergency medical calls are received and incidents are classified according to their priority, i.e., triaged; and controlling, where the best available resources are dispatched to handle the event.[2]

The call-taking process is generally managed by emergency medical dispatchers.[3] 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.[1, 4] Dispatchers usually follow a clinical protocol, established in the medical dispatch center, and periodically verified by medical supervisors.[5]

However, despite preparation and the existence of triage protocols, assigning priorities to emergency medical call incidents (EMCI) is a challenging and stressful task for dispatchers, requiring constant concentration.[6-8] 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.[9] A wrong priority assignment derives either in insufficient medical attention or unnecessary resource deployment.[10-12] In consequence, EMCIs triage protocols are continuously revised and enhanced.

Many triage algorithms, such as the Emergency severity index,[13] the Manchester triage system,[14] the Canadian triage and acuity scale [15] or the Australasian triage scale,[16] have been widely studied and enriched.[17-20] However, they are difficult to benchmark, deriving in no international agreement about their use for EMD.[21] Likewise, these algorithms depend on structured clinical information which is not always available during the call.[22] As such, improvements in EMD processes by redefining this sort of protocols are extremely costly and limited.

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.

The potential capability of deep learning to enhance EMCI classification through the provision of decision support to non-clinical dispatchers, was spotted by the Health Services Department of the Valencian region, aware of the potential of these models: deep learning is at the state of the art of machine learning in tasks involving complex types of data,[23] e.g., high dimensional, unstructured, sequential, multimodal,[24-27] 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,[28-30] prediction of emergency calls

volume,[31] automatic stress detection of the caller,[32] interpretable knowledge extraction,[33] performance monitoring,[34] cardiac arrest calls assistance [35] or triaging unconscious and fainting patients.[36] Therefore, we can argue that deep learning models are a feasible and promising technology to improve EMD through EMCI classification.

In this work, 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, as mentioned in the previous paragraph, to our knowledge, this is the first large-scale study undertaking a general EMCI classification trough deep learning.

MATERIALS

Dataset

A total of 1 244 624 independent EMCI of the Health Services Department of the Valencian Community, comprising during-call and after-call data, were compiled in retrospective from 2009 to 2012. The Health Services Department board of the Valencian Community approved the data use for this project, removing before their analysis any information that may disclose the identity of the person.

During-call data (Figure 1 top) consist of demographics, circumstantial factors, clinical features collected throughout the triage tree navigation and free text dispatcher observations. From a data type perspective, we found structured, i.e., fixed fields, and unstructured, i.e., open fields, data, as well as stationary, i.e., their order is not informative, and sequential, i.e., their order is informative, data. Further details about these data are available in Supplementary material Appendix 1.

After-call data involve physician diagnoses standardized by the International classification of diseases codes,[37] maneuvers, procedures, hospitalizations and emergency department stays linked to each one of the incidents (Supplementary material Appendix 1). These data were used to derive EMCI classification labels.

The inclusion criteria in our study consisted in those EMCI which after-call data were fully available, and which during-call data were registered by non-novice dispatchers, i.e., dispatchers with more than 100 calls managed. The final working dataset size comprised 722 270 EMCI.

Labels derivation

Three different but complementary labels were defined to classify EMCI (Figure 1 bottom): life-threatening level (yes/no), admissible response delay (undelayable, minutes, hours, days) and emergency system jurisdiction (emergency system/primary care). These labels were derived from after-call data, by means of a mapping defined by a panel of 17 physicians from the Health Services Department of the Valencian Community, using a Delphi methodology.[38]

