|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  высшего образования |
| **«Национальный исследовательский ядерный университет «МИФИ»**  **(НИЯУ МИФИ)**  **Предуниверситарий НИЯУ МИФИ** |

# Выпускная работа обучающего IT-класса Предуниверситария НИЯУ МИФИ

«Распознавание фейковых новостей»

|  |  |
| --- | --- |
| Обучающиеся | Долженко Анастасия Тимофеевна, 11 И2  Махмудов Михаил Андреевич, 11 И2 |
|  |  |
| Научный руководитель | Дубровина Марина Игоревна |
| Должность, степень, звание | руководитель IT-классов |
|  |  |
| Дата защиты: |  |
| Результат защиты: |  |

Москва 2022

**Реферат**

* Количество страниц отчета: 26
* Количество иллюстраций: 1
* Количество диаграмм: 2
* Количество таблиц: 3
* Количество используемых источников: 30
* Ключевые слова: фейковые новости, достоверные новости, автоматическое распознавание достоверности, искусственный интеллект, машинное обучение.

*Цель работы*: разработка и реализация эффективного общедоступного инструмента для распознавания недостоверной информации в сети интернет.

*Решённые в рамках работы задачи*:

* Исследование признаков отличия фейковых новостей от достоверных
* Анализ существующих решений, продуктов и исследований на поставленную и смежные темы
* Создание базы данных фейковых и достоверных новостей на английском языке
* Разработан эффективный алгоритм парсинга сайтов новостных источников
* Реализованы алгоритмы генерации признаков текста новости
* Разработана и реализована программа распознавания фейков в сети

**Содержание**

Введение. Актуальность 4-6

Начало работы. Теоретическое исследование проблемы 7-9

Сбор данных для обучения моделей 10-13

Реализация парсера 14-16

Генерация признаков 17-19

Реализация телеграмм-бота 20

Тестирование 21

Результаты 23

Заключение 24

Список литературы и источников 25-26

**Введение. Актуальность**

Количество, достоверность, объективность предоставляемой информации в открытых источниках является важным показателем развитости общества. Тем не менее, проблема распознавания фейковых новостей распространена в мире уже на протяжении многих лет, однако их масштаб и эффективность их влияния на массы поднялись на беспрецедентные уровни именно сегодня.

Фальшивые новости могут быть преднамеренно созданы для продвижения экономических, политических и социальных интересов и могут оказать негативное влияние на убеждения и решения людей, задействоваться в «информационных войнах», поэтому важно иметь возможность оценить уровень доверия к новости и её источнику. Однако благодаря стремительному росту популярности социальных сетей, интернет-СМИ, и ростом возможностей искусственного интеллекта в вопросе генерации текста сделать это становится всё труднее.

Традиционный способ проверки информации посредством «ручной» сверки представленных фактов становится практически невозможным из-за огромного объёма, скорости распространения информации в интернете и конечно же из-за увеличивающегося риска в поисках правды снова натолкнуться на фейк. Следовательно, можно сказать, что масштабы появления дезинформации в онлайн пространстве настолько велики, что справиться с ней без помощи компьютерных алгоритмов уже невозможно.

Уже сегодня учеными и исследователями разрабатываются сервисы и решения в данной области, но большинство из них имеют закрытый программный код или лишь малое описание процесса работы алгоритма. Они умеют с довольно высокой долей вероятности отличить ложную новость по стилистике написания и употребляемым в ней словам, однако если анализу подвергается лишь текст новости, то нет гарантии того, что он может быть написан профессионалом с прекрасным стилем письма и в то же время содержать в себе ложную информацию.

В совместной работе лаборатории искусственного интеллекта MIT-IBM Watson и HarvardNLP была представлена модель для проверки визуального следа автоматически сгенерированного текста. Это позволяет проанализировать, насколько велика вероятность того, что текст автоматически сгенерирован. Знание того, является ли автором новости реальный человек или алгоритм, могло бы сыграть существенную роль в итоговом принятии решения о фейковости новости. Однако в ходе работы тестирование модели на реальных новостях не показало впечатляющих результатов и не выявило корреляции между недостоверностью новости и подобной информацией об её создании.

