



An AI-based Mixture-of-Experts Framework for Multi-Type Iatrogenic Drug Interaction Prediction

Kawther Makhoul¹, Karim Bouamrane² and Djamila Hamdadou³

ABSTRACT

Pharmacovigilance involves preventing adverse effects in patients through the prevention, detection, and evaluation of these effects. Prescription errors, incorrect dosages, and drug interactions cause iatrogenic drug effects. There are two main types of drug interactions: between two drugs (DDI) and between a protein and a drug (DTI). The multitude of molecules and treatments available today makes it difficult to prevent their effects. Deep learning (DL) appears to be a solution to this problem. Current work in DL focuses on a single type of interaction and does not generalize to other types. We propose a multitask model that predicts both types of interactions (DDI and DTI). It deploys two mixture-of-experts (MoE) blocks for each task. This mechanism avoids interference between representation spaces. One module for drugs uses convolutional neural networks (CNNs). A second module for proteins learns multi-granularity patterns. The model test yields the following results for the DTI task: accuracy (0.952), precision (0.959), F1 (0.952), recall (0.944), AUC (0.987), and AUPR (0.987). The DDI task yields accuracy (0.980), precision (0.966), F1 (0.980), recall (0.995), AUC (0.997), and AUPR (0.995).

Article information:

Keywords: Pharmacovigilance, Iatrogenic Drug, Drug-drug Interaction, Drug-target Interaction, Deep Learning

Article history:

Received: November 22, 2025

Revised: March 26, 2026

Accepted: June 12, 2026

Published: July 4, 2026

(Online)

DOI: 10.37936/ecti-cit.2026203.264933

1. INTRODUCTION

Pharmacovigilance is an essential pillar of public health, ensuring the detection, evaluation, and prevention of adverse drug effects [1], [2]. Its major challenge is preventing harmful, unpredictable reactions to guarantee patient safety. These adverse effects constitute what are called “iatrogenic drugs”, in other words, the set of disorders or diseases caused by taking one or more drugs. Iatrogenic effects can result from a prescription error, poor compliance with treatment, or, even more so, drug interactions [3]. Among the main causes of iatrogenic effects are drug interactions, which can be of different types: either caused by two drugs (DDI) or between a drug and a protein (DTI), etc. [3]. The study focuses on these two interaction types (DDI and DTI).

Indeed, with the rapid proliferation of molecules on the market and the growth of combination treatments, the automated prediction of these interactions is becoming a major scientific and medical challenge.

The rise of deep learning (DL) has revolutionized research on DDI and DTI prediction. Several families of neural architectures have been explored for predicting these interactions. For example, in the field of convolutional neural networks (CNNs) [4], CNNs have been shown to detect local patterns in sequences effectively but struggle with long-range dependencies. In works such as [5] and [6], the authors have adopted these architectures. Another widely used type is called graph neural networks (GNN) [7]. They incorporate the molecular structure of chemical compounds as atomic graphs. Studies such as [8], [9], and [10] have used it to model atomic interactions. However, these depend on the quality of the molecular graph and may suffer from overparameterization on complex graphs. We also find sequential architectures such as LSTM (Long Short-Term Memory) [4] and BiLSTM, which help overcome difficulties with long-term dependencies. Recently, transformers have used attention to capture global relationships in sequences without recursion. Works such as [11], [12], and [13] have ex-

^{1,2,3}The authors are with the Laboratory of Informatics of Oran (LIO), Computer Science Department, Faculty of Exact and Applied Sciences, University of Oran 1, Ahmed Ben Bella, El M'Naouer, PB 1524, Oran, Algeria, Email: makhoul.kawther@edu.univ-oran1.dz, bouamrane.karim@univ-oran1.dz and hamdadou.djamila@univ-oran1.dz

¹Corresponding author: makhoul.kawther@edu.univ-oran1.dz and kawther.mk31@gmail.com

exploited these approaches to learn contextual representations from sequential data. However, transformer cost estimation is complex, and sequential models struggle to generalize to nonlinear structures such as molecular graphs.

Despite these advances, most existing approaches focus on a single task, either DDI or DTI prediction. Which leads us to the following question: “Can we overcome this distinction between tasks, which often leads to the development of domain-specific models that are difficult to reuse in other contexts?”

To overcome these limitations, we propose a parallel multitask learning (MTL) model based on a mixture-of-experts (MoE) mechanism. Unlike approaches that focus representations on each type of interaction, our model maintains a clear functional separation between DDI and DTI through two independent MoE blocks, each dedicated to its task. This design favors training stability without introducing interference between the representation spaces of DDI and DTI.

To summarize our work, we proceed as follows:

- We design a drug-encoding module based on a CNN applied to SMILES-derived representations, enabling the extraction of meaningful structural motifs shared across both DDI and DTI tasks.
- We propose a protein module based on three parallel convolutional layers, allowing the model to capture biochemical patterns at different levels of granularity, specifically for the DTI task.
- We introduce an enriched fusion mechanism that combines drug and protein representations using multiple vector operations (concatenation, element-wise product, and absolute difference), enabling the model to capture both strong correlations and structural divergences between entities.
- We propose and compare several model variants, including a dedicated single-task model (DDI or DTI), a multitask model based on a fully connected (FC) block, and, finally, the core of our study: a multitask model based on the MoE mechanism.

The present paper is structured as follows: Section 1 serves as an introduction; Section 2 reviews the literature; Section 3 formulates the problem; Section 4 describes the methodology employed; Section 5 presents and discusses the results in depth; and Section 6 concludes and proposes future work.

2. RELATED WORKS

Below, we review works on DTI and DDI prediction, two fundamental tasks in pharmacological research.

For the DTI task, several studies rely on convolutional networks. Study [5] used CNN to learn drug-protein feature matrices. Attention models allow non-covalent interactions to be modeled by as-

signing vectors to atoms and amino acids. However, SMILES interpretability and binding site localization remain limited. Continuing in this vein, [14] prolongs the idea by adding self-attention to convolution. Their goal is to compensate for CNN weaknesses by improving the detection of relevant sequence regions. However, combining architectures complicates learning and local interaction modeling.

Given limitations, several works use GNN to represent molecular structures. [10] proposed a model combining GAT for structural features with LSTM for modeling protein sequences. However, unknown target pairs remain unexplored, reducing performance. [15] developed a GNN-based model to encode molecules as graphs. CNN extracts multi-level representations from sequence and contact maps. However, the model does not consider protein pocket sites, which limits its interpretability. Thus, although GNNs address gaps in CNNs, they struggle to integrate the modalities involved.

To overcome these limitations, several papers adopt hybrid architectures. Study [8] established a cross-attention to detect different interaction features. They also used a GNN-focused molecule aggregator to model complex dependencies in drug molecular graphs and encode sequences. The limitation of this work is its inability to leverage protein structural information. [16] pursues this evolution using a GCN to capture drug graphs and a BiLSTM to process protein sequences. To better integrate drug-target interactions, the authors use a cross-attention mechanism. Residual blocks are also employed to mitigate the vanishing gradient problem. This model shows good results but suffers from scalability issues. The authors suggest generalizing the model to other domains (disease-gene, protein-protein, and drug-drug interactions). In this context, transformers have enabled models that capture richer representations, whose main contributions we present. [11] combined local and global from molecules and protein sequences. Neural fingerprints extract global drug features. Self-attention captures long-range protein dependencies. Bilinear attention learns local interaction sites, improving performance and interpretability. Model robustness to other biological interactions (protein-protein and disease-gene) can be assessed. [12] introduced a model transferring protein and drug representations via pre-trained models. It leverages GNN for drug graphs and transformers for SMILES. A multi-view strategy enables complementary feature extraction. However, its complexity may reduce training efficiency and neglect substructure positional information. Similarly, [17] suggested a transformed-based model leveraging sub-pocket analysis. It relies on pre-trained GCN models to improve generalization and interaction modeling. However, it lacks clinical evaluation. Finally, [18] developed a multimodal fusion that combines a transformer,

StarNet, and multiple levels of protein representation. A heterogeneous attention block captures binding sites, improving drug–target integration. Limitations include random protein embeddings, the memory cost of sequence-to-2D mapping, and transformer overhead. Study [19] proposed a multimodal model that uses a CNN for sequential features, a GNN for 3D molecular and protein structures, cross-attention for fusion, contrastive learning for representation discrimination, and a MixHop aggregator for neighborhood information in the DTI network. However, it is limited by poor exploitation of 3D features and redundancy or noise in multimodal fusion.

After examining DTI models, we will turn to DDI approaches with a similar progression. For the convolutional network-based method, we refer to [6]. They established a CNN-based model to determine whether two drugs interact and, if so, the type of interaction. For this, they use Jaccard similarity to quantify drug similarity from feature vectors. They plan to improve and reduce computation time. In [20], the authors introduced a model that incorporates deep convolutional learning to derive categorical features for drug pairs. Random forests are used for binary classification. They plan to expand the model to other areas, such as interactions with targets and proteins.

