УДК 004.855.5

СНИЖЕНИЕ ВЛИЯНИЯ УТЕЧКИ ДАННЫХ НА ТОЧНОСТЬ МОДЕЛЕЙ В МАШИННОМ ОБУЧНИИ

В.П. Корячко
Заведующий кафедрой систем автоматизированного проектирования РГРТУ им. В.Ф. Уткина, доктор технических наук, профессор
koryachko.v.p@rsreu.ru

В.И. Орешков
Доцент кафедры систем автоматизированного проектированиия РГРТУ им. В.Ф. Уткина, кандидат технических наук, доцент
vyacheslav.oreshkov@yandex.ru

В.П. Корячко

Окончил Рязанский радиотехнический институт. Область научных интересов связана с применением технологий машинного обучения в системах автоматизированного проектирования, разработкой методов представления знаний с использованием нечёткой логики и мягких вычислений.

В.И.Орешков

Окончил Рязанскую государственную радиотехническую академию. Область научных интересов связана с использованием технологий искусственного интеллекта в системах автоматизированного проектирования, машинным обучением, проектированием интеллектуальных информационных систем.

Аннотация. Рассмотрена проблема утечки данных в задачах машинного обучения и выявлены связанные с ней факторы, негативно влияющие на точность аналитических моделей на этапе их практического применения. Предложены методы обнаружения утечек данных и их предотвращения. Проведены экспериментальные исследования возможности устранения потерь точности моделей, основанных на машинном обучении, связанных с утечкой данных, на примере нейронной сети в задаче предсказания урожайности зерновых по данным агрохимического обследования почв.

Ключевые слова: интеллектуальный анализ данных, машинное обучение, аналитическая модель, обучающие данные, тестовые данные, утечка данных, нейронная сеть.

Введение.

Одной из основных групп методов обработки данных в технологиях Big Data является интеллектуальный анализ данных (Data Mining), в рамках которого могут быть решены задачи численного предсказания (регрессии), классификации, кластеризации, ассоциации, прогнозирования и т.д. Эти задачи являются основой многих сложных решений в области управления и бизнеса [1].

Наиболее популярным подходом к решению перечисленных задач является машинное обучение (machine learning - *ML*), а именно построение моделей, основанных на машинном обучении (*ML*-моделей). В таких моделях происходит автоматическая подстройка параметров в процессе обучения на некоторой выборке данных (называемой обучающей или тренировочной). Модель в процессе обучения «извлекает» из данных скрытые зависимости, закономерности и шаблоны, интерпретация которых человеком (аналитиком, специалистом предметной области) позволяет получать новые, практические полезные знания для целей поддержки принятия решений [2].

Хотя машинное обучение в настоящее время рассматривается как технологическое и инструментальное ядро Data Mining и Big Data, с обучением и практическим использованием *DM*-моделей связан целый комплекс проблем, игнорирование которых может привести к

некорректно работающим моделям и ошибочным решениям. К основным проблемам относятся эвристический характер многих алгоритмов обучения, возможность переобучения *ML*-моделей, завышенная оценка их точности.

Эвристической характер алгоритмов машинного обучения обусловлен тем, что они часто не имеют под собой строгой математической основы, не гарантируют получения единственного и точного решения, и даже решения вообще, хотя и обеспечивают приемлемый результат в большинстве практически значимых случаев. Действительно, градиентные алгоритмы обучения нейронных сетей могут «застрять» в локальном минимуме целевой функции, а начальные значения весов нейронов устанавливаться случайным образом и, как следствие, единственное решение не может быть получено даже на одном и том же наборе обучающих данных. Вместе с тем, эвристические алгоритмы обучения позволяют получить приемлемое решение даже в условиях неполных и даже частично искажённых данных, когда строгие алгоритмы вообще не работают.

Эффект переобучения (overfitting) *ML*-моделей связан со слишком точной подстройкой параметров модели под обучающие данные, что приводит к снижению её точности при практическом использовании на новых данных, не использовавшихся в процессе обучения. Причиной переобучения может стать некорректный выбор архитектуры *ML*-модели (когда число её параметров оказывается сравнимым с числом обучающих примеров) или слишком большое число итераций обучения. Для борьбы с переобучением используются тестовое множество и перекрёстная проверка (крсс-валидация).