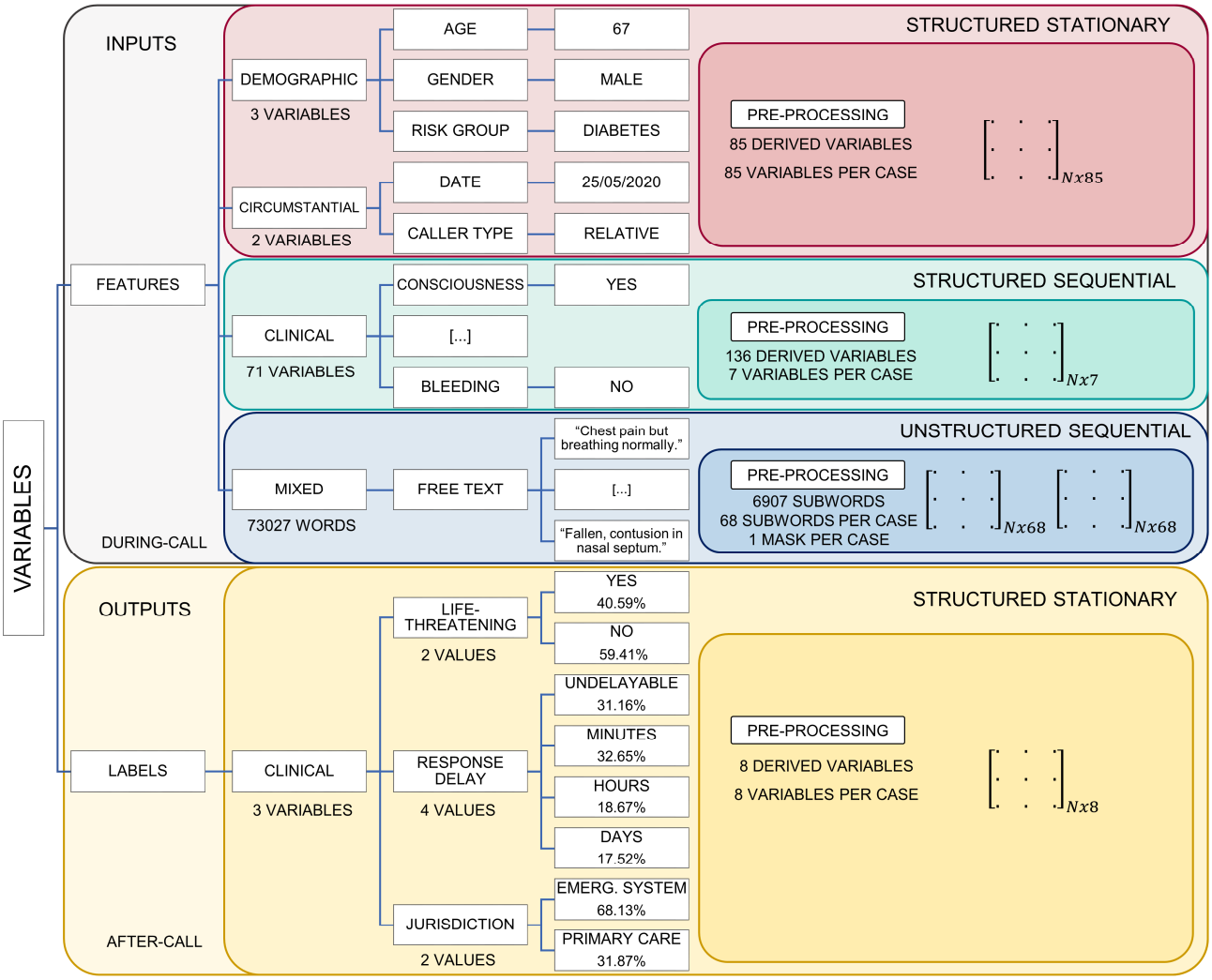

Figure 1. Dataset variables arranged by type. Names and cardinality, before and after pre-processing (derived variables), are presented, indicating how many variables (or subwords, when referring to text features) are available per case after pre-processing. Examples for their values are also included. Class frequencies for each output label are also reported. N is equal to the 722 270 EMCI used in the study.

Framework

The implementation language was Python 3.7.3,[39] making use of libraries Pandas,[40] NumPy,[41] and Fuzzywuzzy,[42] for data pre-processing and Pytorch (version 1.4.0),[43] Hugginface transformers [44] and Hyperopt [45] for modeling.

METHODS

Data pre-processing

Depending on variable type, different pre-processing techniques were applied, mapping the original data to a matrix representation (Figure 1 right, highlighted pre-processing blocks). Details about this pre-processing step can be found in Supplementary material Appendix 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 2).[46] 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*.

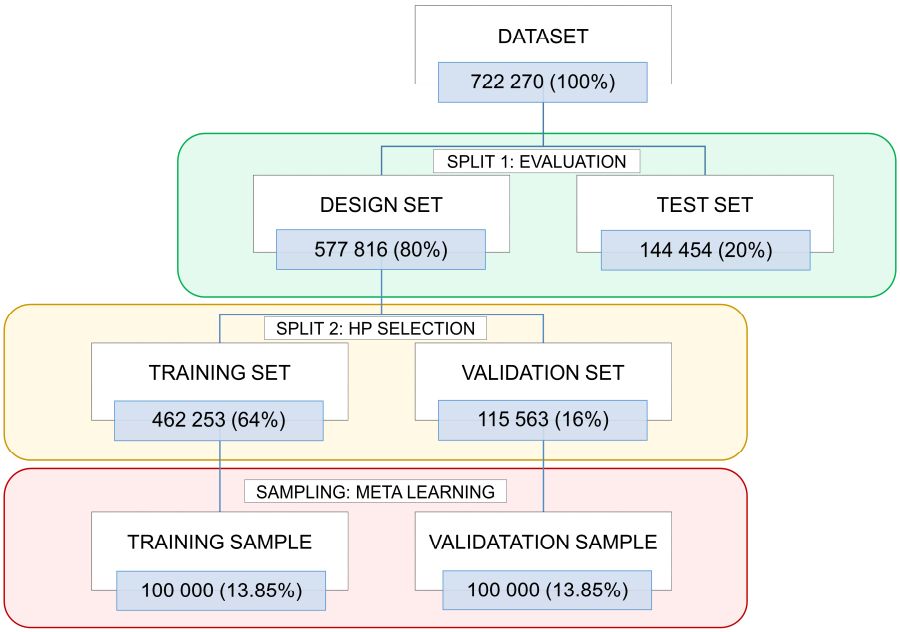

Figure 2. Data splitting and sampling. The number of data of each partition, along with its percentage respect the total number of data, are provided. Abbreviations: HP, hyperparameter.