As with DTI, we find studies centered on GNN. For instance, [9] proposed a model to capture atomic-level interactions and improve precision. The first step is to extract atomic features for each drug using a GNN, then construct an interaction map between atom pairs of the two drugs. Finally, the interaction map and molecular representations are merged to predict the presence and specific type of interaction. However, the model depends solely on the molecular structure. Moreover, [21] proposed a model that incorporates molecular bond information (covalent and non-covalent). Then, they use a GCN to detect these representations in its topology. However, this model does not capture higher-order interactions. In a different direction, some approaches are inspired by natural language processing. [13] presented a model combining the ChemBerta tokenizer (from Transformer) for molecular representation and a BiLSTM for bidirectional contextual dependencies. To improve performance, they also introduced a dual mode. However, this model remains limited by sequential SMILES dependencies and struggles to extract local structural motifs essential for characterizing molecular substructures in interactions. To enrich representation, some recent work combines complementary views. [22] combined two views in their model. The first builds a similarity-based drug network to capture relational contexts. The second extracts molecular substructures to enrich representations. Both views are fused via an attention mechanism. This work is limited by poor generalization in cold-start scenarios, as its mechanisms reduce the

transferability of representations to new molecules. During the same period, [23] established a model that combines two views: molecular structure and interaction graph topology. A GCN extracts topological features, and a contrastive learning module aligns the two representation views. However, the model remains limited by its heavy dependence on the quality and completeness of the interaction graph. Finally, [24] introduced a multi-view encoding approach exploiting SMILES sequences, molecular graphs, and atomic spatial information. These views are fused via an attention module, enabling richer drug representations. However, the model is limited by the quality and completeness of the different data.

Unlike previous works, we propose a multitask model that employs a mixture-of-experts (MoE) mechanism. Existing models focus on a single task, limiting generalization to other biological interactions (as highlighted by [11] or [20]). Our model learns DTI and DDI representations in parallel. This strategy promotes model improvement and robustness. Moreover, several architectures, such as those proposed in [12] and [14], combine various modules (CNNs, GNNs, and Transformers) but exhibit limited learning efficiency.

In contrast, our use of MoE allows us to adapt expert subnetworks based on inputs and improve their flexibility. Regarding multimodal fusion, recent models such as [18] or [19] struggle with noise and repetition when combining multiple views. Our progressive fusion enables a stable representation while capturing both local and global dependencies. Therefore, our model provides a modular solution that addresses these limitations and paves the way for improved inter-domain generalization of biomolecular interactions.

3. PROBLEM FORMULATION

We consider two closely related supervised learning tasks: DDI and DTI prediction. Let: $D = \{d_1, d_2, \dots, d_N\}$ be the set of drugs. Let: $P = \{p_1, p_2, \dots, p_N\}$ be the set of proteins. In the DDI task, our work will consist of predicting the probability that two drugs interact, using the function $f_{DDI} : D \times D \rightarrow [0, 1]$. In the DTI task, the work will also predict the probability of a drug-protein interaction using the following function: $f_{DTI} : D \times P \rightarrow [0, 1]$. Where: 0: no interaction; 1: existence of interaction. These functions are learned in parallel using a multitask approach, in which the model shares some parameters while retaining task-specific submodules. This approach exploits complementary molecular representations and interaction-specific features, promoting better task-specific generalization.

4. METHODOLOGY

In this section, we will highlight the different models and modules developed in our study.

4.1 Overview

We offer different types of models and group them by task type (see Fig. 1).

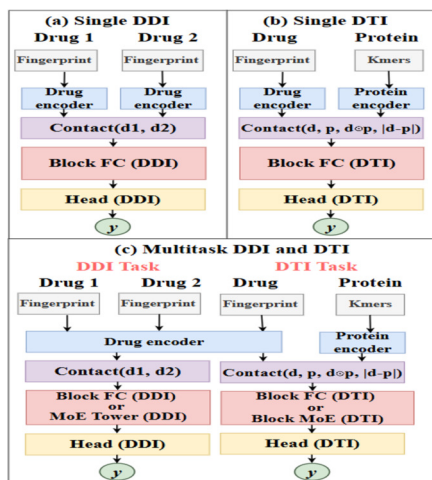


Fig.1: Single and multitask models.

4.1.1 Single task models

One model is dedicated to DDI prediction and another to DTI prediction. These models use FC blocks for fusion and prediction.

4.1.2 Multitask models

These architectures jointly predict DDI and DTI. Two variants have been developed: (i) a version using an FC block partially shared between tasks, (ii) a version integrating an MoE block per task.

The multitask model is our main approach (see Fig. 2). It adopts a hybrid architecture: (i) drug and protein modules. (ii) task-specific fusion blocks (FC or MoE) dedicated to each task, (iii) independent binary classification heads. This design aims to capture

the structural complexity of chemical compounds and proteins while facilitating task-specific processing.

Below, we describe each model component.

4.2 Module dedicated to drugs

Firstly, drugs are all represented by their chemical structures, which are often called SMILES [25]. Each structure is converted into a Morgan fingerprint using RDKit (a Python library) of dimension $d_f = 2048$: $x_d = fMorgan(SMILES) \in \{0,1\}^{d_f}$. Indeed, each bit of x_d gives us the absence or presence of a particular chemical motif identified by a topological radius set to 2. This representation encodes the molecular topology by expanding the circle around each atom. Then, a first 1D-CNN is applied to learn the local atomic patterns of the molecular structure. Its formula is as follows: $h_{d1} = Conv1D(x_d, W_1, b_1)$. With: $W_1 \in R^{32 \times 1 \times 9}$, given that a kernel size of 9 was chosen to capture the medium-sized atomic substructures commonly found in SMILES-derived fingerprints. Smaller kernels were unable to encode meaningful chemical patterns, whereas larger kernels increased model complexity without yielding significant performance gains. The number of filters is 32 to learn independent chemical patterns. Each filter acts as a substructure detector. Then we add a second 1D-CNN layer that combines the patterns to capture longer-range dependencies, as follows: $h_{d2} = Conv1D(h_{d1}, W_2, b_2)$. With: $W_2 \in R^{64 \times 32 \times 9}$. Beyond two layers, we observe overfitting. After the two 1D-CNNs, we normalize the features using the function $h_d = BatchNorm(h_{d2})$ to stabilize the feature distribution [26]. We introduce an activation function $a_d = ReLU(h_d)$, which allows us to keep only the positively activated patterns. Each convolutional layer precedes the max pooling operation to reduce the dimension while retaining the maximum of 4 consecutive positions (the most representative motifs in each segment of the molecule):

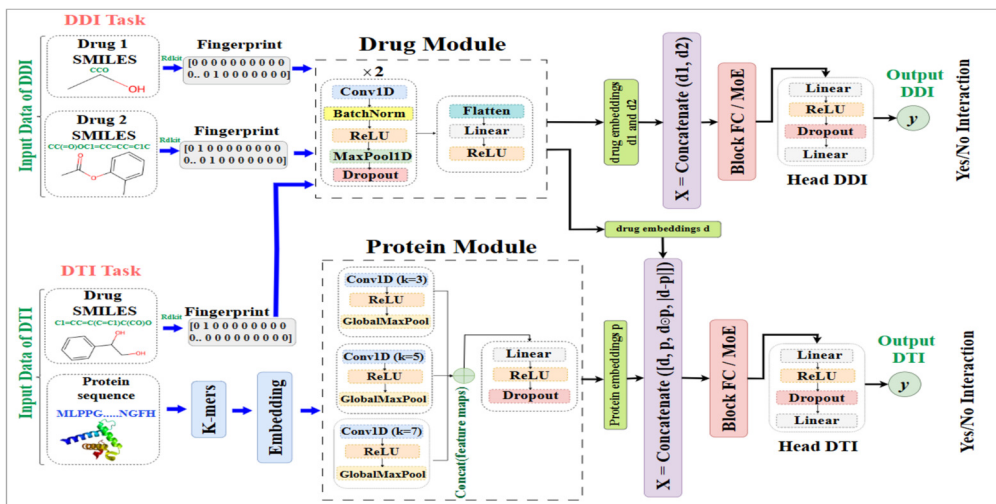


Fig.2: General architecture.

$p_d = \text{MaxPool1D}(a_d, \text{pool_size} = 4)$.

Additionally, a dropout is applied to randomly tunable neurons [27], preventing a specific chemical pattern from dominating the representation. Its formula is as follows: $p'_d = \text{Dropout}(p_d)$. Finally, the output is flattened and projected into a dense latent space: $z_d = \text{ReLU}(W_d \cdot \text{Flatten}(p'_d)) + b_d$, with: $W_d \in R^{64 \times 8192}$.

The embedding vector $z_d \in R^{64}$ captures global chemical regularities. Increasing the integration dimension generally improves the model's ability to capture structural diversity. However, our study suggests that higher-dimensional integrations tend to lead to overfitting, which results in lower generalization performance. For this reason, a 64-dimensional space was chosen as a balanced option, offering sufficient expressive power while ensuring stable learning. Moreover, this module is common between DDI and DTI drugs, whether for single or multitasking.