Следует отметить, что проблемы, связанные с эвристическим характером *ML*-моделей достаточно хорошо изучены, освещены в литературе, а инструменты для их решения включаются в большинство программных средств анализа данных. Вместе с тем, при решении практических задач в различных предметных областях возникают и другие проблемы, связанные с машинным обучением и практическим применением *ML*-моделей. К одной из таких проблем, на которую в сообществе аналитиков данных стали обращать пристальное внимание в последнее время, относится утечка данных (data leakage).

Актуальность.

Утечку данных называют одной из десяти основных проблем интеллектуального анализа данных и машинного обучения [3]. Под утечкой данных в машинном обучении понимается ситуация, когда при построении *ML*-модели используются данные, которые оказываются недоступными на этапе её практического использования. Несмотря на то, что проблема утечки данных способна существенно испортить жизнь аналитикам, в настоящее время она недостаточно широко отражена в литературе

Следует отметить, что утечке в наибольшей степени подвержены данные, формируемые во внешнем окружении организации, доступность которых на зависит от самой организации. Однако, если такие данные игнорировать, потенциально ценная информация из внешнего окружения останется неиспользуемой при обучении модели.

Действительно, эксплуатация любой *ML*-модели производится в двух режимах: обучения (training) и предсказания (prediction). В процессе обучения выполняется настройка параметров модели с использованием обучающего множества по определённому алгоритму обучения. В режиме предсказания модель формирует (предсказывает) значения целевой переменной для новых наблюдений, которые не использовались на этапе обучения.

Пусть, например, модель обучается на наборе данных, в котором присутствует поле «Возраст клиента». Очевидно, что это важная информация с точки зрения предсказания, например, кредитоспособности заёмщика в кредитном скоринге, или уровня лояльности клиента в маркетинге. Но может случится так, что в режиме предсказания данные о возрасте окажутся недоступными. В результате информация, используемая в режиме обучения не может быть использован при предсказании. Это и называется утечкой данных. Ожидаемым результатом утечки данных будет снижение точности предсказания модели на практических данных,

относительно точности предсказания на обучающих и тестовых (где **«**утекшая**»** переменная присутствует). Причины данного явления интуитивно понятны: при предсказании используется меньше информации, чем при обучении. Такой вид утечки называется **«**целевой утечкой**»** (target leakage). Целевая утечка имеет место, когда при построении модели (режим обучения) используется информация, которая не будет доступна в процессе её практического использования (режим предсказания). Целевые утечки необходимо рассматривать в контексте временных периодов доступности данных, а не только той польsз, которые данные могут принести.

Ещё одной причиной, которая может привести к эффекту утечки данных, является их предобработка, а именно нормализация, масштабирование, сглаживание, квантование и другие методы, которые приводят к изменению значений данных. Применение предобработки позволяет сделать процесс обучения более эффективным, а модель более точной но только на обучающих данных. Но при этом «подгонка» модели будет производится именно к изменённым данным, а не к реальным, которые, вероятнее всего будут использоваться на этапе практического использования. Поэтому качество модели на обучающих данных, окажется переоцененным.

Данный тип утечки в литературе иногда называют «утечкой обучающего/тестового множества» (в англоязычном варианте train-test contamination). Чтобы обнаружить этот вид утечки следует проверять модель на тестовом множестве, которое не было подвержено предобработке. Другой вариант - построить модель на предобработанных данных, а затем оценить её качество на части данных, которые не были подвергнуты предобработке. Например, если в процессе предобработки производится подстановка пропущенных значений, то модель может показать хорошие результаты при тестировании, но плохие при работе с практическими данными.

И, наконец, если проверочные или тестовые данные «протекают» в обучающие, то оценка качества модели при валидации окажется завышенной. Поэтому если валидация модели строится на простом разделения исходного набора данных на обучающее и тестовое множества, то важно позаботиться, чтобы проверочные данные не попали в обучающие, в том числе и на этапе предобработки.