Также стоит отметить продукт разработчика Aaron Edell. Он представил систему, определяющую, написан ли текст языком, характерным для достоверной новостной статьи, и достиг действительно высоких результатов на тестах валидационной выборке. Однако сам автор предупреждает о том, что низкая оценка программой статьи может быть и статья, содержащая достоверную информацию, однако не укладывающуюся в «стандарты Associated Press» (перевод с английского).

Разработки проводятся не только в прикладных областях, имеющих перед собой цель выпуск какого-либо продукта. Так, Вероника Перес-Розалес с коллегами из Мичиганского университета в Анн-Арборе, провели анализ подлинных и фейковых статей и пришли к выводу о том, что фейки содержат больше наречий. Поддельные новости в этом анализе также имели тенденцию использовать более позитивный язык и выражать большую уверенность.

По результатам иследований пользователи просматривают достоверные новостные статьи, добираясь до них через web-сайты чаще, чем через социальные сети, и наоборот в случае фейковых новостей.

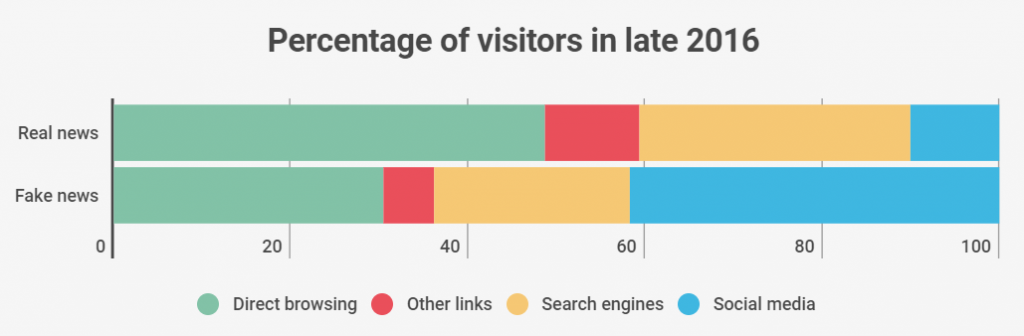


Рисунок 1

Однако логично полагать, что обычно пользователи социальных сетей чаще являются представителями современного поколения или же просто осведомлены о том, что не вся информация, размещаемая в сети является правдивой, поэтому они больше склонны к перепроверке подозрительной информации и меньше подвержены влиянию фейков в сети интернет. Напротив, представители старшего поколения чаще узнают о новостях из браузерных новостных источников, а также не привыкли перепроверять найденную информацию, поэтому разработка инструментов для проверки статей с web-сайтов остаётся актуальной.

Таким образом несмотря на то, что в настоящее время в мире разрабатываются алгоритмы, позволяющие с некоторой точностью устанавливать факт недостоверности новости, однозначного решения этой проблемы до сих пор нет и исследования по решению проблемы всё ещё актуальны.

**Начало работы. Теоретическое исследование проблемы**

Первым этапом работы стал поиск исследований, связанных с распознаванием фейковых новостей. Необходимо было выяснить какие характеристики ложных статей уже известны, какие потенциально могут быть полезны в решении проблемы, какие инструменты используются и так далее.

После изучения сайтов с фейковыми новостями, было обнаружено, что фейки можно разделить на множество различных категорий. Существуют статьи, в которых приведены реальные факты и правдивые данные, однако интерпретируются они совершенно неправильным и нелогичным образом. Есть псевдонаучные тексты, статьи, замаскированные под новостные статьи, однако которые при этом являются сугубо личным изложением мнения автора, сатирические той или иной степени тексты, «собирательные» тексты, копирующие отдельные или целые фрагменты каких-либо источников и формируя их в одну статью. Также некоторые исследователи предпринимали попытки разделить сайты с недостоверными статьями на аналогичные группы - «сатира», «фейковые», «вводящие в заблуждение» и так далее.

