Computer Science • 27(1) 2026 https://doi.org/10.7494/csci.2026.27.1.6899

Samridhi Dev
Aditi Sharan

ENHANCED GNN WITH CAUSAL PROXIMITY VECTORS: BRIDGING CAUSALITY AND PROXIMITY IN GRAPH NEURAL NETWORKS USING TEXTUAL DATA

Abstract *A knowledge graph is a structured representation of entities and their relationships, often used in biomedical domains to model complex interactions. Graph neural networks (GNNs), which utilize these graphs, are effective for predicting interactions missing in the knowledge graph. However, GNN lacks the ability to incorporate causal reasoning, which is crucial to biomedical applications. Additionally, they limit their ability to generalize to unseen data. In oncology, where treatment regimens are intricate and patient responses are highly variable, predicting adverse drug reactions (ADRs) is particularly difficult. Existing models fail to capture the indirect, high-granularity information needed for accurate ADR prediction. To address these challenges, we propose the Causality and Proximity-based Relational Multihead Attention Model (CPRMAM). This model leverages a knowledge graph of ADR-related cancer case studies and introduces a causal proximity vector to prioritize relevant relationships. By employing an inductive GNN approach, CPRMAM generalizes to unseen data, improving ADR prediction.*

Keywords graph neural network, adverse drug reactions, aggregation function, CPRMAM, knowledge graph completion

Citation Computer Science 27(1) 2026: 147–169

1. Introduction

Adverse drug reactions (ADRs) represent a serious concern in health care, as they can lead to severe and potentially fatal outcomes. Therefore, early detection and prediction of ADRs are critical for ensuring patient safety. While numerous deep-learning (DL) and machine learning (ML) models have been developed for various biomedical applications such as drug-drug interaction (DDI), drug-protein interaction (DPI), and association mining, there is a significant gap in models specifically targeting ADR prediction. Recent advancements in graph representation learning, particularly with graph neural networks (GNNs), have demonstrated promising performance in the biomedical domain. GNNs are particularly effective at learning high-quality embeddings that capture the topological properties of graphs, thus overcoming the limitations of traditional methods based on statistical associations, similarity measures, and conventional ML frameworks.

However, GNNs face several challenges. Most GNN embeddings primarily focus on topological and structural features, often overlooking the semantic properties of individual nodes. Additionally, during model training, neighbor nodes are usually treated with equal importance, disregarding the varying degrees of relevance each neighbor may have to the target node. To address these issues, existing models generally rely on simple relational structures, which fail to capture latent and complex information such as intersentential, indirect, and nested relationships. Furthermore, many GNN models are transductive by nature, limiting their ability to generalize to unseen nodes. To enhance inductive capabilities and incorporate semantic richness and heterogeneity, researchers have increasingly turned to knowledge graphs. However, most current GNN models do not integrate causal reasoning due to the inherent complexity of embedding causality within the model architecture.

To overcome these limitations, we proposed a novel GNN-based model trained on a cancer-specific knowledge graph related to ADRs. Our proposed model, the Causality and Proximity-based Relational Multihead Attention Model, aims to predict ADRs by identifying missing links within the knowledge graph, integrating both causality and proximity to improve predictive accuracy and robustness.

The knowledge graph we used for ADR prediction was built from a cancer-specific corpus of case reports [7], [6], in which cancer-related entities and their causal relationships were annotated in a triplet format, consisting of subjects, predicates, and objects. To enhance the model's ability to capture causal semantics and complex relational structures, including nested, indirect, and inter-sentential associations, we extended the triplet format to allow more granular relation extraction and categorization.

Subsequently, these extracted triplets were used to construct a knowledge graph that visually represents relationships, providing a foundation for training our ADR prediction model. In the biomedical field it is essential to consider relationships when training models, as ignoring them would render the model ineffective. The knowledge graph, composed of subject, predicate, and object elements, serves as a

crucial knowledge base for embedding semantic depth and heterogeneity into the model.

We fully integrated this knowledge graph into the GNN framework and introduced a novel causal proximity-aggregation function. This function effectively encodes both causality and proximity information, enabling the generation of more effective and robust embeddings. Crucially, it preserves both the semantic and structural integrity of the data while mitigating the problem of over-smoothing. The key contributions of this study are as follows:

- Integrating node-specific semantic properties into the encoding process while preserving the graph's topological structure.
- Implementing multi-hop indirect message-passing mechanisms to address the issue of over-smoothing in GNNs.
- Capturing latent relational information that is often overlooked in current research.
- Training models to recognize high-granularity causal relations within the data.
- Incorporating proximity considerations into the aggregation function to enhance model performance.
- Developing a precision-focused model for predicting cancer-specific ADRs.
- Advancing knowledge discovery in the relationships between cancer drugs and ADRs through specialized modeling techniques.

2. Related work

This section briefly reviews and summarizes the recent studies related to our work. Adverse drug reaction (ADR) prediction has been approached using various methodologies, encompassing statistical, machine learning (ML), deep learning (DL), graph-based, and similarity-based techniques.

In statistical-based approaches, researchers have developed drug-pair protein interaction profiles using data from the STITCH and TWOSIDES databases, employing the Laplacian-corrected estimator to predict drug-induced effects [28]. Another group of researchers modeled ADR-drug relationships with a three-layer hierarchical Bayesian model [2], utilizing Latent Dirichlet allocation to uncover biochemical mechanisms linking ADRs to drug structures [25]. Additional statistical methods include using proportional reporting ratios, reporting odds ratios, and empirical Bayes geometric mean algorithms to identify drug-associated adverse events [27]. Moreover, researchers collected data from seven databases and implemented methods such as Bayesian confidence propagation neural network, gamma Poisson shrinker, proportional reporting ratio, and reporting odds ratio to detect ADRs [21]. Extended likelihood ratio test methods based on Poisson models were used to identify ADR signals with disproportionately high reporting rates [29], while other studies leveraged Eudra Vigilance data [19], tree-based scan statistics [22], and disproportionate methods to detect unknown causal associations [13].

In the realm of ML/DL-based approaches, researchers focused on integrating data from DrugBank, DrugCentral, and TWOSIDES databases, proposing the HCNS-ADR machine-learning method to predict ADRs from combined medication [30]. Another team developed a structure-enhanced line-graph convolutional network to learn comprehensive representations of drug-disease pairs, transforming the task into a node classification problem using the SEAL architecture for link prediction [15]. A convolutional framework was also proposed to construct chemical fingerprint features and assess their associations with ADRs [8].

