Better Negatives, Better Predictions: Negative Sample Selection Strategies for Enhancing Biomedical KG Edge Classification

Emanuele Cavalleri*
Miad Alavinezhad*
Marco Mesiti
Dario Malchiodi
emanuele.cavalleri@unimi.it
miad.alavinezhad@studenti.unimi.it
marco.mesiti@unimi.it
dario.malchiodi@unimi.it
Department of Computer Science, University of Milano
Milano, Italy

Abstract

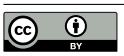
The selection of negative samples plays a crucial role in a wide range of machine learning algorithms and is particularly critical in edge classification tasks, where this choice has a direct impact on predictive performance. In this paper, we propose a set of strategies for generating negative edges in large, heterogeneous biomedical knowledge graphs, tailored to different link prediction scenarios. In these graphs, the absence of an observed edge does not necessarily indicate the absence of a relationship; instead, it may simply reflect missing or undiscovered knowledge. Leveraging latent-space graph embeddings, we analyze the impact of different negative sample selection strategies that account for both node types and edge semantics. Our initial experiments on two biomedical knowledge graphs demonstrate substantial improvements in classification performance, independent of the underlying predictive model, highlighting the robustness and effectiveness of the proposed approach. Results show that our strategies for generating negative edges in a knowledge graph outperform random negative sampling, yielding statistically significant improvements in balanced accuracy. Code and data for reproducing experiments are available at https://github.com/SLIMlaboratory/glow26 and https://zenodo.org/records/18074722.

Keywords

Negative edge selection, graph representation learning, knowledge graphs, link prediction, edge classification

ACM Reference Format:
Emanuele Cavalleri, Miad Alavinezhad, Marco Mesiti, and Dario Malchiodi. 2026. Better Negatives, Better Predictions: Negative Sample Selection Strategies for Enhancing Biomedical KG Edge Classification. In *Companion Proceedings of the ACM Web Conference 2026 (WWW Companion '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3774905.3794655

*Both authors contributed equally to this research.



WWW Companion '26, Dubai, United Arab Emirates

ACM ISBN 979-8-4007-2308-7/2026/04
https://doi.org/10.1145/3774905.3794655

1 Introduction

Knowledge graphs (KGs) [21] are increasingly adopted as a natural means for integrating heterogeneous information originating from diverse domains and application scenarios, including, but not limited to, economics [37], financial analysis [22], and biomedical research [10]. KGs provide a natural formalism to model real-world entities and the various types of relationships that can exist between them. Furthermore, when coupled with an ontological description, they facilitate interoperability across heterogeneous systems and support the development of a broad spectrum of AI-based analysis and processing. This processing commonly relies on supervised Machine Learning (ML) methods that exploit the KG topology. In settings involving *binary* predictions, a supervised ML algorithm requires as input a dataset containing both *positive* and *negative examples*, representing the two possible prediction outcomes. This generally leads to a fundamental problem when working with KGs, especially biomedical ones, because a marked asymmetry emerges: positive instances are straightforward to identify, whereas negative instances are rarely available. For example, an edge between two nodes representing a gene and a pathology encodes established knowledge that the former contributes to influencing the latter. However, the absence of such an edge does not necessarily indicate the opposite relationship; more plausibly, it reflects that no scientific study has yet investigated this specific connection.

Negative sample selection strategies are often applied to KGs to synthesize negative examples to be fed to the ML algorithm being used, typically by constructing nodes, edges, or labels that are not observed in the graph while respecting its structural and semantic constraints. Such strategies aim to approximate the set of false or implausible facts by perturbing existing positive triples (e.g., through entity or relation replacement—under assumptions such as the closed- or partial-closed-world hypothesis [28, 31]). The resulting negative instances are intended to provide a meaningful learning signal, mitigating the absence of explicit negative information in KGs and enabling supervised or semi-supervised models to be effectively trained. Although these types of strategies have been thoroughly examined with respect to how they are incorporated into graph representations [25], their explicit influence on the performance of downstream ML algorithms has received comparatively limited attention in the existing literature.

To address this gap, this paper presents a systematic empirical analysis of how three distinct negative-sample selection strategies affect the task of link prediction, conducted on two biomedical KGs. Through a series of controlled experiments, we evaluate and compare the performances of these strategies. Our findings indicate that topology-aware strategies for generating negative edges outperform random sampling, yielding statistically significant improvements across both binary classifiers and one-class models.

The remainder of this article is structured as follows: Sect. 2 provides a brief overview of the literature on KG representation and introduces the main ML methods that form the basis of our study, while Sect. 3 introduces key concepts related to KGs that are used in our approach. Sect. 4 describes the strategies for the selection of negative samples that we propose. The construction of the datasets and the design of the experimental setup are presented in Sect. 5, and the results of our experiments are shown and discussed in Sect. 6. Some closing remarks end the paper.

2 Related Work

2.1 Graph Representation Learning

Graph Representation Learning (GRL) is a paradigm that aims to transform the complex, non-Euclidean structure of graphs into low-dimensional vector spaces while preserving their essential structural and semantic properties [38]. GRL techniques learn vector representations (named *embeddings*), "living" in continuous spaces, that capture relationships such as proximity, community structure, and node/edge attributes. These embeddings enable the application of standard ML algorithms for downstream tasks (e.g., link prediction, node classification, novelty detection, and clustering).

A prominent category of GRL approaches is based on Random Walks (RWs) and extends the "distributional hypothesis" from linguistics to the graph domain. These methods perform RWs from each node to gather contextual information and train word2vec-style [27] models to learn embeddings. Popular examples include DeepWalk [30] and node2vec [17], which mainly differ in their walk strategies: DeepWalk uses simple first-order random walks, while node2vec introduces biased second-order walks for greater flexibility. Another group of GRL methods focuses on relation-learning models [2, 36], designed primarily for heterogeneous graphs to jointly learn entity and relation embeddings. These models, widely adopted for the representation of KGs, employ contrastive learning to assign high scores to true edges while penalizing "corrupted" ones. TransE [2] is one of the most widely adopted relation-learning approaches, known for its efficiency and strong empirical performance. It represents each relation as a translation operation from the subject to the object in the embedding space.

By bridging the gap between symbolic graph data and continuous ML models, GRL provides a powerful foundation for analyzing and reasoning over complex graph-structured information.

