Обзор и сравнительный анализ алгоритмов атак и защиты на графовые архитектуры ИНС

Д.А. Киржинов, Е.А. Ильюшин

***Аннотация*—Графы окружают нас повсюду, объекты реального мира часто определяются в терминах их связей с другими объектами. Набор объектов и связей между ними естественным образом выражаются в виде графа. В силу содержательности такого представления данных, которое образуется в результате различных искусственных и естественных процессов, обучение нейронных сетей на таких данных является мощным инструментом. Спектр атак на архитектуры ГНС (графовых нейронных сетей) очень широк, и для каждого из методов атаки требуется разработать и определить эффективные методы защиты, а также исследовать атаки с точки зрения вычислительной сложности для их возможного применения на больших графах, используемых при решении реальных прикладных задач. Данная работа является обзором, в рамках которого рассматривается безопасность графовых нейросетевых архитектур, включая алгоритмы атак и способы защиты от них путем повышения устойчивости (робастности). Также приводится некоторая классификация этих методов по различным критериям и обзор существующих работ по данной тематике.**

***Ключевые слова*—Графовые нейронные сети, состязательные атаки и защита, робастность**

I. Введение

О графовых нейронных сетях. Интуитивная идея, лежащая в основе ГНС, заключается в том, что узлы графа представляют объекты или понятия реального и виртуального мира, а ребра – взаимоотношения между ними. Так, можно поставить в соответствие каждому узлу вектор признаков, называемый состоянием, который собирает представление объекта (эмбеддинг). Чтобы определить этот вектор, нужно учитывать, что связанные узлы соединены ребрами с учетом информации, содержащейся как на ребрах, так и внутри вершин. Так, представление об объекте (вершине) естественным образом задается с помощью информации, содержащейся в окрестности вершины (Рисунок 1).

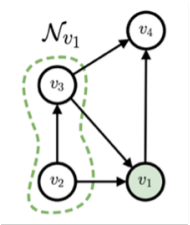

Рис. 1: Окрестность вершины v_1

Статья получена 03 января 2024

Киржинов Довлет Азаматович, МГУ им. М.В. Ломоносова, факультет вычислительной математики и кибернетики, Москва, Россия (email: dovlet.kirzhinov@mail.ru).

Ильюшин Евгений Альбинович, МГУ им. М.В. Ломоносова, факультет вычислительной математики и кибернетики, Москва, Россия (email: eugene.ilyushin@gmail.com).

Графы, при рассмотрении их в виде матрицы смежности, очень похожи на картинки, а в действительности картинка это частный случай графа. Исходя из этого интуитивно кажется целесообразным применять свертку в графовых нейронных сетях для извлечения признаков.

Если представить пиксели изображения вершинами графа, соединить соседние по свертке пиксели ребрами и предоставить относительную позицию пикселей в информации о ребре, то графовая свертка на таком графе будет работать так же, как и свертка над изображением (Рисунок 2).

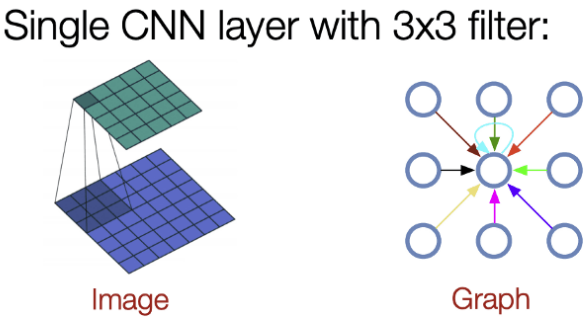

Рис. 2: Иллюстрация аналогии свертки на изображении и при представлении его в виде графа с фильтром 3×3.

В действительности ГНС являются обобщением сверточных нейронных сетей. В отличие от картинок, на вход ГНС подаются сразу графы, что в целом убирает ограничение на фиксированный размер входа объектов. Более того, структура графа более неоднородна, нежели у изображений, а значит нужны более сложные механизмы, которые могли бы сохранять информацию о вершинах, ребрах и графе в целом после этапов свертки.

Одним из основных таких механизмов является передача сообщений (message passing). Для такого обощения нужно отказаться от «прямоугольной природы» свертки, а использовать признаки соседних вершин и агрегировать их с некоторыми весами с признаками в рассматриваемой вершине. Агрегируемую информацию с текущей и соседних вершин называют сообщениями (messages). После агрегации всех сообщений следует этап обновления эмбеддинга текущей вершины. Совокупность этапов построения сообщений и агрегации называется этапом передачи сообщений.

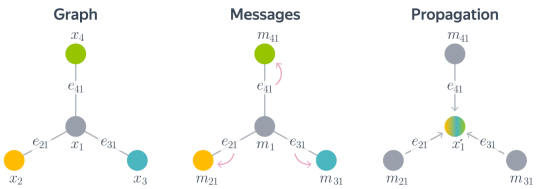

Рис. 3: Схематическое изображение механизма передачи сообщений на одной итерации для одной вершины: для вершины x_1 агрегируются сообщения из соседних вершин – m_{21}, m_{31}, m_{41} используя типы соединяющих их ребер e_{21}, e_{31}, e_{41}, после чего обновляется представление вершины, обозначаемое как x_1'

Вопросы безопасности в машинном обучении. Природа атак на системы машинного обучения и глубокого обучения отличается от других угроз безопасности информации. Состязательные атаки опираются на сложность глубоких нейронных сетей и их статистический характер, чтобы найти способы их использования и изменения их поведения. Нет способов обнаружить действия злоумышленников с помощью классических инструментов, используемых для защиты программного обеспечения от киберугроз [1].

Состязательные атаки также применимы и к графовым архитектурам ИНС. Злоумышленник может генерировать неблагоприятные возмущения графа, манипулируя структурой графа или характеристиками узлов, чтобы обмануть модели ГНС, такие манипуляции проилюстрированны на рисунках 4

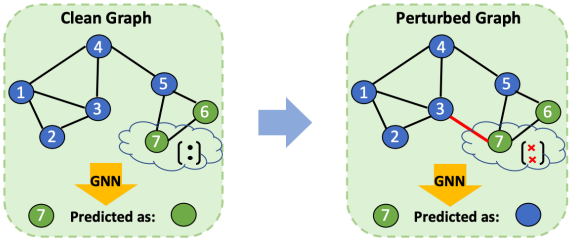

Рис. 4: Пример состязательной атаки на графовые данные. Задача – предсказать цвет узлов. Здесь вершина 7 является целевым узлом. Атакующий стремится изменить предсказание для узла 7 путем модификации ребер и признаков. [2].

