ТЕХНИЧЕСКИЕ НАУКИ

УДК 004

Создание инструмента для анализа текстовых сообщений на основе методов машинного обучения для повышения эффективности обнаружения угроз

Е.В. Дудкина, В.А. Ващенко, А.И. Молодцова
Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Аннотация
Рассмотрены системы, применяемые для анализа текстовых сообщений. Проведено исследование методов машинного обучения с выявлением их точности при поставленной задаче классификации. В результате сравнения был выбран метод дерева решений и разработан собственный инструмент для выявления текста, имеющего негативный окрас. Описана сама разработка, а также приведены примеры использования обученной модели, демонстрирующие, что нейронная сеть в состоянии определить угрозу с точностью 91 %. Целью статьи является создание программного обеспечения для обнаружения угроз в текстовых сообщениях.

Ключевые слова: нейронная сеть, обнаружение угроз, машинное обучение, текстовые сообщения, NLP, Natural Language Processing, дерево решений, адаптивный бустинг, логистическая регрессия

Для цитирования. Дудкина Е.В., Ващенко В.А., Молодцова А.И. Создание инструмента для анализа текстовых сообщений на основе методов машинного обучения для повышения эффективности обнаружения угроз. *Молодой исследователь Дона.* 2024;9(4):11–16.

Creating a Tool for Analyzing Text Messages Using Machine Learning Techniques to Enhance the Efficiency of Threat Detection

Evgeniya V. Dudkina, Vadim A. Vashchenko, Anastasiya I. Molodtsova
Don State Technical University, Rostov-on-Don, Russian Federation

Abstract
A study of machine learning methods was conducted to determine their accuracy in a specific classification task. After comparing various methods, the decision tree approach was chosen, and a custom tool was created to identify text with a negative connotation. The development process was described, along with examples of using the trained model, which demonstrated that a neural network could identify a threat with 91% accuracy. The aim of this article was to develop software for detecting potential threats in text messages.

Keywords: neural network, threat detection, machine learning, text messaging, NLP, Natural Language Processing, decision trees, adaptive boosting, logistic regression

For citation. Dudkina EV, Vashchenko VA, Molodtsova AI. Creating a Tool for Analyzing Text Messages Using Machine Learning Techniques to Enhance the Efficiency of Threat Detection. *Young Researcher of Don.* 2024;9(4):11–16.

Введение. В современном мире большинство людей использует социальные сети, где в процессе коммуникации с другими пользователями нередко возникают конфликтные ситуации. В ходе разногласий в адрес оппонента могут поступить угрозы, причем не все они ограничиваются словами, неся в себе возможность перехода к физическим действиям противника [1].

В нынешних реалиях борьба с подобными сообщениями ведется посредством вынесения жалобы от пострадавшей стороны, либо же сторонних свидетелей конфликта, однако данный метод не является эффективным. Это можно заметить невооруженным глазом, просмотрев раздел комментариев под вызывающим дискуссии или сомнительным контентом. С большим объемом угроз тяжело бороться даже группе пользователей, не говоря уже о единственном оппоненте агрессивного пользователя.

© Дудкина Е.В., Ващенко В.А., Молодцова А.И., 2024

Для автоматизации выявления опасных ситуаций и уменьшения агрессии со стороны пользователей сети Интернет необходимо создание инструмента для анализа эмоциональной окраски текстовых сообщений. Целью статьи является создание такого программного обеспечения для обнаружения угроз.

Определен ряд задач для работы:

1. Исследовать существующие модели.
2. Выбрать наиболее подходящую модель для реализации нейронной сети.
3. Разработать программное средство.
4. Провести ряд экспериментов, доказывающих эффективность разработки.