2.2 Classification Models

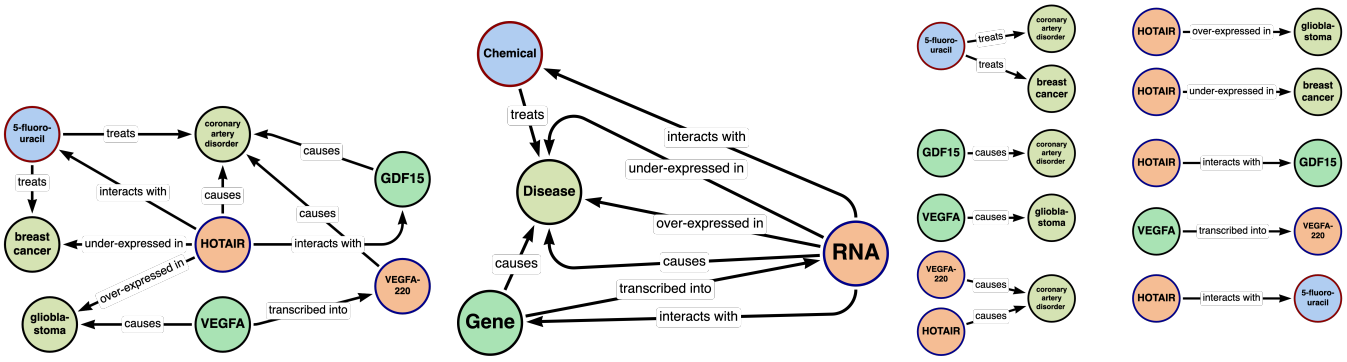
We focus on binary classification induced by a relation-specific partitioning of the KG. Specifically, for each target relation type, the object to be classified is a candidate link represented through its embedding, and the prediction is either *true* or *false*, corresponding to the acceptance or rejection of the candidate as a valid instance of that relation. For example, in a biomedical KG, this corresponds to predicting whether a given gene is associated with a specific disease. Within this setting, we consider two different approaches. The first relies on classical ML models specifically designed for binary classification. These models are trained on a set of candidate links, each associated with a boolean label expressing the ground truth about its membership to the edge set of the KG under study. The second approach is based on one-class classification (OCC), which aims at learning a single target class while treating all other instances as belonging to an undefined or outlier category. OCC methods are trained exclusively using examples from the target class, which makes them particularly suited for tasks such as the detection of anomalies (e.g., fraud detection in financial transactions, fault diagnosis in industrial equipment [34], or network intrusions) or the discovery of novelties (e.g., new species in ecological data [12] or emerging topics in text streams [19]). In our context, their use is motivated by the strong imbalance between links and non-links in a KG: valid links are vastly outnumbered by non-links. Under this assumption, embeddings of the latter can be regarded as anomalies w.r.t. the distribution induced by embeddings of existing links. Once trained, an OCC model can be naturally turned into a binary classifier by interpreting "normal" and "anomalous" outcomes as prediction of links and non-links, respectively.

In our experiments, we consider two classical binary classification models (Decision Trees and Random Forests) and two widely used OCC approaches (One-Class Support Vector Machines and Isolation Forests), which are briefly described in the following.

Decision Trees (DTs). A DT implements a hierarchy of decision rules organized in a tree structure and learned from data [4]. Each non-terminal node is associated with a condition on the data attribute value, while leaves are labeled with one of the available classes, which corresponds to the prediction produced by the model. Accordingly, the path from the root to a leaf of the tree provides a symbolic explanation of the classification process for a given object. A DT is learned by recursively partitioning the training set into increasingly homogeneous subsets. At each step, the learning algorithm selects a feature and a split that best separates the data according to a heterogeneity measure (e.g., the Gini index), with the goal of making the target values within each subset as uniform as possible. This process continues until a stopping condition is met, for instance based on tree depth or leaf impurity.

Random Forests (RFs). A RF is an ensemble of DTs, each learned from a subsample of the original training set [3]. Subsampling typically involves selecting subsets of both training instances and attributes, with the aim of promoting variety among the trees, each specializing in a particular region of the input space. Queries to a RF are answered by suitably aggregating the prediction of the various trees, for instance through majority vote.

One-class Support Vector Machines (OCSVMs). OCSVMs, first formalized in [33], extend the principles of support vector machines (SVMs [13]) to the single-class setting by constructing a decision boundary in a feature space (induced by a nonlinear transformation of the original data) that maximally separates the target data from the origin. This is achieved by solving a constrained optimization problem that also considers a controlled number of soft margin



(a) Knowledge Graph G (b) Schema G_S (c) Partition of G by schema edges

Figure 1: Example of partitioning a KG G according to its schema G_S.

violations. Once this problem has been solved, a *decision boundary* is established so that points falling outside this boundary are deemed outliers or novelties [1, 26]. Variants such as SVDD [35] and distributed OCSVM [9] further adapt this model for scalability and dynamic environments.

Isolation Forests (IFs). An IF is an ensemble-based approach to anomaly detection that relies on the idea that anomalous instances are rare and differ significantly from the majority of the data, making them easier to isolate through random partitioning [24]. The method constructs a collection of binary trees, known as isolation trees, each of which is built by recursively splitting the data using randomly selected attributes and randomly chosen split values. Unlike DTs, these trees are grown without reference to class labels and without optimizing any impurity or information-based criterion. During this process, instances that lie in sparse regions of the input space or exhibit unusual attribute values tend to be separated from the rest of the data after a small number of splits, while normal instances require more partitions to be isolated. As a result, anomalies correspond to shorter paths from the root to a leaf in the isolation trees. For a given instance, an anomaly score is obtained by aggregating and normalizing the path lengths across all trees in the forest, with shorter average path lengths indicating a higher degree of abnormality. Binary classification is ensured by thresholding this abnormality degree.

2.3 Impact of Negative Edges on Graph ML

The impact of negative edge selection in graph ML tasks has been explored along two main directions. The first concerns the selection of negative edges to improve graph embeddings. In this setting, negative edge selection is tightly integrated into the embedding learning process and used to optimize the loss of a GRL model, enforcing separation between the embeddings of selected negative edges and those of graph edges in the latent space. Proposed approaches exploit influence functions [25], degree-aware and other topology-driven strategies [39], as well as corruption-based, dynamic, adversarial, and auxiliary-entity sampling methods [5, 23].

A second direction, instead, leverages negative edge selection to enhance downstream classifiers. In this scenario, rather than aiming at shaping the latent space, negative edge selection is used to improve subsequent classification performance. To the best of our knowledge, the only work following this paradigm for link prediction is [8], which investigates a degree-aware negative sampling strategy on a graph of approximately 255 k edges with few relation types. Homogeneous GRL methods are used for generating embeddings and a perceptron is employed as classifier, using uniform negative sampling as a baseline. Our work builds upon this latter paradigm by taking into account the different kinds of predicates (i.e., relation types) in a KG and by analyzing the impact of different negative sample selection strategies on a heterogeneous link prediction task.

3 Knowledge Graphs

The structure and content of a KG can be described by means of a schema. The schema enables the representation of the types of entities available in a given domain and the types of relationships that can exist among them. In some domains (like the biomedical one), the description can be exploited as a constraint, while in others it can be exploited simply as a data guide for the information to be included in the KG.

Definition 1. *(*KG Schema*). Let N_T be the set of node types of interest in a given domain, E_T the set of edge types (predicates), and ϕ the function that assigns to each node its type. A* Knowledge Graph Schema *G_S is a graph $\langle N_S, E_S, N_T, E_T, \phi \rangle$, in which the sets of nodes is N_S and $E_S = \{(s, p, t) \mid s, t \in N_S, \phi(s), \phi(t) \in N_T, p \in E_T\}$.*

In a schema, there is 1:1 correspondence between the graph nodes and their types. Therefore, schema nodes can be represented by means of their types. In the following, each triple $(s, p, t) \in E_S$ is represented through the triple $(\phi(s), p, \phi(t))$ and denoted *schema fact*. Moreover, Facts(G_S) denotes all possible facts present in G_S.

