Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК
РОП 09.04.03 Медведева М.А.

_____
«01» июня 2024 г.

# МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Разработка алгоритма автоматического обнаружения и классификация спам сообщений с применением машинного обучения и нейронных сетей

Научный руководитель: _____
доцент, к.э.н. Медведев М.А.

Научный руководитель: _____ Балунгу Д.М.
ассистент

Студент группы _____ Ганущак Д.Ю..
РИМ-220980

Екатеринбург
2024

# РЕФЕРАТ

Тема магистерской диссертации:

“Разработка алгоритма автоматического обнаружения и классификации спам сообщений с применением машинного обучения и нейронных сетей”.

Магистерская диссертация выполнена на 78 страницах, содержит 12 таблиц, 27 рисунков, 36 использованных источников.

Целью работы является совершенствование методов и алгоритмов защиты от спама и атак по электронной почте с использованием машинного обучения и нейронных сетей. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ существующих методов и алгоритмов защиты от спама и атак по электронной почте.
2. Исследовать возможности применения современных алгоритмов машинного обучения для фильтрации спама.
3. Разработать и реализовать алгоритмы, использующие нейронные сети для повышения точности фильтрации.
4. Провести оценку разработанных алгоритмов на основе реальных данных и сравнить их эффективность с существующими решениями.
5. Исследовать вопросы, связанные с законодательством и этическими аспектами использования методов фильтрации спама.

Методология исследования включает в себя анализ и обзор литературы, экспериментальную разработку и тестирование алгоритмов, а также сравнительный анализ различных методов защиты от спама. В ходе исследования были использованы как теоретические, так и практические подходы, что позволило получить всестороннюю оценку возможностей и ограничений современных технологий защиты электронной почты.

Структура работы состоит из трех глав. В первой главе рассматриваются существующие методы и алгоритмы защиты от спама и атак по электронной почте, включая вероятностные методы, методы на основе правил, методы машинного обучения и гибридные методы. Также обсуждаются инновационные подходы, такие как гибридные модели, самообучающиеся системы и эволюционные алгоритмы. Во второй главе описывается методология проектирования и реализации алгоритмов машинного обучения, начиная с подготовки данных и выделения признаков и заканчивая обучением, оценкой и развертыванием моделей. В третьей главе рассматривается практическая реализация проекта, включая разработку нового метода или улучшение существующего, а также анализ результатов.

Таким образом, данная дипломная работа направлена на изучение и разработку эффективных методов защиты от спама и атак по электронной почте с использованием современных технологий машинного обучения и искусственного интеллекта, что является актуальной и значимой задачей в области информационной безопасности.

## Введение

С годами коммуникация развивалась, и возможностей для обмена мыслями и идеями у людей становилось все больше. Средства коммуникации развивались веками: начиная с эпохи, когда люди общались только лицом к лицу, писали письма, совершали телефонные звонки, отправляли текстовые сообщения, и до сегодняшнего дня, когда человек подключен к Интернету, был достигнут прогресс, и связь развивалась, становясь дешевой. Популярность мобильных телефонов и их широкое использование привлекли внимание киберпреступников, которые использовали их эгоистично. Они широко используют нежелательную электронную почту, которую также называют спамом.. Эти сообщения могут быть использованы злоумышленниками для заманивания получателей заставляют перейти на вредоносную страницу или отправить ответное сообщение, что может привести к взиманию дополнительной платы или даже за рекламные предложения и товары. Как правило, спам-сообщения представлены в виде различных текстов, изображений и различных мультимедийных форматов, которые рассылаются по мобильным телефонам, электронной почте и даже через социальные сети. Основное внимание в нашей исследовательской работе уделяется спам-текстам и электронным письмам.

Отправка коротких сообщений с мобильного телефона стала мгновенным, экономичным, удобным в использовании и доступным средством коммуникации. Как бы то ни было, доступность, достоверность и безопасность сервиса находятся под угрозой из-за большого количества текстовых сообщений. Нежелательное сообщение - это сообщение, отправленное без разрешения пользователя; иногда такие сообщения также называют спамом. Большое количество спама влияет на пропускную способность, объем памяти, время и энергопотребление. Получение нежелательных сообщений может стоить гораздо дороже, чем просто время и хранение, поскольку они

используются в качестве средства распространения вирусов, маскировки и проведения социальной инженерии, основанной на интеллекте или небрежности конечного пользователя.

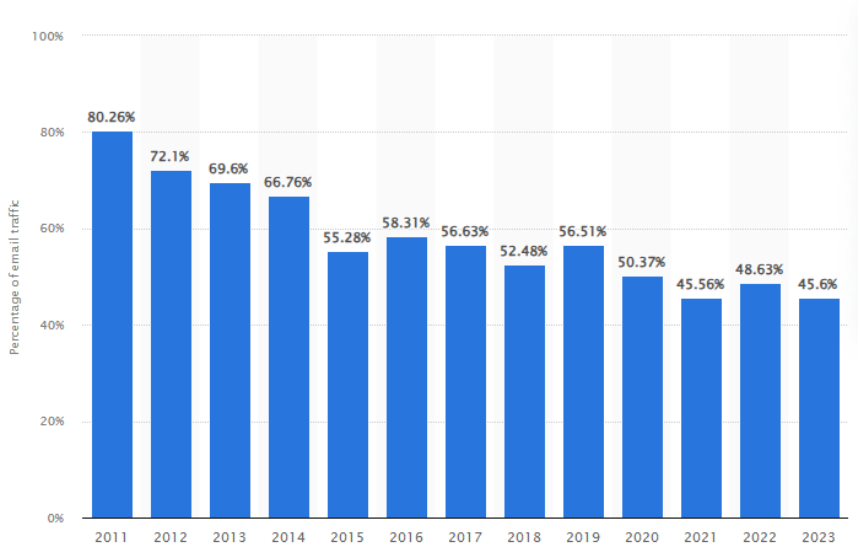



Рисунок 1 - Анализ спам-сообщений в Мире с 2011 по 2023 год (Statista, 2024)

Поставщики услуг электронной почты и пользователи пытались бороться с этой атакой, просто используя фильтр или набор

правил, которые отделяют хорошие электронные письма (ham) от спама, основываясь на предварительном знании привычных

спам-рассылки; это называется подходом, основанным на правилах. Однако спамеры постоянно стремятся доставить это электронное письмо получателю, постоянно изменяя структуру сообщения, чтобы обойти все параметры/фильтры, установленные в пути. Следовательно, существует необходимость постоянно обыгрывать злоумышленника в его игре, чтобы обезопасить почтовый ящик.

Опасность, которую представляют текстовые спам-сообщения, вынудила производителей телефонов разрабатывать встроенные решения для борьбы со всеми видами спама, включая непрошеные короткие служебные сообщения (SMS). Технологические компании, такие как Apple и Google, создали систему

[4] рассылки спама  встроенные в телефоны фильтры позволяют пользователям блокировать нежелательные сообщения, фильтровать их и сообщать о них.

Непрекращающаяся борьба между спамерами и спам-фильтрами привела к появлению нескольких методов фильтрации. Автоматическая фильтрация[29] нежелательной почты с использованием правил позволяет фильтровать электронную почту, разделяя ее на спам и нежелательные (законные) письма. Фильтрация нежелательной почты осуществляется путем настройки правил, созданных пользователем или каким-либо другим провайдером, или приложений для фильтрации нежелательной почты на основе правил, которые применяются к основной массе сообщений, чтобы имитировать подход человека к этому. Главный недостаток  особенностью этого подхода является необходимость постоянного обновления правил в соответствии с меняющимся способом атаки спамера, что может привести к ошибочной классификации ham как спама (ложные срабатывания). Такой подход к созданию правил с бесконечным циклом занимает много времени и довольно неудобен для большинства пользователей. Подход, основанный на машинном обучении, предлагает систему, которая может классифицировать и обновлять саму себя на основе нового трюка, используемого спамером. Это может быть сделано как контролируемым, так и неконтролируемым образом.

# ГЛАВА 1: Анализ методов и алгоритмов защиты от спама и атак по электронной почте

## ОБОСНОВАНИЕ НЕОБХОДИМОСТИ ИССЛЕДОВАНИЯ

### Законодательная основа регулирования спама и электронных атак

В современном мире, где большая часть личной и коммерческой жизни перенесена в цифровое пространство, утечки персональных данных стали одной из наиболее острых угроз. Эти угрозы не только подрывают доверие потребителей к брендам и услугам, но и могут привести к серьезным финансовым потерям и ущербу для репутации компаний. Кроме того, для физических лиц последствия могут быть еще более серьезными, включая возможность финансового мошенничества и нарушения приватности. Утечки данных часто происходят из-за недостатков в системах безопасности, но также могут быть вызваны ошибками сотрудников или целенаправленными атаками хакеров. Эффективность законодательства в области защиты данных постоянно подвергается испытаниям этими новыми и развивающимися угрозами. В этом контексте правовые рамки должны не только реагировать на инциденты после их происхождения, но и создавать надежную основу для предотвращения таких утечек в будущем.

Законодательство, такое как Федеральный закон "О персональных данных" в России, ставит своей задачей защиту граждан от неправомерного использования их данных. Однако растущее количество утечек[10] показывает, что действующие меры могут быть недостаточными. Это подчеркивает необходимость более строгих мер регулирования, включая обязательные требования к защите данных для всех организаций, обрабатывающих персональные данные, ужесточение контроля со стороны регулирующих органов и усиление ответственности за нарушения.

На самом деле, в российском законодательстве нет прямого запрета на рассылку спама. Однако, рассылка спама может быть признана нарушением в следующих случаях:

1. Нарушение Федерального закона от 08.07.2003 № 130-ФЗ "О связи": Статья 25 данного закона запрещает рассылку сообщений электросвязи без согласия абонента. Спам, рассылаемый без согласия получателя, является нарушением этой статьи.
2. Нарушение Федерального закона от 27.07.2006 № 149-ФЗ "Об информации, информационных технологиях и о защите информации": Статья 29 данного закона устанавливает обязанность операторов персональных данных принимать меры по защите персональных данных от несанкционированного доступа. Спам, содержащий персональные данные получателя, может быть признан нарушением этой статьи, если эти данные были получены незаконным способом.
3. Нарушение Гражданского кодекса РФ: Статья 1523 ГК РФ запрещает распространение информации, порочащей честь, достоинство или деловую репутацию гражданина или юридического лица. Спам, содержащий оскорбительную или дискредитирующую информацию, может быть признан нарушением этой статьи.
4. Нарушение других законов и нормативных актов: Существует ряд отраслевых стандартов и рекомендаций по защите от спама, которые могут содержать запреты на рассылку спама в определенных сферах деятельности.

Таким образом, в то время как технологии продолжают развиваться, законодательство должно адаптироваться, чтобы адекватно реагировать на эти изменения. Рассмотрение резонансных случаев утечек данных, их последствий и ответных мер поможет определить ключевые направления для усиления законодательной защиты персональной информации в России и мире.

## Проблематика увеличения количества спам-атак

За последнее десятилетие одним из важнейших факторов стало колоссальное увеличение спама. По данным многочисленных исследований, спам составляет до 60% всего трафика электронной почты, что приводит к огромной трате полосы пропускания и перегрузке почтовых систем. Именно эта серьезная проблема спама лежит в основе данной диссертации и является ее формулировкой. Необходимо разработать и принять меры для борьбы со спамом. Безусловно, одним из лучших решений в этом направлении является фильтрация спама. Существуют два основных подхода к фильтрации спама: инженерия знаний и машинное обучение. Методы фильтрации спама, основанные на инженерии знаний, используют набор заранее определенных и определяемых пользователем правил. Эти правила применяются для идентификации основных характеристик электронного письма. Однако, как было отмечено, фильтрация спама на основе инженерии знаний страдает от плохой обобщенности. Методы машинного обучения строят классификатор, обучая его на наборе электронных писем, называемом обучающей выборкой. Исследования показывают, что фильтры спама на основе машинного обучения достигают лучшей обобщенности по сравнению с фильтрами, основанными на инженерии знаний. В течение многих десятилетий велась интенсивная работа в области классификации спама и категоризации электронной почты, и наиболее популярными методами являются наивный байесовский классификатор и машины опорных векторов (SVM)[15]. В данной работе мы предлагаем подход глубокого обучения для идентификации спам-писем и оцениваем его эффективность на пяти хорошо изученных наборах данных для обнаружения спама. Мы также сравним производительность различных алгоритмов глубокого обучения, чтобы определить, какой из них работает лучше всего.