4.3 Module dedicated to proteins

We proceeded to segment the sequence of each protein into size 3 k-mers [28]: $S_p = \{s_i = a_i a_{i+1} a_{i+2} | 1 \leq i \leq L - k + 1\}$. A vocabulary of k-mers is constructed, then each trigram is vectorized via an embedding layer: $E_p = \text{Emb}(S_p) \in R^{L' \times 64}$. Where: L' means the length of the k-mer sequence, and 64 is the dimension of the embedding space. This representation encodes the local biochemical properties of amino acids. The architecture of the protein module is composed of the following: First, three parallel convolutional layers are used to extract patterns at different scales: $C_k = \text{ReLU}(\text{Conv1D}_k(E_p, W_k, b_2))$. Where: k is the size of the convolution kernels, $k = \{3, 5, 7\}$ and $W_k \in R^{32 \times 64 \times k}$, 32 represents filters and 64 the input dimension. The use of multiple kernel sizes captures protein sequence patterns at different biological scales. Each convolution learns protein patterns at different scales: (i) For $k = 3$: very local patterns, (ii) For $k = 5$: short structural motifs, (iii) For $k = 7$: longer patterns. The feature maps are aggregated by global max pooling: $m_k = \text{GlobalMaxPool}(C_k)$, then concatenated: $h_p = [m_3, m_5, m_7] \in R^{96}$. Finally, the concatenated vector is projected into a common latent space with drugs: $z_p = \text{Dropout}(\text{ReLU}(W_p h_p + b_p))$, with: $W_p \in R^{64 \times 96}$. The vector $z_p \in R^{64}$ constitutes the final protein embedding used by DDI and DTI modules. Thus, molecular and protein representations share the same dimensionality and can be combined directly.

4.4 Multitask module

We developed two multitask variants for DDI and DTI prediction to evaluate their impact. These variants are:

4.4.1 Block FC

The FC block consists of a linear layer succeeded by batch normalization [26], ReLU activation [29], and a dropout [27]. The DDI block takes as input a concatenation of two drug representations, while the DTI block processes drug-protein combinations and element-wise interactions. This block integrates learned representations before the task-specific prediction head. Despite sharing the same architecture, each task learns its own parameters to optimize latent representations.

4.4.2 Block MoE

The approach uses a separate MoE block for each DDI and DTI task. This design is based on the fact that both tasks process structurally heterogeneous input spaces, as previously discussed in the drug and protein modules. The DDI task exclusively adopts chemical representations from SMILES, while the DTI task combines chemical and protein information (SMILES with k-mer embeddings). Due to this fundamental difference, using a shared MoE would have resulted in interference between incompatible feature distributions, leading to suboptimal representation learning. This separation of MoE also allows each task to develop its own experts, given the nature of the data it processes. Moreover, it retains the benefits of dynamic gating $G(x)$, which assigns weights to the experts $E_i(x) = \{E_1, E_2, E_3, \dots, E_{N_e}\}$ for each sample. In the following, we will detail the architecture of the MoE block (see Fig. 3).

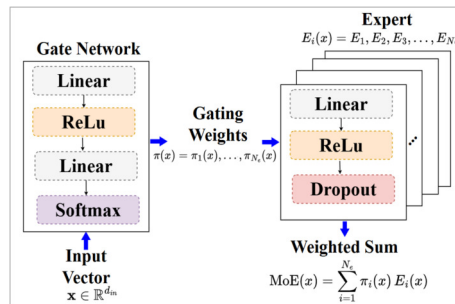


Fig.3: Block MoE.

1. General structure of the MoE block

The MoE block relies on multiple E_i experts working in parallel rather than a single FC block. There is a gating network that assigns weights to the experts based on the input. For an $x \in R^{d_{in}}$, a weighted combination of the expert outputs and the gating weights determines the block's final output. The general formula is as follows: $MoE(x) = \sum_{i=1}^{N_e} \pi_i(x) E_i(x)$. Where: N_e is the number of experts, $E_i(x)$ is the output of the i^e expert, $\pi_i(x)$ is the weight assigned to this expert, and $\sum_{i=1}^{N_e} \pi_i(x) = 1$.

2. Expert structure

Each expert is a small multi-layer perceptron (MLP) [4] composed of a linear layer, a ReLU activa-

tion [29], and a dropout [27] for regularization. For an input x , the output of the i^e expert is given by the following formula: $E_i(x) = \text{Dropout}(\text{ReLU}(W_i x + b_i))$. Given that: $W_i \in R^{d_h \times d_{in}}$ and $b_i \in R^{d_h}$ are the parameters of each expert, d_h corresponds to the hidden dimension of expert output.

The goal of each expert is to learn a different transformation of the input space, thereby modeling the various relationships in molecular or protein data.

3. Network gating

The gating network $G(x)$ selects the relevant experts for a given input. It is implemented as a two-layer fully connected MLP [4] with ReLU activation [29] in the middle, followed by Softmax normalization at the output. Its formula is as follows: $\pi(x) = \text{Softmax}(W_2 \text{ReLU}(W_1 x + b_1) + b_2)$. Given that: W_1 and W_2 are the weights of the two layers, b_1 and b_2 are the biases of the two layers, and $\pi(x) = \pi_1(x), \pi_2(x), \pi_3(x), \dots, \pi_{N_e}(x)$ correspond to the normalized probabilities assigned to each expert. This mechanism identifies the relevant experts for each sample.

4.5 Model output

The model has task-specific heads (DDI and DT), each utilizing the representations provided by the MoE block. Below, we detail the operation of these two heads and the combination of their losses in the multitask model.

4.5.1 Head DDI

In the DDI task, each molecule is encoded using the same drug-specific module, yielding two embedding vectors d_1 and d_2 , which are then concatenated to form $x_{DDI} = [d_1; d_2]$. Then they are processed by the MoE block: $h_{DDI} = \text{MoE}(x_{DDI})$. The DDI head applies a nonlinear projection followed by a regularization dropout: $h'_{DDI} = \text{Dropout}(\text{ReLU}(W_1^{DDI} h_{DDI} + b_1^{DDI}))$. A final linear layer and a sigmoid function produce the final output: $\hat{y}_{DDI} = \sigma(W_2^{DDI} h'_{DDI} + b_2^{DDI})$. Where: σ is the sigmoid function. $\hat{y}_{DDI} \in [0, 1]$: represents the predicted probability of interactions with two drugs.

Finally, the loss function compares this prediction to the actual value using binary cross-entropy (BCE):

$$\mathcal{L}_{DDI} = \frac{1}{N} \sum_{j=1}^N [y_{DDI}^{(j)} \log(\hat{y}_{DDI}^{(j)}) + (1 - y_{DDI}^{(j)}) \log(1 - \hat{y}_{DDI}^{(j)})] \quad (1)$$

4.5.2 Head DTI

In the DTI task, the model combines the embeddings of the drug d and protein p after their respective dedicated modules have processed them. The representations are then merged according to several complementary vector relationships (concatenation, product, and the absolute difference): $x_{DTI} = [d; p; (d \odot p); |d - p|]$. This vector is then

passed into the MoE block: $h_{DTI} = \text{MoE}(x_{DTI})$. Then, in the DTI head, similar in structure to the DDI, with intermediate dropout: $h'_{DTI} = \text{Dropout}(\text{ReLU}(W_1^{DTI} h_{DTI} + b_1^{DTI}))$. The final output is: $\hat{y}_{DTI} = \sigma(W_2^{DTI} h'_{DTI} + b_2^{DTI})$. The loss is a BCE:

$$\mathcal{L}_{DTI} = \frac{1}{N} \sum_{j=1}^N [y_{DTI}^{(j)} \log(\hat{y}_{DTI}^{(j)}) + (1 - y_{DTI}^{(j)}) \log(1 - \hat{y}_{DTI}^{(j)})] \quad (2)$$

4.5.3 Multitask combination

The learning is performed conjointly on both tasks with a weighted total cost function:

$$\mathcal{L}_{tot} = \alpha_{DDI} \mathcal{L}_{DDI} + \alpha_{DTI} \mathcal{L}_{DTI} \quad (3)$$

With: $\alpha_{DDI} = \alpha_{DTI} = 0.5$. Separate heads allow each task to learn its own decision rules. Furthermore, the use of dropout in the heads (h'_{DDI}) and (h'_{DTI}) introduces stochastic regularization, limiting the co-adaptation of the weights of final layers. Thus, the model retains good generalization while efficiently exploiting the representations learned in the MoE blocks.

5. EXPERIMENTAL STUDY AND DISCUSSION

We present our experimental study, starting with data, experimental setup, training strategy, and evaluation metrics. We then discuss results and provide a validation case study. Finally, we analyze the model's theoretical complexity and computational cost.

5.1 Datasets

In our study, we employed three widely used datasets: C. elegans [30] and Human [30] for DTI, and BioSNAP [31] for the DDI task. The C. elegans dataset comprises 1,434 drugs and 2,504 distinct proteins [30], [32]. As for the Human dataset, it comprises 1,052 drugs and 852 proteins [30], [32]. The negative samples from the Human and C. elegans datasets were generated by Liu et al. [30] using a systematic screening approach based on the dissimilarity between drugs and proteins. They were subsequently provided by Tsubaki *et al.* [32]. For the DDI task, we used the BioSNAP dataset, which includes 1,322 approved drugs and 41,520 positive interactions. Negative samples were selected and filtered by Huang *et al.* [33] by considering all possible drug pairs and excluding known positive interactions. The remaining pairs were treated as candidate negatives and randomly sampled. Furthermore, the datasets exhibit a balanced distribution of positive and negative samples, preventing bias toward either class during training (see Table 1 for dataset details).

Table 1: Details of the datasets.

Dataset	Total	Positive	Negative
Human	6,728	3,364	3,364
C.elegans	7,786	3,893	3,893
BioSNAP	83,040	41,520	41,520

5.2 Configuration and training strategy

This section describes data splitting strategies, hyperparameters, and training settings. All experiments were conducted on Google Colab with an NVIDIA T4 GPU (16 GB VRAM) and a CPU (12 GB RAM).