The schema can be used as a footprint to generate KGs.

Definition 2. *(*KG*). Let $G_S = \langle N_S, E_S, N_T, E_T, \phi \rangle$ be a KG Schema. A* Knowledge Graph *G made according to G_S is a pair $\langle N, E \rangle$, where:*

- *N is the set of nodes (entities);*
- *$E \subseteq N \times E_T \times N = \{(s, p, t) \mid \phi(s), \phi(t) \in N_T, p \in E_T\}$ is the set of edges representing typed relationships between nodes.*

Given a KG G, Nodes(G) and Edges(G) denote the corresponding sets of nodes and edges. Moreover, $\text{sp}_G(s, t)$ is the length of the

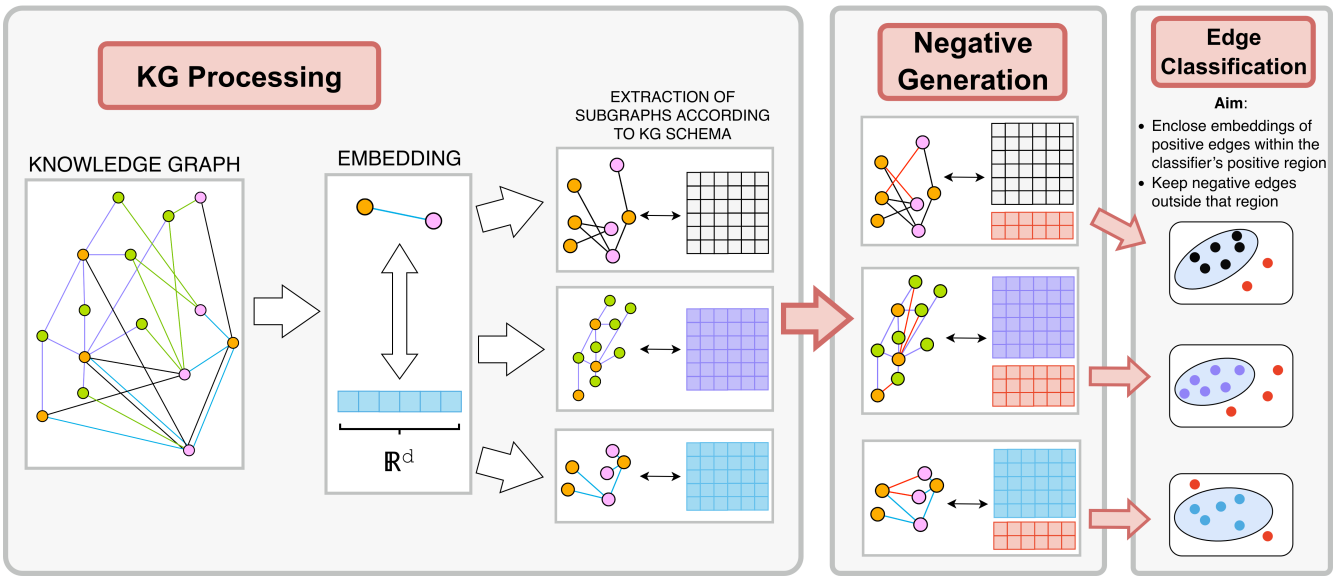


Figure 2: Pipeline for edge classification in a KG. The approach involves (*i*) KG embedding and homogeneous subgraph extraction, (*ii*) negative edge generation, and (*iii*) classification training.

minimal path between the source s and the target t ($\text{sp}_G(s,t) = \infty$ when no path exists), and $\deg_G(n)$ is the number of edges incident to n (i.e., the edges in which n appears either as source or target).

Starting from a schema fact $f = (s_T, p, t_T)$ of a schema G_S, we introduce the *induced subgraph* G_f. It corresponds to the subgraph of G whose nodes are of type s_T and t_T and whose edges are labeled by p. The induced subgraph might contain nodes of two types, but its edges are homogeneous. The induced subgraphs of a graph G constitute a partition of G (i.e., $\cup_{f \in \text{Facts}(G_S)} \text{Edges}(G_f) = E$ and $\cap_{f \in \text{Facts}(G_S)} \text{Edges}(G_f) = \emptyset$).

Example 1. *Figure 1a shows an example of a KG G describing relationships among RNA molecules, genes, diseases, and chemicals. Multiple types of relations can exist between entities of the same type according to the schema G_S in Figure 1b. Finally, Figure 1c illustrates the partitions of G obtained through the schema facts* $\text{Facts}(E_S)$.

A KG G can be embedded into a latent space by means of one of the approaches described in Sect. 2.1. The use of a heterogeneous method (like TransE) allows to consider both the graph topology and the kind of edges in the definition of similarity among nodes and among edges. Through the embedding function $\text{emb}_N : N \to \mathbb{R}^d$, a d-dimensional vector is associated with each graph node. Moreover, the embedding function $\text{emb}_E : E \to \mathbb{R}^d$ is used to generate the embedding of the edges by applying the Hadamard product on the embedding of the source and target nodes.

4 Classification Approach Integrating a Negative Selection Strategy

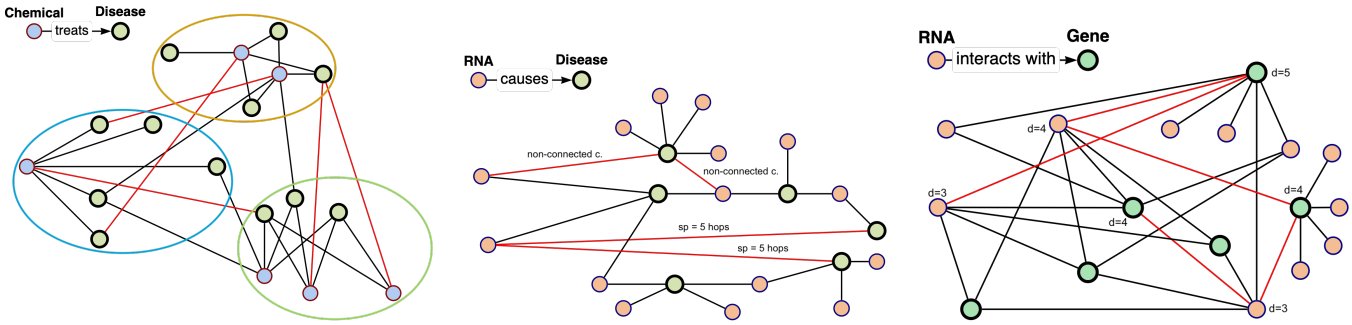
An edge classification approach that exploits a negative sample selection strategy is organized in a three-step pipeline (see Figure 2).

The first step aims to compute and organize the embeddings of the nodes and edges of a KG G according to the schema facts of its schema G_S. Specifically, nodes and edges are embedded in a latent space through the emb_N and emb_E functions described above. Then, edge embeddings are grouped by their associated schema fact f. This grouping allows us to extract homogeneous sets of embeddings while preserving the heterogeneity induced by different edge and node types.