## Необходимость улучшения методов борьбы со спамом

Традиционные методы борьбы со спамом, включая черные и белые списки (blacklists/whitelists) и базовые фильтры содержимого, уже не могут эффективно справляться с постоянно меняющимися стратегиями спаммеров. Киберпреступники используют сложные обфускации и методы маскировки, что делает классические подходы менее эффективными. Современные спам-фильтры должны использовать более продвинутые технологии, такие как машинное обучение и искусственный интеллект, чтобы адаптироваться к новым угрозам и минимизировать ложные срабатывания.

**Роль машинного обучения в классификации спама**

Машинное обучение предлагает мощные инструменты для анализа больших объемов данных, что делает его идеальным для задач фильтрации спама. Алгоритмы машинного обучения могут обучаться на основе примеров спама и не спама, постепенно улучшая свою точность и способность распознавать сложные и изменчивые паттерны. Использование машинного обучения позволяет создавать адаптивные системы, которые могут эффективно справляться с эволюцией спам-атак без постоянного вмешательства человека.

**Сравнительный анализ преимуществ и недостатков различных подходов внедрения решений ИИ**

Сравнительный анализ преимуществ и недостатков различных подходов внедрения решений искусственного интеллекта может быть полезным для понимания, какой подход может быть наиболее подходящим для конкретной ситуации. При выборе подхода к внедрению решений ИИ необходимо учитывать цели организации, доступные ресурсы, уровень экспертизы и готовность к инвестициям. Каждый из подходов имеет свои преимущества и недостатки, и важно провести тщательный анализ перед принятием решения.

На данный момент существует несколько подходов внедрения решений искусственного интеллекта:

1. Разработка собственных решений;
2. Использование коммерчески доступных решений;
3. Использование открытых платформ и инструментов.

Сравнение преимуществ и недостатков различных подходов внедрения решений искусственного интеллекта представлено в таблице 1.

Таблица 1 – Сравнение преимуществ и недостатков различных подходов внедрения решений искусственного интеллекта.

| **Название** | **Преимущества** | **Недостатки** |
|---|---|---|
| Разработка собственных решений ИИ | Полный контроль над разработкой и реализацией решений ИИ. Гибкость в настройке и адаптации решений. Конкурентное преимущество. | Высокие затраты на исследования и разработку. Необходимость наличия экспертных знаний и опыта в области ИИ. Длительное время разработки и внедрения. |
| Использование коммерчески доступных решений ИИ | Быстрая реализация и внедрение решений. Отсутствие необходимости в глубоких знаниях в области ИИ. Поддержка и обновления. | Ограниченные возможности настройки и адаптации. Зависимость от поставщика. Высокие затраты на лицензии и обслуживание. |
| Использование открытых платформ и инструментов ИИ | Большое сообщество и поддержка разработчиков. Бесплатная доступность и низкие затраты. Гибкость и настраиваемость. | Требуется высокая техническая компетенция. Отсутствие гарантии качества и поддержки. |

Рассмотрим подробнее преимущества и недостатки каждого подхода.

**Разработка собственных решений**

**Преимущества:**

1. Полный контроль над разработкой и реализацией решений ИИ: Разработка собственных решений позволяет иметь полный контроль над всем процессом, начиная от исследований и разработки алгоритмов до реализации и внедрения;
2. Гибкость в настройке и адаптации решений: При разработке собственных решений ИИ можно гибко настраивать и адаптировать их под конкретные потребности и цели организации;
3. Конкурентное преимущество: Уникальные решения ИИ могут дать организации конкурентное преимущество на рынке.

**Недостатки:**

1. Высокие затраты на исследования и разработку: Разработка собственных решений ИИ может быть финансово затратной задачей, требующей значительных инвестиций в исследования, технологическую инфраструктуру и обучение специалистов;
2. Необходимость наличия экспертных знаний и опыта в области ИИ: Разработка собственных решений требует наличия специалистов с глубокими знаниями в области искусственного интеллекта, алгоритмов машинного обучения и программирования;
3. Длительное время разработки и внедрения: Процесс разработки и внедрения собственных решений ИИ может занять значительное время, особенно при необходимости проведения исследований и тестирования.

**Использование коммерчески доступных решений**

**Преимущества:**

1. Быстрая реализация и внедрение решений: Коммерчески доступные решения ИИ предлагают готовые платформы и инструменты, что позволяет быстро реализовать и внедрить ИИ-решения без необходимости разработки с нуля;
2. Отсутствие необходимости в глубоких знаниях в области ИИ: Коммерческие решения обычно предоставляют удобные интерфейсы и инструменты, которые не требуют высокой экспертизы в области искусственного интеллекта;
3. Поддержка и обновления: Коммерческие решения обычно предлагают техническую поддержку и регулярные обновления, что облегчает использование и поддержку решений ИИ.

**Недостатки:**

1. Ограниченные возможности настройки и адаптации: Коммерчески доступные решения могут быть ограничены в возможностях настройки и адаптации под специфические потребности организации;
2. Зависимость от поставщика: При использовании коммерческих решений возникает зависимость от поставщика, что может ограничить свободу действий и гибкость внедрения;
3. Высокие затраты на лицензии и обслуживание: Использование коммерческих решений может требовать значительных затрат на покупку лицензий и оплату обслуживания.

**Использование открытых платформ и инструментов**

**Преимущества:**

1. Большое сообщество и поддержка разработчиков: Открытые платформы ИИ часто имеют активное сообщество разработчиков, которые обмениваются знаниями и предоставляют поддержку;
2. Бесплатная доступность и низкие затраты: Многие открытые платформы и инструменты ИИ доступны бесплатно или по низкой стоимости, что снижает затраты на внедрение решений ИИ;
3. Гибкость и настраиваемость: Открытые платформы обычно предоставляют гибкость настройки и адаптации решений под уникальные потребности организации.

**Недостатки:**

1. Требуется высокая техническая компетенция: Работа с открытыми платформами и инструментами ИИ требует высокой технической компетенции и знания в области программирования и алгоритмов машинного обучения;
2. Отсутствие гарантии качества и поддержки: При использовании открытых решений ИИ может быть ограничена гарантия качества и доступность технической поддержки.

**Фильтрация и классификация сообщений**

Одной из ключевых областей применения ИИ в борьбе с электронными атаками является фильтрация и классификация входящих сообщений. С помощью алгоритмов машинного обучения системы могут автоматизировать процессы идентификации подозрительной корреспонденции, что позволяет значительно ускорить процесс фильтрации и повысить его точность. Используя нейронные сети, системы способны анализировать текстовое содержание и метаданные сообщений, выявляя скрытые признаки спама и фишинга, которые могут быть неочевидны при традиционных подходах.

**Обучение на основе поведения**

ИИ также используется для адаптации систем безопасности к специфическим угрозам, которым подвержена организация. Системы, обученные на реальном трафике электронной почты компании, могут эффективно распознавать и блокировать атаки, нацеленные специально на эту организацию. Методы обучения с подкреплением позволяют системам постоянно совершенствовать свои алгоритмы на основе новых данных, тем самым постоянно повышая уровень защиты.

**Прогнозирование и предупреждение атак**

Применение ИИ для прогнозирования атак является ещё одним перспективным направлением. Анализируя поведение отправителей и характеристики предыдущих атак, ИИ может не только реагировать на угрозы, но и предсказывать потенциальные атаки до их осуществления. Это позволяет организациям предпринимать профилактические меры и снижать вероятность успешного проникновения угрозы.

**Адаптация и персонализация защитных механизмов**

Кроме того, ИИ способен адаптировать защитные механизмы к индивидуальным особенностям каждого пользователя. Анализируя, как пользователи взаимодействуют с электронной почтой и какие сообщения они регулярно помечают как спам, системы научаются лучше распознавать, что именно считать спамом для конкретного человека, тем самым уменьшая количество ошибочно помеченных писем.

**Вывод**

В целом, использование искусственного интеллекта в защите электронной почты от спама и атак представляет собой значительный потенциал для улучшения эффективности и безопасности коммуникационных систем. ИИ способствует автоматизации и оптимизации процессов идентификации и фильтрации спама, а также предоставляет инструменты для

предсказания и предотвращения новых атак. Однако, успешная реализация таких систем требует тщательного учета финансовых и информационных ресурсов, а также неукоснительного соблюдения политик конфиденциальности и защиты данных.

В будущем ИИ, безусловно, будет продолжать развиваться и находить все более эффективное применение в области информационной безопасности, особенно в контексте защиты электронной почты. Это будет способствовать не только повышению безопасности данных, но и улучшению оперативности обработки почтовых сообщений, уменьшению числа ошибочно заблокированных легитимных писем и повышению общей удовлетворенности пользователей услугами электронной почты.

## Обоснование выбора технологий ИИ для защиты от спама и атак по электронной почте

### Улучшение точности обнаружения угроз

Использование алгоритмов машинного обучения и нейронных сетей позволяет системам безопасности адаптироваться к новым и изменяющимся видам атак, обучаясь на реальных данных. Это обеспечивает высокую точность в определении как известных, так и новых угроз, минимизируя количество ложных срабатываний.

### Автоматизация процессов фильтрации

Автоматизация с помощью ИИ значительно ускоряет процессы анализа и фильтрации входящих сообщений, освобождая ресурсы и время специалистов по информационной безопасности для более сложных задач. Это также позволяет системам безопасности оперативно реагировать на угрозы без постоянного вмешательства человека.

### Адаптация и персонализация защитных механизмов

Технологии ИИ способны адаптироваться к специфическим требованиям и поведению пользователей в системе. Это позволяет персонализировать параметры безопасности для каждого пользователя или группы, увеличивая эффективность защиты и минимизируя риск нарушения безопасности.

**Повышение прогностической способности систем**

ИИ может анализировать большие объемы данных о поведении отправителей и характеристиках сообщений, что позволяет не только обнаруживать, но и предсказывать потенциальные атаки до их осуществления. Это помогает предпринимать профилактические меры, снижая риск успешных атак.Исходя из этих преимуществ, использование машинного обучения и нейронных сетей в системах защиты от спама и электронных атак представляется логичным и обоснованным шагом. Таким образом, результаты данной главы подтверждают актуальность исследования и обосновывают выбор машинного обучения и нейронных сетей как ключевых технологий ИИ для дальнейшего изучения и реализации в последующих главах работы.

**Результаты и выводы первой главы**

В рамках первой главы был проведен глубокий анализ подходов и методов защиты от спама и электронных атак с использованием искусственного интеллекта. Результаты анализа разделов, составляющих данную главу, позволяют сделать следующие выводы и результаты:

1. В разделе "Анализ подходов использования ИИ" были рассмотрены различные аспекты искусственного интеллекта, включая сложности его определения и классификацию. Было выявлено, что ИИ охватывает широкий спектр технологий и методов, позволяющих машинам анализировать информацию, обучаться на данных и принимать решения. Этот анализ подчеркнул важность ИИ как инструмента для улучшения защиты от спама и электронных атак;

2. В разделе "Сравнительный анализ преимуществ и недостатков различных подходов внедрения решений ИИ" были оценены три основных подхода: разработка собственных решений, использование коммерчески доступных решений и использование открытых платформ и инструментов. Анализ показал, что каждый подход имеет свои уникальные преимущества и недостатки, которые следует учитывать при выборе стратегии защиты электронной почты;
3. В разделе "Обзор подходов внедрения решений ИИ в системы защиты от спама и атак по электронной почте" было рассмотрено, как ИИ может улучшить процессы идентификации и фильтрации спама, обучения на основе поведения пользователей и предсказания атак. Описаны технологии, такие как машинное обучение и нейронные сети, которые обеспечивают эффективную классификацию и фильтрацию сообщений, а также адаптацию к новым угрозам.

В целом, результаты первой главы демонстрируют, что использование искусственного интеллекта в системах защиты от спама и электронных атак не только улучшает их эффективность, но и способствует более глубокому пониманию и анализу угроз. ИИ предоставляет мощные инструменты для повышения безопасности, оптимизации ресурсов и уменьшения финансовых потерь, связанных с электронными атаками, способствуя созданию более защищенной и продуктивной рабочей среды.