5.2.1 Data splitting strategies

The training strategy relies on dividing the dataset into training, validation, and test sets. For the DTI task, the dataset was randomly split into 80% training, 10% validation, and 10% test sets. For the DDI task (BioSNAP), the official test set was kept unchanged (20%), while the remaining data (80%) were used for training and validation, with a ratio of 8:2. The test set was strictly held out and never used during training (see Table 2). In both tasks, the splitting strategy was performed randomly at the pair level, following previous studies [9], [10], [19], and [24]. We applied five-fold cross-validation (CV) [34] to the training data, using a validation set to select the model.

Table 2: Sample count per fraction.

Dataset	Train	Val	Test
Human	5,382	673	673
C.elegans	6,228	779	779
BioSNAP	53,145	13,287	16,608

5.2.2 Hyperparameter configuration

In this study, the model's hyperparameters were empirically tuned through a series of experiments, accounting for available computational limitations. To ensure stable training and fair evaluation, we explored several key hyperparameters within predefined ranges. The learning rate was set to 0.0001 to ensure gradient explosion and stable convergence. The batch sizes were set to 64 for DDI and 128 for DTI to control computation time and cost while avoiding hardware resource overload. The weight decay was set to 0.0001, which helps control model complexity by penalizing excessively large weights. The dropout was evaluated for values of {0.2, **0.3**, 0.4} and acts as a means of regularization by randomly deactivating a proportion of neurons during training [27]. This prevents the model from becoming overly dependent on certain connections. Finally, the number of epochs was set to 60 to ensure adequate convergence without overfitting. To evaluate the robustness of the pro-

posed model in greater detail, we conducted a sensitivity analysis on the most influential hyperparameters. Due to computational constraints, this analysis focuses primarily on the number of experts and the dropout rate, with other parameters held constant. The final hyperparameters selected are shown in Table 3. A detailed sensitivity analysis is provided in Section 5.4.1.

Table 3: Selected hyperparameters.

Hyperparameter	Value
Optimizer	Adam
Epoch size	60
Learning rate	0.0001
Batch size	64 (DDI), 128 (DTI)
Scheduler	ReduceLROnPlateau
Weight Decay	0.0001
Clipping gradients	Yes, max norm = 5
Dropout	0.3
Loss Function	BCE

5.3 Metrics

The metrics used during this study are as follows [35]:

$$AUC = \int_0^1 TPR(FPR)d(TPR) \quad (4)$$

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$AUPR = \int_0^1 Precision(Recall)dr \quad (8)$$

$$F1 - score = 2 \times \frac{Prec \times Rec}{Prec + Rec} \quad (9)$$

Given that: TP: true positives; TN: true negatives; FP: false positives; FN: false negatives.

5.4 Discussion of results

Below, we present the results on the random pair-level split, including sensitivity analyses, five-fold cross-validation, learning curves, confusion matrices, gating interpretability, and ablation and statistical studies.

5.4.1 Sensitivity analysis

Below, we will examine the model's sensitivity to the main learning hyperparameters, as described in Section 5.2.2, including the number of experts and dropout.

5.4.1.1 Influence of the number of experts

To analyze the impact of model capacity on performance, we investigated the influence of the number of experts (N_e) in the MoE block by varying it from 2 to 4 for the DDI and DTI tasks (see Tables 4 and 5). The overall results show a slight improvement in performance as we increase the number of experts, with the best results generally achieved with 4 experts for the DDI and DTI tasks. For the C. elegans dataset, the 4-expert configuration for the DDI and DTI tasks offers a satisfactory compromise, with an AUC of 0.987 and an F1 score of 0.952. For the Human dataset, the same trend is observed. These results indicate that more experts suggest the model has a good ability to capture complex interaction structures. It is important to note that this increase does not appear to introduce noticeable overfitting, possibly due to the gating mechanism that regulates each expert’s contribution. However, the performance gains remain relatively modest, suggesting that increasing the number of experts beyond 4 offers only limited additional benefit. For the DDI task, performance remains stable regardless of the configuration or dataset. On both the Human and C. elegans datasets, the model consistently achieves AUC values around 0.996 and F1 scores close to 0.980, regardless of the number of experts. This stability suggests that the DDI task is less sensitive to model capacity, likely due to the more homogeneous nature of drug interaction data. In contrast, the DTI task appears to benefit more from an increase in the number of experts, suggesting that modeling drug-protein interactions requires greater representational capacity.

5.4.1.2 Influence of the dropout

After analyzing the influence of model architecture based on the number of experts, we focus on the impact of dropout while keeping the other hyperparameters fixed, notably the number of experts (4 for DDI, 4 for DTI), as determined in Section 5.4.1.1. The influence of the dropout rate was evaluated using three values {0.2, 0.3, and 0.4} for the DDI (BioSNAP) and DTI (Human) tasks. For the DDI task, dropout rates of 0.2 and 0.3 yield comparable median AUC values, with 0.2 showing a slight outlier, while 0.4 exhibits higher dispersion (see **Fig. 4**). Overall, performance differences remain marginal across configurations. For the DTI task, a dropout rate of 0.3 tends to achieve strong median AUC values with relatively stable performance across folds. Although a dropout rate of 0.4 yields a slightly higher median F1-score, it comes at the cost of increased variability, indicating reduced stability (see Fig. 4). Overall, these results suggest that model performance remains robust to dropout within this range. Thus, 0.3 is selected as a trade-off between performance and stability.

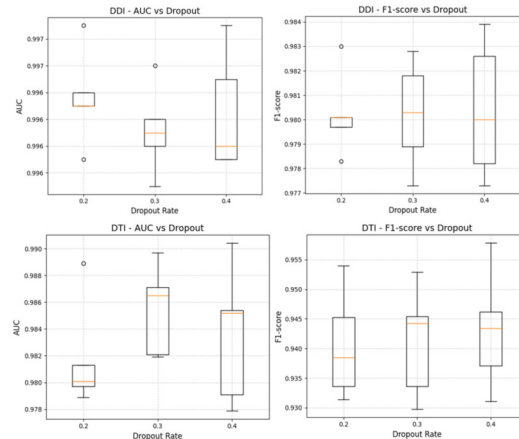


Fig.4: Impact of dropout.

5.4.2 Cross-validation results

In the following, we present the results obtained via five-fold CV for all datasets. On the Human dataset (see Table 6), we observe that the model achieved an average AUC of 0.985 ± 0.003 and accuracy of 0.942 ± 0.009 , with low variance across folds, indicating consistent and reliable performance. Slightly better performance is observed on the C. elegans dataset (see Table 6), with an AUC of 0.987 ± 0.003 and an accuracy of 0.952 ± 0.005 , suggesting that the model better captures this organism’s protein structure. The reduced standard deviations across all metrics for these two datasets indicate minimal inter-fold variance, reinforcing the reliability of the performance estimates. For the BioSNAP dataset (see Table 7), the results show an AUC of 0.996 ± 0.001 and a recall of 0.995 ± 0.001 , regardless of whether training was performed using the Human or C. elegans dataset as the supplementary source. The equivalent results obtained in these two configurations indicate that the model is not sensitive to the auxiliary data source, attesting to transferability and consistency. Overall, these results show that the proposed model maintains stable performance across folds, indicating its effectiveness and ability to generalize.

5.4.3 Learning curves

The evolution of training and validation losses (averaged over the 5 folds) is apparent in **Fig. 5** over 60 epochs for different model variants. We observe that for all configurations, the loss decreases regularly, indicating convergence. However, several differences emerge:

- Single task models (**Fig. 5(a–b)**): the loss decreases rapidly during the initial epochs before stabilizing. On BioSNAP (**Fig. 5a**), the validation loss remains below the training loss, indicating good generalization. On Human (**Fig.**

Table 4: Model performance with different numbers of experts (Human and *C.elegans*).

#Expert DDI	#Expert DTI	AUC	AUPR	Accuracy	Precision	F1	Recall
<i>Human with BioSNAP dataset</i>							
2	2	0.983±0.003	0.984±0.004	0.942±0.004	0.935±0.011	0.942±0.004	0.949±0.014
2	4	0.984±0.003	0.985±0.003	0.941±0.007	0.954±0.007	0.940±0.007	0.926±0.013
4	2	0.984±0.004	0.985±0.004	0.941±0.008	0.938±0.016	0.941±0.007	0.943±0.013
4	4	0.985±0.003	0.986±0.004	0.942±0.009	0.951±0.013	0.941±0.008	0.931±0.013
<i>C.elegans with BioSNAP dataset</i>							
2	2	0.985±0.002	0.986±0.003	0.951±0.008	0.959±0.004	0.950±0.008	0.942±0.016
2	4	0.987±0.002	0.987±0.002	0.947±0.006	0.955±0.006	0.946±0.006	0.938±0.014
4	2	0.986±0.003	0.987±0.003	0.947±0.007	0.962±0.006	0.946±0.007	0.930±0.013
4	4	0.987±0.003	0.987±0.003	0.952±0.005	0.959±0.005	0.952±0.005	0.944±0.012

Table 5: Model performance with different numbers of experts (BioSNAP).