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,[47] defining the *Modular network* (ModNet).

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 [48] 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 [49] 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).

Every subnetwork is described next, supported by Figure 3:

The *Context subnetwork* (Figure 3 left) deals with the demographics and circumstantial factors bound to an EMCI. It consists on a multi-layer perceptron (MLP),[50] due to its adequateness to model structured and stationary data, composed by dense and output blocks. A dense block integrates a fully connected layer [51] a batch normalization layer [52] to manage internal covariate shift, a leaky ReLU [53] activation function to avoid vanishing and exploding gradients, while preventing dead neurons issues,[54] and a dropout layer [55] 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 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,[56] which compresses the sparse input space into a smaller and dense one; a stack of multiple bidirectional long short-term memory (BLSTM)[57] units, which capture long-term dependences far better than standard recurrent models; multiple skip connections [58] across the BLSTM units, to reduce the risk of losing relevant information during BLSTM propagation; a concatenation block which 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 right) deals with the free text dispatcher observations written during an EMCI. It is composed by a bidirectional encoding representations from transformers (BERT)[59] 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 by an embedding block, an encoder block,[60] and a pooler block, while the MLP component is constituted by dense and output blocks.

The *Ensemble subnetwork* (Figure 3 bottom) integrates inner outputs from the ConNet, CliNet and TextNet to generate the final outputs of the *Modular network*. 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.

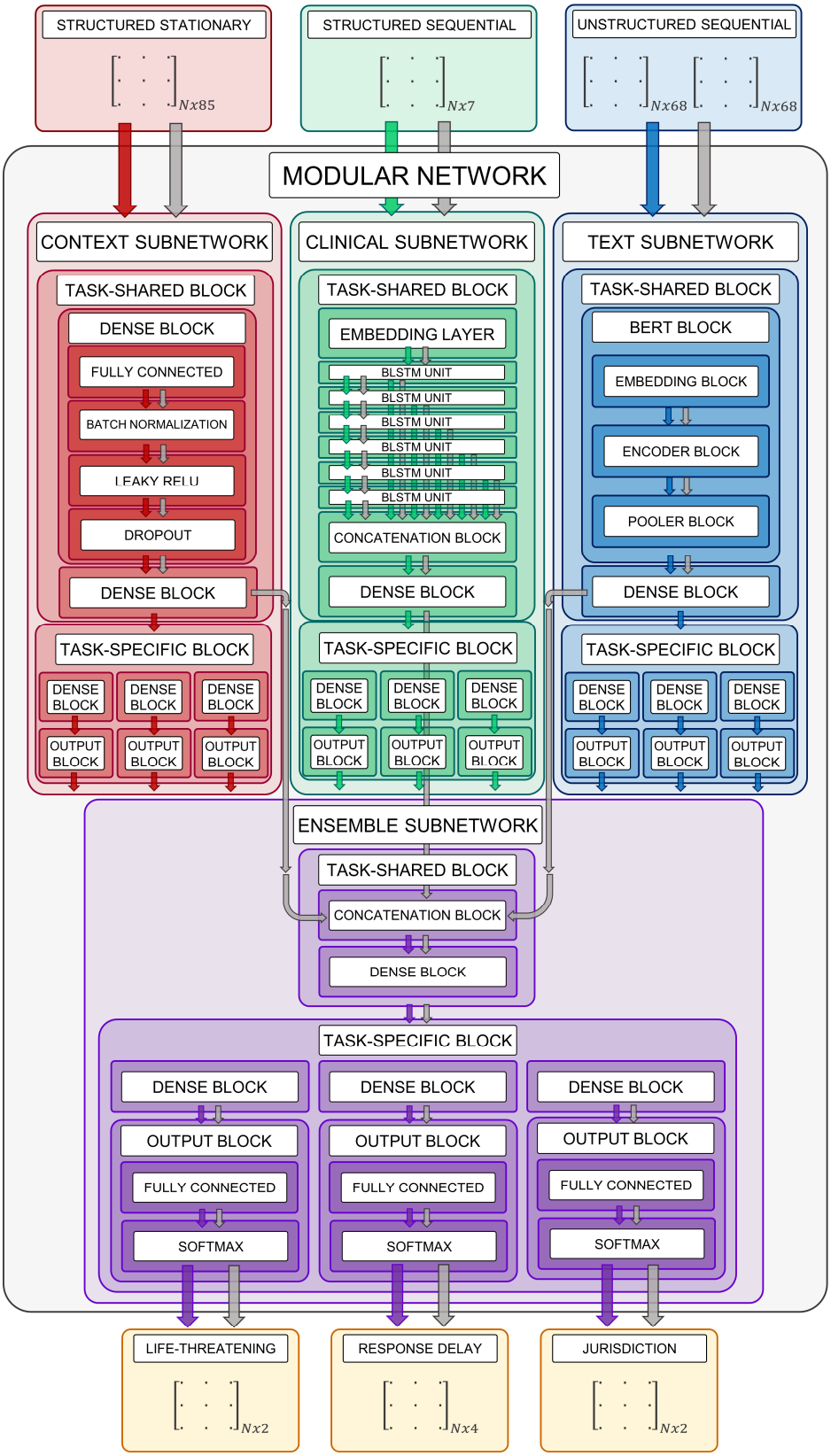

Figure 3. Deep learning model architecture (*Modular network*), including its constituting subnetworks (*Context subnetwork*, *Clinical subnetwork*, *Text subnetwork* and *Ensemble subnetwork*). Arrows indicate the forward propagation direction, for each one of the subnetworks, as well as the global network, colored according to the particular neural network they refer.