В большинстве случаев фейковые новости написаны так, чтобы внешне быть похожими на настоящие. Убеждение в них достигается за счет эвристики, в отличие от подлинных статей, в которых приоритет отдаётся аргументации фактами. Фальшивые новости предназначены для аудитории, которая с меньшей вероятностью прочтёт статью дальше заголовка и нескольких первых абзацев, будет проверять надежность источников, аргументов, содержащихся в тексте новости. Они нацелены на создание ментальных ассоциаций между сущностями и заявлениями.

Как уже упомянуто ранее, многие учёные приходят к выводу о том, что фейки в сети возможно обнаружить в том числе по стилю письма в статье, факту соблюдения журналистской этики.

После анализа существующих данных было сделано несколько выводов о стиле письма, строении, окраске ложных новостей:

1. Существуют стилистические различия между подлинными и выдуманными новостями. Реальные истории содержат больше языкового разнообразия, в то время как ложные выражают большую уверенность в изложенном, используют более позитивно окрашенные фразы.
2. Авторам недостоверных новостных статей свойственно чаще использовать некоторый список слов (Таблица 1), придающих эмоциональную окраску тексту и с помощью которых происходит попытка воздействовать на чувства и эмоции читателя, с целью отвлечения его внимания от лжи в аргументах и фактах, представленных в статье.
3. Были выявлены следующие характеристики, помогающие выявить ложную новость:

* Проверка даты публикации, поскольку фейковые новости, как правило, не датированы, так как сообщают о событиях, которых не было.
* Наличие грамматических и пунктуационных ошибок в тексте.
* Поиск совпадений фрагментов разных статей – основная статья, вышедшая на авторитетном ресурсе, копируется полностью или переписывается с внесением в неё ложных фактов и событий.
* Анализ ссылок-источников в материале, поскольку есть вероятность того, что статья будет поддельной, если она ссылается так же на фейковую новость.
* Анализ заголовка новости. Новости, которые содержат в своих заголовках слова, привлекающие внимание и различные сенсационные утверждения часто могут быть фейковыми.
* Фейки, как правило, короче.
* Наличие обильного использования наречий, цитат, лексических и синтаксических повторов также могут указывать на то, что новость не является достоверной.

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Часто используемые слова в правдивых новостных статьях | | Часто используемые слова в ложных новостных статьях | |
| Слова, выражающие ход мысли | Думать, знать, считать, полагать | Слова, выражающие строгую определённость | Всегда, никогда, доказано |
| Слова, связанные с работой | Работа, класс, босс | Слова, связанные с социальной темой | Поговорить, мы друг |
| Слова с негативной окраской | Нет, без, не | Слова, выражающие эмоции с положительной окраской | Симпатичный, счастливый, хороший, милый |
| Слова, выражающие противопоставление | Но, вместо, против | Слова, относящиеся к рациональным процессам | Причина, знаю, должен (должна) |
| Слова, выражающие эквивалентность | Процент, большинство, часть | Слова, направленные на будущее | Скоро, готовится, будет |

Таким образом были выявлены некоторые паттерны и признаки, характерные фейковым новостям, которые использованы в последующей работе.

**Сбор данных для обучения моделей**

Следующим этапом работы стал сбор данных для программы. Необходимо было собрать огромную базу данных, содержащих URL-адреса новостей, их характеристики. Ручной сбор данных не только неэффективен с точки зрения скорости работы, но и оказался не подходящим, так как при анализе каждой новостной статьи вручную, увеличивается риск ошиби по вине человеческого фактора, ведь нужно было правильно классифицировать статью в отдельный класс – ложная или правдивая, точно выписать информацию о ней, что при большом объёме работы могло бы содержать ошибки, значительно влияющие на процесс работы.

Для оптимизации процесса был произведён поиск существующих готовых датасетов с новостями. В ходе исследования находящихся в открытом доступе наборов данных фейковых и правдивых новостей были изучены наиболее часто встречающиеся в них признаки.