Graph-based approaches have seen significant advancements, with authors proposing a GNN model for adverse drug-event detection [10, 31]. A GNN model trained on clinical data uses SIDER database labels to predict side effect signals between drug-disease pairs [12]. Another innovative method, the contextualized graph embedding model (CGEM), captures cause-effect relations for ADR detection by combining contextualized embeddings, convolutional GNNs, and BertGCN for classification [9]. Additional graph-based techniques include analyzing non-directional heterogeneous health care data for triad prediction [16], combining deep learning with a biomedical tripartite network to predict drug-ADR associations [26], and infer new associations through drug-ADR network topological features [17]. An external link concept was also proposed to infer associations in a heterogeneous network by connecting drugs sharing common ADRs [14].

Similarity-based approaches involve predicting a drug's side effects by combining canonical correlation analysis with network-based diffusion [3], identifying significant correlations between chemical fragments and side effects for ADR prediction [20], and using a naive Bayesian model to infer drug-ADR associations based on known drug-protein and drug-ADR associations [24]. These diverse methodologies illustrate the multifaceted nature of ADR prediction, leveraging a broad spectrum of data sources and analytical techniques to enhance the safety and efficacy of pharmaceutical treatments.

3. Problem definition

We aim to develop a causal proximity-based multi-head attention relational graph neural network trained using causal knowledge graphs to predict adverse drug reactions specific to drugs used in the treatment of cancer. We propose a novel aggregation function to enhance message passing and mitigate over-smoothing. Over-smoothing is the phenomenon where the node representations across the graph become increasingly similar and lose their unique characteristics, which hinders the model's ability to distinguish between different nodes. Our approach aims to improve GNN performance by handling over-smoothing in heterogeneous graphs by adding a proximity vector. We also extend the role2vec algorithm [1] for initial embedding generation by incorporating the node's individual semantic properties and defining the role that the node plays in the knowledge graph. The proposed model is elucidated in Figure 1.

In Figure 1, we consider knowledge graph G and initially generate a five-hop neighborhood for each node, defining the set of its connected neighboring nodes up to five layers. Additionally, we introduce a proximity hop, which includes nodes that are not part of the five-hop neighborhood or are not directly connected but are necessary to capture complex and high-granularity relational information.

Following the hop generation, a feature matrix XN×P is generated in which p is the size of the feature and n is the number of nodes. Using this, we generate the initial node embeddings. To enhance representation learning we initialize proximity vectors for all nodes based on their proximity hop.

The goal is to teach an encoder who maps these initial embeddings to high-level relational embeddings using the proposed causal proximity-based aggregation function A(W, h). For each node, the initial embedding is transformed into a relational embedding, emphasizing the relations through which the node connects to both its hop-included neighbors and proximity nodes. To achieve this, we employ an attention mechanism that determines the importance of each neighboring node in relation to the type of connection it shares with the target node. Furthermore, to effectively incorporate multiple relational dependencies, we utilize a multi-head attention model, enabling the simultaneous capture of diverse relational aspects in the knowledge graph.

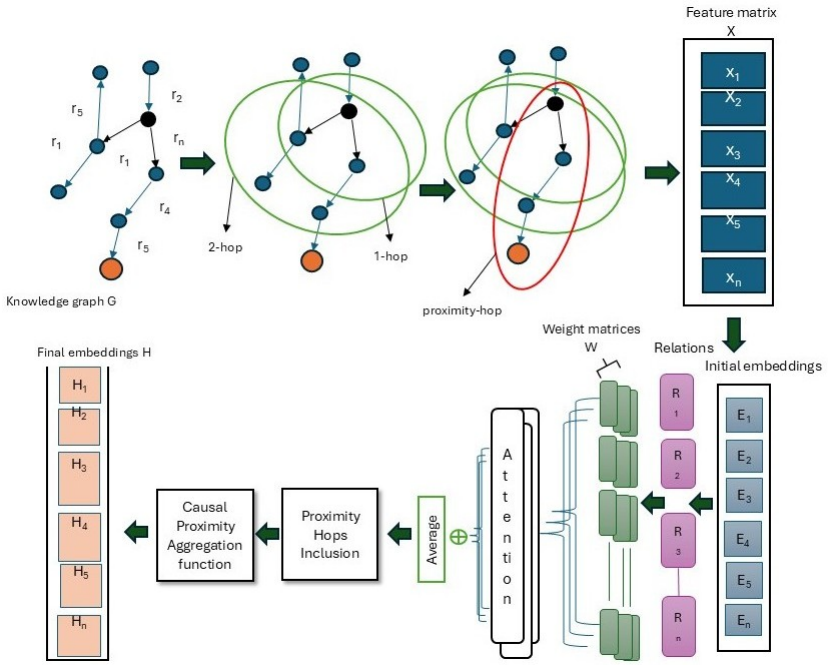

Figure 1. Proposed model

4. Proposed approach

This section delineates the proposed approach, structured into distinct steps. Conventionally, the knowledge graph inference pipeline comprises two stages: first, learning embeddings while maintaining local properties, and second, conducting missing link prediction. However, our method diverges from this norm by partitioning the process into three steps. Initially, we focus on learning initial embeddings while preserving the intrinsic semantic properties of each node, which is crucial for defining its role within the graph. Subsequently, we proceed to teach final embeddings, a process that entails incorporating global context and causality. Finally, we tackle the missing-link prediction task.

Causal relations significantly improve the reliability and interpretability of a model. Causal modeling helps differentiate genuine ADRs from effects caused by underlying conditions, co-medications, or patient demographics, reducing spurious associations. The model is also able to prioritize ADRs with a higher probability of occurrence, improving clinical relevance.

Integrating knowledge graphs with proximity vectors and graph neural networks (GNNs) serves as an effective means to expedite the analysis of clinical data. This fusion not only enhances the quality of the knowledge graph but also mitigates its inherent incompleteness. The algorithm for the proposed model is described in algorithm 1.

4.1. Triplet augmentation

4.1.1. Triplet generation

Initially,, 500 case reports specific to cancer-related adverse drug reactions (ADRs) were sourced from PubMed as raw data. Entities of interest were divided into categories: primary and secondary. Primary entities play the role of subject and object in a triplet. The primary entities are the nodes of a knowledge graph. Secondary entities elucidate the properties of a triplet element. Primary entities (shown in Table 1) are further categorized into two subcategories: higher-order entities and lower-order entities. Higher-order entities define the entities of interest in a corpus and the scope of the corpus, whereas lower-order entities are the support entities required to extract complete information. Subsequently, primary entities were extracted using metamap and variant a settling algorithm [6] [9]. Secondary entities were extracted using clinical Bert trained on the Maccrobat dataset [4]. Relations were extracted and normalized using automated linguistic and dependency-based algorithms [8] [8]. Extracted relations were normalized into 17 classes of relations that include 181 relationship types. Consider the given example in which "hypersensitivity syndrome" and "capecitabine" are the nodes of a knowledge graph, whereas "caused due to" is a relationship type under the ADR "causality relation" class described by the edge of a knowledge graph. Statistics extracted from entities are given in Table 5.