Parameter tuning

Subnetworks were trained in a constructive modularized manner,[47] so they were independently trained and assembled later as loosely coupled models. The optimizer selected for that was ADAM,[61] given its learning adaptability, noisy gradients management and learning process stability. [62-63]. A term of weight decay [64] was included in the parameters upgrading rule expression, to promote regularization. Likewise, it was followed a mini-batch upgrading approach,[65] computing gradients with backpropagation [66] and backpropagation through time.[67] The objective function was a cross-entropy [68] 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,[69] while softmax activation function layers were initialized with Xavier's initialization.[70]

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 (check Supplementary material Appendix 4 for details).

Hyperparameters were tuned following a multi-step strategy (Figure 4):

The first step involved an automatic active learning [71] hyperparameter optimization process (Figure 4 top): four surrogate models (one per subnetwork), based on tree-structured parzen estimators,[72] learned the conditional probability distribution of subnetworks hyperparameters given their associated CEL. Aiming to maximize the Expected Improvement [73] of the CEL, new hyperparameter 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.

Next, the best hyperparameter configurations proposed by the surrogate models were selected (Figure 4 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, i.e., 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 4 bottom). Once trained, its integration into a single architecture defined the global network (ModNet), evaluated later in the *test set*.

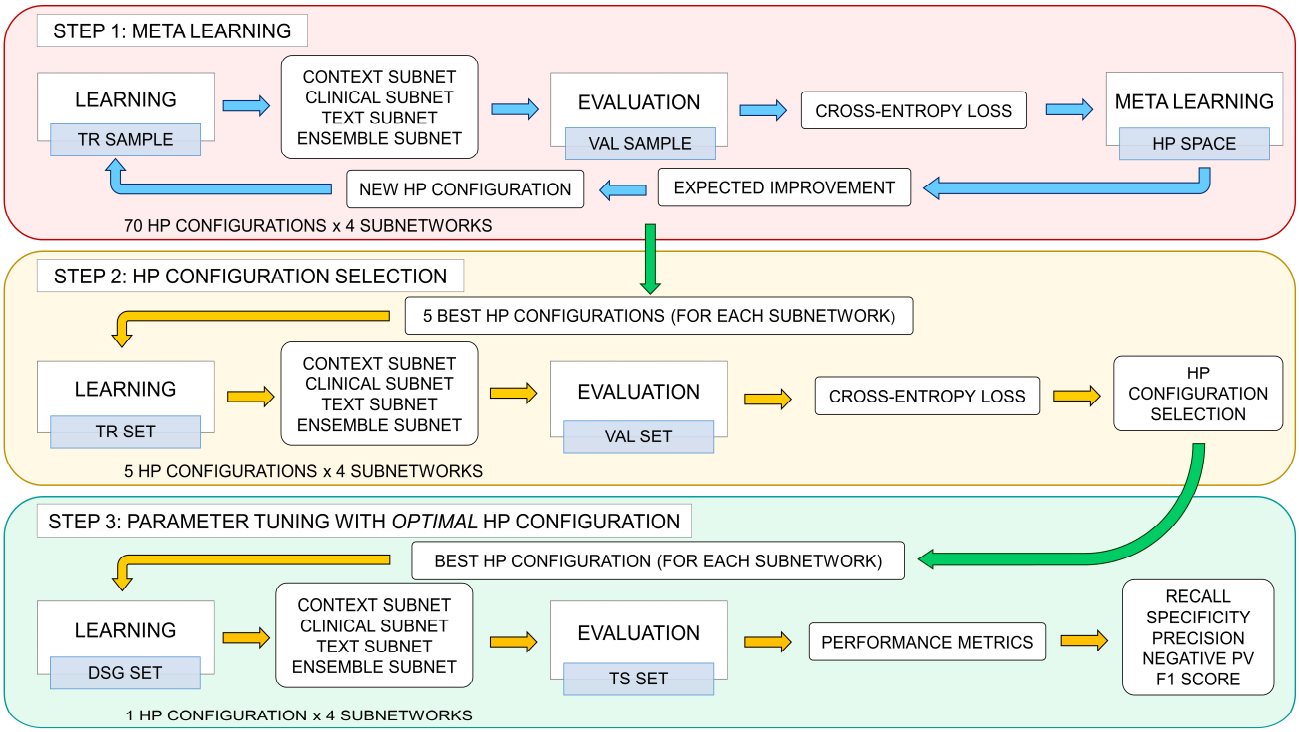

Figure 4. Multi-step hyperparameter tuning strategy. Yellow arrows imply unidirectionality, while blue arrows stand for a feedback loop, both inside a hyperparameter optimization step. Green arrows denote unidirectionality across hyperparameter optimization steps. Abbreviations: HP, hyperparameter; TR, training; VAL, validation; DSG, design TS, test.

