УДК 004.89

ПРИМЕНЕНИЕ ГЕНЕРАТИВНЫХ АТАК ДЛЯ ТЕСТИРОВАНИЯ УСТОЙЧИВОСТИ КЛАССИФИКАТОРОВ

Лейман А.Ф., Старший оператор научной роты, III курс
Военная академия связи имени Маршала Советского Союза С. М. Буденного,
г. Санкт-Петербург

С ростом мощности генеративных моделей, таких как GAN, VAE и трансформерные архитектуры семейства GPT или Diffusion-based подходов, значительно расширились возможности создания реалистичных, но синтетических данных. Это открыло новое направление в исследовании доверенных ИИ-систем – использование генеративных атак для тестирования устойчивости классификаторов [1].

В отличие от классических адверсариальных атак, опирающихся на градиенты или локальные пертурбации входа, генеративные подходы позволяют конструировать глобально правдоподобные, но функционально обманчивые примеры, способные систематически выявлять уязвимости классификационных моделей. Особенно актуальным это становится в высокорисковых сценариях: от анализа медицинских изображений и биометрических данных до модерации текстов, распознавания речи и классификации транзакций. В настоящей работе рассматриваются основные методы генеративных атак, особенности их применения в задачах тестирования и анализируются существующие фреймворки, позволяющие интегрировать такие подходы в процессы разработки и валидации моделей [2].

Генеративные атаки представляют собой стратегию создания входных данных, которые выглядят как нормальные (т.е. соответствуют распределению обучающей выборки), но при этом приводят классификатор к заведомо неверному выводу. В отличие от точечных искажений, они способны подменить семантический смысл или создать новые классы-приманки (decoy classes), оставаясь при этом непротиворечивыми по отношению к контексту задачи. Это делает генеративные атаки особенно опасными и одновременно полезными как инструмент стресс-тестирования моделей, поскольку они затрагивают глобальные аспекты устойчивости – не только локальную гладкость функции, но и концептуальные слабости в представлениях модели [3].

Существуют различные типы генеративных атак, адаптированных под природу данных:

1. Изображения. Наиболее распространённые подходы включают использование GAN (Generative Adversarial Networks) для генерации изображений, которые классификатор ошибочно относит к целевому классу. Например, GAN можно обучить создавать изображения, визуально неотличимые от «кошек», но распознаваемые моделью как «собаки». Такой подход эффективен против моделей в задачах медицинской диагностики (подмена аномалий), распознавания лиц (deepfake), анализа видео (activity spoofing). Особенно популярны так называемые adversarial generative examples, полученные с помощью Conditional GAN или StyleGAN, где генератор оптимизируется не только по правдоподобию, но и по отклику целевой модели.

2. Текст. В области обработки естественного языка генеративные атаки используют языковые модели (GPT, T5, OPT и др.), способные синтезировать тексты, направленные на «обман» классификатора. Например, можно сгенерировать отзыв, который по структуре и лексике выглядит позитивным, но классификатор ошибочно воспринимает его как негативный. Такие атаки позволяют выявлять предвзятость к определённой лексике, недостаточную устойчивость к сарказму, контекстуальные ошибки. В задачах hate speech detection или toxic comment classification генеративные атаки позволяют моделировать реалистичные «пограничные» случаи, критически важные для правовой валидации.

3. Аудио и речь. Здесь применяются VAE и diffusion-модели, способные создавать речевые сигналы, вводящие модель в заблуждение. Например, можно генерировать команды для голосовых помощников, которые воспринимаются моделью как активирующие, но остаются незаметными для человеческого слуха (так называемые скрытые команды). Это особенно актуально в системах безопасности, биометрии и управления.

4. Табличные данные и графы. Генеративные модели (например, табличные GAN, CTGAN, GraphVAE) применяются для синтеза реалистичных, но аномальных транзакций, медицинских показателей, социального поведения. Цель – выявление чрезмерной зависимости модели от отдельных корреляций или статистических шаблонов, которые легко нарушаются генерацией. Такие атаки помогают моделировать сложные случаи fraud detection и рекомендательных систем.