Table 1
High-Order Entity Types

Primary Higher-Order Entities	Explanation	Primary Lower-Order Entities	Explanation
Cancer type	Classification of cancer forms	Gene	A sequence of DNA that encodes for proteins
Drug	Chemical or biological substances used to treat cancer	Protein	A molecule composed of amino acids
Adverse drug reaction	An unintended, harmful response to a drug used in cancer treatment	Gene mutation	A change in the DNA sequence of a gene
Dysn	Entities which cannot be classified as an adverse drug reaction (ADR) or a syndrome	Cell type	Classification of cells based on their function

Entities, along with relationships, were then stored in a triplet format corpus. Considering the given example, the baseline triplet structure proves insufficient in capturing the semantic nuances inherent in complex sentences, as the latent and complex information is missing. For example, consider a sentence from a case report: "Mild hypersensitivity syndrome was caused due to capecitabine during cycle two treatment of metastasis breast cancer".

In this sentence, the extracted relation triplet was

["Hypersensitivity syndrome." "caused due to." "capecitabine"]

But in this triplet format most of the information is lost, such as hypersensitivity syndrome is mild, capecitabine was given during the breast cancer cycle-two treatment, and breast cancer was metastasis.

To address this limitation, we introduce a novel extended triplet structure (highlighted in bold). This structure builds upon the base triplet format by incorporating properties for each triplet component and the proximity links between them. By augmenting triplet elements with properties, we encapsulate detailed and vital information. Concurrently, the inclusion of proximity links establishes connectivity among proximate triplets, facilitating the extraction of comprehensive and meaningful relational data.

Proximity links serve as the foundation for proximity vectors in our proposed aggregation function. Meanwhile, the properties added to triplet elements serve as semantic features during the generation of initial node embeddings. These properties (property values) are the extracted lower-order entities, and property type is the entity type.

[([relation id, ["subject", "predicate", "object"]], proximity linkage), [["property value", "property type", triplet element]]

The introduced novel extended triplet format serves as a robust solution to circumvent this constraint. Here we show how the relational data extracted from Example 1 is significantly enriched through the utilization of our innovative triplet structure.

[(1, ["hypersensitivity syndrome", "caused due to", "capecitabine"]), [2, 3]], [mild, severity, subject]]

[(2, ["capecitabine", "during", "treatment"]), [3], [cycle 2, process, object]]

[(3, ["treatment", "of", "breast cancer"]), [Metastasis, characteristic, object]]

For instance, within Sentence 1, the individual triplets, each denoted by IDs 1, 2, and 3, may initially appear devoid of substantial significance. However, when considered collectively and augmented with incorporated properties and proximity links, they synergistically yield a profound, comprehensive, and contextually rich array of relational information.

4.1.2. Triplet integration

Following the generation of triplets from cancer-specific case reports, additional data in the form of triplets were integrated into existing case report triplets using DrugBank [23] [24]. Relevant information pertaining to drug-drug interactions, drug-gene interactions, drug-food interactions, drug-allele interactions, and drug-protein interactions was extracted from DrugBank and converted into triplet format. These final triplets, amalgamated from both the case reports and DrugBank data, were integrated to form a comprehensive corpus for model training. Data integrated from DrugBank are sorted into classes, which are further categorized into subtypes. To make sure that integration does not cause noise, the whole corpus was normalized using standardized medical vocabulary, and DrugBank is itself a standardized medical vocabulary. To avoid bias we have used negative sampling and adaptive attention weights. Detailed statistics pertaining to this final corpus are presented in Table 6.

4.2. Initial node embedding generation

The efficacy of graph machine learning tasks, such as link prediction, hinges on the acquisition of a useful feature representation for nodes within the graph. Conventionally, node embedding generation methods rooted in structural equivalence have dominated in the literature. However, these methods suffer from two primary drawbacks: First, they are inherently transductive, thereby impeding their generalization to unseen nodes. Second, there exists no standardized approach for seamlessly integrating node attributes into the network representation. Consequently, in these methods proximity between nodes fails to guarantee semantic similarity.

To circumvent these challenges, we propose an extension to the role2vec [1] [1] algorithm, designed to learn a function capable of capturing the semantic role and behavior of nodes within a graph. The basic idea of role2vec is to introduce the notion of attributed random walks that are not tied to structural similarity but instead to a

function that maps a node to a role in a graph. It uses random walks and skip-gram models for embedding generation.

The goal of the proposed algorithm is to generate the embedding of a node based on the role that ii plays in a graph. The role of a node is defined by its features, which are described in Table 2. The notion is that the nodes having similar roles must have similar embedding. It includes structural properties as well as semantic properties. The Role2vec framework integrates vertex mapping and attributed random walks.

Table 2
Node Semantic Features

Node's Semantic Features	Description
Entity_name	Name of the entity
Entity_class	Class/category of the entity
Entity_CUI	Concept Unique Identifier (CUI) of the entity
Entity_property	Properties or attributes of the entity
Entity_synonyms	Synonyms of the entity
Entity_normalized version	Normalized version of the entity

Consider the knowledge graph $G = (V, R, T)$.

V = set of entities, R = set of relations, T = set of triplets in extended format

$$V_N = \{v_1, v_2, v_3, \ldots, v_n\} \tag{1}$$

$$R_K = \{r_1, r_2, r_3, \ldots, r_k\} \tag{2}$$

$$t = (v_i, r_k, v_j; \mathcal{O}, \mathcal{P}) \quad \text{where } v_i, v_j \in V \text{ and } r_k \in R \tag{3}$$

where O is the element properties set and P is the proximity link set.

$$a_{ij}^{(r_k)} \in A_{N\times N} \tag{4}$$

$A_{N\times N}$ = relational adjacency matrix for G

$a_{ij}^{(r_k)}$ = existence of relation r_k between v_i and v_j

where $A_{N\times N}$ is the relational adjacency matrix for G and $a_{ij}^{(r_k)}$ and defines the existence of a relation type r_k between entities v_i and v_j.