# ГЛАВА 2: Модель и методы классификации сообщений для фильтрации «спама»

Различные виды спам-фильтров включают в себя фильтр списка разрешенных/запрещенных адресов, фильтр заголовков, фильтр на основе содержимого и т.д. Стратегия "Список запрещенных" блокирует сообщения с адресов электронной почты и IP-адресов, которые известны как спамеры. Стратегия "Список разрешенных" определяет, от кого разрешено получать сообщения. Фильтры заголовков проверяют заголовок электронного письма на наличие информации об источнике, содержании сообщения и других данных, содержащихся в заголовке. Фильтры на основе содержимого в первую очередь предназначены для проверки содержимого сообщений и определения того, является ли оно спамом. И наоборот, по мере того, как спамеры становятся все более изобретательными в адаптации уникальных способов изменения информации, которая идентифицирует спам — как правило, с большей сложностью, — все алгоритмы сталкиваются с этим.

Для решения таких задач было использовано множество различных классификаторов машинного обучения [31]. Выбор признаков осуществляется с помощью этих процедур, которые извлекают данные из уже подготовленных наборов данных и используют их при обучении классификатора. Алгоритмы машинного обучения, в большинстве случаев хорошо известные в области классификации нежелательной почты, включают в себя наивный байесовский алгоритм, метод опорных векторов, дерево решений и случайный лес. Наивный байесовский классификатор - это классификатор, позволяющий определить вероятность принадлежности наблюдения или объекта к одному из классов. В то же время независимость атрибутов предполагается только в отношении признаков и является естественной для упрощения при проведении сопутствующих вычислений. Вот почему этот метод называется наивным

(простым) Байесовский классификатор [29]. SVM - это обучающая машина, разработанная на основе статистической теории обучения.

Основная идея, лежащая в основе этого подхода, заключается в отображении исходных векторов в пространство более высокой размерности, в котором выполняется поиск разделяющей гиперплоскости с максимальным зазором. По обе стороны гиперплоскости, разделяющей классы, строятся две параллельные гиперплоскости. Разделяющая гиперплоскость - это гиперплоскость в том направлении, которое увеличивает расстояние между двумя параллельными гиперплоскостями. Этот алгоритм работает в предположении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше средняя ошибка классификатора [29]. Дерево решений - это дерево, в котором каждый узел представляет проверку атрибута, а конечные узлы обеспечивают классификацию. Корень упорядочен, и тестовая модель начинается со значений свойств, проверяя значения свойств в каждом узле и упорядочивая соответствующую ветвь до конечного узла, который дает классификацию. Случайный лес - это алгоритм управляемого обучения. Алгоритм случайного леса создает несколько деревьев решений для решения проблемы и, наконец, выбирает наилучшее решение путем голосования. Эти алгоритмы машинного обучения являются лучшими и наиболее эффективными среди других методов обнаружения спама [11].

### 2.1.1 Набор данных №1: Набор данных о сборе SMS-спама

Коллекция SMS-спама - это общедоступный набор помеченных SMS-сообщений, собранных для исследования спама с мобильных телефоно. Коллекция содержит в общей сложности 5574 сообщения как спама, так и нежелательной почты. Она включает в себя четыре набора данных из разных источников. Во-первых, это коллекция из 425 SMS-спам-сообщений, которые были вручную извлечены с веб-сайта Grumbletext.

Вторым выбран случайным образом “не спам” SMS-сообщения. Это подмножество из 3375 из примерно 10 000 допустимых сообщений, список из 450 SMS-сообщений, собранных с сайта Кэролайн Тэг.

Третьим выбран Corpus v.0.1 Big. В ней содержится 1002 SMS-сообщения для новичков и 322 спам-сообщения. Этот набор данных содержит по одному сообщению в строке. Каждая строка содержит два столбца: категория содержит метку (не спам или спам).

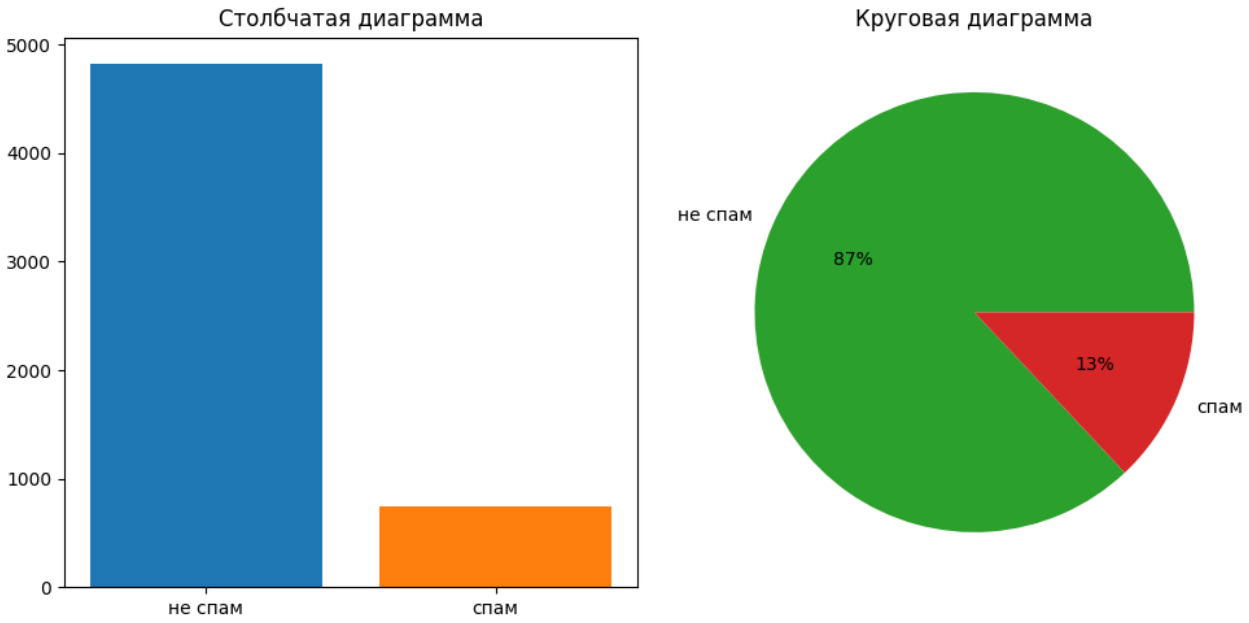



**Рисунок 2 –** Графическое распределение не спам и спам-сообщений в наборе данных

Более подробный анализ набора данных по обеим категориям (не спам и спам) показывает слова, которые часто
встречаются в тексте (рисунок ), а также длину сообщений (рисунок ).

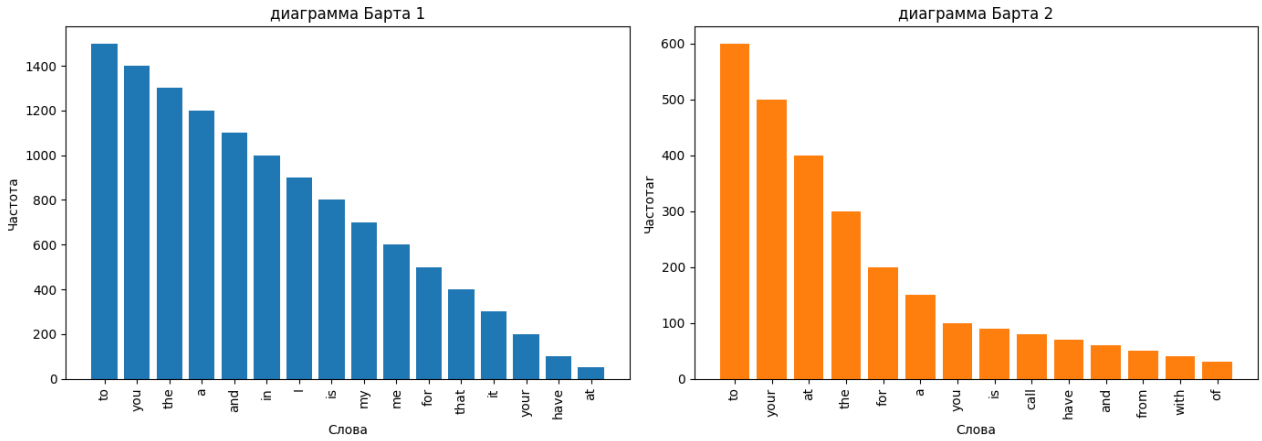



**Рисунок 3 –** Частота “спам” и “не спам” слов в сообщениях

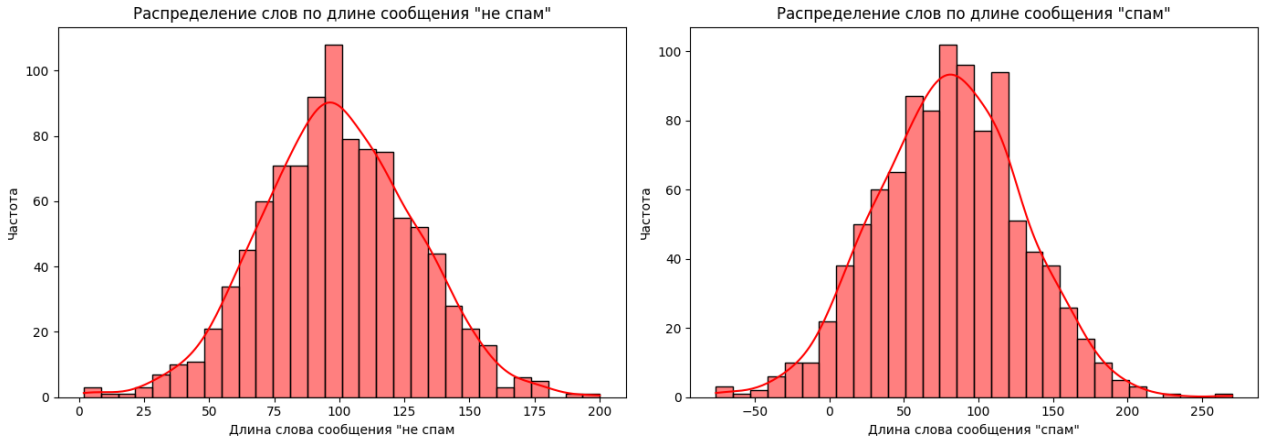



**Рисунок 4 –** Гистограммы длины сообщения как в не спам, так и в спаме.

### 2.1.2 Набор данных №2: Набор данных Enron

Этот набор данных содержит данные от 150 пользователей, в том числе около 500,000 электронных писем, в основном от высшего руководства корпорации Enron. Комиссия получила эти электронные письма в ходе расследования краха Enron. Эти данные представляют собой обширную коллекцию "реальной" электронной почты, которая является общедоступной (Enron Corp, 2015). Версия, использованная в этом исследовании, содержит в общей сложности 5171 электронное письмо, в том числе 1499 сообщений спама и 3672 нежелательных сообщения.