В связи с этим в вопросах безопасности ИНС выделяют четыре проблемы для исследования, а именно: противостояние угрозам (”Устойчивость/Робастность”), выявление опасностей (”Мониторинг”), снижение внутренних опасностей модели (”Выравнивание”) и снижение системных опасностей (”Системная безопасность”) [3].

Данные вопросы безопасности особенно остры для графовых архитектур нейронных сетей в силу их информационной содержательности и большего числа “окон”, или, как еще их называют – бэкдоров (backdoor) для совершения атаки и извлечения/отравления информации, так как эмбеддинги в таких архитектурах представляют собой не просто вектора.

Выведена некоторая классификация типовых атак на системы машинного обучения, и на графовые в том числе. Появление новых атак требует разработки и алгоритмов защит от них.

Надежность существующих методов защиты также сомнительна, и различные исследования показали, что большинство методов защиты неэффективны против конкретной атаки. Именно обнаружение того факта, что модели глубокого обучения не являются ни безопасными, ни надежными, значительно препятствует их развертыванию на практике в критически важных системах, таких, например, как прогнозы в здравоохранении, что, естественно, жизненно важно [4].

К сожалению, пока не выведена единая классификация и методология алгоритмов защиты, т. к. эти алгоритмы в основном направлены на предотвращение конкретных атак. С другой стороны методы защиты представляют собой не столько отражение атак, а по большей части усиление таких систем, повышение устойчивости (робастности).

Из других острых вопросов, для критических приложений стоит задача сертификации систем, моделей и наборов данных, подобно тому, как это делается для традиционных систем программного обеспечения, используемых в таких приложениях. Важно также, что состязательные атаки вызывают проблемы с доверием к алгоритмам машинного обучения, особенно к глубоким нейронным сетям [1].

II. Атаки на графовые нейронные сети

A. Типы задач решаемых ГНС

Основные задачи анализа графов, в которых обычно применяются модели глубокого обучения, делятся на три категории:

- Задачи на уровне узлов. Классификация узлов – одна из наиболее распространенных задач, например, идентификация человека в социальной сети;
- Задачи на уровне связей. Задача уровня связей относится к классификации ребер и предсказанию связей. Среди них предсказание связей является более сложной задачей и широко используется в реальном мире, целью которой является предсказание наличия ребра или силы связи которое интерпретируется ребром, например, предсказание потенциальных отношений между двумя указанными лицами в социальной сети, и даже новых или разрывающихся отношений в будущем;
- Задачи на уровне графов. Рассматриваем граф как особую форму узла (т. е. получается своего рода структура гиперграфа), поэтому задачи на уровне графа похожи на задачи на уровне узла. В качестве примера можно рассмотреть наиболее частое применение – классификацию графов.

B. Атаки на графовые данные и их классификация

Основная идея состязательных атак заключается в том, что во входные данные вносятся помехи (возмущения) и изменения в таких параметрах может быть не просто связать с изменениями реальных характеристик. Учитывая нетривиальную структуру параметров для моделей графовых ИНС, в данном случае имеется больше подходов во внесение изменений в структуру графовых параметров. Можно проследить начало исследований в этом направлении.

С точки зрения атаки на графовую модель обучения, авторы работы [5] впервые исследовали атаки противника на графовые данные, при незначительных возмущениях внесенных в характеристики узлов и структуры графа, целевые классификаторы легко обмануть и, как следствие, они неправильно классифицируют заданные узлы.

Со стороны повышения робастности, в [6] предлагают модифицированную модель: графовые сверточные сети (GCN – Graph Convolutional Network) с системой защиты от атак для повышения ее устойчивости.

И далее, в [7] изучают существующие работы по стратегиям атак и защит на графовые данные. Однако авторы в основном фокусируются на атаках, оставляя работы по защите не до конца изученными.

Формально определить атаку на ГНС можно следующим образом: пусть f модель глубокого обучения, предназначенная для решения задачи. Учитывая набор целевых экземпляров $\tau \subseteq S - S_L$, где S может быть V, E или $\mathcal{G}$ для разных уровней задач соответственно, а S_L обозначает экземпляры с метками, злоумышленник стремится максимизировать потери f, что приводит к ухудшению эффективности предсказания:

$$\underset{\hat{G}\in\Psi(G)}{maximize} \sum_{t_i\in\tau} \mathcal{L}(f_{\theta^*}(\hat{G}^{t_i}, X, t_i), y_i) \ ,$$
$$\text{здесь } \theta^* = \underset{\theta}{argmin} \sum_{v_j\in S_L} \mathcal{L}(f_\theta(\tilde{G}^{v_j}, X, v_j), y_j),$$

где $G \in \mathcal{G}$, $\mathcal{L}$ – функция потерь, t – вершина в графе, $\hat{G}$ – граф с изменениями атакующего, $\tilde{G}$ – исходный граф, $\Psi(G)$ – пространство всех изменений атакущего над графом G, X – матрица признаков, y – истинная метка класса, θ – набор параметров, определяющий модель [8].

Для проведения атак на целевую систему, как правило, злоумышленник обладает определенными знаниями о целевых моделях и наборе данных, что помогает ему достичь поставленной цели. Основываясь на знаниях о целевых моделях, можно выделить различные классы атак.

По уровню владения информацией атакующим:

- ***Атака «белого ящика»*** – считается простейшей атакой, при которой злоумышленники обладают всей информацией о целевой модели, включая её архитектуру, параметры и информацию о градиенте, т.е. целевые модели полностью открыты для злоумышленников. Используя такую информацию, атакующие могут легко повлиять на целевые модели и вызвать разрушительные последствия. Однако в реальном мире это неосуществимо, поскольку обладание такими полными знаниями о целевых моделях крайне затратно. Поэтому атака «белого ящика» менее опасна, но часто используется для приблизительного определения наихудшей производительности атакуемой системы.
- ***Атака «серого ящика».*** Для такого класса атак злоумышленники строго обязаны обладать избыточными знаниями о целевых моделях, что гораздо лучше отражает возможные сценарии практике, поскольку более вероятно, что атакующие имеют ограниченный доступ для получения информации, например, знакомы только с архитектурой целевых моделей. Таким образом, это сложнее, чем проведение атаки «белого ящика», но более опасно для атакуемых моделей.
- ***Атака «черного ящика».*** В отличие от класса «белого ящика», атаки «черного ящика» предполагают, что атакующий ничего не знает о целевых системах. При таких условиях злоумышленникам разрешается выполнять запросы «черного ящика» только на ограниченных выборках, для того чтобы получить информацию о целевой модели. Тем не менее, атаки такого класса самые опасные, поскольку злоумышленники могут атаковать любые модели с ограниченной информацией (или вообще без нее).