#Expert DDI	#Expert DTI	AUC	AUPR	Accuracy	Precision	F1	Recall
<i>With the Human dataset</i>							
2	2	0.996±0.001	0.995±0.001	0.980±0.003	0.966±0.006	0.980±0.003	0.995±0.001
2	4	0.996±0.001	0.995±0.001	0.980±0.002	0.965±0.003	0.980±0.001	0.995±0.001
4	2	0.996±0.001	0.995±0.001	0.978±0.003	0.962±0.008	0.978±0.003	0.995±0.002
4	4	0.996±0.001	0.995±0.001	0.980±0.002	0.966±0.005	0.980±0.002	0.995±0.001
<i>With C.elegans dataset</i>							
2	2	0.996±0.001	0.995±0.001	0.979±0.002	0.964±0.005	0.979±0.002	0.996±0.002
2	4	0.996±0.001	0.995±0.001	0.980±0.002	0.965±0.004	0.980±0.002	0.995±0.001
4	2	0.996±0.001	0.995±0.001	0.978±0.002	0.962±0.005	0.978±0.002	0.995±0.001
4	4	0.996±0.001	0.995±0.001	0.980±0.002	0.966±0.005	0.980±0.002	0.995±0.001

Table 6: CV results on the Human and *C. elegans* datasets.

K Fold	AUC (std)	AUPR (std)	Accuracy (std)	Precision (std)	F1 (std)	Recall (std)
<i>Human with BioSNAP dataset</i>						
1	0.981	0.985	0.928	0.929	0.929	0.930
2	0.987	0.988	0.945	0.969	0.944	0.920
3	0.982	0.979	0.937	0.953	0.933	0.914
4	0.989	0.990	0.954	0.953	0.952	0.952
5	0.986	0.987	0.945	0.951	0.945	0.939
Avg	0.985±0.003	0.986±0.004	0.942±0.009	0.951±0.013	0.941±0.008	0.931±0.013
<i>C.elegans with BioSNAP dataset</i>						
1	0.984	0.982	0.949	0.951	0.949	0.947
2	0.988	0.990	0.955	0.959	0.956	0.953
3	0.982	0.984	0.943	0.964	0.942	0.921
4	0.990	0.991	0.955	0.958	0.955	0.952
5	0.987	0.988	0.956	0.963	0.955	0.947
Avg	0.987±0.003	0.987±0.003	0.952±0.005	0.959±0.005	0.952±0.005	0.944±0.012

Table 7: CV results on the BioSNAP dataset.

K Fold	AUC (std)	AUPR (std)	Accuracy (std)	Precision (std)	F1 (std)	Recall (std)
<i>With the Human dataset</i>						
1	0.996	0.995	0.976	0.958	0.977	0.997
2	0.995	0.994	0.980	0.966	0.980	0.994
3	0.996	0.995	0.978	0.962	0.978	0.995
4	0.996	0.995	0.981	0.970	0.981	0.993
5	0.996	0.994	0.982	0.970	0.982	0.995
Avg	0.996±0.001	0.995±0.001	0.980±0.002	0.966±0.005	0.980±0.002	0.995±0.001
<i>With C.elegans dataset</i>						
1	0.996	0.995	0.980	0.966	0.980	0.995
2	0.995	0.994	0.978	0.962	0.975	0.955
3	0.996	0.995	0.981	0.968	0.981	0.955
4	0.996	0.995	0.982	0.973	0.983	0.993
5	0.996	0.994	0.976	0.960	0.977	0.994
Avg	0.996±0.001	0.995±0.001	0.980±0.002	0.966±0.005	0.980±0.002	0.995±0.001

5b), the validation loss stabilizes slightly above the training loss, suggesting mild overfitting.

- Multitask-FC (Fig. 5(c-d)): follows a similar trajectory to that of the single-task model. On BioSNAP (Fig. 5c), the validation results remain below the training results. On Human (Fig. 5d), a slight discrepancy persists, comparable to that observed in Fig. 5b.
- Multitask-MoE (Fig. 5(e-f)): The training and validation losses follow a quasi-parallel trajectory, indicating good generalization. On BioSNAP (Fig. 5e), the loss decreases sharply during the first epochs before stabilization around a low value, with the validation curve remaining below the training curve, indicating good generalization. On Human (Fig. 5f), convergence remains steady throughout training, without marked fluctuations. Alignment between curves in both cases indicates a reduced generalization gap compared to other variants.

These observations show that introducing the MoE block in the multitask approach improves learning stability and final performance.

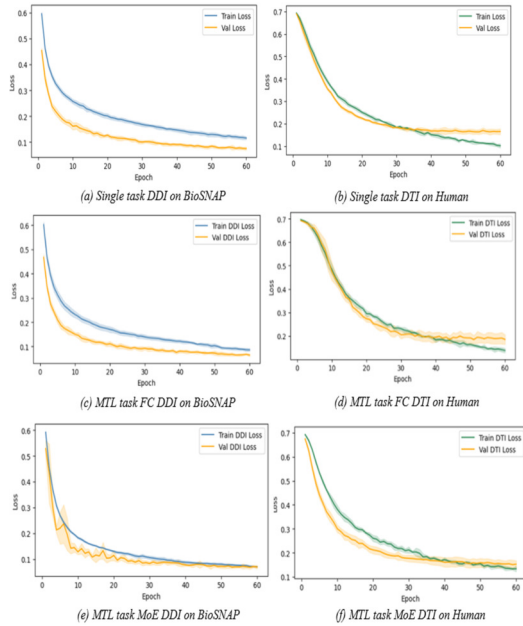


Fig.5: Learning curves.

5.4.4 Confusion matrix

Fig. 6(a-c) shows the confusion matrices for the DDI task on the BioSNAP dataset. We observe that the single-task model achieves stable performance (7,944 TN, 8,192 TP) with moderate errors (387 FP, 85 FN) (Fig. 6a). The FC multitask model shows a slight degradation, with an increase in false positives (421) and false negatives (101) (Fig. 6b). The MoE model, on the other hand, shows a good diagonal of (8,106 TN, 8,237 TP) with a reduction in errors (225 FP, 40 FN) (Fig. 6c).

Fig. 6(d-f) shows the confusion matrices for the DTI task on the Human dataset. We observe that the single-task model already achieves reasonable class separation (295 TN, 323 TP). Still, with a significant number of false positives (30) and false negatives (25) (Fig. 6d), the FC multitask model slightly degrades this balance (Fig. 6e), increasing false negatives (50) while reducing false positives (14). In contrast, the MoE model demonstrates good class separation (308 TN, 325 TP), with the lowest combined error rate (17 FP, 23 FN) (Fig. 6f). Based on these visualizations, we can observe that integrating the MoE mechanism into a multitask approach not only yields numerical improvements but also enhances model stability by consistently reducing false positives and false negatives across both datasets.

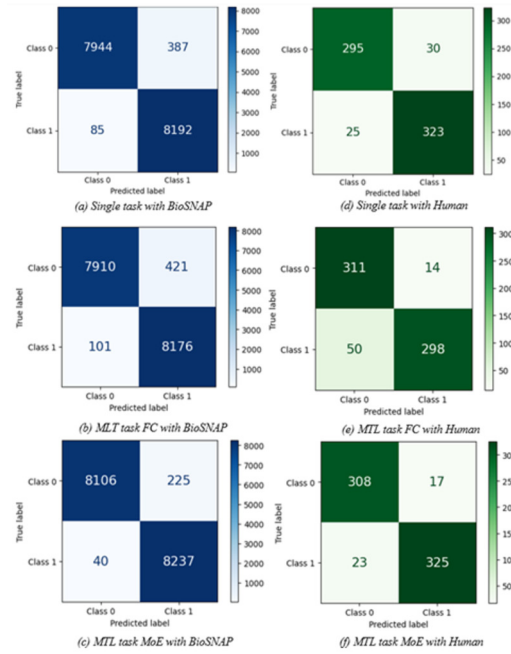


Fig.6: Confusion matrix.

5.4.5 Gating network analysis

To better understand how the selection mechanism works in the MoE architecture, we will illustrate the gating mechanism's behavior in detail using several complementary visualizations.

The heatmaps of gating weights (see Fig. 7 and Fig. 8) show the distribution of expert contributions for the first 100 test samples of the DDI and DTI tasks from the Human and BioSNAP datasets. For the DDI task (Fig. 7), we observe that the distribution of selection weights is largely consistent across the four experts. No single expert dominates the set of samples, and color variations remain moderate. This behavior suggests that the gate adjusts its weights differently for each sample, without converging to an exclusive delegation strategy.

For the DTI task (Fig. 8), we observe a distinct

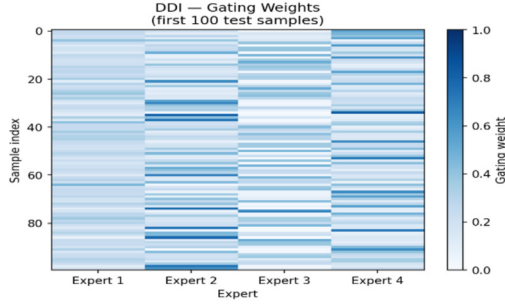


Fig. 7: MoE gating heatmap (DDI task).

pattern in the heatmap. Expert 1 shows significantly higher selection weights than the other three experts (2, 3, and 4) for most samples. This contrast suggests that the gate for the DDI task preferentially directs samples toward a particular expert, reflecting asymmetric expert roles in this task.



Fig. 8: MoE gating heatmap (DTI task).