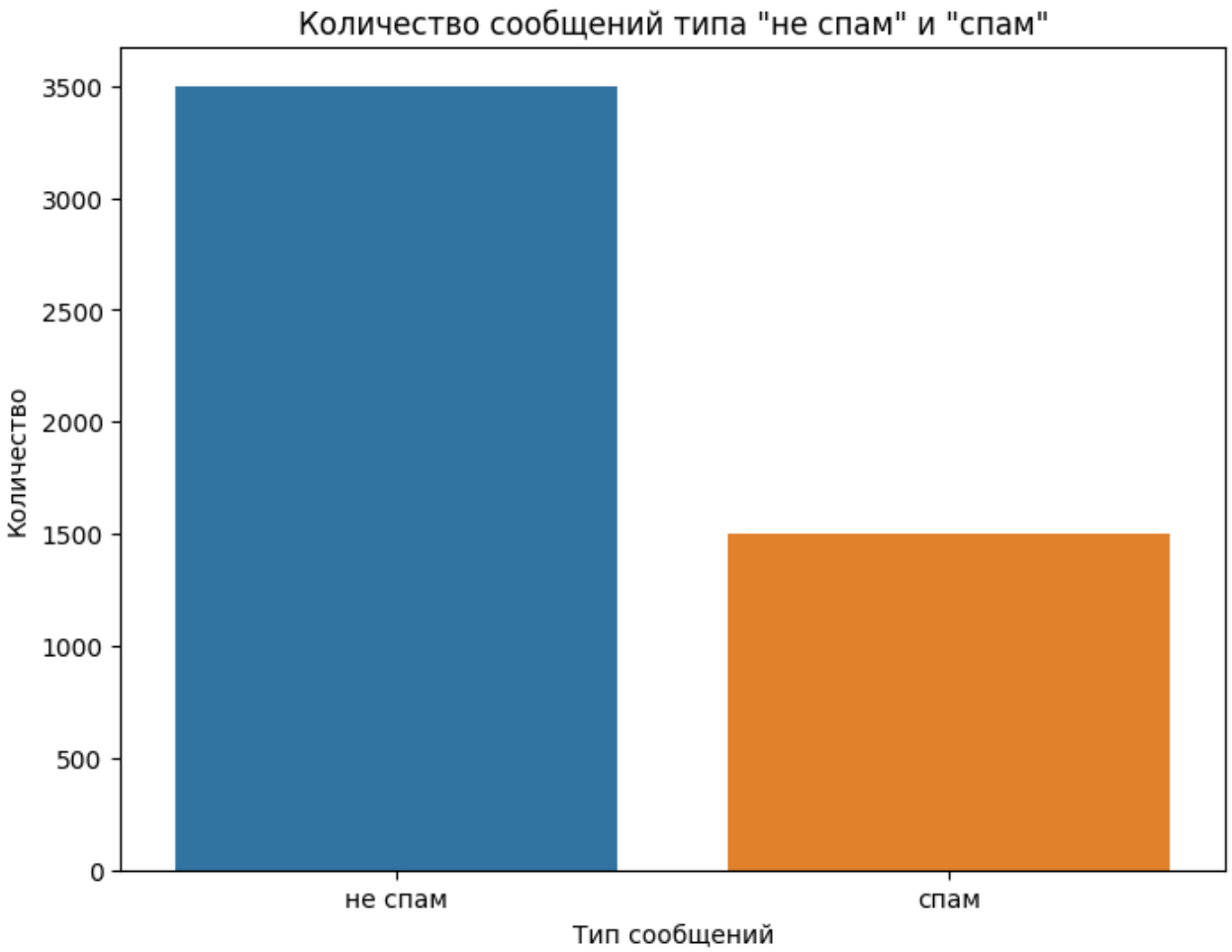



**Рисунок 5 –** Количество нежелательных писем и спама в наборе данных Enron

На рисунке показано графическое распределение нежелательных и спам-сообщений в наборе данных. Он содержит 71% нежелательных и 29% нежелательных сообщений.

### 2.1.3. Набор данных №3: Набор данных для борьбы со спамом

Набор данных Spam assassin представляет собой набор общедоступных данных о спаме по электронной почте, которые могут использоваться исследователями (Apache SpamAssassin, 2004). Assassin corpus содержит 4198 сообщений, собранных с общедоступных форумов. Из этих сообщений 1398

являются спам-сообщениями, которые формируют 33,28% от общего количества сообщений.

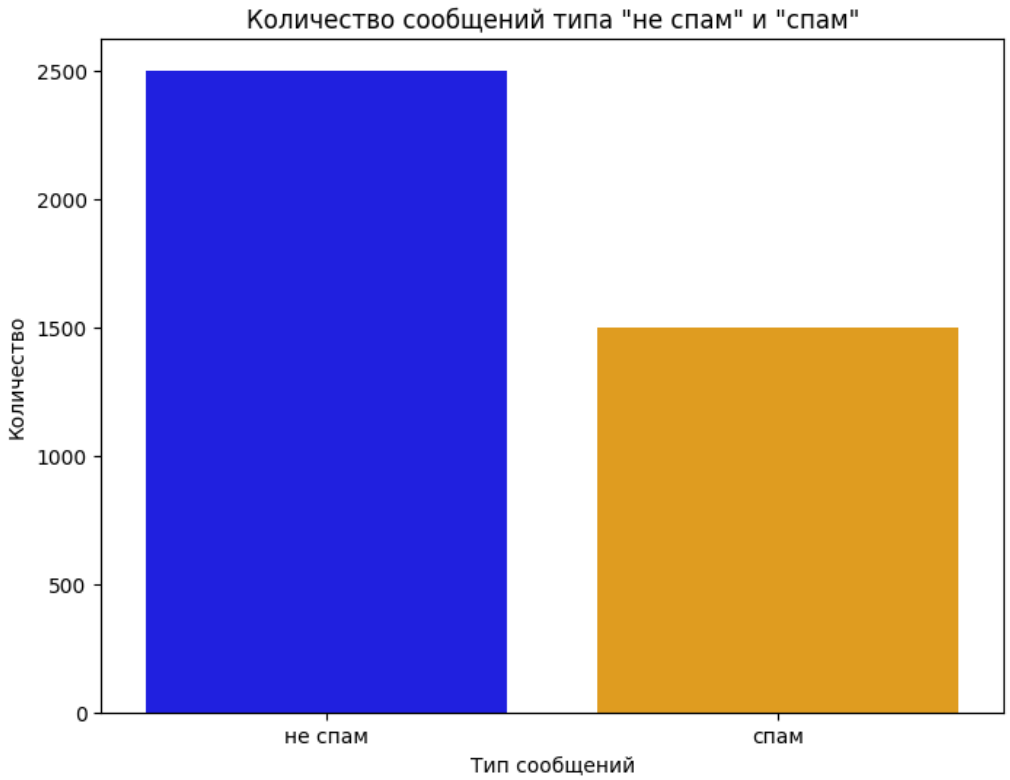



**Рисунок 6 –** Количество нежелательных писем и спама в наборе данных SpamAssassin

### 2.1.4. Набор данных #4: Набор данных о лингвистическом спаме

Набор данных о лингвистическом спаме содержит 2893 сообщения в steam и без спама, собранные из списка лингвистики: моделируемого "списка рассылки о науке и профессии лингвистика". Поскольку большая часть писем, поступающих в этот список, посвящена лингвистическим интересам, таким как объявления о приеме на работу, возможности для исследований и обсуждения программного обеспечения, было легко отделить 2412 сообщений от 481 спама.

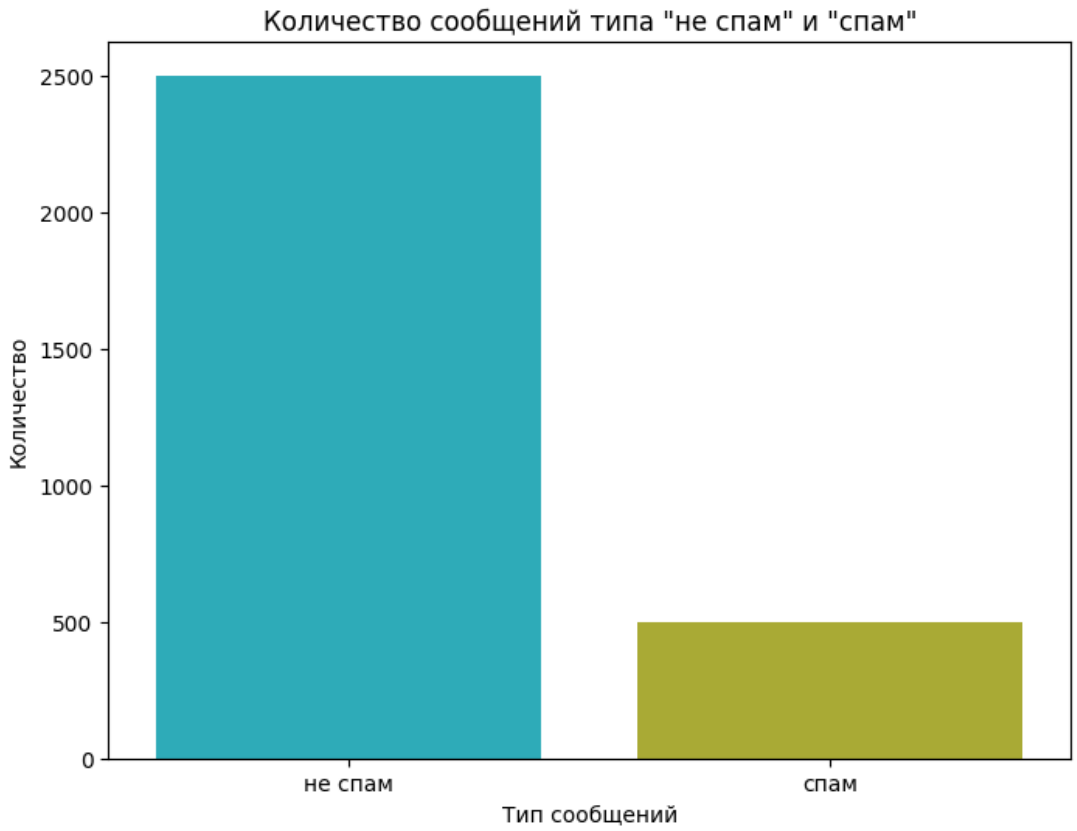



**Рисунок 7 –** Количество нежелательных писем и спама в наборе данных

### 2.2 Модель данных для тренировки модели

Эксперименты проводились с использованием электронных писем из двух наборов данных. Датасет для обучения модели состоит из 31,371 записей, из них  спам-электронных писем и 15,868, "не спам" писем 15,503. Датасет для тренировки модели состоит из 8,175 записей, из них  спам-электронных писем и 4,125, "не спам" писем 4,050.

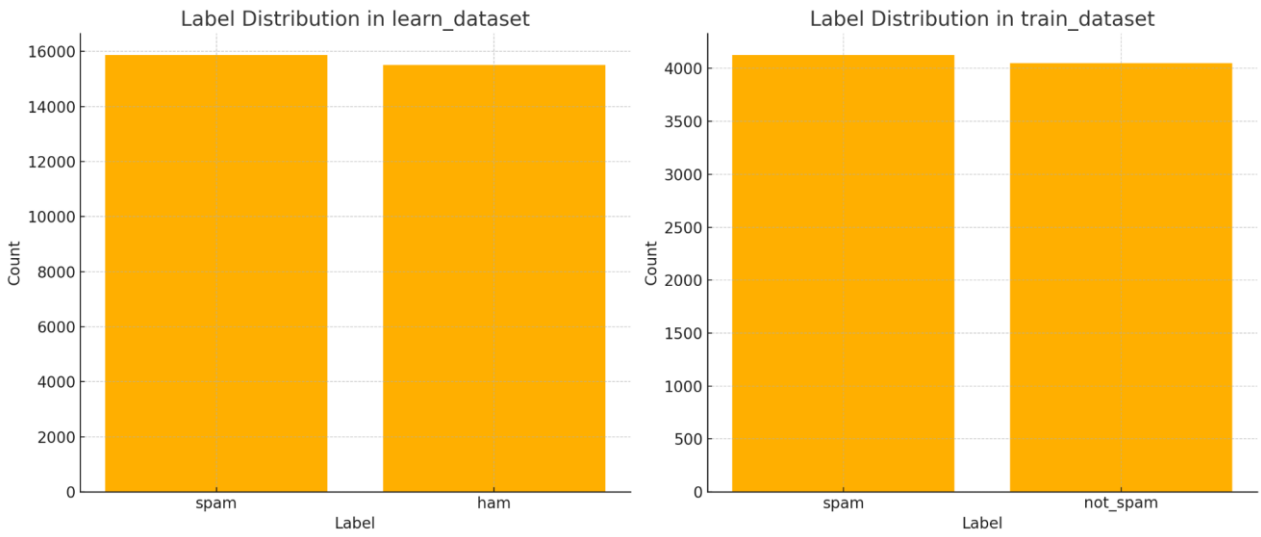



**Рисунок 8 –** Распределение меток в каждом наборе данных

## Визуализация данных

1. Распределение длины текстов.
2. Частота самых популярных слов.