* Текст новости. Является самым распространённым признаком и обязательным для корректной работы программы, так как именно в нём содержится основная информация, позволяющая правильно классифицировать новостную статью. Он есть в 100% датасетов в том или ином виде.
* Тематика новостной статьи. Задача установления тематической принадлежности текста трудна, так как вариантов группировки текстов крайне много. Большинство авторов выделяет две категории: новости и политические новости. Этот признак наблюдается в 55% наборов. Признак может быть полезен в решении поставленной проблемы, однако в связи с невысокой конкретикой и относительно малым количеством данных было принято решение не использовать его.
* Язык, на котором написан текст новости - признак, который встретился лишь в 10% наборов. Было принято решение игнорировать его, поскольку в проекте используются данные некоторых исследований об особенностях лингвистической составляющей в фейковых новостях на английском языке. Следует отметить, что составление нескольких словарей на разных языках - очень сложная задача, в частности, из-за маленького количества готовых наборов данных.
* Дата или время публикации - признак, который встречается в 90% наборов, но на прогнозирование без отдельной сверки представленных фактов не влияет.
* Автор статьи - встречается в 35% датасетов. Полезный признак с точки зрения анализа и составления статистики об “авторитетности” журналистов, ведь чаще всего фейки пишутся одними и теми же авторами. Но было обнаружено, что в наборах данных почти все авторы разные, поэтому принято решение об его игнорировании.
* Источник – признак, приведённый лишь в 25% датасетов, может дать существенный вклад в принятие итогового решения, так как, аналогично авторам статьи, чаще всего фейки публикуются одними и теми же новостными источниками.
* URL-адрес новостной статьи являлся основным критерием при отборе подходящих датасетов, так как именно через него извлекаются характеристики статьи и генерируются другие признаки, необходимые для работы программы.

Также были представлены данные о регионе, стране, городе, в котором опубликована новость, времени загрузки и/или изменения текста новости на портале, ссылки на видео, картинки, используемые в статье однако эти признаки не привносят никакого вклада в решение поставленной проблемы.

Общая статистика по содержанию признаков в датасетах приведена ниже.

Как уже было упомянуто ранее, наличие URL-адреса новости являлось основным признаком отбора данных, но так как около 50% его не содержали, половина всех найденных данных сразу была отброшена.

Далее происходила проверка отобранных данных. Больше половины оказались непригодными по разным причинам. Например, web-страница новости или домен могли больше не существовать.

Однако ещё одна проблема – несвоевременное обновление SSL-сертификатов сайтов новостных источников подразумевала под собой бОльшие трудности. Сайту необходим SSL-сертификат для работы с протоколом безопасного соединения HTTPS. Использование SSL-сертификата обеспечивает безопасное соединение с источником без риска потери данных, целостность передаваемой информации, а также безопасность данных пользователя. На многих, даже на проверенных годами источниках обнаружилась такая проблема, но было принято решение о том, чтобы не использовать подобные источники и уведомлять пользователя в случае, если будет запрошена проверка новости на достоверность с подобных новостных источников, с целью обезопасить информацию как пользователя, так и сайта.

Для оптимизации процесса проверки каждого URL-адреса новости на пригодность, был реализован алгоритм, проверяющий каждую ссылку на это условие и возвращающий соответствующий результат – новость больше не существует или переход на web-страницу новости не является безопасным.

Таким образом еще около трети оставшихся данных была отброшена.

Общая статистика полученных данных представлена на диаграмме ниже.

**Реализация парсера**

Основой работы всей программы является правильно реализованный алгоритм сбора данных с web-страницы новостных статей, без которых невозможен их дальнейший анализ, поэтому следующей задачей стала реализация алогритма для парсинга web-сайтов с целью извлечения из них информации, классифицирующей новость. Такой информацией стали: название и дата регистрации новостного источника, заголовок и текст новости.

Любой web-сайт имеет свой уникальный HTML-код, поэтому задача создания универсального алгоритма для успешного изъятия необходимых данных с любого web-сайта довольно сложна. Есть несколько подходов к решению данной задачи: на основе искусственного интеллекта и «ручного» поиска необходимой инйормации в HTML-коде страницы новости. В проекте были реализованы и сравнены оба из них.

Первый основывается на использовании технологий машинного обучения и извлекает необходимые данные из видимой части web-страницы подобно человеку. С помощью такого подхода можно обойти проблему, которая встречается на некоторых web-сайтах и заключается в том, что тег с именем автора часто заполняется каким-либо стандартным значением, в то время как имя настоящего автора указывается непосредственно в самой новости. Подобные подходы реализованы в библиотеках extractnet 1.0.4 [9] и Dragnet [10].

