В. В. Печеный, А. А. Пономарев, Т. И. Паюсова
Тюменский государственный университет, г. Тюмень
УДК 004.052.2

ЗАЩИТА МОДЕЛЕЙ КОМПЬЮТЕРНОГО ЗРЕНИЯ ОТ ADVERSARIAL-АТАК

Аннотация. В данной работе рассмотрена проблема защиты моделей компьютерного зрения (в частности, нейросетей-классификаторов) от adversarial-атак. Поставлена задача проверки эффективности двух новых методов защиты путем попытки их обхода с помощью нового метода атаки.

Ключевые слова: adversarial-примеры, искусственные нейронные сети, компьютерное зрение, машинное обучение.

Введение. Повсеместное применение машинного обучения не могло не привести к появлению различных видов атак, задача которых заключается в отклонении поведения модели от ожидаемого. В компьютерном зрении особое беспокойство вызывают adversarial-атаки — отправка на вход готовой модели adversarial-примеров. Adversarial-примеры (противоречивые примеры) — это специально модифицированные входные данные (в контексте компьютерного зрения — изображения), которые некорректно классифицируются нейросетью (рис. 1) [1].

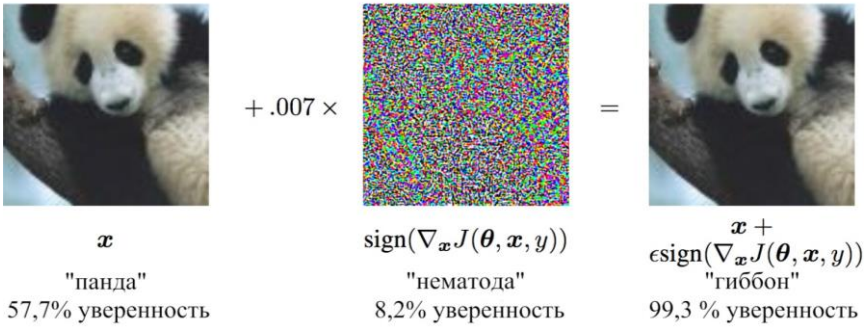

Рис. 1. Adversarial-пример

В этом примере изображение модифицировано целиком. Однако adversarial-примером может быть и частично модифицированное изображение. Дополнительной проблемой является тот факт, что adversarial-пример может быть образован не только модифицированием самого изображения.

Adversarial-атака может быть реализована в реальном мире без применения каких-либо специальных средств путем непосредственного воздействия на распознаваемый объект. Например, на него можно наклеить стикер со специальным изображением, которое может отклонить результат классификации как в случайную, так и в определенную сторону. На рис. 2 на знак «Стоп» было наклеено два стикера. Модель больше не способна обнаружить его и определяет только сами стикеры как множество «бутылок» (bottle) [2].

Рис. 2. Знак «Стоп», на котором реализована adversarial-атака

На 31-м Симпозиуме по Безопасности USENIX было представлено два новых метода защиты нейросетей-классификаторов от adversarial-примеров: **PatchCleanser** — сертифицируемо надежная

защита от adversarial-примеров для любого классификатора изображений [3], **Blacklight** — масштабируемая защита нейронных сетей от blackbox-атак на основе цепочки запросов [4].

PatchCleanser фильтрует входное изображение с помощью алгоритма нанесения двух масок, чтобы оно было определено корректно. Его работоспособность основана на том предположении, что хорошо обученные классификаторы способны корректно определять объект даже если он виден не полностью. Общая схема работы показана на рис. 3 (перевод оригинального рисунка из статьи).

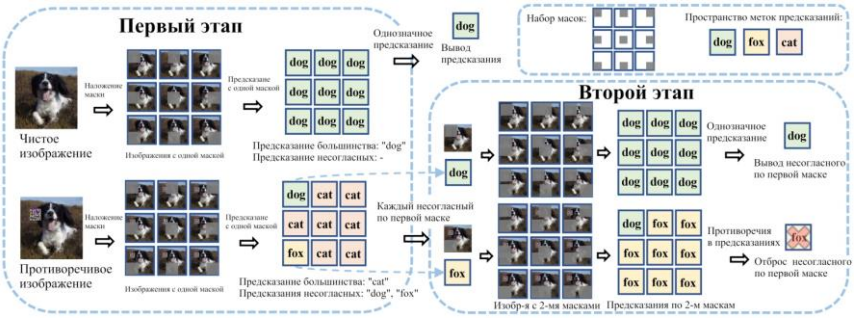

Рис. 3. Общая схема работы PatchCleanser

Первый этап: на базе входного изображения создается набор маскированных. Если исходное изображение изначально «чистое», то результат классификации (предсказание) каждого из полученного набора будет однозначным. Иначе, предсказаний из этого набора будет несколько, поскольку как минимум на одном из маскированных изображений модифицированная часть будет закрыта маской. То предсказание, которое выпало на большинство изображений из набора, отбрасывается из рассмотрения, после чего выполняется второй этап на каждом из оставшихся (на рис. — несогласные).