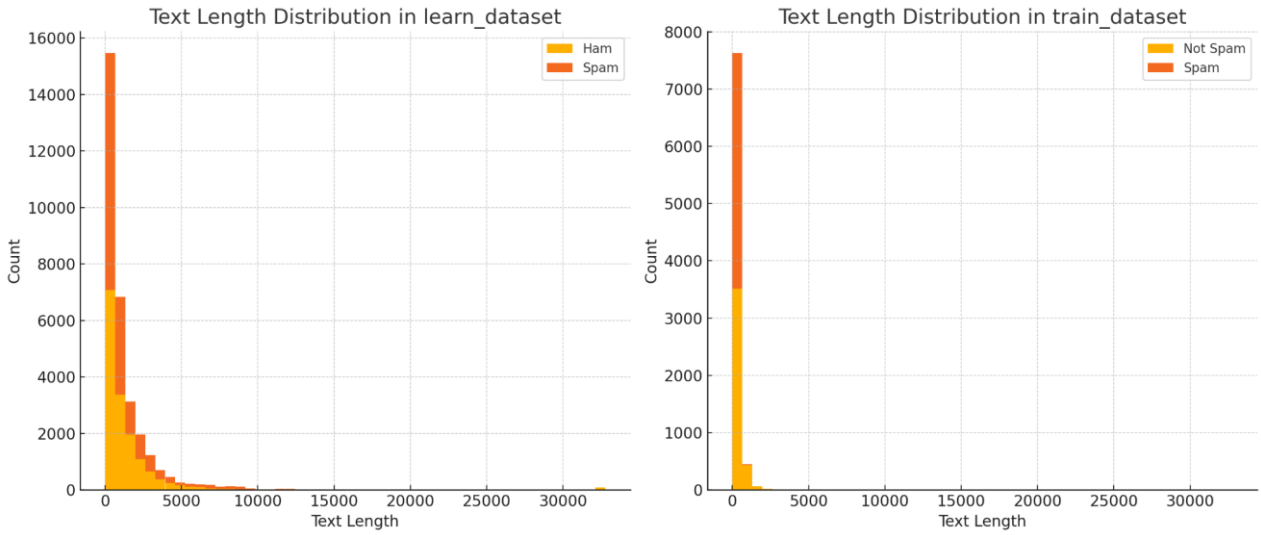



**Рисунок 9 –** Распределение меток в каждом наборе данных

На графиках выше видно, что распределение длины текстов различается для различных меток:

В учебном датасете:

1. "Ham" сообщения имеют более широкий диапазон длин текстов, часто превышающих 2000 символов;
2. "Spam" сообщения, как правило, короче, хотя также встречаются длинные сообщения.

В тренировочном датасете

1. "Not Spam" сообщения имеют более широкий диапазон длин текстов, с некоторыми очень длинными сообщениями;
2. "Spam" сообщения обычно короче, реже превышая 1000 символов.

Теперь визуализируем частоту самых популярных слов для каждой метки в обоих наборах данных.

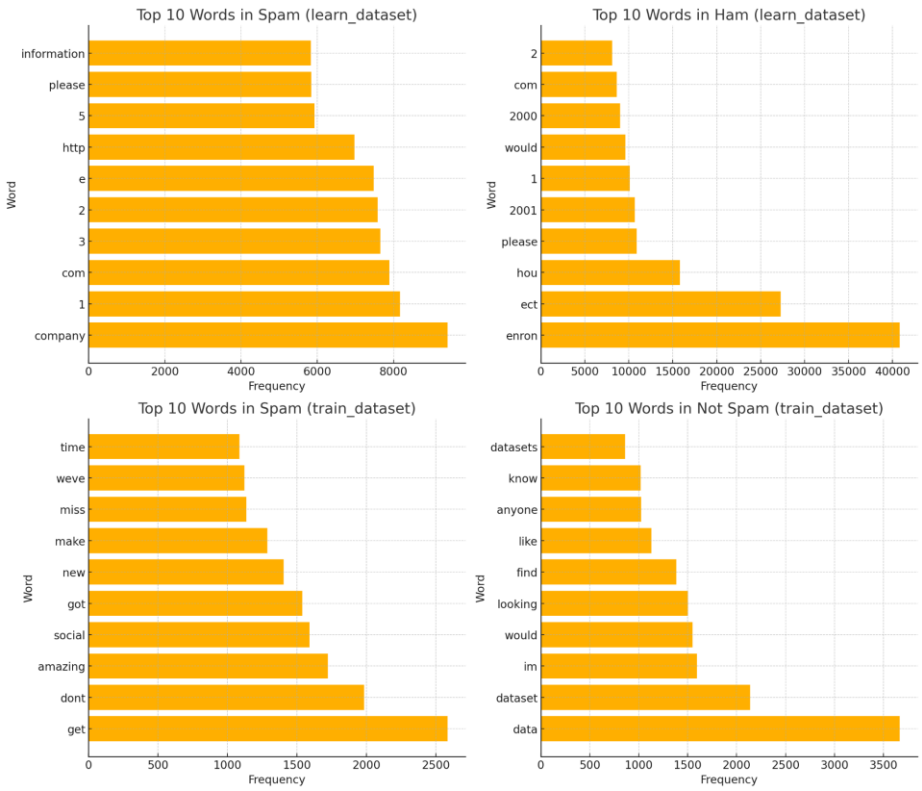



**Рисунок 10 –** Частота слов в датасетах

**Метрики оценки**

Оценочные показатели, приведенные в Таблице, использовались при оценке и сравнении моделей, представлены в таблице 2.

Таблица 2 – Метрики оценки

| **Метрики оценки** | **Описание** |
|---|---|
| Матрица путаницы | Матрица путаницы является одним из наиболее фундаментальных представлений в классификации, поскольку она описывает всю производительность и может использоваться для оценки всех показателей. В матрице это представление правильно и ошибочно классифицированных экземпляров в тестовом наборе. |
| Оценка точности | Процент наблюдений, которые были правильно классифицированы |
| Оценка чувствительности | Это доля реальных положительных случаев, которые наша модель прогнозирует как положительные<br><br>**Чувствительность = истинно положительный результат / (истинный положительный результат + ложный отрицательный результат)** |
| Оценка специфичности | Специфичность также известна как истинно отрицательный показатель, который представляет собой долю фактически отрицательных случаев, которые наша модель прогнозирует как отрицательные.<br><br>**Специфичность = истинно отрицательный результат / (истинно отрицательный + ложноположительный)** |
| Площадь под кривой (AUC) Оценка | При различных пороговых условиях показатель AUC является выходным показателем для задач классификации. Степень или показатель разделимости обозначается AUC. Это показывает, насколько хорошо модель может различать классы. Чем выше AUC, тем точнее модель. |

### 2.3.1 Алгоритмы

#### Наивный Байес

Наивный Байес (NBA)[21] – это алгоритм классификации, основанный на теореме Байеса с предположением о независимости признаков. Например, автомобиль может быть признан спорткаром, если он низкий, двухдверный и его максимальная скорость превышает 250 километров в час. Даже если эти параметры зависят друг от друга, они вносят независимый вклад в вероятность того, что данный автомобиль является спорткаром.. Даже если эти параметры зависят друг от друга, они вносят независимый вклад в вероятность того, что этот фрукт является яблоком. В связи с этим предположением алгоритм называется "наивным". Наивный Байес очень прост и чрезвычайно полезен при работе с очень большими наборами данных. Несмотря на свою простоту, NBA способен превосходить даже некоторые сложные алгоритмы классификации. Все параметры модели можно приблизить с помощью относительных частот из тренировочного набора данных. Это[30] оценки максимально правдоподобных вероятностей. Если данный класс и значение свойства никогда не встречаются вместе в тренировочном наборе, то вероятность, основанная на оценке, будет равна нулю. Это проблема, поскольку умножение на нулевую оценку приведет к потере информации о других вероятностях. Поэтому предпочтительнее вносить небольшие коррекции во все оценки вероятности, чтобы ни одна вероятность не была строго равна нулю.

$$P(A|B) = (P(B|A) * P(A)) / P(B)$$


**Рисунок 11 –** Принцип работы Наивного Байеса

**Машина опорных векторов**

Машина опорных векторов (SVM)[12][10] – это линейный алгоритм, используемый в задачах классификации. Этот алгоритм широко используется на практике для решения как линейных, так и нелинейных задач. Алгоритм SVM предназначен для идентификации экземпляров, ближайших к разделяющей гиперплоскости.[2] Эти точки называются опорными векторами. Затем алгоритм вычисляет расстояние между опорными векторами и разделяющей гиперплоскостью. Это расстояние называется зазором классификатора.

Одна из главных целей этого алгоритма – максимизировать расстояние между опорными векторами и разделяющей гиперплоскостью[26]. Лучшая гиперплоскость считается гиперплоскостью, для которой этот зазор максимально возможен. Большую часть времени набор данных не может быть разделен линейно. Однако этот набор данных можно разделить линейно, проецируя его в более высокое измерение[16]. Один из его параметров - C, регулирует зазор между опорными векторами и разделяющей гиперплоскостью. Чем выше значение C, тем меньше зазор, и больше объектов в тренировочном наборе будут правильно классифицированы. Однако с

большими значениями C вероятность того, что модель будет обобщаться и показывать такие же хорошие результаты на новых данных, очень мала. Чем ниже значение C, тем больше зазор, что приводит к более низкой точности. Поэтому важно корректировать параметры модели для конкретного набора данных, чтобы избежать переобучения, но при этом достичь высокой точности. Параметр gamma определяет, насколько далеко каждый элемент в наборе данных оказывает влияние на определение "идеальной гиперплоскости". Чем ниже gamma, тем больше элементов, даже тех, которые достаточно далеко от разделяющей гиперплоскости, участвуют в процессе ее выбора. Если gamma высока, то алгоритм будет "опираться" только на те элементы, которые находятся ближе всего к гиперплоскости[2]. Если значение gamma установлено слишком высоко, то только элементы, ближайшие к гиперплоскости, будут участвовать в процессе принятия решения о местоположении гиперплоскости. Это поможет игнорировать выбросы в данных.

Алгоритм SVM устроен таким образом, что точки, расположенные ближе друг к другу, имеют больший вес при принятии решения. Однако с правильными настройками для C и gamma можно достичь оптимального результата, который строит более линейную гиперплоскость, игнорирующую выбросы и, следовательно, более обобщенную[14].

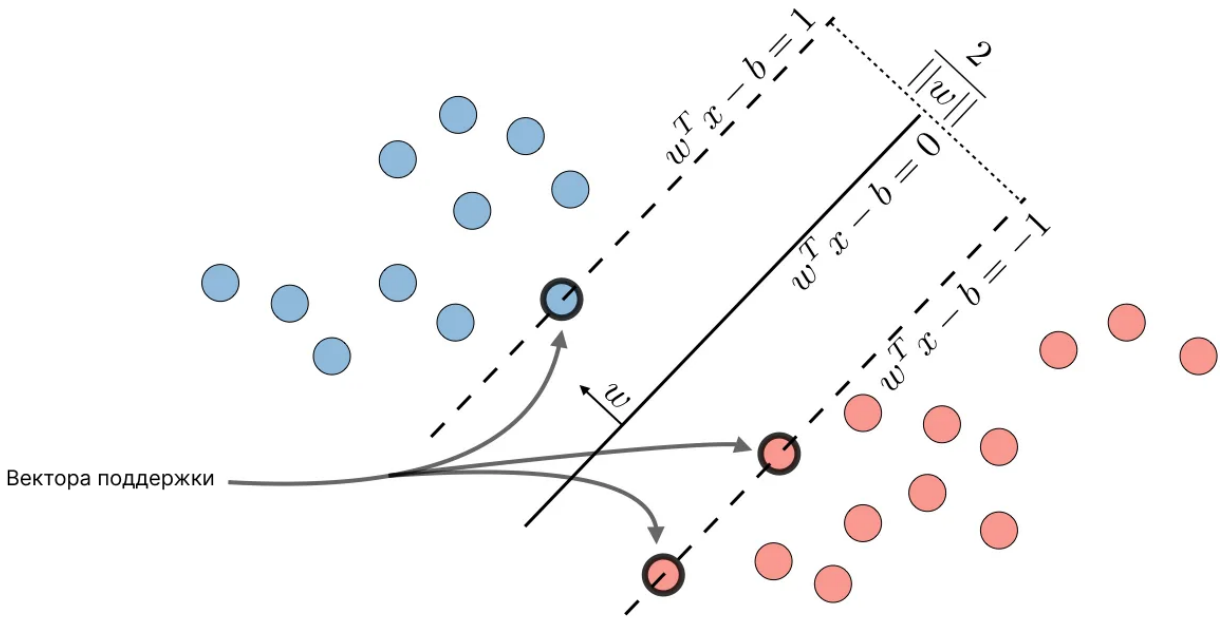


**Рисунок 12 –** Графическое представление работы SVM

## Дерево решений

Деревья решений[36] - это один из лучших инструментов интеллектуального анализа данных и прогнозной аналитики, который позволяет решать задачи классификации и регрессии. Само дерево решений представляет собой метод установления правил принятия решений в виде иерархической структуры, построенной из двух видов элементов: узлов и листьев. Правила принятия решений находятся в узлах[21], и каждый экземпляр оценивается по правилам в узлах, проходя от корня к листу.