Однако в ходе исследования возможностей работы данных библиотек было выяснено, что extractnet неэффективно работает в контексте анализа стуктурированных web-страниц, каковыми являются новостные источники, и извлечения шаблонного содержимого, что также характерно для новостных источников. В свою очередь Dragnet эффективно работает в контексте решения задачи извлечения только основного содержимого статьи или содержимого и любых комментариев, созданных пользователем.

Второй метод основывается на «ручном» поиске необходимой информации с помощью проверки соответствующих тегов в HTML-коде страницы новости.

Как было сказано ранее каждый сайт уникальный HTML-код, соответственно и теги, под которыми находится необходимая информация так же будут разными. По этой причине такой подход является трудным в реализации, так как необходимо собрать большую базу тегов, под которыми будет находится необходимая информация. Однако в ходе исследования было выяснено, что каждый новостной портал использует свою собственную систему тегов. Исключения могут составлять лишь статьи некоторых авторов. Тем не менее, этот факт несколько облегчает процесс создания базы тегов, и таким образом задача свелась к исследованию системы организации HTML-кодов страниц как можно большего количества источников.

Для реализации парсера использовались библиотеки BeautifulSoup4 [11] и requests 2.27.1 [12]. Комбинация функционала библиотек оказалась самой эффективной в контексте решения задачи. Также была попытка реализации парсера с помощью функционала библиотеки Selenium [13]. Однако при сравнительном анализе было выяснено, что при одинаковом качестве обработки данных алгоритм, основывающийся на библиотеке Selenium, обрабатывает некое количество данных в два раза дольше, чем то же количество данных обрабатывает алгоритм на основе библиотек BeautifulSoup4 и requests.

В ходе работы встречались новостные источники, которые использовали специальные алгоритмы защиты от парсинга, что могло бы помешать сбору данных. Любой запрос к web-серверу содержит заголовки, в том числе идентификатор браузера и клиента - юзер-агент (user agent). При работе программы запрос на web-сервер происходит не с браузера пользователя, поэтому алгоритмы защиты новостного источника блокируют такие запросы индентифицируя неправильный юзер-агент. В алгоритме парсера было предусмотрена отправка запроса с юзер-агентом, характерным для браузера любого пользователя. Таким образом удалось обойти защиту и успешно извлекать данные с новостных сайтов.

В результате реализованный алгоритм парсера извлекает все необходимые данные со 100% новостных источников, представленных в собранном наборе данных.

**Генерация признаков**

Генерация признаков из изъятых данных – следующая поставленная в ходе работы задача. Был составлен список признаков, характеризующих новость, а также реализованы вспомогательные алгоритмы с использованием возможностей технологий машинного обучения и функционала библиотек языка программирования Python. В список вошли следующие признаки:

* Анализ заголовка новости. С помощью библиотек tensorflow [19], scikit-learn (sklearn) [21] и keras [20] была создана нейронная сеть, анализирующая заголовок новостной статьи на предмет кликбейта, тем самым прогнозируя фейковость новости на начальном этапе анализа. Сеть содержит 4 слоя – два внутренних, входной и выходной. Точность прогнозирования на валидационной выборке составляет 65%.
* Количество орфографических и грамматических ошибок в тексте новости. С помощью библиотеки JamSpell [14] был реализован алгоритм проверки, подсчёта и исправления (для последующей обработки текста новости) орфографических и грамматических ошибок в тексте новости.
* Анализ тональности и субъективности текста новости. С помощью функционала библиотеки textblob 0.17.1 [17] реализован алгоритм числового прогнозирования и характеристики тональности и субъективности текста новости. Возвращаемые числовые значения впоследствии масштабируются методом библиотеки scikit-learn (sklearn) [20] с целью улучшения качества работы программы.
* Количество наречий и цитат в тексте новости. Во время работы над созданием алгоритма векторизации и обработки текста новости была произведена работа по реализации функции лемматизации текста с помощью библиотеки spaCy [15], в ходе работы которой текст новости токенизируется, а также каждому слову присваивается с помощью специально обученной модели библиотеки nltk [16] POS тег, дающий информацию о части речи обрабатываемого слова. В ходе обработки текста на этапе удаления знаков препинания из текста определяется количество цитат в тексте новости.