По преследуемым целям атакующего:

- ***Нарушение безопасности.*** Атаки, нацеленные на нарушение главных свойств информации: нарушения доступности, целостности и другие. При атаке на доступность злоумышленник пытается нарушить функционирование системы, тем самым блокируя ее нормальную работу. Ущерб носит глобальный характер, то есть нарушается общая производительность всей системы. При атаке на целостность намерение злоумышленников – обойти или обмануть систему обнаружения, что отличается от атаки на доступность тем, что не нарушает нормальную работу системы. Существуют и другие цели, например реверс-инжиниринг информации о модели, таким образом нарушается конфиденциальность.
- ***Специфичность ошибок*** – при данной цели атаки направлены на то, чтобы вызвать определенное неверное поведение атакуемой системы. Для понимания в качестве примера можно рассмотреть задачу классификации вершин графа. Так, например атака, специфичная для ошибки, может быть нацелена на ложную классификацию предсказаний конкретных меток, в то время как атака без специфики не заботится о том, что является предсказанием. Атака считается успешной до тех пор, пока предсказание неверно.
- ***Специфичность атаки.*** Атаки данной направленности могут разделиться на целевые и нецелевые (общая атака). Целевая атака направлена на определенное подмножество узлов (обычно целевой узел), в то время как нецелевая атака является недифференцированной и глобальной.

По атакуемому этапу конвейера: Атаки можно разделить на атаку отравлением и атаку уклонением в зависимости от возможностей противника, которые возникают на разных этапах атаки. Схемы таких атак можно рассмотреть на рисунке 5.

- ***Атака отравлением*** пытается повлиять на производительность целевых моделей путем изменения набора данных на этапе обучения, т. е. целевые модели обучаются на отравленных наборах данных.
- ***Атака уклонением.*** В то время как атаки отравлением сосредоточены на этапе обучения, атаки уклонением применяются на этапе эксплуатации внедряя враждебные примеры во входные данные. Атаки уклонением происходят после того, как целевая модель обучена на исходном графе.

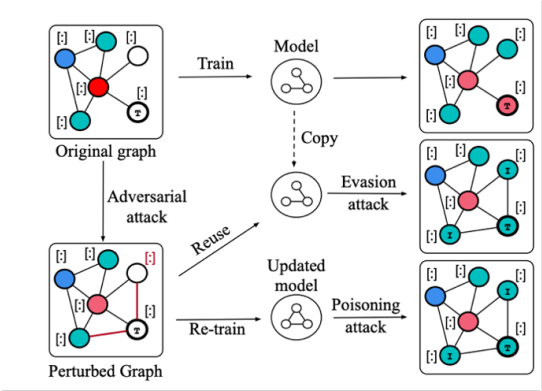

Рис. 5: Схема атак отравлением (poisoning) и уклонением (evasion) [8].

По стратегии: Для атаки на целевую модель на графовых данных злоумышленники могут иметь ряд стратегий для достижения своих злых умыслов. В большинстве случаев они фокусируются на структуре графа или особенностях вершин и ребер.

- ***Атака на топологию графа.*** Злоумышленники в основном фокусируются на топологии графа, так называемая атака на структуру. Например, при таких атаках предоставляется возможность добавлять или удалять некоторые ребра, расположенные между узлами графа, чтобы обмануть целевую систему.
- ***Атака на характеристики графа.*** Хотя топологическая атака является более распространенной, злоумышленники могут проводить и атаки на характеристики узлов и ребер, изменяя эти значения. В отличие от структуры графа, характеристики узлов и ребер могут быть дискретными или непрерывными.
- ***Гибридная атака.*** Обычно злоумышленники используют обе вышеперечисленные стратегии атаки одновременно, чтобы оказать более серьезное воздействие. Кроме того, они могут даже добавить несколько поддельных узлов (с поддельными метками), которые будут иметь свои собственные особенности и отношения с другими неиспорченными экземплярами.

Глобально алгоритмы атак можно разбить на 2 крупных класса в зависимости от того используют ли они информацию о том, как изменяется функция потерь модели по отношению к ее параметрам:

- ***Алгоритмы на основе градиента.*** Интуитивно алгоритм на основе градиента прост, но эффективен. Основная идея заключается в следующем: фиксируются параметры обученной модели, а входные данные рассматриваются как гиперпараметр, который необходимо оптимизировать. Аналогично процессу обучения, злоумышленники могут использовать частную производную потерь по отношению к ребрам (топологическая атака) или признакам (атака по признакам), чтобы решить, как манипулировать набором данных. Однако градиенты не могут быть применены непосредственно к входным данным, вместо этого злоумышленники часто выбирают градиент с наибольшим абсолютным значением и манипулируют им с нужными данными. Хотя большинство моделей глубокого обучения оптимизированы с помощью градиентов, злоумышленники, напротив, могут нарушить их работу также с помощью градиентов.
- ***Алгоритмы, не основанные на градиенте.*** Помимо информации о градиенте, злоумышленники могут генерировать атакующие примеры другими способами. Например, с точки зрения генетического алгоритма, злоумышленники могут выбирать популяцию (враждебные примеры) с наивысшей оценкой пригодности (например, ошибочные выходы целевой/замещающей модели) поколение за поколением. Кроме того, для решения этой проблемы также часто используются алгоритмы обучения с подкреплением. Методы атаки на основе обучения с подкреплением в [9] изучают обобщенные примеры противника в пространстве действий. Более того, примеры противника могут быть даже сгенерированы хорошо разработанной генеративной моделью.

До сих пор большинство существующих работ в сценарии атаки в основном основываются на градиентах, либо матрице смежности, либо матрице признаков, что приводит к топологической атаке и атаке по признакам, соответственно. Однако информацию о градиентах целевой модели получить сложно, поэтому злоумышленники обучают суррогатную модель для извлечения градиентов. В дополнение к алгоритмам на основе градиента, для достижения цели атаки предложено использовать несколько эвристических методов, таких как генетический алгоритм и алгоритмы на основе обучения с подкреплением.

