

УДК 004.89

АВТОМАТИЗАЦИЯ ПРОВЕРКИ УСТОЙЧИВОСТИ МОДЕЛЕЙ К ИЗМЕНЕНИЮ ВХОДНЫХ ДАННЫХ

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

В последние годы всё больше внимания в области искусственного интеллекта и машинного обучения уделяется не только точности моделей, но и их устойчивости — способности адекватно реагировать на изменение входных данных в условиях неопределённости. Это связано с тем, что даже высокоточные нейросети, обученные на обширных выборках, могут демонстрировать аномальное, непредсказуемое или даже опасное поведение при минимальных изменениях входа. В таких случаях модели становятся уязвимыми не только к адверсариальным атакам, но и к случайным шумам, пропущенным значениям, незначительным отклонениям в распределении, или даже к грамматическим и лексическим перестановкам в текстах. Проблема становится особенно актуальной в приложениях, где стабильность и надёжность критичны: медицинская диагностика, системы автономного управления, финансовое моделирование, юридические экспертизы. На этом фоне автоматизация процедур проверки устойчивости моделей к изменению входных данных превращается в стратегически важную задачу как для исследователей, так и для инженеров, занимающихся развёртыванием ИИ-систем в реальной среде [1].

В традиционной практике устойчивость модели оценивается вручную, через подбор искусственных примеров, визуальный анализ, ограниченное число стресс-тестов. Такой подход является трудоёмким, плохо воспроизводимым и заведомо неполным. Он не охватывает всё возможное пространство искажений, не позволяет сравнивать модели между собой в формализованном виде и не предоставляет уверенности в поведении модели за пределами обучающего распределения. В этих условиях возникает необходимость в создании автоматизированных систем тестирования, которые могли бы формализованно, масштабируемо и воспроизводимо оценивать устойчивость моделей к заранее заданному классу изменений.

Под изменениями входных данных здесь понимается широкий спектр возможных трансформаций — от синтетических шумов (гауссовский, пертурбативный, импульсный) и математических искажений (нормализация, масштабирование, обнуление признаков) до более сложных логических, семанти-

ческих и структурных изменений: например, удаление важных признаков, перестановка элементов, подмена единиц измерения, удаление зависимостей. Автоматизированная проверка должна учитывать не только факт изменения выхода модели, но и его степень – например, сдвиг распределения вероятностей, изменение ранга в задаче ранжирования, колебание доверительных интервалов. Кроме того, в ряде задач важно проверять, сохраняется ли интерпретируемость модели при изменениях – особенно в системах, работающих под нормативным контролем.

Современные автоматизированные подходы строятся на основе библиотек и фреймворков, специально предназначенных для генерации тестов и оценки реакций модели. Они включают модули для систематического порождения входов (например, генерация мутаций текста, синтетика изображений, структурные изменения графов), подсистемы автоматического сравнения предсказаний, модули визуализации результатов и отчёты о чувствительности. Такие системы позволяют воспроизводимо проводить проверку по множеству сценариев: тестировать отдельные блоки модели, оценивать чувствительность к отдельным признакам, проверять поведение в краевых случаях, измерять устойчивость к шумам, вводить метрики доверия и варьировать параметры искажения.

Одним из наиболее перспективных направлений является использование *property-based testing* – подхода, в котором вместо конкретных примеров задаются общие свойства, которым должна удовлетворять модель при изменении входа. Например: «если входовое изображение отличается от оригинала не более чем на 2% пикселей, то предсказание должно сохраняться» или «при удалении необязательного признака класс не должен меняться». Такие свойства формализуются и проверяются автоматически при помощи генераторов входов и процедур сравнения. Этот подход широко применяется в тестировании программного обеспечения и всё активнее проникает в область ML.

Другим важным направлением становится применение дифференцируемых метрик чувствительности, которые позволяют измерять гладкость функции модели по отношению к входу. Например, можно анализировать норму якобиана, определять Lipschitz-постоянную или измерять разброс выходов на окрестности входа. Эти численные показатели могут использоваться как для автоматической диагностики проблем, так и в качестве регуляризаторов в процессе обучения модели. На этой основе строятся метрики типа *sensitivity score*, *local robustness score* и *confidence drift*.

Автоматизация проверки устойчивости требует также наличия референсных тестов и эталонных сценариев, по аналогии с *unit-тестами* в программировании. Современные ML-платформы, такие как TensorFlow Extended (TFX), MLflow и DeepChecks, позволяют встраивать проверки устойчивости в

пайплайны развертывания модели. Так, перед выводом новой версии модели в продакшн можно автоматически запускать батч-атаку, проводить симуляции шума, проверять поведение на обобщённых данных, сравнивать метрики доверия с предыдущими версиями и выдавать отчёт об устойчивости. Это делает возможным не только более безопасное развертывание, но и соответствие нормативным требованиям, таким как AI Act, ISO/IEC 23894, GDPR и внутренним протоколам компаний [2].

Отдельное внимание следует уделить использованию автоматических тестов устойчивости в задачах интерпретируемости. Например, при изменении незначительных признаков (или удалении признаков, не имеющих смысла) интерпретация модели не должна радикально меняться. И наоборот — если смысловой признак был убран, то его вклад должен исчезнуть и в объяснении. Такие тесты позволяют выявлять ложную причинность в объяснениях и оценивать доверие к интерпретациям [3].

В заключение стоит подчеркнуть, что автоматизация проверки устойчивости моделей к изменению входных данных становится важнейшим элементом инженерии доверенного ИИ. Это не просто инструмент отладки, а архитектурный компонент, обеспечивающий безопасное внедрение моделей, мониторинг их поведения и соответствие критериям прозрачности, надёжности и подотчётности. Развитие таких систем требует мультидисциплинарного подхода — с участием специалистов в области статистики, тестирования ПО, кибербезопасности, нормативного регулирования и этики. В перспективе, автоматизированная проверка устойчивости будет неотъемлемой частью стандартизированных пайплайнов разработки ИИ, как unit-тесты стали обязательным компонентом промышленной разработки программного обеспечения [4-6].

Список литературы:

1. Zhang, J. Machine Learning Testing : Survey, Landscapes and Horizons / J. Zhang, M. Harman, L. Ma, Y. Liu // IEEE Transactions on Software Engineering. – 2022. – Vol. 48, № 1. – P. 1-36. – DOI 10.1109/TSE.2019.2962027.
2. Bousquet, O. Stability and Generalization / O. Bousquet, A. Elisseeff // Journal of Machine Learning Research. – 2002. – Vol. 2. – P. 499-526.
3. Xu, H. Robustness and Generalization / H. Xu, S. Mannor // Machine Learning. – 2012. – Vol. 86, № 3. – P. 391-423. – DOI 10.1007/s10994-011-5268-1.
4. Murphy, C. Properties of Machine Learning Applications for Use in Metamorphic Testing / C. Murphy, G. E. Kaiser, L. Hu, L. Wu // Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering. – 2008. – P. 867-872.

5. Arcuri, A. Automated Test Case Generation as a Many-Objective Optimization Problem with Dynamic Selection of the Targets / A. Arcuri // IEEE Transactions on Software Engineering. – 2018. – Vol. 44, № 2. – P. 122-158. – DOI 10.1109/TSE.2017.2654255.

6. Протодяконов, А. В. Асимптотический анализ поведения прикладных моделей машинного обучения : учебное пособие / А. В. Протодяконов, А. В. Дягилева, П. А. Пылов. – Вологда : Инфра-Инженерия, 2023. – 144 с. – ISBN 978-5-9729-1455-5. – EDN APHQME.