Также в список вошли такие признаки как:

* Домен сайта новостного источника
* Возраст домена, определяющий возраст новостного источника

Далее была поставлена задача обработки текста для последующей векторизации. В ходе обработки были реализованы функции удаления пунктуации, нахождения и исправления орфографических и грамматических ошибок, лемматизации текста, удаления стоп-слов с помощью модуля stopwords библиотеки nltk [16].

Слудующим шагом стало исследование способов векторизации текста и оценка их эффективности в работе с моделями классификации в существующих решениях. Сводная таблица оценки некоторых решений приведена в Таблице 2.

|  |  |  |
| --- | --- | --- |
| Модель | Модель векторизации | Оценка (accuracy [22]) |
| Нейронная сеть с встроенной LSTM [24] | TF-IDF [23] | 92% |
| Нейронная сеть с встроенной LSTM | Word2Vec [30] | 96% |
| Logistic Regression [25] | TF-IDF | 87% |
| Decision Tree Classifier [26] | TF-IDF | 80% |
| Gradient Boosting Classifier [27] | TF-IDF | 83% |
| Random Forest Classifier [28] | TF-IDF | 85% |
| SVC [29] | TF-IDF | 91% |

Таблица 2

Из таблицы видно, что эффектривность работы на валидационной выборке высока, однако при тестировании на нашем наборе данных результаты были хуже. Вероятнее всего это всязано с переобучением моделей в исслудуемых решениях.

Практически во всех приведенных решениях используется метод TF-IDF [23] для векторизации текста. Однако методы построения моделей на основе векторизатора подсчета (с использованием подсчета слов) или матрицы TF-IDF (подсчет слов относительно того, как часто они используются в других статьях в наборе данных) не учитывают порядок слов и контекст. Существует вероятность того, что две статьи, похожие по количеству слов, могут иметь совершенно разное значение. Также в статье [8] был приведен сравнительный анализ работы этого метода, что показывает его относительную неффективность на фоне других методов векторизации. По этим причинам было принято решение не использовать данный способ векторизации в программе.

Таким образом после исследования эфективности работы вариантов комбинаций моделей классификаии и моделей векторизации текста была реализована Нейронная сеть с встроенной LSTM, а также используется метод веторизации Word2Vec. Программа тестируется для определения точности работы.

**Реализация телеграмм-бота**

Для удобной работы пользователей с программой реализован телеграмм-бот, принимающий ссылку на новость, а затем возвращающий результат анализа фейковости новости, в который интегрируется разработанная программа. Такой выбор обусловлен тем, что таким образом программу можно использовать с любый типов устройств и операционных систем.

**Тестирование**

Финальным этапом работы является тестирование программы и бота на предмет ошибок и исследование для поиска возможнотей для улучшения изменения параметров моделей, задействованных в ходе работы программы, с целью улучшения качества работы программы. Таким образом, было внесено несколько изменений, а также отмечены некоторые особенности работы программы.

В ходе работы был проведен анализ того, какие источники какие новости (фейовые или нет) чащу публикуют. Для этого была взята небольшая выборка, около 4000 новостных статей, а также составлена сводная таблица по этой выборке.

Таблица 3

|  |  |  |  |
| --- | --- | --- | --- |
|  | Источник | Количество опубликованных правдивых новостей | Количество опубликованных фейков |
| 0 | www.reuters.com | 646 | 0 |
| 1 | www.cnn.com | 389 | 0 |
| 2 | www.nytimes.com | 341 | 0 |
| 3 | www.bbc.com | 333 | 0 |
| 4 | abcnews.go.com | 57 | 0 |
| 5 | money.cnn.com | 49 | 0 |
| 6 | edition.cnn.com | 15 | 0 |
| 7 | inhealth.cnn.com | 14 | 0 |
| 8 | www.bbc.co.uk | 10 | 0 |
| 9 | bleacherreport.com | 8 | 0 |
| 10 | clarivate.com | 3 | 0 |
| 11 | thelab.bleacherreport.com | 1 | 0 |
| 12 | cnn.it | 1 | 0 |
| 13 | reuters.com | 1 | 0 |
| 14 | beforeitsnews.com | 0 | 1681 |
| 15 | www.activistpost.com | 0 | 267 |
| 16 | dailybuzzlive.com | 0 | 99 |
| 17 | www.disclose.tv | 0 | 71 |