Первопроходцем в алгоритмах атак можно назвать **алгоритм Nettack**. Авторы [5] представили первую работу по атакам противников на атрибутированные (когда есть параметры и на вершинах, и на ребрах) графы, сфокусировавшись на задаче классификации узлов с помощью графовых сверточных сетей (GCN), используя эффективный метод жадного поиска для внесения изменений в характеристики вершин и структуру графа. На основе обширных экспериментов авторы делают вывод, что даже сложная атака отравлением успешно поддается предложенному подходу. Производительность классификации постоянно снижается, даже когда доступно лишь частичное знание о графе (откуда можно сделать вывод, что это атака относится к классу «серого ящика»). Более того, атаки обобщаются на другие модели классификации узлов.

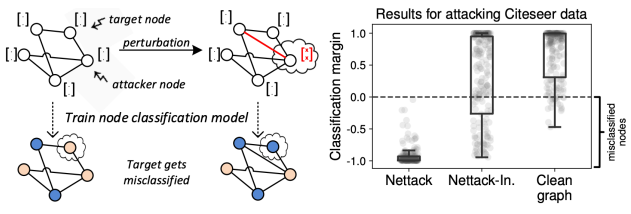

Рис. 6: Иллюстрация результатов работы ГНС после изменения структуры графа в Nettack. [5].

В [5] экспериментально показано, что предложенная модель может значительно ухудшить результаты классификации для целевых узлов, требуя лишь нескольких изменений в графе. Кроме того, авторы продемонстрировали, что эти результаты переносятся на другие известные модели и применимы к различным наборам данных и даже работают, когда наблюдаются только части дан-

ных. В целом, это подчеркивает необходимость борьбы с атаками на графовые данные.

C. Алгоритмы атак, основанные на градиенте, нацеленные на узлы

Алгоритмы на основе градиента в основном нацелены на изменение топологии, добавлению/удалению ребер между узлами на основе информации о градиентах различных суррогатных моделей.

• **RL-S2V** [9] – подход, в основе которого лежит обучение с подкреплением, процедура атаки моделируется как марковский процесс принятия решений с конечным горизонтом. Метод атаки RL-S2V обучается обобщенному методу атаки на графовую структуру, и как следствие его можно отнести к атаке типа «черного ящика».

• **PGD (Projected Gradient Descent), Min-Max** [10] – авторы представляют новый оптимизированный метод атаки на топологию GNN путем внесения изменений в ребра графа, который использует градиенты суррогатной модели и снижает сложность обработки данных графа. Рассматривается 2 сценария атаки: а) атака на предварительно определенную GNN и б) атака на переобучаемую GNN. Это позволило создать две новые топологические атаки: топологическую атаку с проективного градиентного спуска (PGD) и топологическую атаку min-max. Экспериментальные результаты показывают, что предложенные атаки превосходят существующие современные атаки. На основе разработанного подхода авторы также предлагают метод обучения GNN с целью повышения их робастности.

• **Meta-Self, Meta-Train** [11] – (GCN на основе градиента) используют мета-градиенты для решения проблемы отравления графа. Метаградиенты (например, градиенты относительно гиперпараметров) получаются путем обратного распространения ошибки на этапе обучения нейронной сети. Основная идея алгоритма атаки противника заключается в том, чтобы рассматривать матрицу структуры графа как гиперпараметр и вычислять градиент потерь атакующего после обучения по отношению к нему.

• **Greedy GAN** [12] – (жадный GCN и GAN) базируется на создании матриц смежности и характеристик фальшивых вершин, которые будут введены в граф для неправильной классификации целевых моделей.

• **FGA** [13] (Fast Gradient Attack) – (GCN на основе градиента) использует GCN в качестве суррогатной модели для извлечения информации о градиентах и, таким образом, генерирует граф противника. Эксперименты на реальных данных демонстрируют, что FGA ведет себя лучше, чем некоторые базовые методы, т. е. представление (эмбеддинг) сети может быть легко искажено с помощью FGA путем изменения всего нескольких реберных связей, что позволяет достичь самых современных показателей в атаке.

D. Алгоритмы атак связанных с предсказанием ребер

Как уже упоминалось предсказание связей – фундаментальная исследовательская проблема в сетевом анализе, поэтому атаки, нацеленные на эту уязвимую точку, гораздо более интересны с точки зрения атак на ребра.

• **CTR (Closed-Triad-Removal), OTC (Open-Triad-Creation)** [14] – авторы изучают связи между вершинами используя меру соседства на основе структуры графа и предлагают эвристические алгоритмы, позволяющие обойти обнаружение атаки. CTR и OTC два эвристических алгоритма, которые работают за полиномиальное время. Первый, называемый CTR, фокусируется на удалении ребер, так чтобы удалить «треугольники» из графа, а второй, называемый OTC, фокусируется на добавлении ребер, наоборот, чтобы добавить «треугольники» в граф. В первом случае манипуляции производятся для того чтобы скрыть некоторые связи и сходства, а в друго напротив, вынести на поверхность ложные связи. Схема работы таких эвристик проиллюстрирована на рисунках 7 и 8.

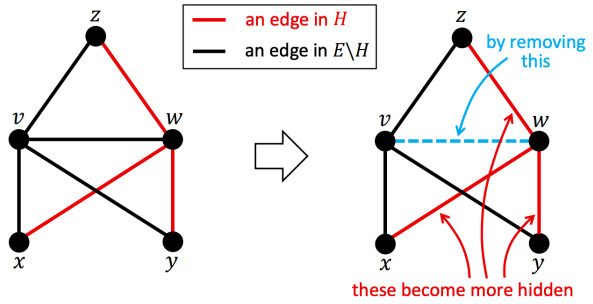

Рис. 7: Иллюстрация основной идеи, лежащей в основе эвристики CTR. Здесь, удаляя ребро (v, w), уничтожаются из сети три треугольника: содержащий вершины v, w, x, другой, содержащий v, w, y, и третий составленный из v, w, z. Это приводит к снижению меры сходства пар (x, w), (w, y) и (w, z). [14].

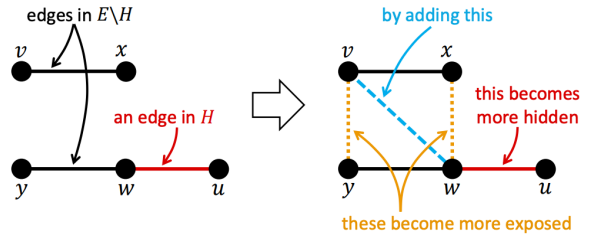

Рис. 8: Иллюстрация основной идеи, лежащей в основе эвристики OTC. Здесь добавление (v, w) создает две открытые триады: одна содержит узлы x, v, w, другая – v, w, y. Соответственно, оценки сходства (x, w) и (y, v) увеличиваются, а для (w, u) – уменьшаются. [14].