Помимо вышеизложенного планируется модификация выбранного метода для эффективной реализации программного средства, анализирующего контекст поступающих сообщений. Модификация связана с такой функцией, как обработка русского текста, что является нововведением в этой области, так как большинство существующих систем основывается на регулярных выражениях и с трудом справляется даже с этой задачей. Примером тому служит список запрещенных слов в такой социальной сети, как ВКонтакте, включающий в себя нецензурную лексику.

Основная часть. Анализ существующих моделей. Для анализа текстовых сообщений возможно применение ряда методов, связанных с обработкой естественного языка (Natural Language Processing — NLP). Мы рассмотрим некоторые из них, чтобы выделить наиболее подходящий нам метод и модифицировать его в рамках поставленной задачи.

Дерево решений. Метод, базирующийся на структуре, похожей на дерево с различными типами узлов: корневыми, внутренними и конечными, где внутренние узлы содержат значение «загрязненности», а конечные узлы представляют окончательные категории классификации [2]. В рамках анализа эта модель показала один из лучших результатов точности — 91 %.

Адаптивный бустинг. Алгоритм, который в процессе обучения строит композицию из базовых алгоритмов обучения для улучшения их эффективности. Это означает, что каждый следующий классификатор строится по объектам, которые не способен классифицировать прошлый. Метод хорош тем, что в сравнении с другими менее склонен к переобучению. К минусам алгоритма относится его чувствительность к статистическим выбросам, а также требование большого объёма обучающей выборки [3]. В нашем исследовании он применялся к методу дерева решений и показал точность 87 %.

Логистическая регрессия. Эта модель способна самостоятельно принимать решения и прогнозировать наступление некоторого события при нескольких переменных [4]. Этот алгоритм, как и предыдущий, чувствителен к выбросам и имеет трудности в случае присутствия в признаках объектов сложных взаимосвязей, но выделяется скоростью работы. В нашем исследовании он показал точность 90 %.

Учитывая точность моделей, а также достоинства и недостатки методов, в основу инструмента определения угроз нами была положена модель, обученная с помощью метода дерева решений.

Описание используемого метода. Метод дерева решений в контексте решения задачи классификации работает по принципу последовательного деления набора данных на подмножества на основе критериев разделения, основанных на признаках. Полученная модель способна принимать решения на основе правил, полученных в процессе обучения.

Основные этапы алгоритма:

Выбор корневого узла. Начинаем с объекта, который становится корневым узлом. Поскольку ни один объект не может точно предсказать окончательные классы из-за так называемой «загрязненности», выбирается метод для ее вычисления.

Вычисление загрязненности. Для объекта с числовыми значениями данные сортируются в порядке возрастания, рассчитываются средние значения соседних значений, и затем вычисляется загрязненность для каждого выбранного среднего значения. Это помогает определить, насколько хорошо объект классифицирует данные.

Разделение на уровни. На каждом уровне дерева выбирается узел с наименьшей загрязненностью. Этот процесс повторяется для разных объектов, чтобы выбрать объект и значение, которые станут узлом. Процесс продолжается на каждом уровне глубины, пока все данные не будут классифицированы [4].

Построение дерева. После завершения процесса разделения, когда все данные классифицированы, строится итоговое дерево решений.

Прогнозирование. Чтобы сделать прогноз для новой точки данных, необходимо спуститься по дереву, используя условия в каждом узле, чтобы получить окончательное значение или классификацию.

Преимущества дерева решений над другими методами:

– он не требует большого количества вычислительных ресурсов;

– способен обрабатывать нелинейные зависимости.

Недостатки метода:

– склонность к переобучению;

– на больших наборах данных скорость обучения уменьшается.

Следует также отметить, что немногочисленные существующие механизмы обнаружения угроз в текстовых сообщениях в основном способны анализировать только англоязычные сообщения. Однако на сегодняшний день существует целый ряд систем для анализа текста, и мы считаем, что современная разработка должна включать в себя их модификацию. Инновационность нашего инструмента заключается в его способности анализировать текст на русском языке.