В простейшем случае по результатам проверки набор примеров, попадающих в узел, делится на два подмножества: в одно входят примеры, удовлетворяющие правилу, а в другое - не удовлетворяющие. Процедура повторяется для каждой подгруппы и продолжается до тех пор, пока алгоритм обучения не достигнет критерия завершения. Итак, для этого последнего узла проверка и разделение не выполняются, и узел объявляется конечным. Лист определяет решение для каждого примера, попавшего в него[19]. Для классификационного дерева это класс, связанный с узлом, а для регрессионного дерева – модальный интервал целевой переменной, соответствующий листу. Это, если, в отличие от узла, лист не содержит правила, но содержит подмножество объектов, которые удовлетворяют всем правилам ветви, заканчивающейся этим листом.

Очевидно, что для того, чтобы перейти к списку, пример должен удовлетворять всем правилам, которые существуют на пути к этому списку. Поскольку путь в дереве, реализованный для каждого из листьев, уникален, это приводит к тому, что пример приходится на один и только один лист, что делает результаты уникальными. Процесс построения деревьев решений состоит в последовательном[22], рекурсивном разделении тренировочного набора на

подмножества с использованием правил принятия решений в узлах. Процесс разделения продолжается до тех пор, пока все узлы в конце всех ветвей не будут объявлены листьями. Объявление узла листом может происходить естественным образом (когда в нем будет содержаться один объект или объекты только одного класса), или при достижении определенного условия остановки, указанного пользователем (например, минимально допустимого количества примеров в узле или максимальной глубины дерева). Алгоритмы построения деревьев решений относятся к так называемым жадным алгоритмам[35].

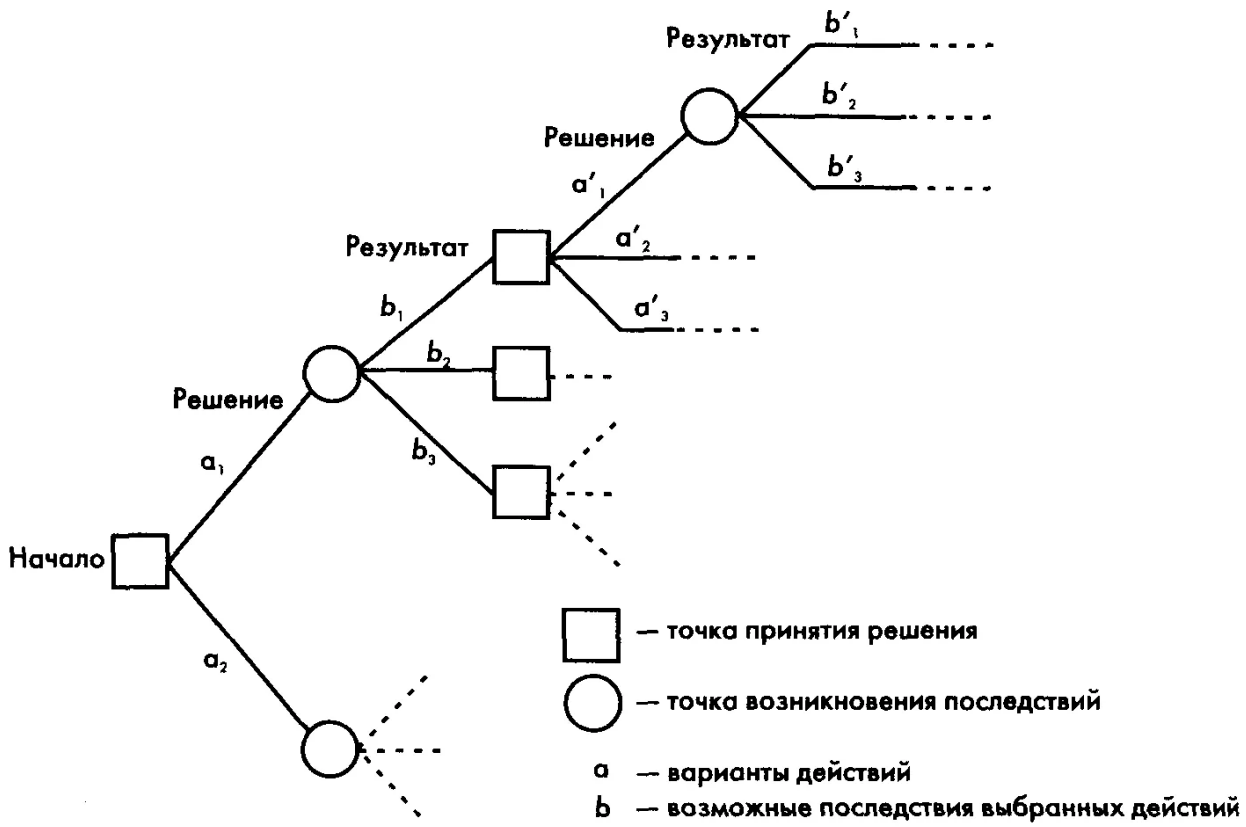



**Рисунок 13 –** Принцип работы алгоритма дерева решений

## Random Forest

Случайный лес – это ансамблевый метод машинного обучения, который создает множество деревьев решений и объединяет их для получения более

точного и стабильного прогноза. Каждый отдельный дерево обучается на случайном подмножестве данных, а прогнозы всех деревьев усредняются для получения окончательного прогноза. Алгоритм Случайный лес основывается на концепции объединения нескольких деревьев решений, создавая «лес» из случайных деревьев решений. Принцип работы этого алгоритма заключается в следующем:

**Выбор подвыборок данных:** Из исходного набора данных с помощью метода бутстреппинга (bootstrap) случайным образом выбираются подвыборки. Каждая подвыборка используется для построения отдельного дерева решений.

**Построение деревьев решений:** Для каждой подвыборки строится дерево решений. В процессе построения дерева на каждом узле случайным образом выбирается подмножество признаков, и из этого подмножества выбирается лучший признак для расщепления узла. Это помогает уменьшить корреляцию между деревьями и повысить общую устойчивость модели.

**Принятие решения:** Для задач классификации каждое дерево в лесу выдает прогноз, и финальное решение принимается на основе голосования большинством (majority voting). Для задач регрессии прогнозы всех деревьев усредняются, чтобы получить окончательный результат.

**Регулировка параметров:** Алгоритм имеет несколько важных параметров, которые необходимо настроить для достижения оптимальной производительности:

1. **Количество деревьев (n_estimators):** Чем больше деревьев в лесу, тем точнее прогноз, но это также увеличивает вычислительную сложность и время обучения;
2. **Максимальная глубина дерева (max_depth):** Ограничение глубины деревьев помогает избежать переобучения. Глубокие деревья могут идеально подогнаться под обучающие данные, но могут плохо обобщаться на новых данных;

3. **Минимальное количество выборок для разделения узла (min_samples_split):** Этот параметр регулирует минимальное количество выборок, необходимых для разделения узла. Увеличение этого значения может помочь уменьшить переобучение;
4. **Минимальное количество выборок в листе (min_samples_leaf):** Этот параметр задает минимальное количество выборок, которые должны быть в листовом узле. Увеличение этого значения также помогает уменьшить переобучение.

Случайный лес устраняет проблемы переобучения и снижает дисперсию модели.

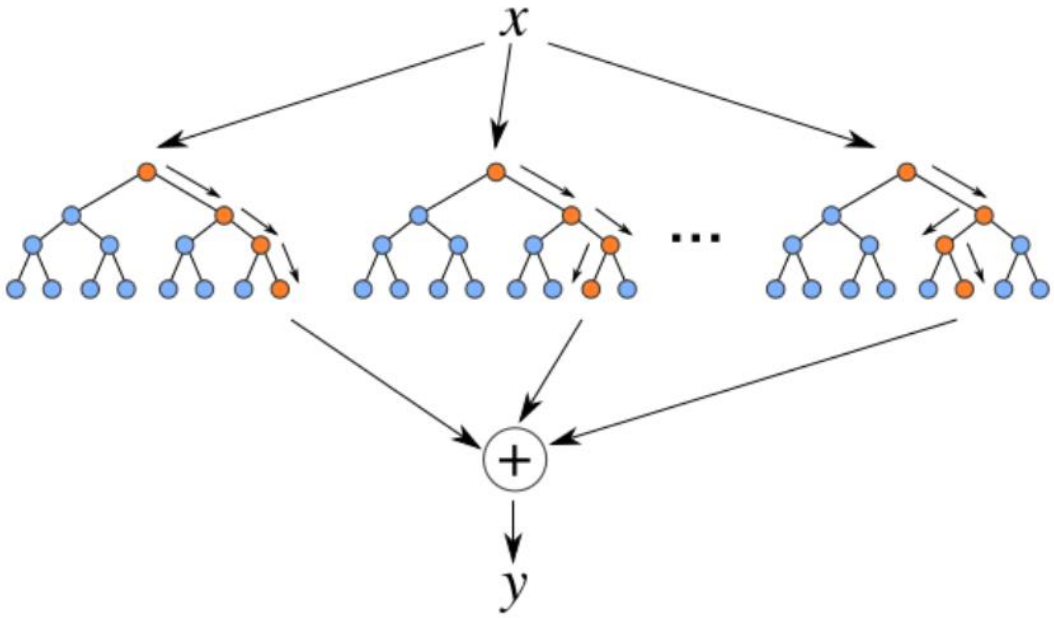



**Рисунок 14 –** Принцип работы алгоритма случайный лес

## 2.4 Метрики оценки работы алгоритмов

### Матрица путаницы(Confusion Matrix)

Описание основ машинного обучения в контексте матрицы путаницы на самом деле весьма полезно. CM показывает количество правильно классифицированных положительных примеров (или не являющихся спамом в случае классификации спама; TP = истинно положительный результат),

количество правильно классифицированных отрицательных примеров (или спама в случае классификации спама; TN = истинно отрицательный результат), количество положительных примеров, классифицированных как отрицательные (или не являющихся спамом классифицируется как спам; FN = ложноотрицательный результат) и количество отрицательных сообщений, классифицированных как положительные (или спам, классифицированный как не спам; FP = ложноотрицательный результат).[23]

**Точность**

Точность – это отношение правильно классифицированных примеров ко всем классифицированным примерам. Другими словами, она показывает, сколько из всех классифицированных примеров были отнесены к правильной метке.[23]

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN},$$

**Прецизионность**

Прецизионность – это отношение количества спам-электронных писем, классифицированных как спам (TP), ко всем электронным письмам, классифицированным как спам (TP + FP). Другими словами, сколько писем, классифицированных как спам, действительно являются спамом.[23]

$$precision = \frac{TP}{TP + FP},$$

**Полнота**

Полнота – это отношение количества найденных спам-электронных писем к общему количеству спам-электронных писем в тестовом наборе данных. Другими словами, из всех спам-электронных писем в тестовом наборе

данных (TP + FN), сколько из них правильно классифицировано как спам (TP).[23]

$$recall = \frac{TP}{TP + FN},$$

**F1-оценка**

F1-оценка определяется как взвешенное гармоническое среднее прецизионности и полноты. Другими словами, она измеряет эффективность поиска с учетом того, сколько раз полнота важнее прецизионности.[23]

$$F1\ score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

**Случайный лес**

Таблица 3 – показатели алгоритма на тренировочном и обучаемом датасете

| | **Прецизионность** | **Полнота** | **F1-оценка** | **Выборка** |
|---|---|---|---|---|
| Не спам | **0.51** | **0.92** | **0.65** | **15553** |
| Спам | **0.65** | **0.14** | **0.23** | **16163** |
| Точность | | | **0.52** | **31716** |
| Macro avg | **0.58** | **0.53** | **0.44** | **31716** |
| Взвешенное среднее значение | **0.58** | **0.52** | **0.44** | **31716** |

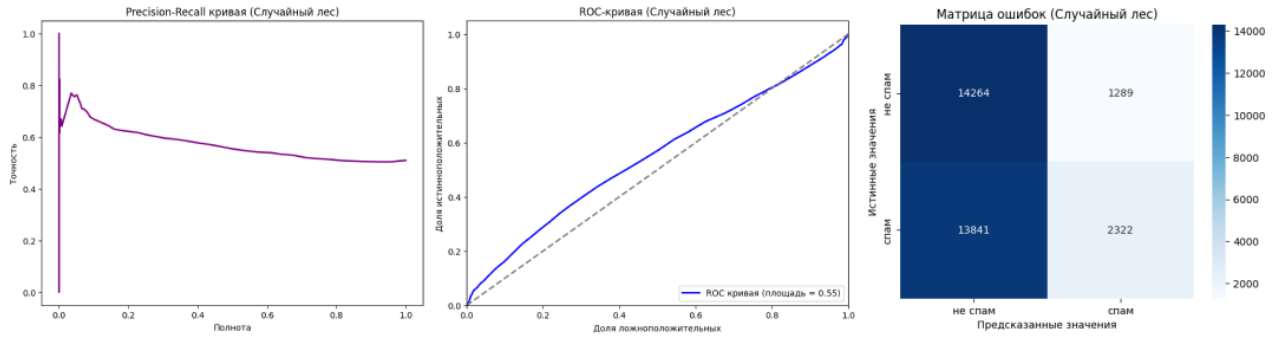



**Рисунок 15 –** Графическое представление результата работы алгоритма

**Наивный Байесовский классификатор**

Прецизионность для обеих категорий (не спам и спам) примерно одинакова и находится на уровне 0.50-0.52. Это указывает на то, что модель делает значительное количество ложноположительных предсказаний в обеих категориях. Наивный Байесовский классификатор показывает умеренные результаты в классификации спам-сообщений. Прецизионность и полнота для категорий спам и не спам близки друг к другу, однако их значения не высоки. Общая точность модели составляет 51%, что говорит о необходимости улучшения модели. Для улучшения производительности можно рассмотреть использование более сложных моделей, таких как нейронные сети или ансамбли методов, а также применение различных техник предобработки данных и балансировки классов.