Эмпирическая оценка на примере социального графа показала, что обе эвристики эффективны на практике, хотя CTR кажется более эффективной, чем OTC, что позволяет предположить, что для того, чтобы скрыть отношения в социальном графе, установление «недружелюбия» тщательно отобранных людей может обеспечить лучшую маскировку отношений в социальном графе, чем дружба с новыми пользователями.

• **IGA** [15] (Iterative Gradient Attack) – (Graph Auto Encoder на основе градиента) итеративный метод атаки, основанный на информации о градиенте в обученном граф-автоэнкодере. Это одна из первых работ, которая посвящена разработке состязательной атаки с предсказанием ссылок. Результаты экспериментов показывают,

что IGA достаточно эффективен для проведения атаки на различные методы предсказания ссылок, включая методы глубокого обучения и классические методы на основе сходства. IGA может быть использована как метод защиты конфиденциальности или метрика для оценки устойчивости методов предсказания связей. Поскольку для вычисления градиента необходимо использовать всю матрицу смежности, атака противника на сеть большого масштаба довольно сложна из-за ограничения по памяти.

• **TGA-Tra, TGA-Gre** [16] (Time-aware Gradient Attack) – в этой работе авторы используют информацию о градиентах суррогатной модели и исследуют атаки противника на предсказание связей для динамических графов (DNLP – dynamic network link prediction). Предлагаемый метод атаки, а именно градиентная атака с учетом времени (TGA), использует информацию о градиенте, полученную в результате глубокого динамического представления сети (DDNE – deep dynamic network embedding) на разных снепшотах, для перезаписи нескольких связей, чтобы заставить DDNE не предсказать целевые связи. Предложены 2 реализации TGA: одна основана на поиске в глубину – TGA-Tra (traverse), а другой упрощен с помощью жадного поиска для повышения эффективности – TGA-Gre (greedy). Эксперименты показывают выдающуюся производительность TGA в атаке на DNLP.

• **UNAttack** [17] – (Атака отравлением вершины) метод внедрения поддельных пользователей для ухудшения показателей рекомендации в сценарии рекомендаций, рассматривая взаимодействие между пользователями и товарами как граф и рассматривая его как задачу прогнозирования связей. В статье описано, как рассчитать оптимальную атаку отравлением данных, а именно UNAttack. Исследователи ввели в рекомендательные системы несколько хорошо продуманных поддельных пользователей, чтобы целевые товары были рекомендованы как можно большему числу пользователей. Основные достижения в данной работе: предложена общая и математическая основа для оптимальной атаки отравлением данных против рекомендательных систем; исследователи формулируют атаку отравлением данных против моделей, основанных на соседстве, в виде оптимизационной задачи, и представили её решение таким образом, чтобы генерировать более эффективных поддельных пользователей. Авторами проведена серия экспериментов на открытых данных в целях демонстрации работы разработанного подхода. Количественный и качественный анализ показывает, что метод атаки может достичь хороших результатов не только в системах рекомендаций на основе соседства, но и в других алгоритмах коллаборативной фильтрации (например, рекомендательная система с использованием байесовского персонализированного рейтинга).

E. Сводная таблица алгоритмов атак на ГНС

Ниже приведена сводная таблица 1 рассмотренных алгоритмов атак на вершины и ребра графа с упоминанием алгоритма, на которой основана атака.

Модель	**Алгоритм**	**Атакуемая цель**
Nettack [5]	Жадный поиск и градиент на основе GCN	Класификация вершин
RL-S2V [9]	Обучение с подкреплением	Классификация вершин и графов
PGD, Min-Max [10]	Градиент на основе GCN	Классификация вершин
Meta-Self, Meta-Train [11]	Градиент на основе GCN	Классификация вершин
Greedy GAN [12]	Градиент на основе GCN + GAN	Классификация вершин
FGA [13]	Градиент на основе GCN	Классификация вершин и детектирование сообществ
CTR, OTC [14]	Мера соседства на основе структуры графа	Предсказание ребра
IGA [15]	Градиент на основе GAE	Предсказание ребра
TGA-Tra, TGA-Gre [16]	Градиент на основе DDNE (Deep Dynamic Network Embedding)	Предсказание ребра
UNAttack [17]	Атака отравлением вершины, градиент на основе меры сходства	Система рекомендаций

Таблица 1: Сводная таблица алгоритмов атак на ГНС.

III. Алгоритмы и методы защиты от атак на ГНС

Предложенные методы атаки заставили исследователей осознать важность устойчивости моделей глубокого обучения. Относительно недавно были предложены и некоторые методы защиты.

Иначе говоря, цель защиты заключается в том, чтобы производительность модели сохраняла определенную стабильность на данных, которые подвергаются злонамеренному нарушению, и такое свойство модели называют робастностью. Хотя были предложены некоторые модели защиты, но четкого и общепринятого определения проблемы защиты не существует.

Формально защиту можно определить следующим образом: пусть $\tilde{f}$ функция глубокого обучения с функцией потерь $\tilde{\mathcal{L}}$ предназначенная для защиты, она получает граф с внедренными изменениями, либо нет.

$$\underset{\hat{G}\in\Psi(G)\cup G}{minimize} \sum_{t_i\in\tau} \tilde{\mathcal{L}}(\tilde{f_{\theta^*}}(\hat{G}^{t_i}, X, t_i), y_i) \ ,$$

$$\text{здесь } \theta^* = \underset{\theta}{argmin} \sum_{v_j\in S_L} \tilde{\mathcal{L}}(\tilde{f_\theta}(\tilde{G}^{v_j}, X, v_j), y_j),$$

где $\tilde{G} \in \Psi(\mathcal{G}) \cup \mathcal{G}$ – робастный граф $\hat{G}$, либо «чистый» граф G (это зависит от того, была ли модель подвержена атаке или нет) [8].

Таким образом наблюдается обратная ситуация: если, атакуя злоумышленник стремится максимизировать потери f, что приводит к ухудшению эффективности предсказания, то для защиты нужно потери минимизировать.

Также, данное формальное определение демонстрирует, что как таковых защит в классическом понимании нет, есть лишь повышение устойчивости, ровно так же, как и атака снижает этот показатель.