Таким образом, утечка данных является значимой проблемой анализа данных с использованием *ML*-моделей и требует разработки и использования методов обнаружения утечек и минимизации их негативного влияния на результаты.

Подходы к предотвращению утечки данных.

Наиболее очевидным подходом к предотвращению утечки данных является организационный. Т.е. управление данными в процессе их интеллектуального анализа должно выстраиваться так, чтобы не допустить утечки. Здесь можно выделить два подхода:

1. Не использовать при обучении модели переменные, которые потенциально могут оказаться недоступными в процессе её практического использования. Недостатки подхода очевидны: мы изначально получим худшую модель, чем могли бы получить, поскольку сознательно отказываемся от части информации, используемой в процессе обучения.

2. Использовать все доступные переменные, в том числе и те, которые могут оказаться подвержены утечке, построив лучшую модель. Затем, если утечка произойдёт, принять меры к организации сбора **«**утекших**»** данных. Недостатками подходя является то, что издержки на сбор данных могут превысить выгоду от их использования, а также то, что такие данные могут оказаться в принципе недоступными или несуществующими. Тривиальным примером такой ситуации является использование целевой переменной в качестве входной при обучении с учителем. Очевидно, что значения целевой переменной в режиме предсказания в принципе не могут быть известны. Другой пример из медицины: если в модели используются данные собранные по выборке ранее наблюдаемых пациентов и среди них есть температура, то новые пациенты просто могут оказаться недоступными для измерения температуры.

3. Восстановление **«**утекших**»** данных на основе доступных в обучающей выборке.

Например, заменять при предсказании недостающее значение в новом наблюдении на значение, искусственно сгенерированное из распределения ранее известных наблюдений. Т.е. «утекшее» значение можно рассматривать как пропущенное и восстанавливать его одним из существующих методов. Однако, здесь есть опасность, что со временем распределение данных меняется (скажем, уровень дохода дрейфует вверх или вниз), в результате вместо пропуска мы получим аномалию.

4. Рассматривать утечку данных как неизбежное зло и смириться с ней, если последствия не наносят неприемлемого ущерба.

Одной из главных проблем, связанных с утечкой данных, является то, что её бывает трудно обнаружить и устранить и она приводит, как минимум, к переоценке качества модели.

Экспериментальные исследования.

Исследования процессов утечки данных авторами работы проводились на нескольких обучающих выборках, связанных с решением практических народнохозяйственных задач. Одной из таких задач было предсказание урожайности зерновых культур (ячменя) на основе данных агрохимического обследования почв [4,5] с использованием нейронной сети. Обучающая выборка содержала следующие признаки, представленные в табл. 1.

Таблица 1. Список признаков обучающей выборки

№ п/п	Наименование	Пояснение	Ед. изм.
1	Площадь	Площадь поля	Га
2.	Кислотность	Средняя кислотность почвы поля	pH
3.	Азот	Содержание азота в почве	Мг/100 г.
4.	Калий	Содержание калия в почве	Мг/100 г.
5.	Фосфор	Содержание фосфора в почве	Мг/100 г.
6.	Угол	Средний угол уклона пашни к югу	градус
7.	Уклон	Доля пашни с уклоном к югу	%
8.	Урожайность	Фактическая урожайность поля	ц/га

Признаки со 2-го по 6-й использовались в качестве входных переменных нейросетевой модели. Урожайность - известное значение урожайности, зафиксированное для полей с наблюдаемыми агрохимическими характеристиками. В процессе обучения нейросети урожайность использовалась в качестве целевой переменной. При практическом использовании модели урожайность должна предсказываться сетью для новых полей.

Результаты применения модели позволили решить следующие задачи:

- оценивать будущую урожайность новых полей, для которых собраны данные агрохимического обследования почвы с целью принятия решения о целесообразности включения их в севооборот;
- изучить зависимость урожайности от агрохимических характеристик с целью разработки агротехнологических мероприятий, направленных на повышение урожайности.