Таблица 4 – показатели алгоритма на тренировочном и обучаемом датасете.

| | Прецизионность | Полнота | F1-оценка | Выборка |
|---|---|---|---|---|
| Не спам | **0.50** | **0.41** | **0.45** | **15553** |
| Спам | **0.52** | **0.61** | **0.56** | **16163** |
| Точность | | | **0.51** | **31716** |
| Macro avg | **0.51** | **0.51** | **0.51** | **31716** |
| Взвешенное среднее значение | **0.51** | **0.51** | **0.51** | **31716** |

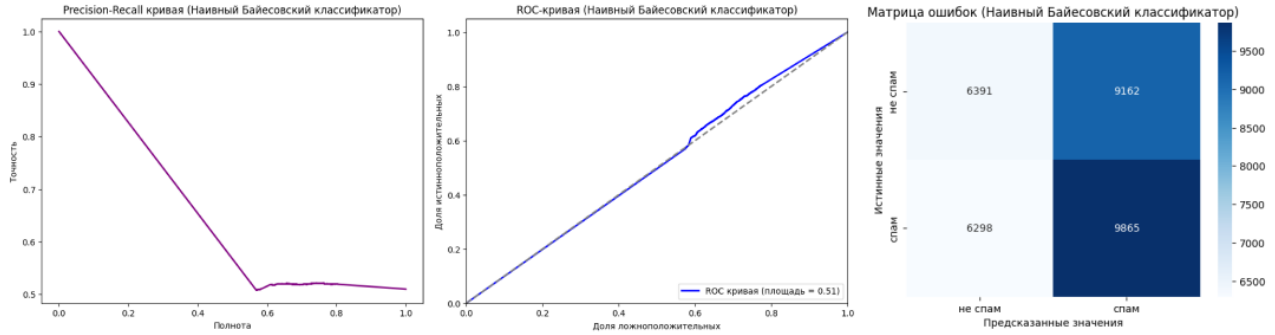



**Рисунок 16 –** Графическое представление результата работы Наивного Байесовского классификатора

**Метод Опорных Векторов (SVM)**

Метод опорных векторов показывает умеренные результаты в классификации спам-сообщений. Прецизионность для спам-сообщений выше, чем для не спам-сообщений, но полнота для спама очень низкая, что указывает на то, что много спам-сообщений не распознаются моделью. Общая точность модели составляет 53%, что говорит о необходимости улучшения модели. Для улучшения производительности можно рассмотреть увеличение количества

данных для спам-сообщений, изменение параметров модели, использование различных ядер для SVM или применение методов балансировки данных.

Таблица 5 – показатели алгоритма на тренировочном и обучаемом датасете

| | **Прецизионность** | **Полнота** | **F1-оценка** | **Выборка** |
|---|---|---|---|---|
| Не спам | **0.51** | **0.90** | **0.66** | **15553** |
| Спам | **0.66** | **0.18** | **0.28** | **16163** |
| Точность | | | **0.53** | **31716** |
| Macro avg | **0.59** | **0.54** | **0.47** | **31716** |
| Взвешенное среднее значение | **0.59** | **0.53** | **0.46** | **31716** |

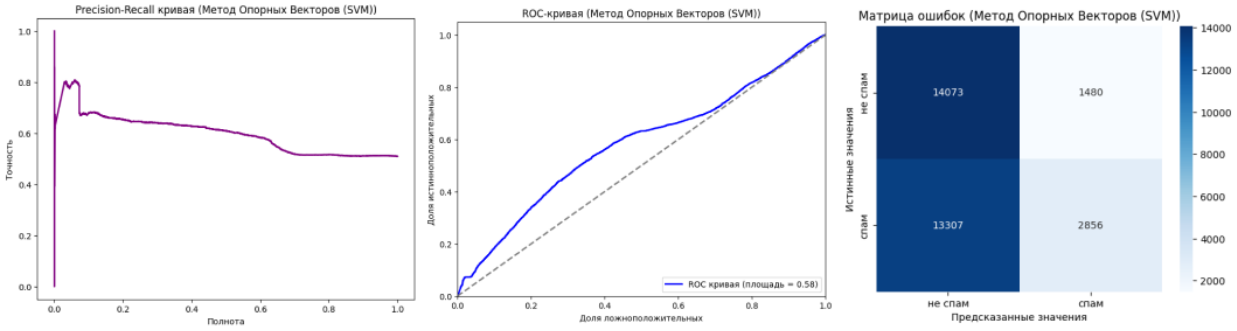



Рисунок 17 – Графическое представление результата работы алгоритма SVM

**Нейронные сети**

Принцип работы заключается в архитектуре нейронной сети, которая может включать слои типа Dense, LSTM, GRU и другие. Входные данные проходят через эти слои, и каждый слой извлекает и обрабатывает признаки для улучшения классификации.

**Обучение модели**: Модель обучается на обучающей выборке с использованием алгоритмов обратного распространения ошибки и оптимизаторов, таких как Adam или SGD. Процесс обучения включает настройку весов модели для минимизации функции потерь.

Данная модель прошла 5 эпох обучения на текущей выборке.Модель показывает хорошие результаты в классификации не спам-сообщений, с высокой полнотой и приемлемой прецизионностью. Однако, для спам-сообщений модель имеет высокую прецизионность, но низкую полноту, что означает, что модель пропускает много спам-сообщений. Для улучшения производительности модели в будущем можно рассмотреть увеличение количества данных для спам-сообщений, изменение архитектуры модели или применение методов балансировки данных, таких как oversampling(избыточная выборка) или undersampling(недостаточная выборка).

Таблица 6 – показатели алгоритма на тренировочном и обучаемом датасете

| | **Прецизионность** | **Полнота** | **F1-оценка** | **Выборка** |
|---|---|---|---|---|
| Не спам | **0.60** | **0.90** | **0.73** | **15553** |
| Спам | **0.88** | **0.18** | **0.53** | **16163** |
| Точность | | | **0.66** | **31716** |
| Macro avg | **0.74** | **0.66** | **0.63** | **31716** |
| Взвешенное среднее значение | **0.74** | **0.66** | **0.663** | **31716** |

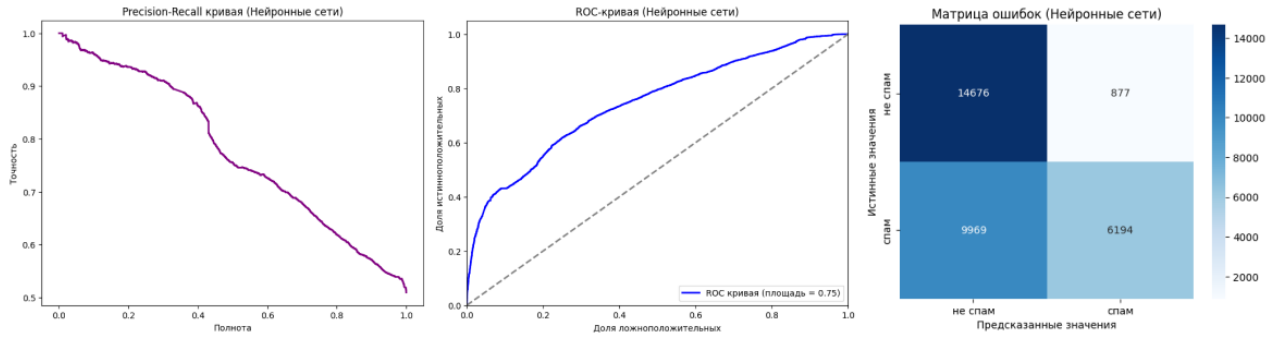



**Рисунок 18 –** Графическое представление результата работы Нейронных сетей

# ГЛАВА 3: Практическая реализация проекта

Для этого подробно описана разработка и внедрение системы фильтрации спама на основе методов машинного обучения и нейронных сетей. В ходе реализации проекта будут последовательно приниматься во внимание следующие аспекты: подготовка данных, выбор модели и обучение, оценка производительности системы, развертывание системы, мониторинг и поддержка. Внимание будет сосредоточено на изучении проблем и ограничений методов исследования, а также на предложении возможных улучшений, основанных на анализе документов, представленных в литературе, и современных научных исследованиях.

## Подготовка данных

Подготовка данных является важнейшим этапом в процессе создания модели машинного обучения, поскольку качество данных напрямую влияет на эффективность модели. В этом разделе мы рассмотрим источники данных, предварительную обработку данных и преобразование текстов в числовые представления[5].

## Предварительная обработка данных

Предварительная обработка данных включает несколько этапов, которые помогают очистить и подготовить данные для обучения модели.

## Очистка данных

Удаление дубликатов и обработка пропущенных значений. Это важно для предотвращения переобучения модели на повторяющихся данных и обеспечения корректности обработки[8].

```
# Удаление дубликатов

df.drop_duplicates(inplace=True)
```

```
# Обработка пропущенных значений

df.dropna(inplace=True)
```

## Преобразование текста

Преобразование текста включает лемматизацию и стемминг, что помогает привести слова к их базовым формам и уменьшить размерность признакового пространства[26].

1. **Лемматизация:** Приведение слов к их начальной форме;
2. **Стемминг:** Отсечение окончаний у слов для получения их основы.

```
import nltk

from nltk.stem import WordNetLemmatizer

from nltk.corpus import stopwords

nltk.download('wordnet')

nltk.download('stopwords')

lemmatizer = WordNetLemmatizer()

stop_words = set(stopwords.words('english'))

def preprocess_text(text):

    words = text.split()

    words = [lemmatizer.lemmatize(word) for word in words if word.lower() not in stop_words]

    return ' '.join(words)

df['processed_text'] = df['Email'].apply(preprocess_text)
```

## Токенизация

Токенизация – это процесс разбиения текста на отдельные слова или токены.

```
from keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(num_words=5000)
```

```
tokenizer.fit_on_texts(df['processed_text'])

sequences = tokenizer.texts_to_sequences(df['processed_text'])
```

## Преобразование текста в числовые представления

После очистки и токенизации текстов необходимо преобразовать их в числовые представления, которые могут быть использованы для обучения модели. Наиболее популярные методы включают TF-IDF и Word2Vec.

## TF-IDF (Term Frequency-Inverse Document Frequency):

TF-IDF взвешивает частоту слов в документе в зависимости от их частоты в корпусе. Это позволяет уменьшить влияние часто встречающихся слов, таких как "the", "and" и т.д.

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=5000)

X_tfidf = vectorizer.fit_transform(df['processed_text']).toarray()
```

## Word2Vec:

Word2Vec создает плотные векторы для слов на основе их контекста в корпусе текстов.