A. Классификация методов защиты

Как таковой четкой и общепринятой классификации методов защит от таких атак нет, так как формально защита предполагает увеличение устойчивости (робастности) моделей, зная поведение конкретных атак. Авторы работы [8] предлагают классифицировать эти способы повышения устойчивости моделей на основе описанных далее интуитивных соображений, которые встречаются в работах по увеличению робастности моделей.

1. Защита на основе препроцессинга. Манипулирование входными данными оказывает большое влияние на

производительность модели. Кроме того, предварительная обработка исходных данных не зависит от структуры модели и методов обучения, что обеспечивает значительную масштабируемость и переносимость. Также предобработка может касаться и структуры графа.

В работе [18] авторы пытаются отбросить ребра, соединяющие узлы с низким показателем меры сходства, что может снизить риск атаки на эти ребра и почти не вредит производительности модели. Данный механизм защиты является общим, в то время как меры сходства могут варьироваться среди различных наборов данных, в частности, для других типов признаков возможно использовать различные меры сходства, такие как косинусное расстояние или коэффициент корреляции. Авторы же используют коэффициент Жаккара.

2. Защита на основе структуры модели. В дополнение к исходным данным структура модели также имеет решающее значение в её производительности. Некоторые существующие подходы изменяют структуру модели, например, GCN, для повышения устойчивости.

RGCN [19] – вместо представления вершин в виде векторов использует Гауссовы распределения в качестве скрытых представлений узлов в каждом сверточном слое. Таким образом, когда граф подвергается атаке, эта модель может автоматически поглотить эффекты изменений в дисперсии Гауссовых распределений. Более того, для устранения распространения атак противника в GCN-ах предлагается механизм внимания на основе дисперсии, т.е. присвоение различных весов окрестностям узлов в зависимости от их дисперсий при выполнении сверток (Рисунок 9). Обширные экспериментальные результаты демонстрируют, что предложенный метод может эффективно повысить устойчивость GCN.

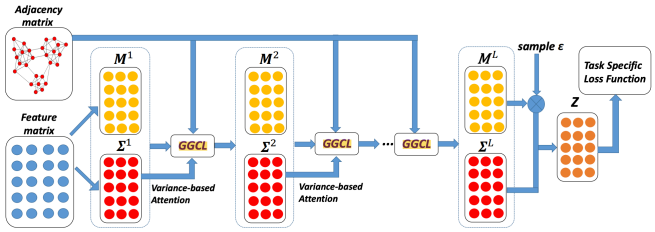

Рис. 9: Схема подхода RGCN. В скрытых слоях используется светочный слой на основе Гауссовых распределений Gaussian-based Graph Convolution Layer (GGCL) использующий механизм внимания на основе дисперсии [19].

3. Защита на основе состязательного обучения. Состязательное обучение показало свою эффективность. Некоторые исследователи успешно адаптируют состязательное обучение из других областей в графовые модели для повышения устойчивости моделей. Выделяют два типа состязательного обучения:

- *Обучение с состязательными целями.* Некоторые методы состязательного обучения постепенно оптимизируют модель непрерывным минимаксным способом под руководством двух противоположных целевых функций (минимизации и максимизации) SVAT, DVAT [20], GCN-GATV [21], беря в основу тот факт, что обновление параметров в GCN происходит только на основе вершин с метками классов, без использования непомеченных данных. В этой работе применяется Virtual Adversarial Training (VAT), состязательный метод регуляризации, основанный на данных как с метками, так и без них, для снижения потерь GCN. Sparse и Dense обозначают, на какие признаки (плотные или разреженные – те, где много нулей в векторе) применяются внедрения шумов.
- *Обучение на состязательных примерах.* В других моделях на основе состязательного подхода во время обучения модели подаются состязательные примеры, что помогает модели научиться адаптироваться к состязательным примерам и, таким образом, уменьшить негативное воздействие этих потенциальных образцов атак [22].

4. Защита на основе целевой функции. Как простой и эффективный метод, в основе которого изменение целевой функции играет важную роль в повышении устойчивости модели. Во многих существующих работах в данном направлении исследователи пытаются обучить устойчивую модель против состязательных атак путем оптимизации целевой функции.

5. Защита на основе обнаружения атаки. Некоторые подходы сосредоточены на обнаружении атак противника или сертификации устойчивости модели или узла графа. Хотя эти методы, основанные на обнаружении, не могут напрямую улучшить устойчивость модели, они могут служить в качестве мониторинга, который постоянно следит за безопасностью модели и подает сигнал тревоги при обнаружении атаки.

GNN обученная с помощью RH-U [23] (Robust Hinge Loss with Unlabeled) – определяет, является ли вершина устойчивой, а устойчивый узел означает, что он не пострадает при проведенной атаке. Авторы предлагают своеобразный сертификат – метод подтверждения (не)робастности графовых сверточных сетей по отношению к помехам атрибутов вершин.

6. Гибридная защита. Как следует из названия, это методы защиты, которые состоят из двух или более типов различных алгоритмов защиты, упомянутых выше. Многие исследователи гибко комбинируют несколько типов алгоритмов защиты для достижения лучшей производительности, тем самым смягчая ограничения, связанные с использованием только одного метода. В данную категорию, в качестве примера, можно отнести тот же GNN обученный с помощью RH-U [23].

В работах [24], [25] исследователи предлагают три основные категории методов защиты GNN, не ограничиваясь тем, что некоторые подходы к защите могут соответствовать нескольким категориям.

1. Методы, улучшающие граф. К этому классу относят подходы предобработки графа с целью повысить робастность модели в процессе обучения. Эти изменения касаются механизмов, предшествующих самой GNN, например передача сообщений в графе. Авторы подразделяют эти методы на контролируемые и неконтролируемые.

Контролируемые включают в себя такую предобработку графа, которая не связана с обучением, а опирается на какие-либо свойства графа, его структуру, характеристики вершин. К этой категории можно отнести рассмотренный ранее подход Jaccard-GCN [18].

Неконтролируемые же напротив, имеют цель сделать граф обучаемым, а именно сделать таковой матрицу смежности, зачастую с дополнительными условиями регуляризации вводящие экспертные предположения о робастности. В качестве примера приводится Property GNN [26] (или ProGNN), суть которого в том, чтобы очистить граф от пертурбаций и рассматривать такую матрицу смежности в качестве обучаемого параметра. Этапы проиллюстрированы на рисунке 10

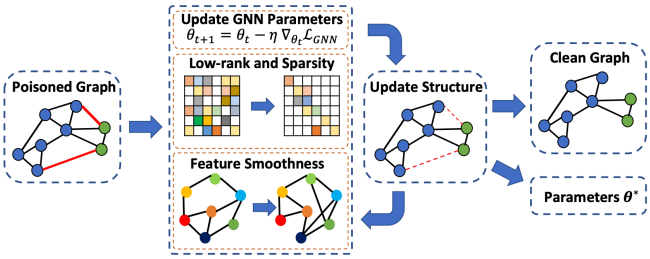

Рис. 10: Общая схема Pro-GNN. Пунктирные линии указывают на меньшие веса [26].

