Сравнительный анализ моделей машинного обучения для классификации водителей на основе данных датчиков микроэлектромеханических систем

Р.Р. Киямов, М.С. Мосева

Московский технический университет связи и информатики

Аннотация: В данной статье представлен сравнительный анализ моделей машинного обучения, применяемых для классификации водителей на основе данных датчиков микроэлектромеханических систем (МЭМС). Исследование основано на открытом наборе данных “UAH-DriveSet”, содержащем свыше 500 минут записей вождения с разметкой событий агрессивного поведения, таких как резкое торможение, резкий поворот и резкое ускорение. Рассмотрены алгоритмы градиентного бустинга, рекуррентной нейронной сети и сверточной нейронной сети. Особое внимание уделено анализу влияния на эффективность классификации параметров разделения данных методом скользящего окна, таких как размер окон и степень их перекрытия. Проведенное исследование вносит вклад в развитие систем машинного обучения для анализа поведения водителей и создание интеллектуальных решений на основе датчиков МЭМС.

Ключевые слова: анализ поведения водителей, датчики микроэлектромеханических систем, машинное обучение, агрессивное вождение, градиентный бустинг, рекуррентные нейронные сети, сверточные нейронные сети, скользящее окно, классификация водителей.

Введение

Агрессивное поведение водителей на дорогах является одним их ключевых факторов, способствующих увеличению числа ДТП и снижению уровня безопасности дорожного движения. Выявление агрессивного поведения с использованием методов машинного обучения может помочь в разработке систем активной безопасности, персонализированных рекомендаций для водителей и более точных страховых моделей [1].

Современное развитие технологии микроэлектромеханических систем (МЭМС) открывает новые возможности для анализа поведения водителей на основе данных, получаемых в реальном времени. В данной работе был использован открытый набор данных движения автомобилей в реальных условиях, который был собран с помощью МЭМС-датчиков смартфонов — “UAH-DriveSet” [2]. Данные более 500 минут вождения были собраны на

нескольких типах дорог от водителей разных возрастных групп с различными типами транспортных средств, включая полностью электрический автомобиль. Датасет имеет готовую разметку, которая включает в себя три вида событий агрессивного вождения: резкое торможение, резкий поворот и резкое ускорение.

Цель данной работы — исследовать и сравнить эффективность различных моделей машинного обучения для выявления событий агрессивного вождения.

Выбор алгоритмов классификации

Для задачи классификации по данным МЭМС-датчиков требуются алгоритмы машинного обучения, способные работать с табличными данными, представляющими собой как сырые параметры движения, так и агрегированные статистические параметры, полученные методом скользящего окна. Так как для полноценного исследования целесообразно рассмотреть оба подхода, то для дальнейшего исследования были выбраны следующие алгоритмы:

- градиентный бустинг на решающих деревьях (Gradient Boosted Decision Trees - GBDT);
- долгая краткосрочная память (Long Short-Term Memory - LSTM);
- сверточная нейронная сеть (Convolutional Neural Network - CNN).

GBDT представляет собой ансамблевый метод машинного обучения, в котором множество слабых моделей - решающих деревьев - объединяются для создания одной сильной. Каждое последующее дерево обучается на ошибках предыдущих, минимизируя функцию потерь с использованием градиентного спуска. Для эффективной работы GBDT целесообразно выполнить предварительную агрегацию статистических параметров данных методом скользящего окна [3, 4]. В качестве наиболее подходящих

реализаций градиентного бустинга на решающих деревьях были выбраны модели “LightGBM” и “CatBoost”. Модель “LightGBM” использует метод обучения “рост по листьям”, который строит деревья, выбирая для разделения узел с максимальным уменьшением ошибки в отличие от традиционного метода, который добавляет новые узлы равномерно на всех уровнях. Это позволяет обучать модель быстрее и достигать большей точности на тех же данных. “CatBoost” демонстрирует высокую устойчивость к несбалансированным данным, задавая вес для каждого объекта обучающей выборки (объекты редкого класса получают больший вес, а объекты часто встречающегося класса получают меньший вес).

LSTM — это разновидность рекуррентных нейронных сетей, которая специально разработана для обработки временных данных. Основной особенностью LSTM является наличие так называемых "ячеек памяти", которые позволяют сети запоминать информацию на длительные временные интервалы, избегая проблемы затухания градиентов [5, 6]. LSTM обновляет состояние своей памяти с использованием трёх основных компонентов: входного, забывающего и выходного гейтов. Это позволяет модели учитывать временную зависимость в данных. Этот алгоритм хорошо подходит для работы с сырыми данными МЭМС-датчиков поскольку может учитывать их временную корреляцию.