Evaluation

The performance of the ModNet, as well as ConNet, CliNet and TextNet subnetworks (EnsNet outputs are the same as the ModNet), were evaluated in the *test set* (144 454 independent EMCI) for each label prediction task. Likewise, performance metrics were also obtained for the current triage protocol of the Valencian emergency medical dispatch service, as a comparative baseline. The evaluation metrics included recall, specificity, precision, negative predictive value (NPV) and the F1 score. For binary labels (life-threatening, jurisdiction), recall, specificity, precision and NPV were referencing the interest class, i.e., life-thread and emergency system jurisdiction. Regarding the multiclass label (response delay), recall, specificity, precision and NPV were calculated for each class and then averaged following a macro approach. Likewise, micro F1 score was computed for both the binary and multiclass labels, to dispose of an overall performance descriptor, not restricted to a single label class while taking into account the total number of true positives, true negatives, false positives and false negatives across all the classes. Finally, for all metrics, 95% confidence intervals were calculated by 1000 bootstrap samples [74] extracted from the test set.

RESULTS

Tables 1, 2 and 3 show the classification performance results for the life-threatening, response delay and jurisdiction labels. Metrics are calculated in the *test set*, for the protocol, the ConNet, the CliNet, the TextNet and the ModNet. Percentage differences (Δ) between the ModNet (final deep learning model) and the protocol are also reported.

Table 1. In-house triage protocol and deep learning models performance in life-threatening prediction (test set). Bootstrapped 95% confidence intervals are shown between brackets.

Model	Life-threatening level (yes/no)				
	Single-class metrics (yes)				Two-class metric (yes/no)
	Recall	Specificity	Precision	NPV	$F1^{micro}$
Protocol	0.644 [0.641, 0.647]	0.636 [0.633, 0.638]	0.547 [0.544, 0.551]	0.723 [0.72, 0.725]	0.639 [0.637, 0.641]
ConNet	0.44 [0.436, 0.443]	0.785 [0.782, 0.787]	0.583 [0.579, 0.587]	0.672 [0.669, 0.674]	0.644 [0.642, 0.646]
CliNet	**0.79 [0.787, 0.793]**	0.61 [0.607, 0.612]	0.581 [0.578, 0.584]	**0.809 [0.807, 0.812]**	0.683 [0.681, 0.685]
TextNet	0.638 [0.635, 0.642]	**0.844 [0.842, 0.846]**	0.737 [0.734, 0.74]	0.773 [0.771, 0.775]	0.76 [0.759, 0.762]
ModNet	0.671 [0.668, 0.675]	0.84 [0.838, 0.842]	**0.742 [0.739, 0.745]**	0.789 [0.786, 0.791]	**0.771 [0.77, 0.773]**
Δ (%)	2.7	20.4	19.5	6.6	13.2

Abbreviations: NPV, negative predictive value; Δ, ModNet difference respect the protocol.

Table 1 shows that the ModNet outperforms the current protocol in the life-threatening prediction task. It captures more true life-threatening situations (higher recall) being much more precise, i.e., with less false positives (much higher precision). Respect to non-life-threatening incidents, it detects many more true cases of this type (much higher specificity) also with less false negatives (higher NPV). Referring to the overall performance in both classes, the ModNet beats the protocol by far (13.2% of micro F1 score improvement).

Focusing on the subnetworks, the ConNet is the weakest deep learning model, although its F1 is superior to that attained by the protocol. 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.

Table 2 shows that ModNet outcomes are substantially above those achieved by the protocol in the response delay prediction task. Overall detection of situations with a specific true admissible response delay (undelayable, minutes, hours, days) is amply improved by the ModNet (15.8% increment in macro recall) while

remarkably enhancing overall precision (17.3% raise). Regarding to the overall capturing of events which do not exhibit certain true admissible response delay, the ModNet is also superior (5.8% increase in macro specificity) showing less false negatives in this task (5.5% increment in macro NPV). Concerning to the general performance in all classes, the ModNet significantly exceeds the protocol (16.4% of micro F1 score improvement).

Table 2. In-house triage protocol and deep learning models performance in response delay prediction (test set). Bootstrapped 95% confidence intervals are shown between brackets.

	Admissible response delay (undelayable, minutes, hours, days)				
Model	$Recall^{macro}$	$Specificity^{macro}$	$Precision^{macro}$	NPV^{macro}	$F1^{micro}$
Protocol	0.411 [0.409, 0.413]	0.8 [0.799, 0.801]	0.416 [0.414, 0.419]	0.805 [0.804, 0.806]	0.428 [0.426, 0.43]
ConNet	0.376 [0.374, 0.378]	0.791 [0.79, 0.792]	0.415 [0.412, 0.418]	0.793 [0.792, 0.794]	0.413 [0.411, 0.415]
CliNet	0.477 [0.475, 0.479]	0.824 [0.823, 0.825]	0.53 [0.527, 0.532]	0.829 [0.828, 0.829]	0.506 [0.504, 0.508]
TextNet	0.544 [0.542, 0.546]	0.851 [0.85, 0.851]	0.583 [0.58, 0.585]	0.854 [0.853, 0.855]	0.576 [0.574, 0.578]
ModNet	**0.569 [0.567, 0.571]**	**0.858 [0.857, 0.859]**	**0.589 [0.587, 0.591]**	**0.86 [0.859, 0.86]**	**0.592 [0.59, 0.594]**
Δ (%)	15.8	5.8	17.3	5.5	16.4

Abbreviations: NPV, negative predictive value; Δ, ModNet difference respect the protocol.