Let X be the set of node attributes defining the semantic role of each node. Let $F_{N\times K}$ be the feature matrix for a set of entities having N entities and K features, and e_i^v denotes the initial embedding of a node v_i, generated by $\mathcal{H}(v_i)$ function using the extended role2vec algorithm [**?**]. Extended role2vec focuses on maximizing cosine similarity S that relates a node to the nodes having similar roles using the mapping function Υ.

$$\{x_1, x_2, x_3, \ldots, x_k\} \in X$$
$$\mathcal{H}(v_i) = x_1^i \circ x_2^i \circ x_3^i \ldots \circ x_k^i \quad \text{where } \circ \text{ denotes concatenation}$$
$$\mathcal{H}(v_i) \to e_i^v$$
$$E_v = \{e_{v1}, e_{v2}, e_{v3}, \ldots, e_{vn}\}$$

For v_i and $v_j \in V$, the two nodes having a similar semantic role; the goal should be:

$$\mathcal{H}(e_i^v \mid e_j^v) \to \max\left(S(e_i^v \mid e_j^v)\right) = \max\left(\frac{e_i^v \cdot e_j^v}{\|e_i^v\|\|e_j^v\|}\right) \tag{5}$$

4.3. Model training

Graph neural networks have been employed for training. A heterogeneous causal relational multi-head attention-graph neural model has been developed. The basic graph attention network is given by Equation 6:

$$h_u^{(L)} = \sigma\left(\sum_{u \in N(v)} \alpha_{uv} W^{(L)} h_u^{(L-1)}\right) \tag{6}$$

σ = nonlinearity function, u = neighbor node, v = current node, α_{uv} = attention weight.

4.3.1. Attention vector calculation

The attention vector α_{uv}, the result of the attention mechanism, defines how much attention, or weight, must be given to a neighbor u of node v. The applied attention mechanism Q computes the attention vector α_{uv} equation 7, indicating the importance of node u to node v. Q incorporates the weight matrix W, neighbor node embedding at layer L defined by $h_u^{(L)}$, and current node embedding at layer $L-1$, defined by $h_v^{(L-1)}$:

$$\alpha_{uv} = Q(W^{(L)} h_u^{(L-1)}, W^{(L)} h_v^{(L-1)}) = \frac{e^{\mathrm{ac}_{uv}}}{\sum_{k \in N(v)} e^{\mathrm{ac}_{vk}}} \tag{7}$$

Node embeddings (Equations 8 to 10) at layer L with multiple α_{uv} with different parameters are given by $h_{u,r}^{(L)}[n]$:

$$h_{u,r}^{(L)}[1] = \sum_{u \in N(v)} \alpha_{uv1} W_r^{(L)} h_u^{(L-1)} \quad \text{(v)} \tag{8}$$

$$h_{u,r}^{(L)}[2] = \sum_{u \in N(v)} \alpha_{uv2} W_r^{(L)} h_u^{(L-1)} \quad \text{(vi)} \tag{9}$$

$$h_{u,r}^{(L)}[n] = \sum_{u \in N(v)} \alpha_{uvn} W_r^{(L)} h_u^{(L-1)} \quad \text{(vii)} \tag{10}$$

4.3.2. Proposed aggregation function

After the initial embedding generation, we introduced a novel aggregation function tailored for message passing and final node generation. This pioneering aggregation function is specifically designed to integrate a proximity vector, a crucial addition to address the over-smoothing dilemma pervasive in graph neural networks (GNNs). Typically, GNN architectures limit the number of layers to 2-5 hops to mitigate over-smoothing. However, this restricted depth fails to capture the intricate semantics and contextual intricacies inherent in complex relations.

To overcome this limitation we propose the inclusion of a proximity vector, which aggregates all nodes interconnected and proximate to a given relation. These proximity relations, derived from the training set produced by our novel triplet format, enrich the model's understanding by incorporating contextual information from neighboring nodes.

In general, in the GNN framework, each layer L consists of a message function M and aggregation function A:

$$\mathcal{M}_u^{(L)} = \mathrm{mf}^{(L)}(h_u^{(L-1)}) = W^{(L)} h_u^{(L-1)} \tag{11}$$

$W^{(L)}$ = weight matrix for layer L. W is updated at each layer L. The proposed aggregation function (Equations 9 and 10) is defined by A:

$$A = H^{(L)} = \mathrm{ReLU}\left(\sum_{r\in R}\sum_{u\in N(v)} \frac{1}{c(v,r)} H_{u,r}^{\mathrm{MH}^{(L)}} + w_0^{(L)} h_v^{(L-1)} + \Delta \sum_{a\in Y} P_a\right) \tag{12}$$

For $v_i = Z_i = H_i^1 + H_i^2 + \cdots + H_i^L$

$H^{(L)}$ will be the updated embedding at layer L, and Z will be the final embedding. $c_{v,r} = |N_{vr}|$ where N_{vr} is the relational node degree. Δ is the normalization function. P, the proximity vector, is the summation of final embeddings of proximity triplets, and Y is the set of proximity triplets:

$$P = \mathrm{ReLU}\left(\sum_{r\in R}\sum_{b\in N(Y)} \frac{1}{c(a,r)} \alpha_{uv} w_r^{(L)} h_b^{(L-1)} + w_0^{(L)} h_a^{(L-1)}\right) \tag{13}$$

$M = \mathrm{mf}^{(L)}(h_u^{(L-1)}) = \mathrm{mf}^L(H_u^L, \theta_G)$, where θ_G is the shared network parameter. Each relation has L relational weight matrices w. For relation r, the relational weight matrices are $W_{r1}, W_{r2}, \ldots, W_{rL}$.

4.3.3. Positive sampling training

We trained the model using the proposed aggregation function on the triplet corpus. We set the number of hidden layers to two (k = 2). The final embedding for each node wa's encoded. Four available translational and deep-learning graph models were

trained on our cancer-specific triplet corpus. We compared our proposed causal proximity relational multi-head attention model (CPRMAM) with four available models: TranseE, DistMult, ComplexEx, and RAGAT. CPRMAM gave the best results, as the encoders are trained on complex and indirect relationships, which makes the process of knowledge discovery more efficient in generating accurate and precise information. For training, 132184 triplets in novel format were considered in training data.

4.3.4. Negative sampling training

Negative sampling is also important to reduce overfitting and makes the model capable of differentiating between true and false predictions. In knowledge-graph completion tasks, negative sampling is used to train models by providing incorrect (negative) examples along with correct (positive) ones. We opted for a 1:N negative sampling strategy [18][19]. 1: N negative sampling is a specific strategy in which, for each positive triplet (h, r, t) multiple negative triplets are generated by corrupting either the subject or object entity.

4.3.5. Loss computation and weight updatimg

We trained our CPRMAM model on two different loss functions: cross entropy loss and pairwise hinge loss. It was seen that the model performed better with the cross-entropy loss function (Equation 14). Then we applied gradient descent to update the weights:

$$\mathcal{L}_{\mathrm{CE}} = -\log\left(\mathrm{ReLU}(f_{r_k}(h_i, h_j))\right) - \log\left(1 - \mathrm{ReLU}(f_{r_k}(h_i, h_j))\right) \tag{14}$$

4.3.6. Selection of most suitable parameters

Efficient model training depends on the selection of correct parameters and avoiding overfitting and underfitting. To ensure that the model gives the best results, we used grid search to find the suitable embedding size: z = 128, learning rate lr = 0.001, the number of corruptions must be made for each triplet eta = 20, and the maximum epochs (500) to run.