Реализация прототипа нейронной сети. В связи с тем, что в Российской Федерации имеется малое количество систем, способных фильтровать негатив в сообщениях и комментариях, из-за чего пользователям социальных сетей и других ресурсов приходится указывать жалобами на угрозы, нами было принято решение самостоятельно реализовать прототип нейронной сети для анализа текста [5].

Для получения датасета было решено объединить наборы данных Toxic Russian Comments [6] и Russian Language Toxic Comments [7] в соответствии с задачей обнаружения угроз, а именно выбрать записи с меткой toxic или threat (негативные комментарии) и уравнять их количество с не содержащими данных меток записями (нейтральные комментарии). Полученный датасет имел размер 94 924 строки.

Далее мы импортировали серию библиотек, самыми важными из которых являются nltk, pandas, pymorphy3 и sklearn. Ntlk предназначена для работы с естественным языком, его обработкой. Pandas позволяет работать с большим объёмом данных, предоставляя огромный инструментарий для его анализа. Pymorphy3 способна проводить морфологический анализ слов русского языка. Sklearn необходима для машинного обучения: позволяет подготовить данные, использовать алгоритмы обучения и оценивать модели.

Затем с помощью библиотеки nltk загрузили список стоп-слов, чтобы удалить из датасета наиболее распространённые слова, не несущие в себе смысловой нагрузки, которые могут помешать обучению модели.

Следующий шаг заключался в написании методов, которые будут участвовать в дальнейшей подготовке и обучении модели. Первая процедура — remove_emojis: с помощью библиотеки demoji мы удалили из текста все эмодзи, чтобы модель не определяла их как часть агрессивного окраса текста. Вторая процедура — lemmatize — для обработки поступающих на вход предложений, она включает в себя несколько шагов: удаление символов с помощью регулярного выражения, обращение к процедуре удаления эмодзи, разбиение предложений на слова, из которых стоп-слова удаляются, а оставшиеся приводятся в свою начальную форму и собираются обратно в предложение, которое возвращает метод.

Самой важной процедурой является trainmodel. Она предназначена для обучения модели классификации с использованием дерева решений. Первоначально мы разделили датасет на тестовую и обучающие выборки, после чего обучили модель и возвратили результат. Затем следует функция gen_model, которая использует все вышеперечисленные методы для генерации и сохранения модели. Финальная процедура анализирует загруженное пользователем предложение, переводя его в вектор и помещая в модель, после чего выдаётся результат 0 или 1, где 1 — подтверждение присутствия агрессии в тексте, 0 — её отсутствия.

Таким образом, созданный инструмент в состоянии определить, содержит ли текстовое сообщение негативный окрас, с точностью 91 %.

Эксперименты. После реализации нейронной сети был проведен ряд проверок, чтобы удостовериться в эффективности обученной модели. Ниже перечислены вводимые в модель предложения (орфография и пунктуация оригиналов сохранены) и результат их анализа.

Предложение № 1: «Будь человеком — сходи в больницу и встань на учёт в качестве донора органов. И мне хорошо, и твоя смерть не будет напрасной». Результат предсказывания продемонстрирован на рис. 1.

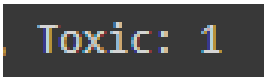

Рис. 1. Анализ предложения № 1

Предложение № 2: «Кстати! Только сегодня читала 10 способов избавится от трупа. Всегда сильно радуюсь, когда теоретические знания пригождаются на практике». Результат предсказывания продемонстрирован на рис. 2.

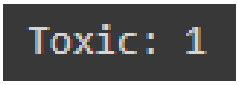

Рис. 2. Анализ предложения № 2

Предложение № 3: «Он просто притворяется тупым, пока строит свои планы по захвату Земли». На рис. 3 показан результат анализа.

https://mid-journal.ru

```
Toxic: 1
```

Рис. 3. Анализ предложения № 3

На вышеперечисленных примерах можно убедиться в том, что модель достаточно точно способна определить негативный окрас текста. Теперь необходимо убедиться в том, что предложения с нейтральным настроением не будут распознаны как несущие негатив.