The second step is devoted to the selection of negative edges, i.e., a subset of edges that are absent in the KG and are assumed to represent *incorrect* relationships. Formally, given a KG G, let $\text{Positive}(G)$ denote the set of edges in G, and $\text{Negative}(G)$ the set of artificially generated edges that should be considered wrong relationships in G ($\text{Negative}(G) \subseteq N \times E_T \times N$, with $\text{Positive}(G) \cap \text{Negative}(G) = \emptyset$). Negative edges are embedded in the same latent space as positive edges and grouped according to their schema fact (represented as red embeddings in Figure 2). Their size should not exceed 3 times the number of positive edges to limit class imbalance.

In the third step, we train one ML model for each schema fact to learn the distribution of positive edges in the KG. The goal is to train a model Model_f using the edge embeddings associated with schema fact f, so that Model_f captures the latent features (i.e., the embeddings) of edges of that type. Each model is expected to enclose positive edges within its positive decision region while keeping negative edges outside. Negative edges are required since without them an optimal Model_f would trivially classify all edges in $\text{Positive}(G)$ (and potentially any other edge) as positive, providing no meaningful information about its discriminative ability. Depending on the chosen learning paradigm, Model_f may be trained using both positive and negative samples in the full ML pipeline—e.g., DTs and RFs —or using only positive samples—e.g., OCSVMs and IFs. In the latter case, negative edges are used exclusively for hyperparameter tuning and model evaluation.

In the remainder of this section, we provide formal representations and examples of the strategies for selecting the negatives.



(a) Community. (b) Shortest path. (c) Degree-aware.

Figure 3: Negative strategies.

4.1 Community-based Strategy

This strategy leverages the community structure identified by a graph clustering algorithm (e.g., the Louvain algorithm [15]) on the graph induced by the schema fact f, assuming that nodes within the same community are more likely to be interconnected. Conversely, inter-community links are comparatively infrequent and can therefore be regarded as suitable candidates for the introduction of negative facts.

Definition 3. *(*Community-Based Strategy*). Let $f = (s_G, p, t_G)$ be a schema fact of a schema G_S associated with a KG G. Let G_f be the graph induced by instances(f) and $\{C_1, \ldots, C_n\}$ the clusters of nodes in G_f computed according to a graph clustering algorithm. A negative fact $(s_N, q, t_N) \notin$ Edges(G_f) is introduced in G_f according to Community-Based Strategy if and only if:*

- $s_N, t_N \in$ Nodes(G_f), $\phi(s_N) = s_G$, $\phi(t_N) = t_G$, $q \in E_T$,
- $((s_N, q, t_N) \in E \wedge q \neq p) \vee ((s_N, q, t_N) \notin E \wedge q = p)$,
- $s_N \in C_i, t_N \in C_j$ *and* $i \neq j$,
- $|\text{Negative}(G_f)| \leq 3 \cdot |\text{Positive}(G_f)|$.

Figure 3a shows an example of how the Community-Based Strategy works on a partition G_f associated with a KG G induced by the instances of the schema fact *(Chemical, treats, Disease)*. Blue nodes represent entities of type *Chemical*, green nodes represent entities of type *Disease*, and black links between them represent KG edges with predicate *treats*. Each ellipse corresponds to a community in G_f detected by the Louvain algorithm. According to the Community-Based Strategy, negative examples (shown in red) are sampled between nodes belonging to distinct communities (e.g., a drug in the blue cluster and a disease in the green one).

4.2 Shortest Path Based Strategy

This strategy selects negative edges by selecting pairs of nodes that are not directly connected and are at least $\tau_{\text{hop}} \in \mathbb{N}$ hops apart in the graph induced by the schema fact f, under the assumption that nodes far apart are unlikely to be linked. If two nodes belong to different connected components, the edge between them is automatically treated as a negative example. Although this strategy may be computationally expensive for large partition graphs, we remark that negatives are sampled only once in our pipeline and can be reused across runs.

Definition 4. *(*Shortest Path-Based Strategy*). Let $f = (s_G, p, t_G)$ be a schema fact of a schema G_S associated with a KG G. Let G_f be the graph induced by instances(f). Moreover, let $\tau_{\text{hop}} \in \mathbb{N}$ be a distance threshold for negative sampling. A negative fact $(s_N, q, t_N) \notin$ Edges(G_f) is introduced in G_f according to Shortest Path-Based Strategy if and only if:*

- $s_N, t_N \in$ Nodes(G_f), $\phi(s_N) = s_G$, $\phi(t_N) = t_G$, $q \in E_T$, $q \neq p$,
- $sp_{G_f}(s_N, t_N) \geq \tau_{\text{hop}}$,
- $|\text{Negative}(G_f)| \leq 3 \cdot |\text{Positive}(G_f)|$.

Figure 3b shows an example of how the Shortest Path-Based Strategy works on a partition G_f associated with a KG G induced by the instances of the schema fact *(RNA, causes, Disease)*. Orange nodes represent entities of type *RNA*, green nodes represent entities of type *Disease*, and black links between them represent KG edges with predicate *causes*. According to the Shortest Path-Based Strategy, negative examples (shown in red) are sampled between nodes that are not directly connected and are separated by at least $\tau_{\text{hop}} = 5$ hops, or belong to different connected components of G_f.

4.3 Degree-Aware Strategy

This strategy exploits the node degree distribution of the graph induced by a schema fact f to generate negative samples. Nodes are first assigned their degree within the graph induced by the instances of f. Negative edges are then sampled starting from highest-degree nodes. The rationale is that nodes with high degrees are typically well-connected, so pairs of high-degree nodes that are not linked are more likely to represent implausible edges [16].

Definition 5. *(*Degree-Aware Strategy*). Let $f = (s_G, p, t_G)$ be a schema fact of a schema G_S associated with a KG G, G_f be the graph induced by G_f, and $\tau_{\text{neg}} \in \mathbb{N}$ be a minimum degree threshold. A negative fact $(s_N, q, t_N) \notin$ Edges(G_f) is introduced in G_f according to the Degree-Aware Strategy if and only if:*

- $s_N, t_N \in$ Nodes(G_f), $\phi(s_N) = s_G$, $\phi(t_N) = t_G$, $q \in E_T$, $q \neq p$,
- $\deg_{G_f}(s_N) \geq \tau_{\text{neg}}$, $\deg_{G_f}(t_N) \geq \tau_{\text{neg}}$,
- $|\text{Negative}(G_f)| \leq 3 \cdot |\text{Positive}(G_f)|$.

Figure 3c shows an example of the Degree-Aware Strategy applied to the partition G_f associated with the schema fact f =*(RNA, interacts with, Gene)*. Negative edges are shown in red and sampled from nodes characterized by $\tau_{\text{neg}} \geq 3$. This ensures that negative

edges are sampled between nodes with the highest connectivity, which are less likely to be connected in reality.