4.4. Triplet scoring

After training the encoder part and generation of final embeddings, the decoder played an important role in scoring the sample triplets and generating nonexistent knowledge. We used four different scoring functions with each encoder. Table 3 shows the possible combinations of encoders and decoders, or we can say aggregation function and scoring functions. It was noticed that the proposed aggregation function, as encoder and ConvE as the decoder, or scoring function, gave the best results.

4.4.1. Corrupt triplet generation

To test the model against false positives, we generated corrupted triplets in test data using a filtered strategy (Equation 15). In the filtered strategy, initially, only the

element of the triplet will be corrupted by replacing that element with random entities and relations. Corruption parameter eta defines the total number of corruptions that must be generated for each triplet. It was found that the most suitable value for eta is 20. Practically generating corruption for every triplet is not feasible. Therefore, we corrupted triplets that contain the relations among the entities of interest: drug, ADR, cancer types, food, genes, proteins, and alleles:

$$\text{For } t = (v_i, r_k, v_j), \quad \text{create the corrupted triplet } t_c = (v_i, r_k, v_j^c) \tag{15}$$

4.4.2. Scoring of triplets

After generating corrupted triplets in the test data, every triplet would then be scored based on scoring functions or decoders. We used four available scoring functions: TransE, DistMult, ComplEx, and ConvE. The score indicates how likely two nodes are connected by a particular relation. Each scoring function mentioned has been used in a possible combination with the available algorithms and the proposed model. It was noticed that ConvE performed better, as shown in Table 3. We computed the score for triplet t using the function f_{r_k} (Equation 16), which took the relationally weighted final layer embeddings h as input:

$$f_{r_k}(h_i, h_j) = \text{ReLU}\left(\text{vec}\left(\text{ReLU}\left([H_i; e_r] \odot \mathcal{T}\right) W_r\right)\right) H_j \tag{16}$$

4.5. Ranking of triplets against corrupt data

For ranking and evaluation, we created test data containing 1,000 triplets along with their corrupt triplet version. Every triplet in the test data was ranked against its corrupted triplet variants based on the score generated. Our proposed model CPRMAM gave the best results when combined with the ConvE scoring function. We used the tied ranking strategy (Equation 17) for triplet rank generation:

$$\text{rank} = \text{COUNT}\left(\text{corruption score} \geq \text{hypothesis score}\right) + 1 \tag{17}$$

4.6. Model evaluation

In the absence of benchmark data, mean reciprocal rank (MRR) [5] and Hits@K [11] have been suggested and utilized by researchers for model evaluation. Our work also utilizes the following evaluation metrics: MRR, Hits@1, Hits@3, Hits@10, and Hits@100 (collectively denoted as Hits@K), as shown in Tables 3 and 4. These metrics measure how well the model ranks the correct ADR predictions. They are important for high-stakes domains like oncology, where incorrect predictions can have significant consequences. The constructed corpus is divided into training data and test data. Our model is evaluated on the test data using these metrics. The test data includes 1,000 triplets and their corrupted triplet version.

MRR (equation 18) measures the ranking quality of the correct triplet among the predicted scores. The mean reciprocal rank evaluates ranking quality by averaging

the reciprocal rank of the first correctly predicted triplet among the predicted triplets. It is expressed as:

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}(i)} \tag{18}$$

where Q represents the set of all queries and rank (i) denotes the position of the first correct triplet for the i-th query. A higher MRR value signifies that correct predictions appear closer to the top, which is crucial in clinical applications in which lower-ranked correct predictions may be overlooked.

Hits@K (equation 19) measures the proportion of test samples in which the correct entity appears in the top K predictions. It indicates the model's ability to make highly confident predictions. The Hits@K metric measures the percentage of test samples in which the correct entity appears within the top K-ranked predictions. It is defined as:

$$\text{Hits@K} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\mathbf{1}(\text{rank}(i) < K) \tag{19}$$

where 1(.) is an indicator function that returns 1 if the condition inside is met and 0 otherwise. A higher Hits@K value indicates that the model effectively ranks correct ADR predictions within the top K results, helping to ensure that critical drug interactions are not overlooked. In Table 3, the left side represents the encoders for final embedding generation, whereas the top side represents the decoders to score the triplets based on the translation method.

Table 3
Evaluation Metrics for Translational-Based Models

Decoder	Metrics	TransE	DistMult	ComplEx
TransE	MRR	0.236	0.284	0.297
	Hits@1	0.004	0.005	0.008
	Hits@3	0.087	0.471	0.071
	Hits@10	0.125	0.145	0.101
	Hits@100	0.184	0.247	0.185
DistMult	MRR	0.284	0.298	0.128
	Hits@1	0.003	0.012	0.010
	Hits@3	0.051	0.124	0.784
	Hits@10	0.150	0.145	0.121
	Hits@100	0.189	0.189	0.210
ComplEx	MRR	0.258	0.248	0.225
	Hits@1	0.005	0.007	0.052
	Hits@3	0.087	0.018	0.907
	Hits@10	1.089	0.105	0.101
	Hits@100	2.258	0.210	0.287

These encoders and decoders are used in combinations. For each possible combination of encoder and decoder, the model has been evaluated using metrics. Table 4 shows the metrics evaluation of the GNN-based model and the proposed model in combination with translational and convolutional decoders. The proposed model with the ConvE scoring function gave the best results. 2

Table 4
Evaluation Metrics for GNN-Based Models

Decoder	Metrics	RAGAT	CPRMAM without proximity vectors	CPRMAM
TransE	MRR	0.236	0.311	0.488
	Hits@1	0.244	0.221	0.338
	Hits@3	0.345	0.243	0.351
	Hits@10	0.410	0.398	0.425
	Hits@100	0.487	0.410	0.497
DistMult	MRR	0.477	0.324	0.458
	Hits@1	0.248	0.244	0.238
	Hits@3	0.245	0.267	0.321
	Hits@10	0.418	0.406	0.425
	Hits@100	0.489	0.444	0.597
ComplEx	MRR	0.469	0.451	0.558
	Hits@1	0.234	0.312	0.375
	Hits@3	0.354	0.356	0.381
	Hits@10	0.497	0.431	0.595
	Hits@100	0.357	0.324	0.699
ConvE	MRR	0.498	0.467	0.658
	Hits@1	0.301	0.320	0.538
	Hits@3	0.341	0.351	0.621
	Hits@10	0.398	0.543	0.695
	Hits@100	0.414	0.654	0.877

4.7. Knowledge discovery

Using link prediction, we try to predict all possible adverse drug reactions for all cancer-related drugs that were not mentioned earlier in the training text data by predicting the missing edges in the knowledge graph. Our model predicted 200 missing links that have been ranked based on the score generated by the scoring function. The mentioned triplets are generated through the knowledge discovery process and were not present earlier in the corpus.