Focusing on ModNet 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 ModNet subnetwork in all metrics, with a substantial increase respect to the CliNet.

Table 3 shows that the ModNet outperforms the protocol in the jurisdiction prediction task. It captures more situations which are truly jurisdiction of the emergency system (better recall) being more precise, i.e., with less false positives (better precision). In relation with incidents which should be derived to primary care, i.e., non-emergencies, the ModNet detects more true cases of this type (higher specificity) also with less false negatives (better NPV). Respect to the overall performance in both classes, the ModNet surpasses the protocol (4.5% of micro F1 score improvement).

Regarding to ModNet subnetworks, although the ConNet presents the highest recall values, its specificity is fairly poor, 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.

Table 3. In-house triage protocol and deep learning models performance in jurisdiction prediction (test set). Bootstrapped 95% confidence intervals are shown between brackets.

Model	Emergency system jurisdiction (yes/no)				
	Single-class metrics (yes)				Two-class metric (yes/no)
	Recall	Specificity	Precision	NPV	$F1^{micro}$
Protocol	0.855 [0.854, 0.857]	0.541 [0.537, 0.545]	0.8 [0.798, 0.803]	0.635 [0.631, 0.639]	0.756 [0.754, 0.758]
ConNet	**0.945 [0.943, 0.946]**	0.288 [0.285, 0.292]	0.741 [0.739, 0.743]	0.708 [0.702, 0.713]	0.736 [0.734, 0.738]
CliNet	0.9 [0.899, 0.902]	0.521 [0.517, 0.525]	0.802 [0.8, 0.804]	0.708 [0.704, 0.712]	0.78 [0.778, 0.782]
TextNet	0.917 [0.916, 0.919]	0.519 [0.515, 0.523]	0.804 [0.802, 0.806]	**0.745 [0.741, 0.749]**	0.791 [0.789, 0.793]
ModNet	0.895 [0.894, 0.897]	**0.597 [0.593, 0.601]**	**0.827 [0.825, 0.829]**	0.726 [0.722, 0.729]	**0.801 [0.799, 0.802]**
Δ (%)	4	5.6	2.7	9.1	4.5

Abbreviations: NPV, negative predictive value; Δ, ModNet difference respect the protocol.

DISCUSSION

Relevance

The superior performance of the ModNet against the triage protocol suggests the existence of information provided during the emergency medical call not considered by the current protocol, but captured by the deep learning model. According to TextNet outcomes, far better than those attained by the ConNet and CliNet, most of this information would be present in the free text dispatcher observations. Since text fields are unbounded, they would embrace wider casuistry, allowing more precision in the EMCI description, lowering, consequently, its uncertainty.

Clinical variables stand as excellent life-threatening detection (about 80% of total cases) features, since dispatchers ask them to reduce chances of missing situations where patient's life is at risk. Likewise, 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.

The hardest classification problem is 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. Likewise, within these binary labels, the less frequent class is tougher to predict than the most frequent one.

The modular approach followed in this work, assembling four specialized subnetworks into a single global network, 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.

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.

Limitations

The main limitation of this work is the inherent uncertainty of the problem: in the studied dataset it was likely to find rather similar input combinations presenting completely different label values. In other words, different disorders presented 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, i.e., Bayes error.[75]

Future work

Next steps include the evaluation of the deep learning model with prospective cases from the Valencia region and its deployment and integration in the emergency medical dispatch center. For that, we will propose a graphical user interface to allow the interaction between the dispatcher and the model during the call. Finally, the resulting tool will be implemented in the emergency medical dispatch center of the Valencian Community.

CONCLUSIONS

A novel deep multitask ensemble model, 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 notably superior to that attained by the current in-house triage protocol of the emergency medical dispatch service of the Valencian Community, achieving an improvement of 13.2%, 16.4%, 4.5% in life-threatening, response delay and jurisdiction classification, respectively, with regard to the micro F1 score metric.

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 wellbeing and health services sustainability.

FUNDING

This work has been supported by the *Agència Valenciana de Seguretat i Resposta a les Emergències* project A180017304-1, the *Ministerio de Ciencia, Innovación y Universidades* of Spain program FPU18/06441 and the *EU Horizon 2020* project InAdvance 825750.

AUTHOR CONTRIBUTIONS

PF, CS, AFC and JMGG conceived the design of the study. AFC and PSC led the definition of medical requirements of the study, exported data from database, conducted data labeling and review. PF and CS pre-processed data. PF, JJA, VBS and CS developed the deep learning model. PF, CS, AFC and JMGG wrote the main manuscript. All the authors contributed to the review and revisions of manuscript, and have seen and approved the final version of the manuscript.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *medRxiv* online.