5 Datasets and Metrics

Table 1 reports the two KGs used to evaluate our approach and their main topological statistics like the cardinality of nodes ($|N|$), edges ($|E|$), node types ($|N_T|$), and edge types ($|E_T|$). Moreover, we propose some statistics (average degree, average shortest path length, and average clustering coefficient) useful to assess the effectiveness of the proposed strategies. The first one, *miRNAdisease* (RNA), is a *view* derived from RNA-KG [11], covering different subsets of RNA-related interactions and annotations. The other, *PheKnowLator benchmark KG* [6] (PKT), is a KG designed for benchmarking biomedical link prediction, integrating molecular, clinical, and disease-level associations.

All the graphs exhibit the typical structural characteristics of large-scale biomedical KGs. Despite their size, both datasets maintain low average degrees and clustering coefficients, while the average shortest-path lengths remain relatively small. This is consistent with the small-world effect [29] commonly observed in biological interaction networks. Moreover, Figure 4 shows the highly skewed degree distribution of the PKT KG (the same pattern appears in RNA). Only a small fraction of nodes act as hubs, displaying very high degrees, whereas the vast majority remain weakly connected. This behavior reflects the presence of a few highly central biological entities surrounded by many low-degree ones.

Standard classification metrics such as precision, recall, specificity, balanced accuracy, F_β-score, and the Matthews correlation coefficient were considered to evaluate the performance of different ML models in the proposed pipeline. Moreover, we introduce a metric named $\hat{F}_\beta$-score, which combines the false positive rate (FPR) and the false negative rate (FNR): $\hat{F}_\beta\text{-score} = (1+\beta^2)\cdot\frac{FNR\cdot FPR}{(\beta^2\cdot FNR)+FPR}$. The parameter β allows control over the importance assigned to the two types of errors: larger values of β deeply penalize the reduction of false assertions (FPR), whereas smaller values prioritize missed negatives (FNR).

6 Experiments

Several experiments have been conducted to validate the proposed approach. Specifically, the quality of the negative sample selection strategies has been evaluated in terms of the overlap of the generated negative samples. Minimal overlap means that the strategies identify complementary edges and that one strategy is not a generalization/specialization of another strategy.

Moreover, we wish to evaluate the quality of the generated classification approach in terms of the metrics described in the previous section. A random negative sampling strategy was adopted as a baseline, with an upper bound on the number of negatives set to three times the number of positive edges.

KG	|N|	|E|	$\|N_T\|$	$\|E_T\|$	Avg. Degree	Avg. Path Length	Avg. Clus. Coef.
RNA	$99k$	$1.6M$	27	138	32.62	5.22	0.11
PKT	$560k$	$5.5M$	23	320	19.71	4.57	0.05

Table 1: Datasets.

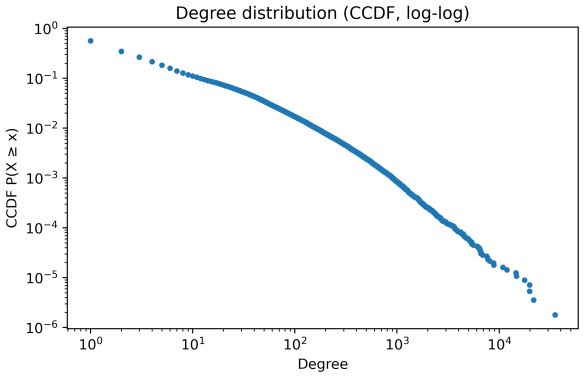


Figure 4: Degree distribution for the PKT KG.

Not all schema facts in the KGs were considered. Partitions (i) with very small cardinality (i.e., fewer than 1 k edges) or (ii) encoding well-established ontological knowledge (e.g., indicating that a gene *is a* protein-coding gene) were discarded, as they do not constitute meaningful or informative classification tasks. After this filtering step, we retained 12 schema facts for RNA and 18 for PKT.

In conducting experiments, we set both τ_{hop} and τ_{neg} to 10, adopted a 70–30 train–test edges split, and used TransE for embedding the KGs since it is a heterogeneous method and efficiently implemented in the GRAPE [7] framework (we chose an embedding size of 32). We set $\beta = 1$ for both the F_β-score and $\hat{F}_\beta$-score metrics to give equal importance to false positives and false negatives. Moreover, $\hat{F}_1$-score is used as the objective metric for hyperparameter selection (we used a nested cross-validation procedure, using a 5-fold outer loop for model evaluation and a 3-fold inner loop for hyperparameter tuning). The complete hyperparameter grids are reported in Appendix A.

6.1 Evaluating Negative Selection Strategies

Figure 5 reports a Venn diagram comparing the negative edges generated by the three strategies on the considered KGs. The diagram highlights both the distinct portions produced by each method and the intersections among them. Despite the large amount of generated negative edges (more than 12 M), the strategies show little overlap. In particular, only 77.1 k negative edges (0.59% of the total) are selected as negatives by all strategies.

This comparison quantifies the heterogeneity of the strategies and shows that the different heuristics lead to different negative sets. Such variability is relevant because it may influence the downstream behavior of the ML models (i.e., differences in performance can be attributed to the strategies).

Finally, all strategies are computationally efficient and suitable for parallel execution. In our experiments (conducted on a server equipped with an Intel Xeon CPU—32 threads—and 251 GB of RAM), only five strategy instances required more than one minute (but less than five) to complete: three shortest path based runs on partitions with more than 140 k edges, and two degree-aware runs.

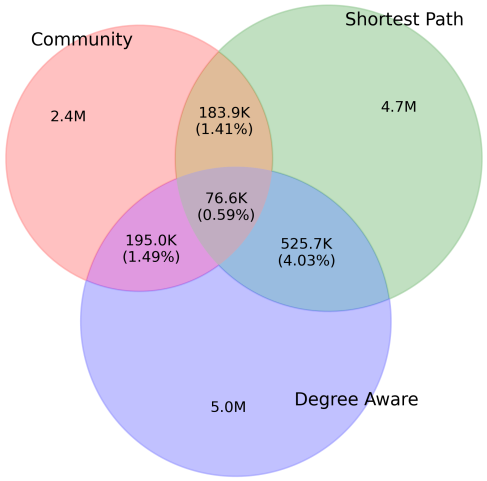


Figure 5: Overlaps of negative edges among strategies.

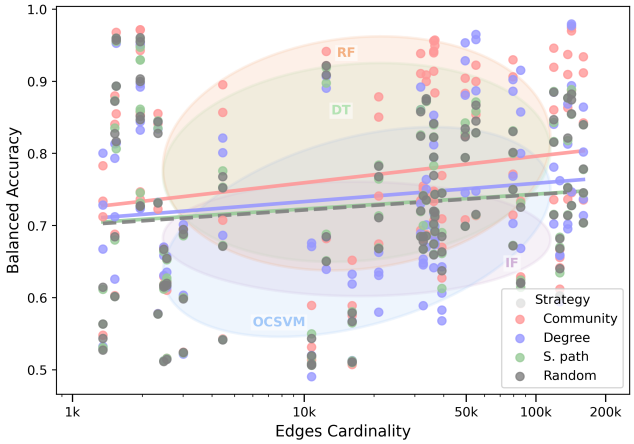


Figure 7: Regression of balanced accuracy as a function of edge cardinality (log) for the considered classification models and negative sampling strategies.