Обучение нейросетевой модели производилось с применением аналитической платформы Deductor (в настоящее время Loginom) российской компании BaseGroup Labs (в настоящее время Loginom). В качестве базовой нейросетевой архитектуры использовалась плоскослоистая сеть прямого распространения с обучением по методу обратного распространения ошибки с оптимизацией параметров алгоритма (крутизны активационной функции, коэффициента скорости обучения и момента) [6]. Граф используемой нейросети представлен на рис. 1.

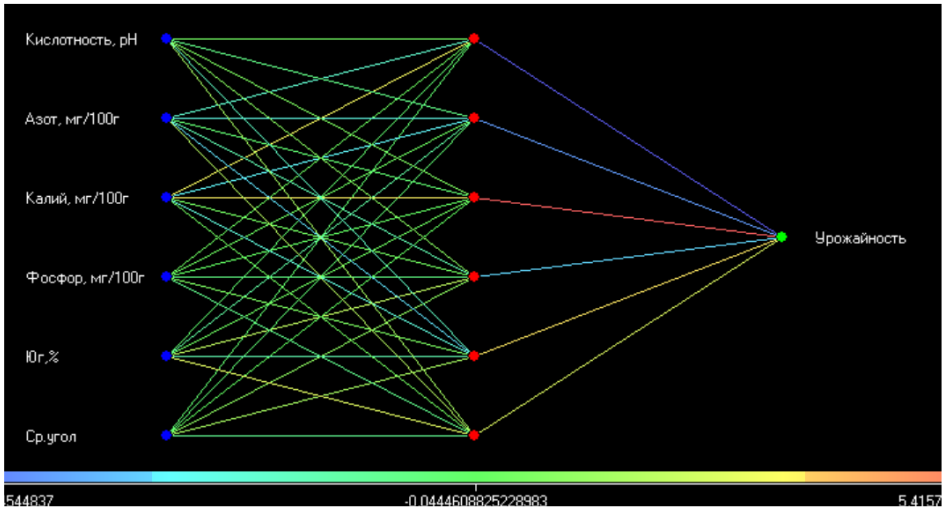

Рисунок 1. Граф нейросети для предсказания урожайности

Обучающий набор донных содержал 150 наблюдений, из которых 90% были случайным образом отобраны в обучающее множество, а 10% в тестовое. Предобработка обучающего и тестового множеств не производилась (т.е. данные использовались для обучения «как есть»). Фрагмент визуализатора с результатами обучения представлен на рис. 2.

Граф нейросети | Обучающий набор

1 / 55

№ поля	Площадь, га	Кислотность, pH	Азот, мг/100г	Калий, мг/100г	Фосфор, мг/100г	Юг,%	Ср.угол	Урожайность	Урожайность_OUT
30	81,95	5,1	4	19,59	17,56	33,02	0,7	3,6	2,9
47	128,4	5,9	8	11,98	25,19	46,99	1,15	9,7	9,1
15	191	6	9	17,78	19,74	14,34	1,21	11,6	11,3
19	58,26	6,1	9	14,42	23,28	43,6	1,01	11,6	11,3
17	17,7	4,4	1	6,8	10	46,39	2,32	0,4	0,7
39	21,65	4,7	2	8,55	13,88	7,52	0,91	1,1	1,4
55	24,68	6,1	9	19	20,5	14,47	1,28	11,6	11,4
3	47,52	5,6	6	12,76	16	44,24	0,92	6,3	6,5
40	60,13	5	3	10,25	19,83	5,63	1,03	2,2	2,0
37	21,65	4,7	2	8,55	13,88	7,52	0,91	1,2	1,4
34	14,85	5,1	4	15,59	15,71	62,63	0,82	3,5	3,3
7	18,15	5	3	7,4	6	1,33	1,78	2,2	2,1
33	82,05	4,8	3	10,59	9,75	2,41	0,81	2,3	2,2
20	84,62	5,4	5	8,64	9,33	0,06	1,01	4,8	4,9
26	30,85	6,1	9	17,4	25,07	99,75	1,37	11,6	11,5
36	118,96	5,1	4	15,59	15,71	18,05	0,81	3,4	3,5
35	17,24	4,8	3	10,59	9,75	3,51	0,55	2,1	2,2
53	151,65	4,8	3	10,89	13,23	5,84	1,07	2,2	2,1
23	26	5,9	8	18,58	20,5	0	1,46	9,7	9,8
6	100,06	5,2	4	14,89	10,8	20,05	1,51	3,4	3,5
54	38,23	5,6	6	13,08	25,2	0,08	1,66	6,3	6,4
18	79,67	4,9	3	14,23	14,21	0,9	1,07	2,2	2,3
12	128,73	6,1	9	6,91	16,11	50,15	1,28	5,6	5,7