ACKNOWLEDGEMENTS

We thank the support of physicians and experts from the *Dirección General de Asistencia Sanitaria de la Conselleria de Sanitat Universal i Salut Pública* and from the *Dirección General de la Agencia de Seguridad y Respuesta a las Emergencias*.

CS acknowledges the support of NVIDIA GPU Grant Program.

REFERENCES

1. Clawson JJ, Dernocoeur KB. Principles of emergency medical dispatch. *Priority Press* 2003.

2. Blandford A, Wong BW. Situation awareness in emergency medical dispatch. *International journal of human-computer studies* 2004; 61(4): 421-452.

3. Stratton SJ. Triage by emergency medical dispatchers. *Prehospital and disaster medicine* 1992: 263-269.

4. Clawson JJ. Dispatch priority training: strengthening the weak link. *JEMS* 1981; 6: 32-35.

5. Palumbo L, Kubincanek J, Emerman C, *et al*. Performance of a system to determine EMS dispatch priorities. *The American journal of emergency medicine* 1996; 14: 388-390.

6. Weibel L, Gabrion I, Aussedat M, *et al*. Work-related stress in an emergency medical dispatch center. *Annals of emergency medicine* 2003; 41: 500-506.

7. Forslund K, Kihlgren A, Kihlgren M. Operators' experiences of emergency calls. *Journal of telemedicine and telecare* 2004; 10: 290-297.

8. Ek B, Edström P, Toutin A, *et al*. Reliability of a Swedish pre-hospital dispatch. *International emergency nursing* 2013: 143-149.

9. Leprohon J, Patel VL. Decision-making strategies for telephone triage in emergency medical services. *Medical Decision Making* 1995; 15: 240-253.

10. Telephone triage of cardiac emergency calls by dispatchers: a prospective study of 1386 emergency calls. Srámek M, Post W, Koster RW. *Heart* 1994; 71: 440-445.

11. Pearce AP. Emergency medical services at the crossroads. *British Association for Accident and Emergency Medicine* 2009.

12. Hjälte L, Suserud BO, Herlitz J, *et al*. Why are people without medical needs transported by ambulance? A study of indications for pre-hospital care. *European Journal of Emergency Medicine* 2007; 14: 151-156.

13. Gilboy N, Tanabe T, Travers D, *et al*. Emergency Severity Index (ESI): A triage tool for emergency department. *Rockville, MD: Agency for Healthcare Research and Quality* 2011.

14. Mackway-Jones K, Marsden J, Windle J. Emergency triage: Manchester triage group. *John Wiley & Sons 2014*.

15. Murray M, Bullard M, Grafstein E. Revisions to the Canadian emergency department triage and acuity scale implementation guidelines. *Canadian Journal of Emergency Medicine* 2004; 6(6): 421-427.

16. Forero R, Nugus P. Australasian College for Emergency Medicine (ACEM) literature review on the Australasian triage scale (ATS). *Institute of Health Innovation* 2012.

17. Christ M, Grossmann F, Winter D, *et al*. Modern triage in the emergency department. *Deutsches Ärzteblatt International* 2010; 107: 892.

18. Storm-Versloot MN, Ubbink DT, Kappelhof J, *et al*. 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* 2011; 18: 822-829.

19. Seiger N, van Veen M, Steyerberg EW, *et al*. Undertriage in the Manchester triage system: an assessment of severity and options for improvement. *Archives of disease in childhood* 2011; 96: 653-657.

20. Zachariasse JM, Seiger N, Rood PP, *et al*. Validity of the Manchester Triage System in emergency care: A prospective observational study. *PloS one* 2017; 12(2): e0170811.

21. FitzGerald G, Jelinek GA, Scott D, *et al*. Emergency department triage revisited. *Emergency Medicine Journal* 2010; 27: 86-92.

22. Farand L, Leprohon J, Kalina M. The role of protocols and professional judgement in emergency medical dispatching. *European journal of emergency medicine: official journal of the European Society for Emergency Medicine* 1995; 2: 136-148.

23. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015; 521: 436-444.

24. Hinton G, Deng L, Yu D, *et al*. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 2012; 29: 82-97.

25. Russakovsky O, Deng J, Su H, *et al*. Imagenet large scale visual recognition challenge. *International journal of computer vision* 2015; 115(3): 211-252.

26. Hirschberg J, Manning CD. Advances in natural language processing. *Science* 2015; 349: 261-266.

27. Silver D, Huang A, Maddison CJ, *et al*. Mastering the game of Go with deep neural networks and tree search. *Nature* 2016; 529(7587): 484-489.

28. Maxwell MS, Henderson SG, Topaloglu H. Ambulance redeployment: An approximate dynamic programming approach. *Winter Simulation Conference (WSC)* 2009: 1850-1860.

29. McLay LA, Mayorga ME. A model for optimally dispatching ambulances to emergency calls with classification errors in patient priorities. *IIE Transactions* 2013; 45(1): 1-24.