Второй этап: из «несогласного» маскированного изображения создается еще один набор. В этом наборе у каждого изображения по две маски. Если выбранное «несогласное» изображение было очищено на первом этапе, то предсказание нового набора будет однозначным и совпадать с предсказанием изображения из первого

этапа. Если предсказание все еще неоднозначное, то выбранное изображение из первого цикла не является очищенным и нужно повторить второй этап на следующем изображении из первого этапа. Далее повторять, пока либо не будет обнаружено однозначное предсказание, либо не закончатся изображения из первого цикла. Во втором случае в качестве результата возвращается предсказание большинства.

Blacklight использует иной подход, который, по сравнению с PatchCleanser, несколько ограничивает широту его применения до систем ML-as-a-Service (машинное обучение как услуга). Такие системы обычно получают на вход данные непосредственно от пользователей (например, они отправляют на систему изображение — запрашивают классификацию), а затем возвращают либо конечный ответ (decision-based), либо характеристику распределения возможных ответов (score-based). Blackbox-атака на основе цепочки запросов заключается в постоянной отправке системе похожих изображений, отличающихся друг от друга лишь примененным изменением, с целью создания работающего adversarial-примера, причем атакующий имеет доступ к модели только на уровне пользователя — возможность отправки запросов и получения ответов.

Схема работы Blacklight представлена на рис. 4 (перевод оригинального рисунка из статьи). Авторы реализовали движок, вычисляющий «отпечаток содержания», по принципу работы сопоставимый с хэш-функцией, но в отличие от нее, значение «отпечатка» не изменяется при достаточно похожих изображениях. Blacklight ведет историю запросов, также сохраняя в ней «отпечатки» изображений.

Рис. 4. Общая схема работы Blacklight

Если на вход было подано достаточно похожее на любое из предыдущих изображение, с помощью «отпечатка содержания» Blacklight определит такое действие как атаку и заблокирует обработку запросов с этим «отпечатком». Таким образом, даже если атакующий может себе позволить производить атаку из нескольких учетных записей системы, защита будет работать, поскольку блокировка реализована непосредственно на уровне запросов.

Помимо методов защиты был представлен новый метод реализации adversarial-атак: **AutoDA** — автоматизированные итеративные adversarial-атаки на decision-based системы [5].

Авторы этого метода атаки определили набор из десяти простых операций, определяющих поле для генерации нового алгоритма атаки, оптимизированного под целевую модель. Алгоритмы, генерируемые AutoDA, по эффективности способны превзойти лучшие из уже существующих алгоритмов.

Проблема исследования. Так как PatchCleanser и Blacklight были представлены на той же конференции, что и AutoDA, их эффективность не была протестирована против последнего. Вследствие этого была поставлена следующая задача: попытаться с помощью AutoDA обойти PatchCleanser и Blacklight, тем самым определив степень гибкости и расширяемости этих методов защиты.

Материалы и методы. Исходные коды прототипов PatchCleanser, Blacklight и AutoDA выложены их авторами на GitHub [6-8]. На данном этапе ведется работа по разворачиванию стенда, позволяющего протестировать представленные методы защиты.

Результаты. Мы определили следующие гипотезы для проверки на стенде:

1. PatchCleanser возможно обойти, если adversarial-атака реализована на входных изображениях целиком (рис. 1);
2. Можно успешно обойти Blacklight, если модифицируемые изображения будут отличаться друг от друга достаточно сильно, чтобы на каждый из них был вычислен уникальный «отпечаток».

Заключение. Определенные выше гипотезы требуют проверки на стенде, над разворачиванием которого ведется работа. Дополнительно планируется исследование по расширению функционала

PatchCleanser по части борьбы с целиком модифицированными (рис. 1) изображениями, если Гипотеза 1 будет подтверждена.

СПИСОК ЛИТЕРАТУРЫ

1. Goodfellow I. J., Shlens J., Szegedy C. Explaining and harnessing adversarial examples // arXiv preprint arXiv:1412.6572. — 2014.
2. Eykholt K. et al. Note on attacking object detectors with adversarial stickers // arXiv preprint arXiv:1712.08062. — 2017.
3. Xiang C., Mahloujifar S., Mittal P. {PatchCleanser}: Certifiably Robust Defense against Adversarial Patches for Any Image Classifier // 31st USENIX Security Symposium (USENIX Security 22). — 2022. — С. 2065-2082.
4. Li H. et al. Blacklight: Scalable Defense for Neural Networks against {Query-Based}{Black-Box} Attacks // 31st USENIX Security Symposium (USENIX Security 22). — 2022. — С. 2117-2134.
5. Fu Q. A. et al. {AutoDA}: Automated Decision-based Iterative Adversarial Attacks //31st USENIX Security Symposium (USENIX Security 22). — 2022. — С. 3557-3574.
6. GitHub — inspire-group/PatchCleanser: Code for "PatchCleanser: Certifiably Robust Defense against Adversarial Patches for Any Image Classifier". — Текст : электронный // GitHub: [сайт]. — URL: https://github.com/inspire-group/PatchCleanser (дата обращения: 27.05.2023).
7. GitHub — Huiying-Li/blacklight. — Текст: электронный // GitHub : [сайт]. — URL: https://github.com/Huiying-Li/blacklight (дата обращения: 27.05.2023).
8. GitHub — Fugoes/AutoDA. — Текст : электронный // GitHub: [сайт]. — URL: https://github.com/Fugoes/AutoDA (дата обращения: 27.05.2023).