6.2 Evaluating ML approaches

Figure 6 reports the mean balanced accuracy for the considered classification models, aggregated over the two KGs, when combined with the associated negative sample selection strategy. Overall, the application of the proposed strategies produces better results with respect to the random baseline in many of the adopted models. The largest performance gains are observed for binary classifiers. In particular, RFs achieve the best results with an average improvement of up to +6% in balanced accuracy when the community-based strategy is adopted. Anomaly detection approaches also benefit from negative strategies, with improvements of up to +2%, indicating that negatives (i.e., "anomalies") from these strategies provide a more "informative" decision boundary.

We also evaluated other binary classifiers, including logistic regression [14], SVMs, and perceptron [32] trained via stochastic gradient descent. However, they underperformed RFs and DTs, achieving a 2–3% lower average balanced accuracy compared to

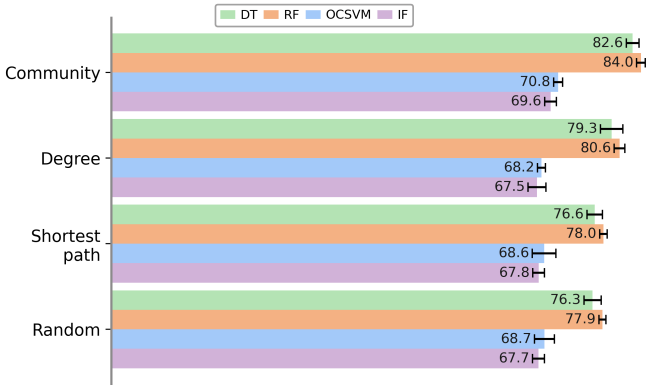


Figure 6: Average balanced accuracy with standard deviation of the considered classification models and negative sampling strategies.

RFs when using the community-based negative sampling strategy. Details are reported in Appendix B.

Moreover, Figure 7 shows that classification performance tends to increase with the number of triples in a partition. Each point in the figure represents the performance of a model on a specific schema fact, colored according to the adopted negative sampling strategy. Regression lines summarize the average performance trend for each strategy. Results confirm that, on average, the community-based strategy yields the best performance, followed by the degree-aware strategy. The shortest path-based strategy performs slightly better than the random baseline, with its advantage becoming more evident as relation cardinality increases. To quantify these observations, we computed the proportion of models outperforming their random-negative counterpart for each strategy. The community-based strategy improves performance in 108 out of 120 models (90.0%), the degree-aware strategy in 62 (51.7%), and the shortest path strategy in 61 (50.8%). A two-sided Wilcoxon signed-rank test comparing each strategy against the random baseline confirms the statistical significance of the observed improvements for community-based ($p \approx 10^{-18}$), while the effect of the shortest path based and degree-aware strategies is weaker ($p \approx 0.01$, $p \approx 0.17$). The figure also includes model-specific ellipses, centered at the empirical mean and scaled according to the covariance structure. OCSVMs benefit the most from the proposed negative sampling strategies when the number of positive edges increases, likely because more positives provide a clearer separation between normal and anomalous regions in the embedding space. IFs exhibit a more stable behavior across relation sizes. Although they benefit less from an increasing number of positive instances, this makes them a good choice when scalability w.r.t. the cardinality of the relation under study is to be accounted for.

Detailed performance results for each strategy, schema fact, and model are reported in Appendix C.

7 Concluding Remarks

In this paper, we have presented a study on the impact of negative sample selection strategies for edge classification in biomedical KGs. By analyzing three strategies that exploit community structure, node distance, and degree information, we showed that a topologically-aware negative sampling can improve link prediction performance. The experimental results on two large-scale biomedical KGs confirm that the proposed strategies outperform random negative sampling and provide robust gains independently of the adopted ML paradigm.

As future work, we plan to apply the approach to other biomedical KGs (e.g., PrimeKG [18], Hetionet [20]), investigate a broader range of parameter settings for the shortest path–based and degree-aware strategies (i.e., τ_{hop} and τ_{neg}), and develop new strategies based on the topology and semantics of the underlying KG. Finally, we plan to extend this work to other ML models, in particular neural-based classifiers.

Acknowledgments

This work was supported by the "Data Science and Decision Science for Digital Health" grant under the PSR-2025 funding scheme of the University of Milano, and by the "National Center for Gene Therapy and Drugs based on RNA Technology", PNRR-NextGeneration EU program [G43C22001320007]. Support to D.M. has been granted by the National Plan for NRRP Complementary Investments (PNC) in the call for the funding of research initiatives for technologies and innovative trajectories in the health–project n. PNC0000003–AdvaNced Technologies for Human–centrEd Medicine (project acronym: ANTHEM). Computational resources were provided by INDACO Core facility, which is a project of High Performance Computing at the University of MILAN. All the authors would like to thank Prof. A. Tettamanzi for the fruitful discussions and insightful suggestions on strategies for negative sample generation.

References

[1] Mohamed Amer et al. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 337–340, 2013.
[2] Antoine Bordes et al. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
[3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
[4] Leo Breiman et al. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.
[5] Ming Cai et al. IF-NS: A New negative sampling framework for knowledge graph embedding using influence function. *Know.-Based Syst.*, 315(C), April 2025.
[6] Tiffany J Callahan et al. An open source knowledge graph ecosystem for the life sciences. *Sci. Data*, 11(1):363, April 2024.
[7] L. Cappelletti et al. GRAPE for Fast and Scalable Graph Processing and random walk-based Embedding. *Nature Computational Science*, 2023.
[8] Luca Cappelletti et al. Node-degree aware edge sampling mitigates inflated classification performance in biomedical random walk-based graph representation learning. *Bioinformatics Advances*, 4(1), January 2024.
[9] Enrique Castillo et al. Distributed one-class support vector machine. *International Journal of Neural Systems*, 25(07):1550029, 2015.
[10] Emanuele Cavalleri et al. An ontology-based knowledge graph for representing interactions involving RNA molecules. *Scientific Data*, 11(1):906, Aug 2024.
[11] Emanuele Cavalleri et al. RNA-KG v2.0: an RNA-centered Knowledge Graph with Properties. *NAR Genomics and Bioinformatics*, 8(1):lqaf194, 01 2026.
[12] M. Ciranni et al. Anomaly detection in feature space for detecting changes in phytoplankton populations. *Frontiers in Marine Science*, 2024.
[13] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
[14] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.
[15] Pasquale De Meo et al. Generalized Louvain method for community detection in large networks. In *2011 11th International Conference on Intelligent Systems Design and Applications*, pages 88–93, 2011.
[16] Jesse Gillis and Paul Pavlidis. The Impact of Multifunctional Genes on "Guilt by Association" Analysis. *PLOS ONE*, 6(2):1–16, 02 2011.
[17] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 855–864, New York, NY, USA, 2016. Association for Computing Machinery.
[18] W. Hämäläinen. Scientific writing for computer science students. http://www.cs.joensuu.fi/~whamalai/teaching.html. Accessed: 2019-04-10.
[19] Mostafa Hejazi and Yogendra P. Singh. One-class support vector machines approach to anomaly detection. *Applied Artificial Intelligence*, 27(4):351–366, 2013.
[20] Daniel Scott Himmelstein et al. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *Elife*, 6, September 2017.
[21] Aidan Hogan et al. Knowledge Graphs. *ACM Computing Surveys*, 54(4):1–37, July 2021.
[22] Natthawut Kertkeidkachorn et al. FinKG: A Core Financial Knowledge Graph for Financial Analysis. In *IEEE 17th International Conference on Semantic Computing (ICSC)*, pages 90–93, 2023.
[23] Bhushan Kotnis and Vivi Nastase. Analysis of the Impact of Negative Sampling on Link Prediction in Knowledge Graphs, 2018.
[24] Fei Tony Liu et al. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 413–422, 2008.
[25] Tiroshan Madushanka and Ryutaro Ichise. Negative Sampling in Knowledge Graph Representation Learning: A Review, 2024.
[26] Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.
[27] Tomas Mikolov et al. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.
[28] Amihai Motro. Completeness information and its application to query processing. *IEEE Transactions on Knowledge and Data Engineering*, 9(5):799–813, 1997.
[29] M E J Newman. Models of the small world. *J. Stat. Phys.*, 101(3-4):819–841, November 2000.
[30] Bryan Perozzi et al. Deepwalk: Online learning of social representations. *CoRR*, abs/1403.6652, 2014.
[31] Raymond Reiter. On closed world data bases. In Hervé Gallaire and Jack Minker, editors, *Logic and Data Bases*, NATO Advanced Study Institute Series, pages 55–76. Springer, 1978.
[32] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
[33] Bernhard Schölkopf et al. Estimating the support of a high-dimensional distribution. In *Advances in Neural Information Processing Systems (NIPS 13)*, pages 582–588. MIT Press, 2001.
[34] Hyun-Jong Shin et al. One-class support vector machines—an application in machine fault detection and classification. *Computers & Industrial Engineering*, 48(2):395–408, 2005.
[35] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.
[36] Carl Yang et al. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4854–4873, 2020.
[37] Yucheng Yang et al. The knowledge graph for macroeconomic analysis with alternative big data. Papers 2010.05172, arXiv.org, Oct 2020.
[38] Hai-Cheng Yi et al. Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, 23(1):bbab340, 09 2021.
[39] Hyunsik Yoo et al. Directed network embedding with virtual negative edges. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 1291–1299, New York, NY, USA, 2022. Association for Computing Machinery.