Рисунок 2. Фрагмент представления с результатами обучения

Для оценивания точности модели можно использовать среднеквадратическую ошибку на выходе сети:

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(Y - \hat{Y}\right)^2 = 0{,}036,$$

где N - число обучающих примеров, Y - наблюдаемое значение целевой переменной, $\hat{Y}$ - значение целевой переменной, предсказанное моделью. Значение, предсказанное сетью, расположено в поле Урожайность_OUT.

После того, как модель была передана в практическую эксплуатацию выяснилось, что в силу организационных проблем данные по признакам 6 и 7 перестали быть доступными. Т.е. при формировании предсказания урожайности для новых полей информация, связанная с уклоном

пашни к югу перестала учитываться. Поскольку для новых полей целевое значение урожайности неизвестно, произвести расчёт среднеквадратической ошибки для оценки того, насколько снизилась точность модели, не представляется возможным. Однако косвенно оценить снижение точности для модели можно с использованием тестового множества, из которого исключены значения «утекших»признаков.

В результате, расчёты показали, что среднеквадратическая ошибка после «утечки»двух признаков составила $E = 0{,}042$, т.е. увеличилась на 17,4 %. Таким образом качество модели на обучающем наборе оказалось значительно переоцененным. Чтобы компенсировать результаты "утечки" было предложено ввести в модель на этапе предсказания фиктивные значения "утекших" признаков, вычисляемые на основе известных значений обучающего множества. Т.е. когда на вход модели подаётся новое наблюдение, вместо «утекших» значений на соответствующие входы поступают значения, сгенерированные по заданному алгоритму. Используемые алгоритмы и результаты их использования приведены в таблице 2.

Таблица 2. Таблица результатов устранения утечки данных

№ п/п	Наименование	Среднеквадратическая шибка	% ухудшения
1	Без устранения	0,042	17,4
2.	Среднее	0,041	13.8
3.	Медиана	0,040	11,1
4.	Замена наиболее вероятным	0,038	5,5
5.	Замена случайным	0,041	13,8

Таким образом, в данном случае эксперимент показал, что ухудшение точности модели вследствие утечки данных имеет место в любом случае и полностью устранить её последствия предложенными методами не удалось.

Наилучший результат достигнут при замене значений «утекших» признаков на наиболее вероятное значение и составляет $E = 0{,}038$, т.е. ухудшение точности составляет 5,5%, что, тем не менее даёт определённый выигрыш по сравнению с ситуацией, если на "утечку" никак не реагировать.

Наиболее вероятное значение оценивается по формуле Бернулли:

$$P_n(m) = C_n^m \cdot p^m \cdot q^{n-m},$$

где $P_n(m)$ - вероятность появления значения m на числе наблюдений n, C_n^m - количество сочетаний из n по m, p и q - вероятности появления и непоявления произвольного значения (p=1-q).

Выводы.

Показано, что при возникновении эффекта утечки данных при обучении и практическом использовании *ML*-моделей (на примере нейросети) существуют возможность частично компенсировать потери точности модели с помощью замены значений «утекших» признаков при работе модели в режиме предсказания, на значения, сгенерированные на основе доступных значений из обучающих данных. В экспериментальном примере удалось достичь снижения потери точности модели с 17,4% в случае, если никакие меры не предпринимаются, до 5,5% пи использовании замены значений "утекших" признаков на наиболее вероятные по обучающему набору.