По данным таблицы 3 можно судить о том, что источники либо публикуют только фейки, либо не публикуют их вообще. Так, зная источник, можно однозначно классифицировать новость. Однако теория не может быть подтверждена и являться полностью правдивой, поэтому было принято решение всегда считать, что фейки могут быть размещены на любом web-сайте, следовательно информация об источнике не будет использоваться в ходе работы программы.

Была произведена работа по подбору параметров моделям, участвующим в работе программы. Так, удалось добиться 75% точности модели, анализирующей новость, в сравнении с 52%, полученными в начале разработки.

**Результаты**

В ходе работы были реализоваын алгоритм парсинга англоязычных новостных источников, алгоритмы генерации признаков и векторизации текста. Также разработана и реализована программа распознавания фейковых новостей с web-сайтов англоязычных источников. Удалось добиться 97% точности распознавания фейков на тестовом наборе данных. Создан специальный телеграмм-бот, предназначенный для комфортного взаимодействия пользовтеля и программы.

**Заключение**

Изучение и анализ существующих решений и исследований показывают, что до сих пор не найден универсальный механизм определения поддельных новостей. Выявлены некоторые признаки, характеризующие фейки, однако с развитием технологий у злоумышленников, создающих фейки будут появляться всё новые способы для маскировки лжи под правду.

В данной работе нами были рассмотрены решения разработчиков и исследователей, их методы и подходы. На их основе, а также на основе собственных исследований было создано уникальное решение.

В ходе работы нами была реализована программа, которая опирается не только на текст, но и на вторичные признаки, которые могут повлиять на прогнозирование.

Для удобного взаимодействия с пользователем программа будет интегрирована в телеграмм-бот, который принимает ссылки на новости, вызывающих сомнение в их достоверности и возвращает результат работы программы.

**Список литературы и источников**

1. <https://arxiv.org/abs/1708.07104>
2. <https://arxiv.org/abs/1703.09398>
3. <https://github.com/mansoor9743/Fake-News-Detection>
4. <https://arxiv.org/abs/1804.09088>
5. <https://www.americanvoiceforfreedom.org/wp-content/uploads/2018/03/The-spread-of-true-and-false-news-online.pdf>
6. Некрасов Г.А., Романова И.И. «Разработка поискового робота для обнаружения веб-контента с фейковыми новостями // Инновационные, информационные и коммуникационные технологии» 2017. № 1. С. 128-130]
7. <https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12128>
8. https://cyberleninka.ru/article/n/vektorizatsiya-teksta-s-ispolzovaniem-metodov-intellektualnogo-analiza-dannyh/viewer -
9. <https://dl.acm.org/doi/10.1145/2487788.2487828> - «Извлечение контента с использованием различных наборов функций», Питерс, Мэтью Э. и Д. Лекок
10. <https://pypi.org/project/extractnet/>
11. <https://github.com/dragnet-org/dragnet>
12. <https://pypi.org/project/beautifulsoup4/>
13. <https://pypi.org/project/requests/>
14. <https://github.com/SeleniumHQ/selenium>
15. <https://github.com/bakwc/JamSpell/>
16. <https://github.com/explosion/spaCy>
17. <https://www.nltk.org/>
18. <https://pypi.org/project/textblob/>
19. <https://www.tensorflow.org/>
20. <https://keras.io/>
21. <https://scikit-learn.org/stable/index.html>
22. <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html>
23. <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>
24. <https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM>
25. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>
26. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
27. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>
28. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
29. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>
30. <https://www.tensorflow.org/tutorials/text/word2vec?hl=ur>