To better characterize this behavior, we calculated Pearson correlations between each expert's selection weights and the average activation of the convolutional kernels applied to protein representations across three k-mer scales ($k = 3, 5$, and 7). The results are presented as a heatmap (see **Fig. 9**). Expert 1 has positive correlations with the k-mer activations for all three scales considered: $r = +0.52$ for $k = 3$, $r = +0.49$ for $k = 5$, and $r = +0.72$ for $k = 7$. These values suggest that the DTI gate tends to assign a higher weight to expert 1 as the protein representation's k-mer scale increases. In contrast, experts 2, 3, and 4 show negative correlations with these same activations, with the most pronounced values observed for expert 2 ($r = -0.51$ for $k = 7$), expert 4 ($r = -0.49$ for $k = 7$), and expert 3 ($r = -0.39$ for $k = 7$). The anti-correlation pattern suggests that when expert 1 is assigned a high weight, the other three experts are less frequently activated. These results, combined with the pattern observed in the heatmap (**Fig. 8**), suggest that expert 1 has a distinct function in protein representation.

This asymmetry suggests operating modes that vary with the nature of the task.

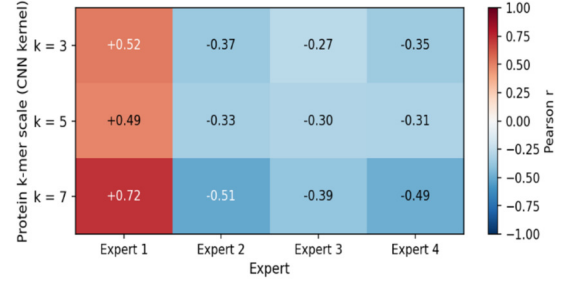


Fig. 9: Expert Correlations vs. k-mer Scales.

5.4.6 Ablation study

We conduct an ablation study to understand the impact of model configurations on overall performance. We compare variants presented previously in Section 4:

- Model trained separately for each task (DTI or DDI).
- Multitask model with an FC block.
- Multitask model with a separate MoE block per task.

This study aims to quantify the contribution of the MoE mechanism and verify whether it provides a meaningful improvement over the selective sharing of each task's representations. On the Human dataset (see Table 8), the multitask model with MOE performs well with an AUC of 0.985, an accuracy of 0.942, and an F1 score of 0.941, compared to the single-task and MTL-FC variants. While the multitask model with FC has a slightly higher precision (0.963), this is accompanied by a significantly lower recall (0.813), indicating less balanced performance. In contrast, MTL-MoE offers a good compromise between precision (0.951) and recall (0.931). On the C. elegans dataset (see Table 8), the same trend is observed.

Regarding the BioSNAP dataset (see Table 9), for both configurations (Human and C. elegans), the multitask model with MoE achieved significant results, with an AUC of up to 0.996, an accuracy of 0.980, and an F1-score of 0.980. Compared to previous observations, MoE also improves recall here, so it performs well compared to single-task models and MTL-FC models. These results demonstrate that multitasking with MoE represents the most efficient and stable configuration for numerical performance.

5.4.7 Statistical test

We used the corrected resampled t-test proposed by Nadeau and Bengio [36] to assess the statistical significance of the performance differences between the MoE model and the other models. The results of this test, conducted on the BioSNAP and Human datasets employing the F1 score metric (see Table 10). The significance threshold is set at $\alpha = 0.05$. For the DDI task, the MoE model obtained a delta of 0.0130 compared to the Single-Task model ($p =$

Table 8: Ablation results for the Human and *C.elegans* dataset .

Model \ Metrics	AUC	AUPR	Accuracy	Precision	F1	Recall
Human						
Single task	0.971±0.004	0.974±0.004	0.913±0.003	0.908±0.016	0.913±0.004	0.918±0.016
MTL-FC	0.972±0.005	0.975±0.004	0.890±0.015	0.963±0.033	0.880±0.021	0.813±0.060
MTL-MoE	0.985±0.003	0.986±0.004	0.942±0.009	0.951±0.013	0.941±0.008	0.931±0.013
C.elegans						
Single task	0.970±0.002	0.973±0.002	0.908±0.007	0.917±0.026	0.907±0.010	0.898±0.032
MTL-FC	0.973±0.007	0.976±0.007	0.908±0.020	0.970±0.014	0.908±0.020	0.841±0.055
MTL-MoE	0.987±0.003	0.987±0.003	0.952±0.005	0.959±0.005	0.952±0.005	0.944±0.012

Table 9: Ablation results for the BioSNAP dataset.

Model \ Metrics	AUC	AUPR	Accuracy	Precision	F1	Recall
Single task	0.994±0.001	0.993±0.001	0.969±0.003	0.951±0.006	0.969±0.003	0.989±0.001
With the Human dataset						
MTL-FC	0.994±0.001	0.993±0.001	0.967±0.004	0.948±0.006	0.967±0.003	0.987±0.003
MTL-MoE	0.996±0.001	0.995±0.001	0.980±0.002	0.966±0.005	0.980±0.002	0.995±0.001
With C.elegans dataset						
MTL-FC	0.994±0.001	0.993±0.001	0.967±0.003	0.948±0.005	0.968±0.003	0.988±0.004
MTL-MoE	0.996±0.001	0.995±0.001	0.980±0.002	0.966±0.005	0.980±0.002	0.995±0.001

Table 10: Comparison of the statistical performance of the MoE model with the corrected resampled *t*-test.

Task	Comparison	Δ	p-value	95% Confidence Interval (CI)	Significant? ($\alpha < 0.05$)
DDI	Single Vs. MoE	0.0130	0.0056	[0.0101, 0.0157]	Yes
	FC Vs. MoE	0.0037	0.0313	[0.0024, 0.0050]	Yes
DTI	Single Vs. MoE	0.0284	0.0100	[0.0193, 0.0364]	Yes
	FC Vs. MoE	0.0617	0.0131	[0.0408, 0.0808]	Yes

0.0056, 95% CI [0.0101; 0.0157]) and 0.0037 compared to the FC model ($p = 0.0313$, 95% CI [0.0024; 0.0050]). For the DTI task, the MoE model shows a delta of 0.0284 compared to the Single-Task model ($p = 0.0100$, 95% CI [0.0193; 0.0364]) and 0.0617 compared to the FC model ($p = 0.0131$, 95% CI [0.0408; 0.0808]). Furthermore, the 95% confidence intervals do not include zero, which indicates that the observed differences are not attributable to sampling variance.

5.5 Comparison results

Below, we present two comparisons: against state-of-the-art (SOTA) methods and against a GNN baseline trained under the same experimental protocol.

5.5.1 Comparison with state-of-the-art (SOTA)

The performance of our model was evaluated by comparing it with SOTA methods, namely [10], [15], [19], [9], [23], and [24]. The comparisons were carried out on the same three datasets. The metrics considered are primarily the AUC and the F1-score, which reflect the discriminative capacity and the balance between precision and recall, respectively [35].

On the Human dataset, MTL-MoE achieves competitive performance with an AUC of 0.985 and an F1 score of 0.941, outperforming [19] and remaining comparable to [10] and [15]. This highlights the effectiveness of the MoE mechanism for separating complex molecular representations (see Table 11).

On the *C. elegans* dataset, our model achieves an AUC of 0.987, outperforming [15] (0.986) and [19] (0.960), while remaining close to [10] (0.990). For the F1 score, our model achieves 0.952, which is

comparable to [10] and [15] (0.956) (see Table 11). These results show that the MoE-based model offers a consistent advantage across datasets, even in scenarios where SOTA models already achieve high performance.

Table 11: SOTA comparison (Human and *C.elegans*).

Model \ Metrics	Human		C.elegans	
	AUC	F1	AUC	F1
[10]	0.979	0.935	0.990	0.956
[15]	0.978	0.940	0.986	0.956
[19]	0.946	0.855	0.960	0.894
MTL-MOE	0.985	0.941	0.987	0.952

On the BioSNAP dataset, our model achieves an AUC of 0.996 and an F1 score of 0.980, outperforming [23] (AUC 0.974, F1 0.926) and equal to [9] in F1. While [24] achieves a slightly higher F1 score (0.982 vs. 0.980), our model outperforms it in AUC (0.996 vs. 0.992) (see Table 12). These results show that the proposed model maintains stable performance on complex DDI tasks with high chemical redundancy.

Table 12: SOTA comparison (BioSNAP).

Model \ Metrics	AUC	F1
[9]	0.990	0.980
[23]	0.974	0.926
[24]	0.992	0.982
MTL-MOE	0.996	0.980

5.5.2 Comparison with baseline

To enhance the credibility of our comparative evaluation, we reimplemented two architectures based on GNN [7], which are as follows:

- GraphDTA [37] (for the DTI task): represents each molecule using graph convolutional layers and encodes proteins using a 1D CNN. Two variants were evaluated: GCN and GAT-GCN.
- SSI-DDI [38] (for the DDI task): encodes each molecule using GAT layers and models interactions between their substructures via a co-attention mechanism. These two methods were trained and evaluated using the same experimental protocol and on the same datasets as our MTL-MoE model.

For the DTI task, the GCN variant of GraphDTA achieves an AUC of 0.966 and an F1-score of 0.908 on the Human dataset, while the GAT-GCN variant improves upon these results with 0.975 and 0.920, respectively. Our MTL-MoE model outperforms these two variants, achieving an AUC of 0.985 and an F1-score of 0.941. On *C. elegans*, the trend is similar (see Table 13).

Table 13: Baseline comparison (Human and *C.elegans*).