Предложение № 4: «на этих обугленных сковородка картошка была очень вкусная, а если ещё с чесночком👍». Результат продемонстрирован на рис. 4.

```
Toxic: 0
```

Рис. 4. Анализ предложения № 4

Как видно из приведённых примеров, наша модель оказалась достаточна точна для анализа различного рода текстовых сообщений, в том числе содержащих ошибки и опечатки. Однако этого недостаточно для утверждения, что наша модель более точна, чем упомянутые ранее адаптивный бустинг или логистическая регрессия. Для доказательства мы провели ряд проверок на каждой из этих моделей, используя одни и те же входные данные с преимущественно негативным содержанием, где функция analyze_sentence_tree() передавала предложение модели дерева решений, analyze_sentence_adapt() – адаптивному бустингу, analyze_sentence_logreg() – логистической регрессии. Каждое предложение выводилось в его начальной форме.

Сравнение моделей № 1: «А тебе не тяжело будет поломанными руками зубы с пола собирать?» (рис. 5).

По результатам предсказаний очевидно, что модель, обученная на методе дерева решений, выдала наиболее точный результат в сравнении с другими моделями.

```
analyze_sentence_tree('А тебе не тяжело будет поломанными руками зубы с пола собирать?')
Sentence: ты тяжело поломать рука зуб пол собирать, Toxic: 1

analyze_sentence_adapt('А тебе не тяжело будет поломанными руками зубы с пола собирать?')
Sentence: ты тяжело поломать рука зуб пол собирать, Toxic: 0

analyze_sentence_logreg('А тебе не тяжело будет поломанными руками зубы с пола собирать?')
Sentence: ты тяжело поломать рука зуб пол собирать, Toxic: 0
```

Рис. 5. Первое сравнение предсказаний трех моделей

Сравнение моделей № 2: «Я вычислю через IP где ты живёшь, а потом сожгу твой дом». На рис. 6 продемонстрирован вывод каждой из моделей.

```
analyze_sentence_tree('Я вычислю через IP где ты живёшь, а потом сожгу твой дом.')
Sentence: вычислить жить сжечь твой дом, Toxic: 1

analyze_sentence_adapt('Я вычислю через IP где ты живёшь, а потом сожгу твой дом.')
Sentence: вычислить жить сжечь твой дом, Toxic: 0

analyze_sentence_logreg('Я вычислю через IP где ты живёшь, а потом сожгу твой дом.')
Sentence: вычислить жить сжечь твой дом, Toxic: 0
```

Рис. 6. Второе сравнение предсказаний трех моделей

При введении новых данных дерево решений также оказалось более точным. Мы продолжили эксперимент по сравнению моделей, чтобы определить критерии оценивания другими моделями текста с негативным окрасом.

Сравнение моделей № 3: «Застрелю». Результат показан на рис. 7.

```
analyze_sentence_tree('Застрелю')
Sentence: застрелить, Toxic: 1

analyze_sentence_adapt('Застрелю')
Sentence: застрелить, Toxic: 0

analyze_sentence_logreg('Застрелю')
Sentence: застрелить, Toxic: 1
```

Рис. 7. Третье сравнение предсказаний трех моделей

Результат проверки наглядно показал, что адаптивный бустинг — наименее точная модель, которая, в отличие от других изученных, не в состоянии определить агрессивный текст, состоящий из одного ключевого слова. А вот логическая регрессия стала более точной при коротких сообщениях, позволяя определить агрессию лишь при прямых угрозах, а не по контексту, как дерево решений.