('colon cancer', 'undergo', 'galantamine')
['Ipilimumab', 'adverse effects', 'fever']
['Ipilimumab', 'adverse effects', 'increased orthostatic hypotensive activities']

5. Data statistics

This section shows the statistics of entities and relations in a corpus. Figure 2 shows the count of entity categories, whereas Figure 3 shows the statistics of relations integrated from DrugBank. Table 5 shows the random corpus triplet division into training and testing sets. Table 6 shows the entities and relations total and unique count. Table 7 shows the DrugBank triplet classes.

Table 5
Corpus Triplet Statistics

Set	Count
Corpus triplet	133,684
Training set	132,184
Validation set	500
Testing set	1,000

Table 6
Corpus Entities and Relation Statistics

Element	Unique Count	Total Count
Corpus entities	55	3,756
Corpus relations	184	428,451

Table 7
DrugBank Triplet Classes Statistics

Triplet Class	Triplet Subtypes Count
Drug-drug interaction	73
Drug-food interaction	9
Drug-protein interaction	19
Drug-allele interaction	7
Drug-gene interaction	13

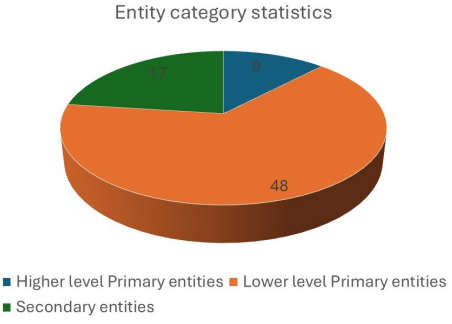

Figure 2. Entity category statistics

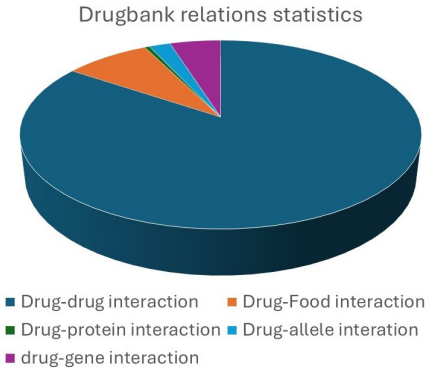

Figure 3. Drug-bank relation statistics

Algorithm 1 CPIRMAM Model

Input: Corpus triplets
Output: Missing links in knowledge graph

Initialize feature set for a node v
Generate feature matrix: $F_{N\times P}$ for V, $X_p = \{x_1, x_2, x_3, \ldots, x_p\}$
Calculate initial node embeddings e for each node v, $E_d = \{e_1, e_2, e_3, \ldots, e_n\}$
Set initial relational weight matrix W
Initialize proximity nodes and embedding set Y
for each $r_k \in R$ **do**
    for each $v_n \in V$ **do**
        Initialize causal proximity aggregation function
        Calculate multi-head attention vector for different heads
        Calculate final embedding for v based on steps 7 and 8
    end for
end for
Initialize decoder (scoring function)
Initialize final embeddings for V
Generate negative embeddings for V using negative sampling
Compute triplet loss and update weight W
Train relational model
for each $t \in T$ **do**
    Generate corrupted triple set T_c
    Initialize final embeddings for V
    Initialize scoring function $f_{r_k}(h_i, h_j)$
    Calculate scores for testing triplet t and T_c
    Generate rank for t against T_c
end for
Evaluate model
Predict missing links

6. Comparison and validation

The proposed model was compared with the state-of-the-art model RAGAT. It was noticed that the proposed model performed better. A comparison can be seen in Table 8 and Figure 4. Links generated through the proposed model that are not present in the corpus are validated by domain experts and medical practitioners.

Table 8
Model Comparison Table

Model	MRR	Hits@1	Hits@3	Hits@10	Hits@100
RAGAT	0.498	0.301	0.341	0.398	0.414
CPRMAM	0.658	0.538	0.621	0.695	0.877

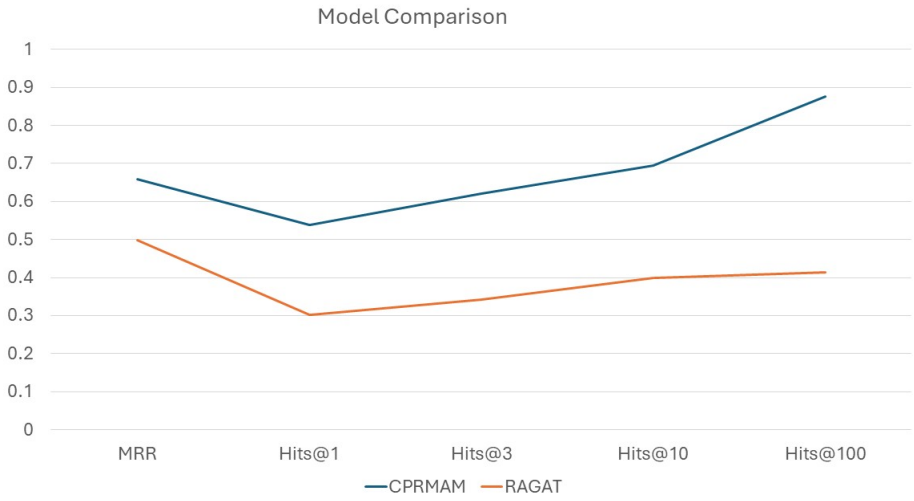

Figure 4. Comparison graph

Table 9 presents a comparative analysis of adverse drug reactions (ADRs) identified by CPRMAM and the baseline models (TransE, DistMult, and ComplEx). We observed that while all models capture some common ADRs, only CPRMAM provides additional medical context such as body organ, age-related factors, and dose dependency, which are crucial for clinical applications.

For instance, baseline models detect "peeling" as an ADR of 5-Fluorouracil, but only CPRMAM specifies "peeling at palms and soles," making it more clinically interpretable. Similarly, age-related nausea and constipation (increased in patients aged 65+) and dose-dependent seizures are uniquely captured by the proposed model.