30. Chen AY, Lu TY. A GIS-based demand forecast using machine learning for emergency medical services. *Computing in Civil and Building Engineering* 2014: 1634-1641.

31. Channouf N, L'Ecuyer P, Ingolfsson A, *et al*. The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health care management science* 2007; 10: 25-45.

32. Lefter I, Rothkrantz LJ, Van Leeuwen DA, *et al*. Automatic stress detection in emergency (telephone) calls. *International Journal of Intelligent Defence Support Systems* 2011; 4: 148-168.

33. Barrientos F, Sainz G. Interpretable knowledge extraction from emergency call data based on fuzzy unsupervised decision tree. *Knowledge-based systems* 2012; 25: 77-87.

34. Klement, P, Snásel V. Using SOM in the performance monitoring of the emergency. *Simulation Modelling Practice and Theory* 2011; 19: 98-109.

35. Blomberg SN, Folke F, Ersbøll AK, *et al*. Machine learning as a supportive tool to recognize cardiac arrest in emergency calls. *Resuscitation* 2019; 138: 322-329.

36. Tollinton L, Metcalf AM, Velupillai S. 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* 2020; 104179.

37. International classification of diseases, ICD-9. *World health organization* 2015.

38. Dalkey NC. The Delphi method: an experimental study of group opinion. *RAND CORP SANTA MONICA CALIF* 1969.

39. Python Language Reference, version 3.7.3. *Python Software Foundation* 2019. https://www.python.org.

40. McKinney W. Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference* 2010; 445: 51-56.

41. Walt SVD, Colbert SC, Varoquaux G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering* 2011; 13: 22-30.

42. Gonzalez J, Rodrigues P, Cohen A. Fuzzywuzzy: Fuzzy string matching in python 2017.

43. Paszke A, Gross S, Chintala S, *et al*. Automatic differentiation in pytorch. *31st Conference on Neural Information Processing Systems (NIPS)* 2017.

44. Wolf T, Debut L, Sanh V. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *Arxiv* 2019: arXiv-1910.

45. Bergstra J, Komer B, Eliasmith C, *et al*. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery* 2015; 8(1): 014008.

46. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai* 1995; 14(2): 1137-1145.

47. Chen K. Deep and modular neural networks. *Springer Handbook of Computational Intelligence* 2015: 473-494.

48. Caruana R. Multitask learning. *Machine learning* 1997; 28(1): 41-75.

49. Ruder S. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv preprint* 2017: arXiv:1706.05098.

50. Rosenblatt F. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. *Cornell Aeronautical Lab Inc Buffalo NY* 1961: VG-1196-G-8.

51. Goodfellow I, Bengio Y, Courville A. Deep learning. *MIT press* 2016.

52. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint* 2015: arXiv:1502.03167.

53. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. *Proc. Icml* 2013; 30(1): 3.

54. Nwankpa C, Ijomah W, Gachagan A, *et al*. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint* 2018: arXiv:1811.03378.

55. Hinton GE, Srivastava N, Krizhevsky A, *et al*. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint* 2012: arXiv:1207.0580.

56. Bengio Y, Ducharme R, Vincent P. A neural probabilistic language model. *Journal of machine learning research* 2003; 3: 1137-1155.

57. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 1997; 45(11): 2673-2681.

58. He K, Zhang X, Ren S. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition* 2016: 770-778.

59. Devlin J, Chang MW, Lee K, *et al*. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* 2018: arXiv:1810.04805.

60. Vaswani A, Shazeer N, Parmar N. Attention is all your need. *Advances in neural information processing systems* 2017: 5998-6008.

61. Kingma DP, Ba J. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)* 2015.

62. Ruder S. An overview of gradient descent optimization algorithms. *arXiv preprint* 2016: arXiv:1609.04747.

63. Sun S, Cao Z, Zhu H, et al. A Survey of Optimization Methods from a Machine Learning Perspective. *IEEE transactions on cybernetics* 2019.

64. Krogh A, Hertz JA. A simple weigth decay can improve generalization. *Advances in neural information processing systems* 1992: 950-957.

65. Bertsekas DP. Incremental least squares methods and the extended Kalman filter. *SIAM Journal on Optimization* 1996; 6(3): 807-822.

66. Hecht-Nielsen R. Theory of the backpropagation neural network. *Neural networks for perception*. Academic Press 1992: 65-93.

67. Werbos PJ. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 1990: 1550-1560.

68. Janocha K, Czarnecki WM. On Loss Functions for Deep Neural Networks in Classification. *arXiv preprint* 2017*: arXiv:1702.05659*.

69. He K, Zhang X, Ren S. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision* 2015: 1026-1034.

70. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics* 2010: 249-256.

71. Settles B. Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences* 2009.

72. Bergstra JS, Bardenet R, Bengio Y, *et al*. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems* 2011: 2546-2554.

73. Jones DR. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization* 2001; 21(4): 345-383.

74. Efron B, Tibshirani RJ. An introduction to the bootstrap. *CRC press* 1994.

75. Fukunaga K. Introduction to statistical pattern recognition. *Elsevier* 2013.