Работоспособность адаптивного бустинга мы проверили ещё на одном примере: «Хочешь, чтобы я выехал за твоими родственниками и сжег их заживо?» (рис. 8).

```
analyze_sentence_tree('Хочешь, чтобы я выехал за твоими родственниками и сжег их заживо?')
Sentence: хотеть выехать твой родственник сжечь заживо, Toxic: 1

analyze_sentence_adapt('Хочешь, чтобы я выехал за твоими родственниками и сжег их заживо?')
Sentence: хотеть выехать твой родственник сжечь заживо, Toxic: 1

analyze_sentence_logreg('Хочешь, чтобы я выехал за твоими родственниками и сжег их заживо?')
Sentence: хотеть выехать твой родственник сжечь заживо, Toxic: 0
```

Рис. 8. Пример, где анализ адаптивного бустинга положителен

Как видим, на рис. 8 представлен положительный вывод модели адаптивного бустинга. Однако сравнительный анализ методов убедительно показал, что во всех вышеописанных случаях выбранный нами метод дерева решений оказался наиболее эффективным.

Заключение. Таким образом, изучив некоторые из существующих методов анализа текста, мы выбрали один из наиболее точных — дерево решений, и на его основе создали модель, способную выявлять агрессию в текстовых сообщениях, написанных на русском языке. Актуальность подобной системы не вызывает сомнений по причине того, что фильтров, способных распознавать негатив в социальных сетях, крайне мало, и при этом они не способны определить его по контексту, а наша модель успешно с этим справляется.

Список литературы

1. Ravinder Singh, Sudha Subramani, Jiahua Du, Yanchun Zhang, Hua Wang, Khandakar Ahmed, et al. Deep Learning for Multi-Class Antisocial Behavior Identification From Twitter. *IEEE Access*. 2020;8:194027–194044. https://doi.org/10.1109/ACCESS.2020.3030621

2. Некрасов М.В. Автоматизация метода «Дерево решений». *Актуальные вопросы экономических наук.* 2013;32:66–70.

https://mid-journal.ru

3. Акинина Н.В. Нейросетевой метод дешифрации спутниковых снимков в задачах обнаружения несанкционированных свалок. *Известия Тульского государственного университета. Технические науки.* 2017;2:25–31.

4. Горев С.В. Исследование методов и алгоритмов искусственного интеллекта при определении стоимости произведений искусства. *Известия высших учебных заведений. Серия: экономика, финансы и управление производством.* 2022:4(54):21–28. https://doi.org/10.6060/ivecofin.2022544.622

5. Shukrity Si; Anisha Datta; Somnath Banerjee; Sudip Kumar Naskar. Aggression Detection on Multilingual Social Media Text. In:10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). 2019. https://doi.org/10.1109/ICCCNT45670.2019.8944868

6. *Toxic Russian Comments*. URL: https://www.kaggle.com/datasets/alexandersemiletov/toxic-russian-comments (дата обращения: 02.04.2024).

7. *Russian Language Toxic Comments.* URL: https://www.kaggle.com/datasets/blackmoon/russian-language-toxic-comments (дата обращения: 02.04.2024).

Об авторах:

Евгения Владимировна Дудкина, студент кафедры кибербезопасности информационных систем Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), evgesha.dudkina.02@mail.ru

Вадим Александрович Ващенко, студент кафедры кибербезопасности информационных систем Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), vadimrobot@yandex.ru

Анастасия Игоревна Молодцова, студент кафедры кибербезопасности информационных систем Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), zefi_chan@mail.ru

***Конфликт интересов:* авторы заявляют об отсутствии конфликта интересов.**

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the Authors:

Evgeniya V. Dudkina, Student of the Cybersecurity of Information Systems Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), evgesha.dudkina.02@mail.ru

Vadim A. Vashchenko, Student of the Cybersecurity of Information Systems Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), vadimrobot@yandex.ru

Anastasiya I. Molodtsova, Student of the Cybersecurity of Information Systems Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), zefi_chan@mail.ru

***Conflict of Interest Statement:* the authors do not have any conflict of interest.**

All authors have read and approved the final manuscript.