These enhancements improve ADR specificity and real-world applicability in oncology pharma-covigilance. This demonstrates that CPRMAM surpasses baseline models by incorporating richer semantic and contextual information, making it a more reliable model for ADR prediction in clinical settings.

Table 9
Drug and Adverse Drug Event (ADE) Classification

Drug	Adverse Drug Event (ADE)	CPRMAM	TransE	DistMult	ComplEx	Clinical relevance
Erlotinib	Hepatorenal syndrome	No	No	No	No	Rare ADR
	Internal bleeding	No	No	No	No	Severe ADR
5-Fluorouracil	Swelling	No	No	No	No	Common ADR
	Palmar-plantar erythrodysesthesia	No	No	No	No	Drug-specific ADR
	Redness	No	No	No	No	Common ADR
	Peeling (palms and soles)	Yes	No	No	No	Location-enhanced ADR (unique to CPRMAM)
	Blisters (skin)	Yes	No	No	No	Location-enhanced ADR (unique to CPRMAM)
Cabazitaxel	Nausea (increased when 65+ years)	No	No	No	No	Age-related ADR
	Constipation (increased in 65+ years)	No	No	No	No	Age-related ADR
Prednisone	Weight gain (belly)	Yes	No	No	No	Location-enhanced ADR (unique to CPRMAM)
Doxorubicin	Pain (belly)	Yes	No	No	No	Location-enhanced ADR (unique to CPRMAM)
Fludarabine	Seizures (with high doses)	No	No	No	No	Dose-dependent ADR
	Stroke	No	No	No	No	Severe ADR
	Heart attack	No	No	No	No	Severe ADR
	Blindness	No	No	No	No	Rare ADR

7. Result

The proposed model CPRMAM overcomes the existing limitations in literature and performs better than the two existing models. Different GNN algorithms were implemented in combination with different scoring functions. We applied different scoring function combinations and found that the proposed aggregation function with the ConvE scoring function gave the best results in the evaluation. Our model achieved 0.658 MRR, 0.538 Hits@1, 0.621 hits@3, 0.695 hits@10, and 0.877 Hits@100.

8. Conclusion

In this study we have addressed the critical need for an automated adverse drug reaction (ADR) prediction model specifically tailored for cancer treatment. The proposed Causality and Proximity-based Multihead Attention Relational Model (CPRMAM) incorporates causal and semantic proximity information into graph neural networks (GNNs), overcoming several limitations of existing models, such as over-smoothing and equal neighbor weighting. By incorporating a novel extended triplet format and a causal proximity aggregation function, our model captures complex, indirect, and nested relational information that traditional models overlook. This approach enhances the quality and completeness of knowledge graphs, allowing for more robust embeddings and improved message passing in GNNs. Furthermore, the inductive nature of CPRMAM allows it to generalize to unseen nodes, a significant improvement over the transductive nature of current models. Evaluation of CPRMAM demonstrated superior performance compared to state-of-the-art models, achieving notable metrics, such as 0.658 MRR and 0.877 Hits@100, confirming its effectiveness in predicting ADRs and completing knowledge graphs. Our model not only advances ADR prediction in the biomedical domain, it also facilitates efficient knowledge discovery. Future work will focus on further refining the model, exploring its application to other biomedical domains and integrating additional data sources to enhance its predictive capabilities and generalizability.

Acknowledgements

We are truly grateful to Dr. Neera Samar, professor of general medicine, RNT Medical College and Hospital, Udaipur, for her invaluable expertise and validation of our findings. Her insightful guidance and expert validation have significantly contributed to the rigor and relevance of this research. Finally, we thank the reviewers for their constructive feedback and suggestions, which have greatly enhanced the quality of this manuscript.

References

[1] Ahmed N.K., Rossi R., Lee J.B., Willke T.L., Zhou R., Kong X., Eldardiry H.: Learning Role-based Graph Embeddings, *arXiv preprint arXiv:180202896*, 2018. https://arxiv.org/abs/1802.02896.

[2] Atias N., Sharan R.: An Algorithmic Framework for Predicting Side Effects of Drugs, *Journal of Computational Biology*, vol. 18(3), pp. 207–218, 2011. doi: 10.1089/cmb.2010.0255.

[3] Bate A., Lindquist M., Edwards I., Olsson S., Orre R., Lansner A., DeFreitas: A Bayesian Neural Network Method for Adverse Drug Reaction Signal Generation, *European Journal of Clinical Pharmacology*, vol. 54, pp. 315–321, 1998. doi: 10.1007/s002280050466.

[4] Caufield J.H.: MACCROBAT, Dataset, 2019. doi: 10.6084 / m9.figshare.9764942.v2.

[5] Craswell N.: Mean Reciprocal Rank. In: L. Liu, M.T. Özsu (eds.), *Encyclopedia of Database Systems*, Springer, Boston, MA, 2009. doi: 10.1007/978-0-387-39940-9_488.

[6] Dev S., Sharan A.: Annotated and Normalized Causal Relation Extraction Corpus for Improving Health Informatics. In: *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pp. 417–422, NLP Association of India (NLPAI), 2023.

[7] Dev S., Sharan A.: Automatic Construction of Named Entity Corpus for Adverse Drug Reaction Prediction. In: A. Bhattacharya, S. Dutta, P. Dutta, V. Piuri (eds.), *Innovations in Data Analytics. ICIDA 2022*, Advances in Intelligent Systems and Computing, vol. 1442, Springer, Singapore, 2023. doi: 10.1007/978-981-99-0550-8_20.

[8] Dey S., Luo H., Fokoue A., Hu J., Zhang P.: Predicting Adverse Drug Reactions through Interpretable Deep Learning Framework, *BMC Bioinformatics*, vol. 19(Suppl 21), p. 476, 2018. doi: 10.1186/s12859-018-2544-0.

[9] Gao Y., Ji S., Zhang T., Tiwari P., Marttinen P.: Contextualized Graph Embeddings for Adverse Drug Event Detection. In: M. Amini, S. Canu, A. Fischer, T. Guns, P. Kralj Novak, G. Tsoumakas (eds.), *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2022*, Lecture Notes in Computer Science, vol. 13714, Springer, Cham, 2023. doi: 10.1007/978-3-031-26390-3_35.

[10] Gao Y., Zhang X., Sun Z., Chandak P., Bu J., Wang H.: Precision Adverse Drug Reactions Prediction with Heterogeneous Graph Neural Network, *Advanced Science*, vol. 12(4), p. e2404671, 2024. doi: 10.1002/advs.202404671. Epub ahead of print.

[11] Khan M., Mello G., Habib L., Engelstad P., Yazidi A.: HITS-based Propagation Paradigm for Graph Neural Networks, *ACM Transactions on Knowledge Discovery from Data*, vol. 18(4), 2024. doi: 10.1145/3638779.