A Hyperparameter Configuration

This appendix reports the hyperparameter grids adopted for the classification models considered in the experimental evaluation.

For DTs, we optimized the minimum number of samples required to split an internal node, the minimum number of samples per leaf, and the maximum tree depth. Specifically, *min_samples_split* $\in \{2, 5, 10\}$, *min_samples_leaf* $\in \{1, 2, 4\}$, and *max_depth* $\in \{None, 10, 20, 30, 50\}$.

For RFs, we tuned the number of estimators and the maximum depth of individual trees. The number of estimators ranged in $\{100, 200, 300, 400, 500\}$, while *max_depth* $\in \{None, 10, 20, 30, 50\}$.

For OCSVMs, we adopted the radial basis function (RBF) kernel and optimized ν and the kernel coefficient γ. The ν parameter was varied in the range $[0.1, 0.2]$ using 10 uniformly spaced values, while $\gamma \in \{10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$.

Finally, for IFs, we tuned the number of trees and the expected contamination level. The number of estimators was set to $\{100, 200, 400\}$, while contamination values $\{0.2, 0.3, 0.4, 0.5\}$ were explored.

B Other Binary Classifiers

Table A.1 reports results obtained for additional binary classifiers trained via stochastic gradient descent and combined with the community-based negative sampling strategy, and compares their performance against RF in terms of average balanced accuracy. Models were tuned using the cross-validation procedure described in Sect. 6.

Model	Avg. BA	Δ Avg. BA vs RF
Logistic Regression	81.5 ± 0.01	-2.5%
SVM	81.3 ± 0.01	-2.7%
Perceptron	81.2 ± 0.02	-2.8%
RF	84.0 ± 0.01	–

Table A.1: Average balanced accuracy with standard deviation of additional binary classifiers evaluated using the community-based negative sampling strategy.

C Strategies' Detailed Performances

Tables A.2–A.5 detail the classification results obtained for each negative sampling strategy. We report the best five and worst five schema facts in terms of balanced accuracy, together with edge cardinality (`Edges`), precision, recall, specificity, F_1-score, Matthews correlation coefficient (`MCC`), and $\hat{F}_1$-score. Results are reported as mean and standard deviation over the outer folds of the nested cross-validation procedure.

Model	KG	Schema Fact	Edges	Precision	Recall	Spec.	B. Acc.	F_1	MCC	$\hat{F}_1$
RF	PKT	Gene - interacts with - Protein	142.1k	0.980±0.000	0.971±0.001	0.977±0.000	0.974±0.001	0.975±0.000	0.946±0.001	0.026±0.000
RF	PKT	Gene - genetically interacts with - Gene	2.0k	0.981±0.005	0.988±0.004	0.955±0.011	0.972±0.007	0.984±0.003	0.947±0.009	0.019±0.004
RF	RNA	Gene - genetically interacts with - Gene	2.0k	0.981±0.005	0.987±0.003	0.956±0.012	0.972±0.007	0.984±0.003	0.947±0.011	0.019±0.004
DT	PKT	Gene - interacts with - Protein	142.1k	0.977±0.001	0.968±0.000	0.972±0.001	0.970±0.001	0.973±0.000	0.940±0.000	0.029±0.000
RF	RNA	miRNA - in similarity relationship with - miRNA	1.5k	0.971±0.007	0.997±0.001	0.939±0.015	0.968±0.008	0.984±0.004	0.949±0.012	0.007±0.003
		Mean	42.8k	0.834±0.005	0.825±0.007	0.710±0.010	0.768±0.008	0.805±0.005	0.576±0.012	0.097±0.005
RF	PKT	Protein - located in - Cell	10.8k	0.720±0.001	0.996±0.001	0.039±0.005	0.517±0.003	0.836±0.001	0.135±0.009	0.008±0.002
OCSVM	PKT	GO - positively regulates - GO	2.5k	0.775±0.002	0.996±0.002	0.036±0.010	0.516±0.006	0.872±0.001	0.127±0.031	0.008±0.004
IF	PKT	Protein - located in - Cell	10.8k	0.295±0.002	0.798±0.006	0.229±0.003	0.514±0.005	0.430±0.003	0.029±0.009	0.320±0.008
OCSVM	PKT	GO - negatively regulates - GO	2.5k	0.774±0.002	0.996±0.002	0.029±0.008	0.512±0.005	0.871±0.001	0.109±0.029	0.008±0.003
IF	PKT	Protein - located in - Anatomy	16.1k	0.261±0.003	0.800±0.013	0.215±0.003	0.508±0.008	0.393±0.005	0.016±0.013	0.319±0.016

Table A.2: Detailed performance of the community-based strategy across the considered KGs.