CNN изначально разработаны для анализа данных с локальными зависимостями, таких как изображения, однако их принципы успешно адаптированы для работы с временными рядами. Основной компонент CNN — сверточные слои, которые применяют фильтры (свертки) для выявления локальных особенностей в данных. В случае временных рядов фильтры сканируют последовательность значений, обнаруживая характерные паттерны [7, 8]. Алгоритм отлично подходит для анализа кратковременных изменений в данных МЭМС-датчиков.

Подготовка обучающей и тестовой выборки

Так как сырые данные МЭМС датчиков включали в себя только ускорения и угловые положения, то для расширения набора признаков были рассчитаны дополнительные характеристики: скорости изменения ускорений (производные от ускорений) и угловые скорости (производные от угловых положений). Перед обучением выбранных моделей была проведена предварительная нормализация данных, которая заключается в преобразовании каждого признака так, чтобы его значения имели нулевое среднее и единичное среднеквадратическое отклонение (стандартизация). Такая нормализация гарантирует, что значения всех признаков будут распределены вокруг нуля в диапазоне, пропорциональном их отклонению. Это обеспечивает сравнимость признаков и улучшение сходимости моделей, что особенно важно для алгоритмов машинного обучения, которые чувствительны к масштабам признаков [9].

Для сохранения временной структуры данных они были разделены на части фиксированной длины методом скользящего окна. Такие части позволяют использовать их для анализа и обучения моделей. Размер окон и степень их перекрытия варьировались, чтобы исследовать влияние этих параметров на эффективность моделей и выбрать наиболее оптимальные значения. Например большие окна могут улавливать контекст и длительные события и снижают чувствительность к шумам, но в тоже время могут пропускать кратковременные события. Окна небольшого размера напротив хорошо фиксируют кратковременные события и обеспечивают большее количество обучающих примеров, но более чувствительны к шумам и могут терять контекст. Использование перекрытия окон позволяет снизить риск потери важных событий, которые могут частично выпадать из окон, и дополнительно увеличивать количество обучающих примеров. При этом большое перекрытие может приводить к избыточности данных, так как

соседние окна содержат значительную долю одинаковой информации [10].

Для “LightGBM” и “CatBoost” у каждого окна были рассчитаны следующие статистические показатели для каждого признака (ускорений, скоростей изменения ускорений, угловых скоростей): среднее значение, среднеквадратическое отклонение, минимальное и максимальное значения, медиана, 20-й и 80-й процентили. Такая агрегация параметров позволяет представить динамику данных в каждом окне компактным числовым описанием, подходящим для обучения деревьев решений.

Для работы с нейронными сетями (LSTM и CNN) данные оставались в исходной форме и были сегментированы на окна без какой-либо агрегации. Каждое окно представляло собой массив данных для каждого признака в исходной временной последовательности.

Далее окна были подвергнуты процедуре перемешивания для повышения обобщения моделей. Данная процедура случайным образом перемешивает окна датасета, сохраняя соответствие между признаками (входными данными) и метками классов (целевыми переменными). После этого данные были разделены на обучающую и тестовую выборки в соотношении 80:20, где 80% данных использовались для обучения модели, а оставшиеся 20% — для тестирования.

Сравнение эффективности моделей

В первую очередь был выполнен поиск параметров разделения данных на окна, которые показывают наибольшую эффективность моделей. Для этого параметры разделения варьировались в следующих пределах: размер окна от 6,4 до 1,2 с и степень перекрытия от 0 до 75 %. Наиболее высокие значения точности и F1-меры наблюдаются при размере окна от 2 до 3 с. При слишком больших окнах (более 5 с) метрики начинают снижаться. Это может быть связано с потерей локальных паттернов из-за избыточной агрегации

данных для GBDT-моделей и возможного переобучения для нейронных сетей. При слишком малых окнах (менее 2 с) также наблюдается снижение метрик, вероятно из-за недостаточного объема данных в каждом окне для обучения модели. Для всех моделей максимальное перекрытие (75%) дает наилучшие результаты. Это объясняется тем, что модели обучаются на большем количестве данных и получают больше информации из пересекающихся участков окон. Низкая степень или отсутствие перекрытия негативно влияет на метрики, так как временная информация между окнами становится разрозненной. Метрики для размера окна 2,5 с и перекрытия 80% (то есть окна с шагом 0,5 с) представлены в таблице №1.

Таблица №1

Метрики моделей

Размер окна, с	Степень перекр., %	Точность / F1-мера			
		“LightGBM”	“CatBoost”	LSTM	CNN
2,5	80	0,98 / 0,84	0,97 / 0,73	0,98 / 0,83	0,97 / 0,64

Из полученных результатов видно, что модели “LightGBM” и LSTM лучше справляются с задачей в условиях дисбаланса классов, чем “CatBoost” или CNN. Самое низкое значение F1-меры получено у модели CNN, что указывает на то, что CNN испытывает серьёзные трудности с распознаванием редких классов.