Авторы делают упор на том, что реальные графы часто обладают определенными свойствами, такими как разреженность и низкий ранг матрицы смежности, а также смежные вершины имеют схожие атрибуты.

2. Методы, улучшающие обучение. Эти подходы позволяют улучшить обучение - без изменения архитектуры с целью повышения робастности обучаемых θ^*-параметров. Подразделяют на робастное и последующее обучение. Первое подразумевает альтернативные схемы обучения и функций потерь, которые нацелены на правильную классификацию синтетических состязательных данных в обучающей выборке. Принципы дальнейшего обучения не имеют строгого математического определения и объекта обучения. Примером такого подхода является предварительное обучение весов GNN на возмущенных графах в работе.

3. Улучшение архитектуры. Авторы работы [24] подразделяют эти методы на подходы:

- основанные на правилах, которые обычно используют некоторые метрики;
- статистические подходы, например рассмотренный ранее RGCN [19];
- робастные агрегации, заключаются в замене стандартной агрегирующей функции передачи сообщений (обычно среднего значения) на более робастную альтернативу, например усеченное среднее или медиана.

B. Резюме по методам повышения устойчивости (робастности) ГНС

В настоящее время большинство методов повышения робастности ГНС основаны на двух аспектах: устойчивость метода обучения или устойчивость структуры модели. Среди них методы обучения в основном представляют собой состязательное обучение, а многие улучшения структуры модели осуществляются с помощью механизма внимания, используемого, например в рекурентных и сверточных сетях. Кроме того, есть некоторые исследования, которые напрямую не улучшают устойчивость ГНС, но пытаются проверить устойчивость или пытаются обнаружить данные, которые отравлены. Такой подход крайне интересен, т. к. идейно похож на методы, используемые для мониторинга атак на инфраструктуру информационных систем.

Две классификации, приведенные авторами в [8] и [24] в некоторой степени пересекаются, но в то же время однозначно отнести какой-либо подход в одну из предлагаемых категорий кажется затруднительным – один и тот же подход можно отнести во многие классы.

IV. Заключение

Хотя модели глубокого обучения на графах достигли замечательных результатов в различных задачах анализа графов, например, в классификации узлов, предсказании связей и кластеризации графов, однако они демонстрируют неопределенность и ненадежность против состязательных примеров. Данная проблема касается не только моделей машинного обучения на графах, но и моделей глубокого обучения вообще.

Данная тематика в последние несколько лет активно исследуется, уже есть и отчетливая таксономия атак, но видно, что все не так хорошо обстоит с методами защиты (или же повышением робастности). На сегодняшний день использование систем машинного обучения в критических приложениях невозможно без решения вопросов об устойчивости используемых моделей. Также на данный момент нет универсальных решений в данной области, поэтому высока актуальность исследований конкретно в этом направлении.

Для критических применений остро стоит вопрос сертификации систем, моделей и наборов данных, а учитывая информационную содержательность моделей графовых нейронных сетей, для них этот вопрос встает еще более остро (например, уже разработаны атаки на рекомендательные системы, что является актуальной проблемой для многих бизнесов).

Библиография

[1] Д.Е. Намиот, Е.А. Ильюшин и И.В. Чижов. “Атаки на системы машинного обучения - общие проблемы и методы”. В: *International Journal of Open Information Technologies* 10.3 (2022), с. 17—22.

[2] Wei Jin и др. “Adversarial Attacks and Defenses on Graphs”. В: *SIGKDD Explor. Newsl.* (2021), с. 19—34.

[3] Dan Hendrycks и др. “Unsolved problems in ml safety”. В: *arXiv preprint arXiv:2109.13916* (2021).

[4] Д.Е. Намиот, Е.А. Ильюшин и И.В. Чижов. “Текущие академические и индустриальные проекты, посвященные устойчивому машинному обучению”. В: *International Journal of Open Information Technologies* 9.10 (2021), с. 35—46.

[5] Daniel Zügner, Amir Akbarnejad и Stephan Günnemann. “Adversarial attacks on neural networks for graph data”. В: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, с. 2847—2856.

[6] Shen Wang и др. “Adversarial defense framework for graph neural network”. В: *arXiv preprint arXiv:1905.03679* (2019).

[7] Lichao Sun и др. “Adversarial attack and defense on graph data: A survey”. В: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[8] Liang Chen и др. “A survey of adversarial learning on graphs”. В: *arXiv preprint arXiv:2003.05730* (2020).

[9] Hanjun Dai и др. “Adversarial attack on graph structured data”. В: *International conference on machine learning*. PMLR. 2018, с. 1115—1124.

[10] Kaidi Xu и др. “Topology attack and defense for graph neural networks: An optimization perspective”. В: *arXiv preprint arXiv:1906.04214* (2019).

[11] Daniel Zügner и Stephan Günnemann. *Adversarial Attacks on Graph Neural Networks via Meta Learning*. 2019. arXiv: 1902.08412 [cs.LG].

[12] Xiaoyun Wang и др. “Attack graph convolutional networks by adding fake nodes”. В: *arXiv preprint arXiv:1810.10751* (2018).

[13] Jinyin Chen и др. “Fast gradient attack on network embedding”. В: *arXiv preprint arXiv:1809.02797* (2018).

[14] Marcin Waniek и др. “Attack tolerance of link prediction algorithms: How to hide your relations in a social network”. В: *arXiv preprint arXiv:1809.00152* (2018).

[15] Jinyin Chen и др. “Link prediction adversarial attack”. В: *arXiv preprint arXiv:1810.01110* (2018).

[16] Jinyin Chen и др. “Time-aware gradient attack on dynamic network link prediction”. В: *IEEE Transactions on Knowledge and Data Engineering* (2021).

[17] Liang Chen и др. “Data poisoning attacks on neighborhood-based recommender systems”. В: *Transactions on Emerging Telecommunications Technologies* 32.6 (2021), e3872.

[18] Huijun Wu и др. “Adversarial examples on graph data: Deep insights into attack and defense”. В: *arXiv preprint arXiv:1903.01610* (2019).