Следует отметить, что приведённую методику не следует рассматривать как общую, которая будет работать во всех случаях. Это связано с тем, что результаты мероприятий, направленных на предотвращения последствий утечки данных будут зависеть от особенностей самой задачи, решаемой с помощь *ML*-модели, вида самой модели, природы обучающих данных и характера самих утечек. Тем не менее, приведённый пример показывает, что потери точности

ML-моделей, связанные с утечкой данных могут в определённой степени компенсироваться, поэтому оставлять утечки данных без внимания не следует.

Заключение.

Таким образом в работе произведён анализ причин и последствий такого явления как утечка данных в моделях, основанных на машинном обучении, и предложен ряд методов, которые могут способствовать снижению потери точности моделей при её работе с новыми данными, не использовавшимися в процесс обучения.

В эксперименте, проведённом с использованием нейросетевой модели для оценивания потенциальной урожайности зерновых на основе данных агрохимического обследования почв показано, что потери точности модели, связанные с эффектом утечки данных могут быть частично скомпенсированы путём замены значений признаков, подвергшихся утечке, фиктивными значениями, сгенерированными на основе известных значений признаков обучающего набора данных. Наилучшие результаты были получена при использовании замены на наиболее вероятное значение по обучающей выборке.

Список литературы

[1] Паклин Н.Б. Бизнес-аналитика: от данных к знаниям (+ CD): учеб. пособие. / Паклин Н.Б., Орешков В.И. 2-е изд., испр. – СПб.: Питер, 2013. – 704 с.

[2] Корячко В.П. Интеллектуальные системы и нечеткая логика / В.П. Корячко, М.А. Бакулева, В.И. Орешков. – М.: КУРС, 2017. – 346 с.

[3] S. Kaufman. Leakage in Data Mining: Formulation, Detection, and Avoidance / S. Kaufman, S. Rosset, C. Perlich. KDD'11, August 21 – 24, 2011, San Diego, California, USA.

[4] Васильев Е.П. Интеллектуальные системы бизнес-аналитики в сельскохозяйственном производстве / Васильев Е.П., Евстропов А.С., Орешков В.И. Проблемы механизации агрохимического обслуживания сельского хозяйства. 2011. № 2. С. 189-208.

[5] Васильев Е.П. Моделирование урожайности на основе данных агрохимического обследования почв с помощью метода ассоциативного анализа / Васильев Е.П., Орешков В.И. – Вестник РГАТУ. – 2012 – № 4 (16) – С. 8 - 13.

[6]В.И. Орешков. Интеллектуальный анализ данных: учеб. пособие / Орешков В.И. – Изд-во Рязан. гос. радиотехн. ун-та. Рязань, – 2016. 160 с.

REDUCING THE IMPACT OF DATA LEAKAGE ON MODEL ACCURACY IN MACHINE LEARNING

V.P. Koryachko
Head of the Department of Computer-Aided Design Systems, Ryazan State Radio Engineering University named after V.F. Utkin, Doctor of Technical Sciences, Professor

V.I. Oreshkov
Associate Professor of the Department of Computer-Aided Design Systems, Ryazan State Radio Engineering University named after V.F. Utkin, candidate of technical sciences

Department of Computer-Aided Design Systems
Faculty of Computer Science
Ryazan State Radio Engineering University named after V.F. Utkin
E-mail: vyacheslav.oreshkov@yandex.ru

Abstract. The problem of data leakage in machine learning problems is considered and the factors associated with it that negatively affect the accuracy of analytical models at the stage of their practical application are identified. Methods for detecting data leaks and preventing them are proposed. Experimental studies have been carried out on the possibility of eliminating the loss of accuracy of models based on machine learning associated with data leakage, using the example of a neural network in the problem of predicting grain yields based on agrochemical soil survey data.

Keywords: data mining, machine learning, analytical model, training data, test data, data leakage, neural network.