Model \ Metrics	Human		<i>C.elegans</i>	
	AUC	F1	AUC	F1
GraphDTA (GCN) [37]	0.966 ±0.006	0.908 ±0.013	0.978 ±0.006	0.936 ±0.010
GraphDTA (GAT-GCN) [37]	0.975 ±0.005	0.920 ±0.008	0.985 ±0.001	0.945 ±0.003
MTL-MOE	0.985 ±0.003	0.941 ±0.008	0.987 ±0.003	0.952 ±0.005

For the DDI task, SSI-DDI achieves an AUC of 0.981 ± 0.001 and an F1 score of 0.934 ± 0.002 on the BioSNAP dataset. Our MTL-MoE model outperforms the baseline model, with an AUC of 0.996 ± 0.001 and an F1 score of 0.980 ± 0.002 (see Table 14).

Table 14: Baseline comparison (BioSNAP).

Model \ Metrics	AUC	F1
SSI-DDI [38]	0.981 ± 0.001	0.934 ± 0.002
MTL-MOE	0.996 ± 0.001	0.980 ± 0.002

5.6 Case study

To assess the generalizability of our model, we tested it on the most commonly used drugs provided by the pharmacovigilance laboratory at the university health institution in Oran, Algeria. The model shows good overall agreement between real values and predictions on the DTI task (see Table 15, which lists each drug by its DrugBank ID and each protein by its UniProt ID). On this new data, only one interaction (DB00717–P06401) was misclassified. Norethis-

terone (DB00717) interacts with the progesterone receptor (P06401), and this pair is labeled as positive in the Human dataset. However, the model predicts a negative interaction. A bias in the training data can explain this error. The molecules that contain a terminal alkyne ($C\equiv C$), such as norethisterone, are associated with a non-interaction in 71.6% of the cases in the dataset, compared to only 28.4% of positive interactions. The model has therefore implicitly learned that this chemical group is predominantly incompatible with protein binding. When evaluating the DB00717–P06401 pair, this negative bias dominates its prediction, despite the actual interaction present.

On the other hand, P06401 is implicated in only six pairs in the dataset, all of which are positive, limiting the model’s ability to learn a reliable binding profile for this target. When evaluating the DB00717–P06401 pair, these two biases may be the cause of the observed misclassification. On the DDI task (see Table 15), only one interaction (DB00421–DB00717) was wrongly predicted as positive. From the analysis of the BioSNAP training dataset, we observe that neither norethisterone (DB00717) nor spironolactone (DB00421) is present in the DDI training data. The model has no direct signal to evaluate their combination. It is forced to generalize only from structurally similar molecules, which leads to the observed misclassification. These results suggest that our model can be generalized to local data. However, these results remain exploratory, and further validation on larger, more diverse datasets would be essential.

Table 15: Case study.

DTI task			
Drug	Protein	Real	Predicted
DB00331	Q9Y478	True	True
DB00945	P23219	True	True
DB00091	P49069	True	True
DB00717	P06401	True	False
DB00080	P02768	False	False
DB00252	P11712	False	False
DB00264	P07550	False	False
DB00295	P10632	False	False
DDI task			
Drug1	Drug2	Real	Predicted
DB00421	DB01048	True	True
	DB00331	True	True
	DB00945	True	True
	DB00091	True	True
	DB00717	False	True
	DB14480	False	False
	DB03692	False	False
	DB01050	False	False

5.7 Computational complexity

We describe the algorithmic complexity of our proposed model in Table 16.

Table 16: Proposed model complexity.

Component	Complexity
DDI representation	$O(d)$
DTI representation	$O(d)$
Drug module	$O(d_f.k)$
Protein module	$O(L'.k)$
MoE (DDI)	$O(N_e^{DDI}.2d.d_h)$
MoE (DTI)	$O(N_e^{DTI}.4d.d_h)$
Gating network	$O(d_{in}.N_e)$

Given that: where d, d_f, L', d_h, d_{in} , and N_e : Note the embedding dimension, fingerprint size, sequence length, hidden size, input dimension, and number of experts, respectively. The dedicated modules for drugs and proteins are moderately complex due to the use of convolution operations. This complexity is mitigated by the pooling layers, which significantly reduce feature dimensionality. The MoE modules influence the overall complexity of the model. The DDI task operates on a $2d$ input, whereas the DTI task uses a richer $4d$ representation. The overall complexity can therefore be written as follows:

$$O(N_e^{DDI}.2d.d_h + N_e^{DTI}.4d.d_h) \quad (10)$$

5.8 Computational cost

Table 17 illustrates the computational analysis of the MoE model compared to the FC model. We observe that the MoE introduces a moderate computational overhead compared to the FC baseline. In fact, the MoE model contains approximately 1.31 million parameters compared to 1.16 million for the FC model (an increase of about 14%), with a corresponding model size of 5.04 MB versus 4.44 MB. In terms of theoretical complexity, both models exhibit identical FLOP (Floating Point Operations) [39] for the DDI task (1.40G) and nearly identical values for the DTI task (5.34G vs. 5.36G), indicating that the MoE architecture does not significantly increase arithmetic complexity. In terms of inference latency, the MoE model shows a moderate increase of approximately 15% for the DDI task (53.054 vs. 45.934 μ s/sample) and about 4% for the DTI task (144.888 vs. 139.556 μ s/sample). GPU memory consumption increases moderately, from 61.01 to 81.29 MB for the DDI task and from 135.26 to 155.54 MB for the DTI task. Overall, these results suggest that the MoE architecture provides competitive predictive performance while maintaining a reasonable computational cost.

Table 17: MoE vs FC Cost.

Model / Metrics	DDI		DTI	
	FC	MoE	FC	MoE
Params	1,159,586	1,316,072	1,159,586	1,316,072
Size (MB)	4.44	5.04	4.44	5.04
FLOP	1.40G	1.40G	5.34G	5.36G
Inference Time (μ s/sample)	45.934	53.054	139.556	144.888
GPU Memory (MB)	61.01	81.29	135.26	155.54

6. CONCLUSION

In this work, we proposed a multitask architecture for parallel prediction of DDI and DTI interactions. The model is based on a combination of MoE blocks, enabling good generalization. The results across several datasets show that the multitask MoE model achieves competitive performance compared to single-task approaches and multitask architectures that rely solely on dense layers. This model shows good robustness across diverse chemical and protein representation spaces. In addition, we validated the model using real local data from the Oran pharmacovigilance laboratory, thereby highlighting its generalizability. For prospects, we plan to expand our scope of study to other types of interactions (drug-food, disease-drug, etc.). From a technical perspective, the modular design of the proposed MoE architecture facilitates such extensions. For example, food compounds can be represented using molecular structures similar to those of drugs (e.g., SMILES-based representations), allowing the reuse of the existing drug encoding module. For disease-related interactions, additional modules can be designed to incorporate disease-specific features, such as clinical or biological descriptors, and integrated into the framework via the same modular pipeline. We also intend to explore additional datasets and investigate self-supervised learning to further improve the model’s transferability. Moreover, we will consider implementing more systematic hyperparameter optimization strategies, such as grid search or Bayesian optimization, to further improve the model’s performance and reproducibility. We also wish to explore more rigorous evaluation settings, including cold-start scenarios (drugs or proteins not used in the experiment), to better assess generalization under real-world conditions. Finally, we plan to add another MoE variant that supports multiple gates.

ACKNOWLEDGEMENT

The authors thank the Directorate General for Scientific Research and Technological Development (DGRSDT) for its support of the national project to develop “An intelligent clinical decision support system dedicated to pharmacovigilance”, carried out in collaboration between the LIO laboratory and the pharmacovigilance laboratory of the EHU of Oran.

AUTHOR CONTRIBUTIONS

Conceptualization, K.M.; methodology, K.M.; software, K.M.; validation, K.M., K.B., and D.H.; formal analysis, K.M.; investigation, K.M.; data curation, K.M.; resources, K.M.; writing—original draft preparation, K.M.; writing—review and editing, K.M., K.B., and D.H.; visualization, K.M.; supervision, K.B. and D.H. All authors have read and agreed to the published version of the manuscript.