[19] Dingyuan Zhu и др. “Robust graph convolutional networks against adversarial attacks”. В: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, с. 1399—1407.

[20] Ke Sun и др. “Virtual adversarial training on graph convolutional networks in node classification”. В: *Pattern Recognition and Computer Vision: Second Chinese Conference, PRCV 2019, Xi’an, China, November 8–11, 2019, Proceedings, Part I 2*. Springer. 2019, с. 431—443.

[21] Fuli Feng и др. “Graph adversarial training: Dynamically regularizing based on graph structure”. В: *IEEE Transactions on Knowledge and Data Engineering* 33.6 (2019), с. 2493—2504.

[22] Jinyin Chen и др. “Can adversarial network attack be defended?” В: *arXiv preprint arXiv:1903.05994* (2019).

[23] Daniel Zügner и Stephan Günnemann. “Certifiable robustness and robust training for graph convolutional networks”. В: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, с. 246—256.

[24] Felix Mujkanovic и др. “Are Defenses for Graph Neural Networks Robust?” В: *Advances in Neural Information Processing Systems* 35 (2022), с. 8954—8968.

[25] Stephan Günnemann. “Graph neural networks: Adversarial robustness”. В: *Graph Neural Networks: Foundations, Frontiers, and Applications* (2022), с. 149—176.

[26] Wei Jin и др. “Graph structure learning for robust graph neural networks”. В: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, с. 66—74.

Review and comparative analysis of attack and defence algorithms on graph-based ANN architectures

Dovlet Kirzhinov, Eugene Ilyushin

***Abstract*—Graphs are all around us; objects in the real world are often defined in terms of their relationships to other objects. A set of objects and the relationships between them are naturally expressed as a graph. Due to the meaningfulness of such a representation of data, which is generated by various artificial and natural processes, training neural networks on such data is a powerful tool. The spectrum of attacks on GNN (Graph Neural Network) architectures is very wide, and for each of the attack methods, it is required to develop and define effective defense techniques, and to investigate the attacks in terms of computational complexity for their possible application on large graphs used in real application cases. This paper is a survey in which the security of such graph neural network architectures is discussed, including attack algorithms and how to defend against them by improving robustness. It also provides some classification of these methods according to various criteria and a review of existing works on this topic.**

***Keywords*—Graph Neural Networks, Adversarial Attack and Defense, Robustness.**

References

[1] D.E. Namiot, E.A. Ilyshin, and I.V. Chizhov. “Attacks on Machine Learning Systems - Common Problems and Methods”. In: *International Journal of Open Information Technologies* 10.3 (2022), pp. 17–22.

[2] Wei Jin et al. “Adversarial Attacks and Defenses on Graphs”. In: *SIGKDD Explor. Newsl.* (2021), pp. 19–34.

[3] Dan Hendrycks et al. “Unsolved problems in ml safety”. In: *arXiv preprint arXiv:2109.13916* (2021).

[4] D.E. Namiot, E.A. Ilyshin, and I.V. Chizhov. “On-going academic and industrial projects dedicated to robust machine learning”. In: *International Journal of Open Information Technologies* 9.10 (2021), pp. 35–46.

[5] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. “Adversarial attacks on neural networks for graph data”. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 2847–2856.

[6] Shen Wang et al. “Adversarial defense framework for graph neural network”. In: *arXiv preprint arXiv:1905.03679* (2019).

[7] Lichao Sun et al. “Adversarial attack and defense on graph data: A survey”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[8] Liang Chen et al. “A survey of adversarial learning on graphs”. In: *arXiv preprint arXiv:2003.05730* (2020).

[9] Hanjun Dai et al. “Adversarial attack on graph structured data”. In: *International conference on machine learning*. PMLR. 2018, pp. 1115–1124.

[10] Kaidi Xu et al. “Topology attack and defense for graph neural networks: An optimization perspective”. In: *arXiv preprint arXiv:1906.04214* (2019).

[11] Daniel Zügner and Stephan Günnemann. *Adversarial Attacks on Graph Neural Networks via Meta Learning*. 2019. arXiv: 1902.08412 [cs.LG].

[12] Xiaoyun Wang et al. “Attack graph convolutional networks by adding fake nodes”. In: *arXiv preprint arXiv:1810.10751* (2018).

[13] Jinyin Chen et al. “Fast gradient attack on network embedding”. In: *arXiv preprint arXiv:1809.02797* (2018).

[14] Marcin Waniek et al. “Attack tolerance of link prediction algorithms: How to hide your relations in a social network”. In: *arXiv preprint arXiv:1809.00152* (2018).

[15] Jinyin Chen et al. “Link prediction adversarial attack”. In: *arXiv preprint arXiv:1810.01110* (2018).

[16] Jinyin Chen et al. “Time-aware gradient attack on dynamic network link prediction”. In: *IEEE Transactions on Knowledge and Data Engineering* (2021).

[17] Liang Chen et al. “Data poisoning attacks on neighborhood-based recommender systems”. In: *Transactions on Emerging Telecommunications Technologies* 32.6 (2021), e3872.

[18] Huijun Wu et al. “Adversarial examples on graph data: Deep insights into attack and defense”. In: *arXiv preprint arXiv:1903.01610* (2019).

[19] Dingyuan Zhu et al. “Robust graph convolutional networks against adversarial attacks”. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 1399–1407.

[20] Ke Sun et al. “Virtual adversarial training on graph convolutional networks in node classification”. In: *Pattern Recognition and Computer Vision: Second Chinese Conference, PRCV 2019, Xi’an, China, November 8–11, 2019, Proceedings, Part I 2*. Springer. 2019, pp. 431–443.

[21] Fuli Feng et al. “Graph adversarial training: Dynamically regularizing based on graph structure”. In: *IEEE Transactions on Knowledge and Data Engineering* 33.6 (2019), pp. 2493–2504.

[22] Jinyin Chen et al. “Can adversarial network attack be defended?” In: *arXiv preprint arXiv:1903.05994* (2019).

[23] Daniel Zügner and Stephan Günnemann. “Certifiable robustness and robust training for graph convolutional networks”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 246–256.

[24] Felix Mujkanovic et al. “Are Defenses for Graph Neural Networks Robust?” In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 8954–8968.

[25] Stephan Günnemann. “Graph neural networks: Adversarial robustness”. In: *Graph Neural Networks: Foundations, Frontiers, and Applications* (2022), pp. 149–176.

[26] Wei Jin et al. “Graph structure learning for robust graph neural networks”. In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, pp. 66–74.