[12] Kwak H., Lee M., Yoon S., Chang J., Park S., Jung K.: Drug-Disease Graph: Predicting Adverse Drug Reaction Signals via Graph Neural Network with Clinical Data. In: *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, vol. 12085, pp. 633–644, 2020. doi: 10.1007/978-3-030-47436-2_48.

[13] Lerch M., Nowicki P., Manlik K., Wirsching G.: Statistical Signal Detection as a Routine Pharmacovigilance Practice: Effects of Periodicity and Resignalling Criteria on Quality and Workload, *Drug Safety*, vol. 38, pp. 1219–1231, 2015. doi: 10.1007/s40264-015-0345-1.

[14] Lin J., Kuang Q., Li Y., Zhang Y., Sun J., Ding Z., Li M.: Prediction of Adverse Drug Reactions by a Network Based External Link Prediction Method, *Analytical Chemistry*, vol. 5(21), p. 6120, 2013. doi: 10.1039/c3ay41290c.

[15] Liu B.M., Gao Y.L., Li F., Zheng C.H., Liu J.X.: SLGCN: Structure-enhanced Line Graph Convolutional Network for Predicting Drug–Disease Associations, *Knowledge-Based Systems*, vol. 283, p. 111187, 2023. doi: 10.1016/j.knosys.2023.111187.

[16] Liu R., AbdulHameed M., Kumar K., Yu X., Wallqvist A., Reifman J.: Data-driven Prediction of Adverse Drug Reactions Induced by Drug-Drug Interactions, *BMC Pharmacology and Toxicology*, vol. 18(1), p. 44, 2017. doi: 10.1186/s40360-017-0153-6.

[17] Manzi S., Reis B.: Predicting Adverse Drug Events Using Pharmacological Network Models, *Science Translational Medicine*, vol. 3(114), p. 114ra127, 2011. doi: 10.1126/scitranslmed.3002073.

[18] Miao R., Yang Y., Ma Y., Juan X., Xue H., Tang J., Wang X.: Negative Samples Selecting Strategy for Graph Contrastive Learning, *Information Sciences*, vol. 613, pp. 667–681, 2022. doi: 10.1016/j.ins.2022.09.024.

[19] Monaco L., Melis M., Biagi C., et al.: Signal Detection Activity on EudraVigilance Data: Analysis of the Procedure and Findings from an Italian Regional Centre for Pharmacovigilance, *Expert Opinion on Drug Safety*, vol. 16, pp. 271–275, 2017. doi: 10.1080/14740338.2017.1284200.

[20] Pauwels E., Stoven V., Yamanishi Y.: Predicting Drug Side-Effect Profiles: A Chemical Fragment-Based Approach, *BMC Bioinformatics*, vol. 12(1), p. 169, 2011. doi: 10.1186/1471-2105-12-169.

[21] Schuemie M.J., Ryan P.B., DuMouchel W., Suchard M.A., Madigan D.: Using Electronic Health Care Records for Drug Safety Signal Detection: A Comparative Evaluation of Statistical Methods, *Medical Care*, vol. 50, pp. 890–897, 2012. doi: 10.1097/mlr.0b013e31825f63bf.

[22] Wang S.V., Maro J.C., Baro E., et al.: Data Mining for Adverse Drug Events with a Propensity Score-Matched Tree-Based Scan Statistic, *Epidemiology*, vol. 29, pp. 895–903, 2018. doi: 10.1097/ede.0000000000000907.

[23] Wishart D.S., Knox C., Guo A.C., Cheng D., Shrivastava S., Tzur D., Gautam B., Hassanali M.: DrugBank: A Knowledgebase for Drugs, Drug Actions and Drug Targets, *Nucleic Acids Research*, vol. 36(Database issue), pp. D901–D906, 2008. doi: 10.1093/nar/gkm958.

[24] Xiang Y.P., Liu K., Cheng X.Y., Cheng C., Gong F., Pan J.B., Ji Z.L.: Rapid Assessment of Adverse Drug Reactions by Statistical Solution of Gene Association Network, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12(4), pp. 844–850, 2015. doi: 10.1109/tcbb.2014.2338292.

[25] Xiao C., Zhang P., Chaowalitwongse W.A., Hu J., Wang F.: Adverse Drug Reaction Prediction with Symbolic Latent Dirichlet Allocation. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17)*, pp. 1590–1596, AAAI Press, 2017. doi: 10.1609/aaai.v31i1.10717.

[26] Xue R., Liao J., Shao X., Han K., Long J., Shao L., Ai N., Fan X.: Prediction of Adverse Drug Reactions by Combining Biomedical Tripartite Network and Graph Representation Model, *Chemical Research in Toxicology*, vol. 33(1), pp. 202–210, 2020. doi: 10.1021/acs.chemrestox.9b00238.

[27] Yildirim P., Majnarić L., Ekmekci O., Holzinger A.: Knowledge Discovery of Drug Data on the Example of Adverse Reaction Prediction, *BMC Bioinformatics*, vol. 15 Suppl 6, p. S7, 2014. doi: 10.1186/1471-2105-15-S6-S7.

[28] Zhang F., Sun B., Diao X., Zhao W., Shu T.: Prediction of Adverse Drug Reactions Based on Knowledge Graph Embedding, *BMC Medical Informatics and Decision Making*, vol. 21(1), p. 38, 2021. doi: 10.1186/s12911-021-01402-3.

[29] Zhao Y., Yi M., Tiwari R.C.: Extended Likelihood Ratio Test-Based Methods for Signal Detection in a Drug Class with Application to FDA's Adverse Event Reporting System Database, *Statistical Methods in Medical Research*, vol. 27, pp. 876–890, 2018.

[30] Zheng Y., Peng H., Zhang X., et al.: Predicting Adverse Drug Reactions of Combined Medication from Heterogeneous Pharmacologic Databases, *BMC Bioinformatics*, vol. 19(Suppl 19), p. 517, 2018. doi: 10.1186/s12859-018-2520-8.

[31] Zhou F., Khushi M., Brett J., Uddin S.: Graph Neural Network-Based Subgraph Analysis for Predicting Adverse Drug Events, *Computers in Biology and Medicine*, vol. 183, p. 109282, 2024. doi: 10.1016/j.compbiomed.2024.109282.

Affiliations

Samridhi Dev
Jawaharlal Nehru University, New Delhi 11067,
textttsamrid21Ω_scs@jnu.ac.in

Aditi Sharan
Jawaharlal Nehru University, New Delhi 11067, aditisharan@mail.jnu.ac.in

Received: 13.01.2025
Revised: 11.02.2025
Accepted: 15.08.2025