References

- [1] G. Jeetu and G. Anusha, “Pharmacovigilance: A Worldwide Master Key for Drug Safety Monitoring,” *Journal of Young Pharmacists*, vol. 2, no. 3, pp. 315–320, 2010.
- [2] D. Hamdadou, H. Toumi, F. Zemri, Y. Omari and A. Bengueddach, “Intelligent Clinical Decision Support System for Monitoring and Preventing Adverse Effects of Drug-Drug Interaction,” in *Intelligent and Fuzzy Systems*, Springer, pp. 250–257, 2024.
- [3] K. Makhoulouf, S. Madouri, D. Hamdadou and K. Bouamrane, “An Intelligent Clinical Decision Support System for Managing Iatrogenic Drug Risks Using Electronic Medical Records,” in *Intelligent and Fuzzy Systems*, Springer, pp. 368–376, 2025.
- [4] T. Perumal, N. Mustapha, R. Mohamed and F. M. Shiri, “A Comprehensive Overview and Comparative Analysis on Deep Learning Models,” *Journal on Artificial Intelligence*, vol. 6, no. 1, pp. 301–360, 2024.
- [5] Q. Zhao, H. Zhao, K. Zheng and J. Wang, “HyperAttentionDTI: improving drug–protein interaction prediction by sequence-based deep learning with attention mechanism,” *Bioinformatics*, vol. 38, no. 3, pp. 655–662, 2022.
- [6] C. Zhang, Y. Lu and T. Zang, “CNN-DDI: a learning-based method for predicting drug–drug interactions using convolution neural networks,” *BMC Bioinformatics*, vol. 23, no. S1, p. 88, 2022.
- [7] J. Zhou *et al.*, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020.
- [8] Y. Pan, Y. Zhang, J. Zhang and M. Lu, “CS-DTI: an interpretable cross-attention network with GNN-based drug molecule aggregation for drug-target interaction prediction,” *Applied Intelligence*, vol. 53, no. 22, pp. 27177–27190, 2023.
- [9] S. Gupta, S. Laghuvarapu and U. D. Priyakumar, “GraphDDI: Graph Neural Network for Prediction of Drug-Drug Interaction,” in *Artificial Intelligence in Healthcare*, Springer, pp. 17–30, 2024.
- [10] M. Zhao, M. Yuan, Y. Yang and S. X. Xu, “CPGL: Prediction of Compound-Protein Interaction by Integrating Graph Attention Network With Long Short-Term Memory Neural Network,” in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 3, pp. 1935–1942, 1 May–June 2023.
- [11] F. Liu, H. Xu, P. Cui, S. Li, H. Wang and Z. Wu, “NFSA-DTI: A Novel Drug–Target Interaction Prediction Model Using Neural Fingerprint and Self-Attention Mechanism,” *International Journal of Molecular Sciences*, vol. 25, no. 21, p. 11818, 2024.
- [12] X. Ouyang, Y. Feng, C. Cui, Y. Li, L. Zhang and H. Wang, “Improving generalizability of drug–target binding prediction by pre-trained multi-view molecular representations,” *Bioinformatics*, vol. 41, no. 1, p. btaf002, 2024.
- [13] J. Qiu *et al.*, “PTB-DDI: An Accurate and Simple Framework for Drug–Drug Interaction Prediction Based on Pre-Trained Tokenizer and BiLSTM Model,” *International Journal of Molecular Sciences*, vol. 25, no. 21, p. 11385, 2024.
- [14] X. Zeng, W. Chen and B. Lei, “CAT-DTI: cross-attention and Transformer network with domain adaptation for drug-target interaction prediction,” *BMC Bioinformatics*, vol. 25, no. 1, p. 141, 2024.
- [15] X. Gong, Q. Liu, J. He, Y. Guo and G. Wang, “MultiGranDTI: an explainable multi-granularity representation framework for drug-target interaction prediction,” *Applied Intelligence*, vol. 55, no. 2, p. 107, 2025.
- [16] K. Makhoulouf, D. Hamdadou, Y. Omari, K. Bouamrane and F. A. Zemri, “Drug-Target Interaction Prediction Based on GCN and BiLSTM with a Cross-Attention Mechanism,” in *Proc. 2024 International Conference of the African Federation of Operational Research Societies (AFROS)*, Tlemcen, Algeria, pp. 1–5, 2024.
- [17] S. Liu, Y. Liu, H. Xu, J. Xia and S. Z. Li, “SP-DTI: subpocket-informed transformer for drug–target interaction prediction,” *Bioinformatics*, vol. 41, no. 3, p. btaf011, 2025.
- [18] Y. Zhang, Q. Wang, C. Zhang, B. Feng, J. Shang and L. Zhang, “IHDFN-DTI: Interpretable Hybrid Deep Feature Fusion Network for Drug–Target Interaction Prediction,” *Interdisciplinary Sciences: Computational Life Sciences*, 2025.
- [19] D. Gao and F. Zhu, “GraphCF: Drug–target interaction prediction via multi-feature fusion with contrastive graph neural network,” *Artificial Intelligence in Medicine*, vol. 167, p. 103196, 2025.
- [20] P. Gupta, A. Majumdar, E. Chouzenoux and G. Chierchia, “DeConDFFuse: Predicting drug–drug interaction using joint deep convolutional transform learning and decision forest fusion framework,” *Expert Systems with Applications*, vol. 227, p. 120238, 2023.

- [21] Z. Jiang, P. Ding, C. Shen and X. Dai, “Geometric Molecular Graph Representation Learning Model for Drug-Drug Interactions Prediction,” *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 12, pp. 7623–7632, 2024.
- [22] D. Niu, L. Zhang, B. Zhang, Q. Zhang and Z. Li, “DAS-DDI: A dual-view framework with drug association and drug structure for drug–drug interaction prediction,” *Journal of Biomedical Informatics*, vol. 156, p. 104672, 2024.
- [23] Z. Jiang, Z. Gong, X. Dai, H. Zhang, P. Ding and C. Shen, “Deep graph contrastive learning model for drug-drug interaction prediction,” *PLoS ONE*, vol. 19, no. 6, p. e0304798, 2024.
- [24] L. Wang, Y. Li, Y. Zhou, L. Guo and C. Chen, “MFE-DDI: A multi-view feature encoding framework for drug-drug interaction prediction,” *Computational and Structural Biotechnology Journal*, vol. 27, pp. 2473–2480, 2025.
- [25] D. Weininger, “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules,” *Journal of Chemical Information and Computer Sciences*, vol. 28, no. 1, pp. 31–36, 1988.
- [26] J. L. Ba, J. R. Kiros and G. E. Hinton, “Layer Normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [28] C. Moeckel *et al.*, “A survey of k-mer methods and applications in bioinformatics,” *Computational and Structural Biotechnology Journal*, vol. 23, pp. 2289–2303, 2024.
- [29] S. Kılıçarslan, K. Adem, and M. Çelik, “An overview of the activation functions used in deep learning algorithms,” *Journal of New Results in Science*, vol. 10, no. 3, pp. 75–88, 2021.
- [30] H. Liu, J. Sun, J. Guan, J. Zheng and S. Zhou, “Improving compound–protein interaction prediction by building up highly credible negative samples,” *Bioinformatics*, vol. 31, no. 12, pp. i221–i229, 2015.
- [31] M. Zitnik, S. R. Sosc and J. Leskovec, “BioS-NAP Datasets: Stanford Biomedical Network Dataset Collection,” 2018. [Online]. Available: <https://snap.stanford.edu/biodata/>
- [32] M. Tsubaki, K. Tomii and J. Sese, “Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences,” *Bioinformatics*, vol. 35, no. 2, pp. 309–318, 2019.
- [33] K. Huang, C. Xiao, T. Hoang, L. Glass and J. Sun, “CASTER: Predicting Drug Interactions with Chemical Substructure Representation,” in *Proc. AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 702–709, 2020.
- [34] D. Berrar, “Cross-Validation,” in *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, pp. 542–545, 2019.
- [35] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [36] C. Nadeau and Y. Bengio, “Inference for the generalization error,” *Advances in Neural Information Processing Systems*, vol. 12, 1999.
- [37] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le and S. Venkatesh, “GraphDTA: predicting drug–target binding affinity with graph neural networks,” *Bioinformatics*, vol. 37, no. 8, pp. 1140–1147, 2021.
- [38] A. K. Nyamabo, H. Yu and J.-Y. Shi, “SSI-DDI: substructure–substructure interactions for drug–drug interaction prediction,” *Briefings in Bioinformatics*, vol. 22, no. 6, p. bbab133, 2021.
- [39] S. Abnar, H. Shah, D. Busbridge, A. M. E. Ali, J. Susskind and V. Thilak, “Parameters vs FLOPs: Scaling Laws for Optimal Sparsity for Mixture-of-Experts Language Models,” *arXiv preprint arXiv:2501.12370*, 2025.



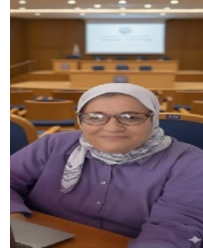
Kawther Makhoulf received her master's degree in “Decision Support and Intelligent Systems” from the University of Oran 1 Ahmed Ben Bella, Algeria in 2022. She is currently pursuing a PhD in “Artificial Intelligence” at the same university. Her research interests include intelligent systems, artificial intelligence, in particular machine learning, deep learning, transformer models. She can be contacted at email:

makhoulf.kawther@edu.univ-oran1.dz



Karim Bouamrane received the PhD Degree in computer science from the Oran University in 2006. He is full Professor of computer Science at the same university. He is member of the Laboratory of Informatics of Oran (LIO). He is the head of the team diagnosis and decision support system. His current research interest's deal with decision support system in maritime and urban transportation system, production

system, application of bio-inspired based optimization meta-heuristic, health systems. He participates in several scientific international/national committees conferences in Algeria and others countries in the same domain and collaborates in many scientific projects. He is co-author of more than 70 scientific publications and communication. He can be contacted at email: bouamrane.karim@univ-oran1.dz



Djamila Hamdadou specializes in Artificial Intelligence, Decision Support Systems, Multi-Criteria Analysis, Collaborative and Spatio-Temporal Decision Systems, and Business Intelligence. She is a Full Professor at University Oran 1 and Head of the Laboratory of Computer Science of Oran (LIO), where she leads the research team "Artificial Intelligence Tools for Spatio-Temporal and Medical Decision Support". She is

also the author and co-author of numerous scientific publications in international journals and conferences, and has participated as a speaker and invited lecturer in several national and international scientific events related to Artificial Intelligence and decision support systems. She can be contacted at email: hamdadou.djamila@univ-oran1.dz