Для более глубокого анализа поведения моделей в отношении классификации редких классов был проведен анализ матриц ошибок для каждой из моделей. Матрицы ошибок для рассматриваемых моделей представлены далее в таблицах, при этом классы соответствуют событиям: 0 - отсутствие события, 1 - резкое торможение, 2 - резкий поворот, 3 - резкое ускорение.

Таблица №2

Матрица ошибок для “LightGBM”

		Расчетные классы			
		0	1	2	3
Реальные классы	0	14001	24	18	15
	1	39	242	0	0
	2	110	4	136	2
	3	49	4	0	149

Таблица №3

Матрица ошибок для “CatBoost”

		Расчетные классы			
		0	1	2	3
Реальные классы	0	13951	41	16	50
	1	45	234	1	1
	2	164	9	78	1
	3	67	5	0	130

Таблица №4

Матрица ошибок для LSTM

		Расчетные классы			
		0	1	2	3
Реальные классы	0	13958	29	41	30
	1	62	218	0	1
	2	70	4	178	0
	3	51	1	2	148

Таблица №5

Матрица ошибок для CNN

		Расчетные классы			
		0	1	2	3
Реальные классы	0	13898	43	35	82
	1	73	202	2	4
	2	182	8	60	2
	3	78	2	6	116

Из полученных результатов следует, что у всех моделей наибольшее количество ошибок происходит для события “резкий поворот”, которое часто ошибочно классифицируется как “отсутствие события”. Стоит отметить, что LSTM имеет самую высокую точность выявления события “резкий поворот” по сравнению с другими моделями, но при этом и повышенное количество ложных срабатываний для этого класса. “CatBoost” и CNN имеют самое большое число ошибок при выявлении события “резкий поворот”, что делает их менее предпочтительными. “LightGBM” показывает самые сбалансированные результаты.

Заключение

На основе проведенного исследования моделей машинного обучения с учетом полученных метрик эффективности можно сделать выводы:

- “LightGBM” показала самый высокие уровени точности и F1-меры при простоте использования и низких требованиях к вычислительным ресурсам;

- “CatBoost” имеет высокие показатели, но уступает “LightGBM”;

- LSTM обеспечила лучший результат среди рассмотренных нейронных сетей и целесообразность ее дальнейшего использования следует исследовать при наличии большего количества данных для обучения;

- CNN продемонстрировала самые низкие метрики из рассматриваемых моделей и высокую чувствительность к дисбалансу классов;

“LightGBM” является наиболее подходящей моделью благодаря ее высокой точности, универсальности и низким вычислительным требованиям.

Литература (References)

1. Bouhsissin S., Sael N., Benabbou F. Driver behavior classification: a systematic literature review. IEEE Access. 2023. 11. pp. 14128-14153.

2. Romera E., Bergasa L.M., Arroyo R. Need Data for Driver Behaviour Analysis?. IEEE Int. Conf. on Intelligent Transportation Systems (ITSC). 2016. pp. 387-392.

3. Nguyen T.T., Doan P.T., Le A.N., Kolla B.P., Chowdhury S., Tran D.N., Tran D.T. Develop algorithms to determine the status of car drivers using built-in accelerometer and GBDT. International Journal of Electrical and Computer Engineering. 2022. 12. 1. pp. 785-792.

4. Lu Y., Fu X., Guo E., Tang, F. XGBoost algorithm-based monitoring model for urban driving stress: Combining driving behaviour, driving environment, and route familiarity. IEEE Access. 2021. 9. pp. 21921-21938.

5. Kadri N., Ellouze A., Ksantini M., Turki, S.H. New LSTM deep learning algorithm for driving behavior classification. Cybernetics and Systems. 2023. 54. 4. pp. 387-405.

6. Kouchak S.M., Gaffar A. Detecting driver behavior using stacked long short term memory network with attention layer. IEEE Transactions on Intelligent Transportation Systems. 2020. 22. 6. pp. 3420-3429.

7. Chan T.K., Chin C.S., Chen H., Zhong, X. A comprehensive review of driver behavior analysis utilizing smartphones. IEEE Transactions on Intelligent Transportation Systems. 2019. 21. 10. pp. 4444-4475.

8. Shahverdy M., Fathy M., Berangi R., Sabokrou M. Driver behavior detection and classification using deep convolutional neural networks. Expert Systems with Applications. 2020. 149. p. 113240.

9. Ahsan M.M., Mahmud M.P., Saha P.K., Gupta K.D., Siddique Z. Effect of data scaling methods on machine learning algorithms and model performance. Technologies. 2021. 9. 3. p. 52.

10. Xie J., Hu K., Li G., Guo Y. CNN-based driving maneuver classification using multi-sliding window fusion. Expert Systems with Applications. 2021. 169. p. 114442.

Дата поступления: 20.12.2025 Дата публикации: 1.02.2025