Model	KG	Schema Fact	Edges	Precision	Recall	Spec.	B. Acc.	F_1	MCC	$\hat{F}_1$
RF	PKT	Gene - interacts with - Protein	142.1k	0.991±0.000	0.988±0.000	0.971±0.001	0.980±0.001	0.990±0.000	0.956±0.001	0.017±0.000
DT	PKT	Gene - interacts with - Protein	142.1k	0.990±0.000	0.987±0.000	0.967±0.001	0.977±0.001	0.989±0.000	0.952±0.001	0.018±0.000
RF	RNA	miRNA - under expressed in - Disease	55.0k	0.985±0.000	0.981±0.001	0.949±0.001	0.965±0.001	0.983±0.001	0.927±0.002	0.027±0.001
RF	RNA	miRNA - in similarity relationship with - miRNA	1.5k	0.976±0.006	0.999±0.001	0.917±0.023	0.958±0.012	0.987±0.004	0.944±0.015	0.002±0.002
DT	RNA	miRNA - under expressed in - Disease	55.0k	0.982±0.004	0.975±0.001	0.941±0.013	0.958±0.007	0.979±0.001	0.909±0.006	0.034±0.001
		Mean	42.8k	0.801±0.007	0.802±0.010	0.675±0.013	0.739±0.012	0.776±0.006	0.511±0.013	0.114±0.007
OCSVM	PKT	Protein - located in - Cell	10.8k	0.436±0.032	0.139±0.020	0.901±0.008	0.520±0.014	0.210±0.027	0.059±0.026	0.178±0.012
OCSVM	PKT	GO - positively regulates - GO	2.5k	0.774±0.001	0.994±0.002	0.036±0.007	0.515±0.005	0.870±0.001	0.111±0.021	0.012±0.003
IF	PKT	Protein - located in - Anatomy	16.1k	0.286±0.002	0.800±0.013	0.226±0.004	0.513±0.009	0.421±0.004	0.029±0.010	0.317±0.016
OCSVM	PKT	GO - negatively regulates - GO	2.5k	0.774±0.002	0.996±0.002	0.029±0.007	0.512±0.005	0.871±0.001	0.108±0.033	0.008±0.004
IF	PKT	Protein - located in - Cell	10.8k	0.637±0.004	0.801±0.010	0.180±0.015	0.491±0.013	0.709±0.005	-0.023±0.016	0.321±0.012

Table A.3: Detailed performance of the degree-based strategy across the considered KGs.

Model	KG	Schema Fact	Edges	Precision	Recall	Spec.	B. Acc.	F_1	MCC	$\hat{F}_1$
RF	RNA	miRNA - in similarity relationship with - miRNA	1.5k	0.972±0.003	0.998±0.001	0.922±0.008	0.960±0.005	0.985±0.002	0.943±0.006	0.004±0.001
RF	RNA	Gene - genetically interacts with - Gene	2.0k	0.973±0.009	0.994±0.002	0.922±0.027	0.958±0.014	0.984±0.005	0.936±0.021	0.011±0.004
DT	RNA	miRNA - in similarity relationship with - miRNA	1.5k	0.972±0.008	0.988±0.005	0.923±0.024	0.956±0.015	0.980±0.004	0.925±0.017	0.020±0.007
RF	PKT	Gene - genetically interacts with - Gene	2.0k	0.965±0.008	0.997±0.002	0.898±0.025	0.948±0.014	0.981±0.004	0.925±0.015	0.005±0.003
DT	PKT	Gene - genetically interacts with - Gene	2.0k	0.960±0.010	0.977±0.006	0.883±0.029	0.930±0.018	0.968±0.002	0.875±0.011	0.037±0.005
		Mean	42.8k	0.804±0.005	0.808±0.010	0.648±0.014	0.728±0.012	0.783±0.007	0.489±0.012	0.119±0.007
IF	PKT	Protein - located in - Anatomy	16.1k	0.247±0.002	0.800±0.013	0.225±0.006	0.513±0.009	0.378±0.004	0.026±0.011	0.318±0.016
OCSVM	PKT	GO - negatively regulates - GO	2.5k	0.774±0.001	0.996±0.001	0.029±0.007	0.512±0.004	0.871±0.001	0.103±0.029	0.009±0.002
RF	PKT	Protein - located in - Anatomy	16.1k	0.762±0.001	0.998±0.001	0.022±0.004	0.510±0.003	0.864±0.000	0.102±0.006	0.004±0.002
RF	PKT	Protein - located in - Cell	10.8k	0.734±0.001	0.998±0.001	0.020±0.004	0.509±0.003	0.846±0.000	0.095±0.018	0.005±0.001
IF	PKT	Protein - located in - Cell	10.8k	0.273±0.002	0.798±0.006	0.215±0.005	0.507±0.005	0.407±0.003	0.015±0.010	0.321±0.008

Table A.4: Detailed performance of the shortest path-based strategy across the considered KGs.

Model	KG	Schema Fact	Edges	Precision	Recall	Spec.	B. Acc.	F_1	MCC	$\hat{F}_1$
RF	PKT	Gene - genetically interacts with - Gene	2.0k	0.983±0.003	0.977±0.005	0.944±0.011	0.960±0.008	0.980±0.003	0.915±0.013	0.032±0.006
RF	RNA	miRNA - in similarity relationship with - miRNA	1.5k	0.977±0.007	0.999±0.000	0.921±0.024	0.960±0.012	0.988±0.003	0.947±0.016	0.001±0.001
DT	RNA	miRNA - in similarity relationship with - miRNA	1.5k	0.978±0.006	0.990±0.004	0.924±0.021	0.957±0.013	0.984±0.002	0.929±0.011	0.017±0.004
RF	RNA	Gene - genetically interacts with - Gene	2.0k	0.982±0.003	0.970±0.001	0.940±0.010	0.955±0.005	0.976±0.002	0.899±0.007	0.039±0.002
RF	PKT	Chemical - interacts with - Gene	12.4k	0.959±0.001	0.983±0.002	0.861±0.005	0.922±0.004	0.971±0.002	0.871±0.007	0.030±0.002
		Mean	42.8k	0.804±0.005	0.806±0.009	0.647±0.013	0.726±0.011	0.782±0.006	0.485±0.011	0.120±0.006
OCSVM	PKT	GO - negatively regulates - GO	2.5k	0.773±0.001	0.996±0.001	0.027±0.005	0.511±0.003	0.871±0.001	0.101±0.022	0.008±0.003
RF	PKT	Protein - located in - Anatomy	16.1k	0.773±0.000	0.998±0.000	0.024±0.002	0.511±0.001	0.871±0.000	0.111±0.008	0.004±0.001
IF	PKT	Protein - located in - Anatomy	16.1k	0.236±0.003	0.800±0.013	0.222±0.006	0.511±0.009	0.364±0.004	0.023±0.012	0.317±0.016
IF	PKT	Protein - located in - Cell	10.8k	0.234±0.001	0.801±0.010	0.215±0.005	0.508±0.007	0.363±0.003	0.016±0.007	0.318±0.012
RF	PKT	Protein - located in - Cell	10.8k	0.772±0.000	0.999±0.001	0.014±0.002	0.506±0.002	0.871±0.000	0.088±0.011	0.002±0.002

Table A.5: Detailed performance of the random-based strategy across the considered KGs.