Рассмотрим теперь основные методы интеграции генеративных атак в тестирование:

1. Fine-tuned generator-based pipeline: генератор обучается на полном датасете и дополнительно оптимизируется по функции потерь, зависящей от реакции классификатора. Это позволяет «настраивать» генератор на создание именно тех примеров, которые ломают модель [4].

2. Black-box генерация с последующей фильтрацией: модели вроде GPT используют для генерации большого числа вариантов, а затем отбираются только те, которые вызывают изменение классификации или нарушают порог доверия. Это применимо при отсутствии доступа к внутренним параметрам модели.

3. Zero-shot генерация с шаблонной атакой: используется заранее подготовленный набор prompt-шаблонов, запускаемых в генеративной модели, что позволяет строить корпус «пограничных» примеров – например, токсичный комментарий в форме вопроса [5].

4. Iterative feedback loop: генератор и классификатор работают в цикле – каждая неудачная атака используется для улучшения генерации, а успешные – для пополнения набора тестов.

Метрики оценки для количественной оценки воздействия генеративных атак применяются следующие метрики:

- Attack Success Rate (ASR) – доля успешных атак (смена класса).
- Semantic Similarity – степень семантической близости с оригинальным примером (BERTScore, USE).
- Perplexity (для текста) – сохраняется ли правдоподобие высказывания.
- Fidelity – насколько атака сохраняет легитимность в контексте задачи.
- Confidence Drop – изменение уверенности модели под действием сгенерированного примера.

Для практической реализации генеративных атак активно применяются:

- TextAttack, OpenAttack – для текстов;
- Foolbox, ART – для изображений;
- DeepFool-GAN, AdvGAN – специализированные архитектуры;
- Diff-Attack и DiffusionBench – для диффузионных атак;
- Testron, RobustBench, CheckList – как мета-платформы для организации тестирования.

Интеграция таких атак в жизненный цикл ML-моделей позволяет создавать расширенные тестовые сеты, строить оценки worst-case поведения, проверять модели на устойчивость к сложным сценариям и, в конечном счёте, повышать надёжность и доверие к ИИ-системам. Это особенно важно в контексте регулирования ИИ и сертификации: генеративные атаки позволяют выявлять неочевидные риски, которые не фиксируются обычным тестированием.

Применение генеративных атак для тестирования устойчивости классификаторов становится неотъемлемой частью современной инженерии доверенного ИИ. Эти атаки позволяют выходить за пределы локальных границ

входного пространства и тестировать модели на уровне концептуальной устойчивости – выявляя не только численные, но и смысловые, лексические и логические уязвимости. В перспективе, генеративные методы станут не только инструментом атак, но и основой для формализации worst-case тестирования, автоматической валидации моделей и построения устойчивых архитектур нового поколения.

Список литературы:

1. Ilyas, A. Adversarial Examples Are Not Bugs, They Are Features / A. Ilyas, S. Santurkar, D. Tsipras [et al.] // Advances in Neural Information Processing Systems. – 2019. – Vol. 32.

2. Gilmer, J. Adversarial Spheres / J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, G. E. Dahl // International Conference on Learning Representations Workshop. – 2018.

3. Schott, L. Towards the First Adversarially Robust Neural Network Model on MNIST / L. Schott, J. Rauber, M. Bethge, W. Brendel // International Conference on Learning Representations. – 2019.

4. Song, Y. Constructing Unrestricted Adversarial Examples with Generative Models / Y. Song, R. Shu, N. Kushman, S. Ermon // Advances in Neural Information Processing Systems. – 2018. – Vol. 31.

5. Xiao, C. Generating Adversarial Examples with Adversarial Networks / C. Xiao, B. Li, J. Zhu [et al.] // Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. – 2018. – P. 3905-3911. – DOI 10.24963/ijcai.2018/543.