```
from gensim.models import Word2Vec

sentences = [text.split() for text in df['processed_text']]

word2vec_model = Word2Vec(sentences, vector_size=100, window=5, min_count=1, workers=4)
```

## Выделение признаков

Выделение признаков – это процесс преобразования данных в формат, который может быть использован для обучения моделей машинного обучения[7]. В данном разделе рассмотрим текстовые, мета-признаки и структурные признаки.

**Текстовые признаки**

Слова "TF" (Частота использования термина) и "IDF" (Обратная частота использования документа) означают, как часто то или иное слово встречается в данном документе. Таким образом, TF количественно определяет релевантность слова на основе одного документа. IDF определяет, насколько уникальным является слово в коллекции всех документов. Слова, встречающиеся в большинстве документов, имеют низкий идентификатор, поскольку они не имеют информационной ценности[3].

Формула для вычисления частоты термина TF-IDF, обратной частоте документа, объединяет понятия TF и IDF для вычисления важности каждого слова в каждом документе. Формула выглядит следующим образом:

$$\textbf{TF-IDF(t,d) = TF(t,d) * IDF(t)}$$

Модель “bag of words” - это простой формат, используемый для представления в виде текста в области обработки естественного языка и сбора информации. В этой модели текст — будь то одно предложение или весь документ целиком — представляется в виде набора слов, без учета грамматики и порядка слов, но с информацией о количестве этих слов. В основном, набор слов используется в методах классификации документов и при обучении классификаторов[1]. Одно из самых ранних упоминаний о нем в лингвистическом контексте было найдено в 1954 году в статье Зеллига Харриса "Дистрибутивная структура".

В Word Embedding отображаются слова и документы. Концепция Word Embedding, или вектора слов, представляет собой числовой векторный ввод,

который представляет слово в некотором пространстве меньшего размера. Это позволяет словам со схожими значениями иметь идентичные представления. Встраивание слов представляет собой еще одно представление функций из текста, позволяющее нам применять эти функции в моделях машинного обучения для работы с текстовыми данными. Они пытаются сохранить синтаксическую и семантическую информацию. Такие методы, как Bag of Words, CountVectorizer и TFIDF, основаны на подсчете слов в предложении, но они не сохраняют никакой синтаксической или семантической информации.

В таких алгоритмах размер вектора равен количеству элементов в словаре. Мы можем получить разреженную матрицу, если большинство элементов равны нулю. Большие входные векторы подразумевают большое количество весовых коэффициентов, что приводит к большим вычислительным затратам, необходимым для обучения. Встраивание слов решает эти проблемы.

Текстовые признаки являются основными для задач фильтрации спама. Основные методы выделения текстовых признаков включают:

**Мешок слов (Bag of Words):**

1. Преобразование текстов в векторы, где каждый элемент представляет частоту слова в тексте[5].

**TF-IDF:**

1. Взвешивание частоты слов в зависимости от их важности в корпусе текстов[5].

**Word Embeddings:**

1. Создание плотных векторов для слов, которые учитывают их контекст и взаимосвязь[5].

**Пример кода для TF-IDF:**

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=5000)

X_tfidf = vectorizer.fit_transform(df['processed_text']).toarray()
```

### Мета-признаки

Мета-признаки включают информацию о самом письме, которая может быть полезна для классификации:

**Информация о отправителе:**

1. Адрес электронной почты отправителя, домен, IP-адрес и т.д.

**Время отправки:**

1. Дата и время отправки письма.

**Размер письма:**

1. Длина текста, количество вложений и их размер.

```
df['sender_domain'] = df['Email'].apply(lambda x: x.split('@')[1])

df['email_length'] = df['Email'].apply(len)
```

### Структурные признаки

Структурные признаки включают элементы, которые могут быть специфичны для спам-писем:

**Наличие вложений:**

1. Вложенные файлы и их типы.

**Количество ссылок:**

1. Ссылки в тексте письма и их количество.

**HTML-теги:**

1. Использование HTML-разметки в письме.

```
df['has_attachments'] = df['Email'].apply(lambda x: 1 if 'attachment' in x else 0)

df['link_count'] = df['Email'].apply(lambda x: x.count('http'))
```

### Наличие вложений

Спам-письма часто содержат вложения, которые могут быть вредоносными файлами, такими как вирусы или трояны. Анализ наличия и типов вложений может помочь в выявлении спам-писем.

**Типы вложений:** Вредоносные вложения часто имеют специфические расширения, такие как **.exe**, **.zip**, **.rar**, **.docm**, **.xlsm**, которые могут содержать вредоносный код.

```
def has_attachment(email):
    # Пример проверки наличия вложений по ключевым словам
    return 1 if 'attachment' in email.lower() else 0
df['has_attachments'] = df['Email'].apply(has_attachment)
```

### Количество ссылок

Спамеры часто включают множество ссылок в свои письма, чтобы перенаправить получателей на вредоносные или фишинговые сайты. Анализ количества ссылок в письме может быть полезен для выявления спама.

**Ссылки в тексте:** Подсчет количества ссылок в тексте письма.

```
def count_links(email):
    # Пример подсчета количества ссылок в тексте письма
    return email.lower().count('http')
df['link_count'] = df['Email'].apply(count_links)
```

### HTML-теги

Спам-письма часто используют HTML-разметку для маскировки ссылок и вставки вредоносного кода. Анализ использования HTML-тегов может помочь в обнаружении таких писем.

**HTML-разметка:** Проверка наличия HTML-тегов в тексте письма.

```
def has_html(email):
    # Пример проверки наличия HTML-тегов в тексте письма
    return 1 if '<html>' in email.lower() else 0
df['has_html'] = df['Email'].apply(has_html)
```

## Обучение и настройка модели

В данном разделе мы подробно рассмотрим процесс выбора, обучения и настройки модели машинного обучения для фильтрации спама. Этот процесс включает выбор алгоритмов, обучение моделей, настройку гиперпараметров и их оценку на обучающих данных.

### Выбор алгоритмов

Выбор алгоритма машинного обучения является критически важным этапом, так как различные алгоритмы могут показывать разную производительность на одном и том же наборе данных. Рассмотрим наиболее популярные алгоритмы для задачи фильтрации спама:

**Наивный байесовский классификатор:**

1. **Преимущества:** Простота и скорость обучения, хорошая производительность на текстовых данных;
2. **Недостатки:** Предположение о независимости признаков, что не всегда выполняется в реальных данных.

**Метод опорных векторов (SVM):**

1. **Преимущества:** Высокая точность на хорошо структурированных данных, эффективен при высоких размерностях;
2. **Недостатки:** Вычислительная сложность на больших наборах данных, необходимость выбора правильного ядра.

**Случайный лес:**

1. **Преимущества:** Высокая точность, устойчивость к переобучению, возможность обработки данных с большим числом признаков;
2. **Недостатки:** Модель может быть медленной при предсказании на новых данных, особенно для больших лесов.

**Нейронные сети:**

1. **Преимущества:** Способность обрабатывать сложные и неструктурированные данные, высокая производительность на больших наборах данных;
2. **Недостатки:** Требования к большим объемам данных и вычислительным ресурсам, сложность настройки и обучения.

**Гибридный режим**

Далее, сравниваем модели классификации двух методов машинного и глубокого обучения. Для этого были использованы предварительно обработанные наборы данных, а именно: набор данных для сбора SMS-сообщений о спаме, набор данных Enron, набор данных для борьбы со спамом и набор данных Linq. Они были взяты в качестве исходных данных как для моделей классификации машинного, так и для глубокого обучения, и полученные результаты были сопоставлены с использованием обеих моделей. Другими словами, мы рассмотрели три комбинации: статическое встраивание с классификатором машинного обучения, статическое встраивание с классификатором глубокого обучения и динамическое встраивание с классификатором глубокого обучения.

Главное - получить результат правильной классификации, поэтому множество заданий помогает выбрать наилучшую модель. Для сравнения производительности всех классификаторов машинного обучения была использована матрица путаницы. Матрица путаницы - это назначение, которое очень хорошо подходит для сравнения классификаторов ML, поскольку абсолютное число, указанное в матрице путаницы, может вводить в

заблуждение, учитывая, что все четыре набора данных имеют разные размеры. Результаты для всех моделей в четырех наборах данных приведены в таблице. Модели SVM неизменно набирали самые низкие баллы по всем наборам данных.

Таблица 7 – Сравнение матрицы пуцтаницы всех моделей в разных наборах данных.

| Оптимизатор | Байес | SVM | Случайные лес | Нейронные сети |
|---|---|---|---|---|
| Тренировочная модель | 0.0 | 0.0 | 0.016 | 0.1 |
| Ling | 0.0 | 0.0 | 0.21 | 0.0 |
| Spam Assasin | 0.039 | 0.009 | 0.063 | 0.18 |
| Enron | 0.078 | 0.016 | 0.023 | 0.016 |

**Статическое встраивание с использованием нейронных сетей в качестве классификатора**

Предварительно обученная модель GloVe 6B была использована для преобразования слов в векторы, то есть для встраивания. Это было сделано в попытке улучшить встраивание во все наборы данных. Модель НС представляет собой рекурсивную LSTM-сеть с плотным выходным слоем, состоящим из двух нейронов с активацией SoftMax. В этой модели по-прежнему сохраняется вероятность отсева, равная 0,2. В экспериментах, описанных в этом разделе, соотношение обучения и тестирования составляло 75% к 25%, поскольку модели глубокого обучения требуют большего от соотношения обучения и тестирования для получения хорошего результата. Архитектура использованной модели показана на рисунке далее.

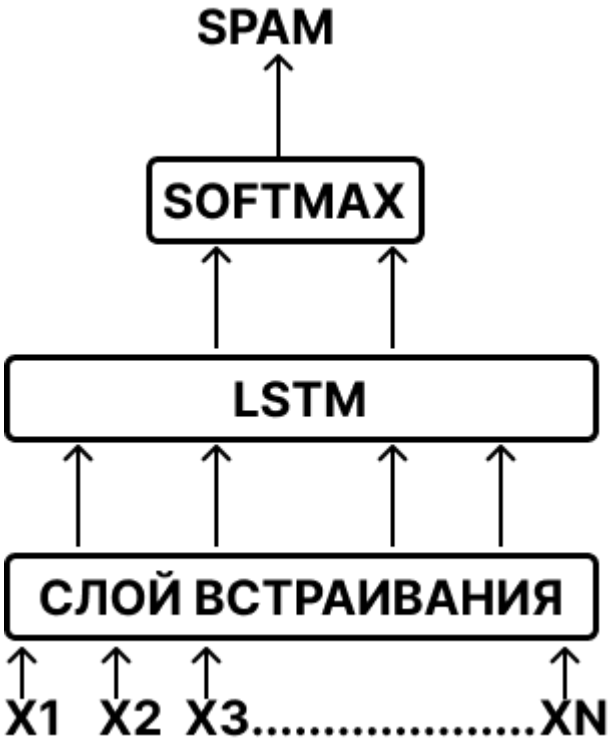



**Рисунок 19 –** Архитектура модели

Итак, этот первый уровень довольно прост. Это всего лишь перчаточный режим, предварительно обученный преобразованию слов в пространственный объект размером 100. Затем второй уровень представляет собой рекуррентную нейронную сеть с единицами измерения LSTM. Наконец, в качестве выходного слоя были использованы два нейрона, представляющие "спам" и "не спам", с функцией активации SoftMax. Функция оптимизатора(Adam) повышает точность нейронной сети за счет изменения атрибутов сети, таких как скорость обучения и весовые коэффициенты, что позволяет снизить общие потери и повысить точность. Критерием является следующее: результаты различных оптимизаторов для 25%-ного раздельного тестирования набора обучающих данных.

Таблица 8 – Результат LSTM с тестовым разделением на 0,25 для набора тренировочных данных

| Оптимизатор | Точность(%) | Полнота(%) | Прециозность(%) |
|---|---|---|---|
| Adadekta | 97.00 | 85.18 | 96.53 |
| Adam | 97.28 | 86.33 | 96.67 |
| sgd | 96.63 | 84.533 | 97.10 |

Исходя из вышеизложенного,Adam показал наилучшие результаты по точности, запоминанию и прецизионности.

Таблица 9 – Результат LSTM с тестовым разделением на 0,25 для набора Enron

| Оптимизатор | Точность(%) | Полнота(%) | Прециозность(%) |
|---|---|---|---|
| Adam | 97.25 | 94.74 | 94.71 |


**Рисунок 20 –** Результат алгоритма по показателям путаницы в наборе данных Enron

Таблица 10 – Результат LSTM с тестовым разделением на 0,25 для набора Linq.

| Оптимизатор | Точность(%) | Полнота(%) | Прециозность(%) |
|---|---|---|---|
| Adam | 98.95 | 96.23 | 97.45 |


**Рисунок 21 –** Результат алгоритма по показателям путаницы в наборе данных Linq

Таблица 11 – Результат LSTM с тестовым разделением на 0,25 для набора SpamAssasin.

| Оптимизатор | Точность(%) | Полнота(%) | Прециозность(%) |
|---|---|---|---|
| Adam | 97.62 | 96.31 | 96.52 |


**Рисунок 22 –** Результат алгоритма по показателям путаницы в наборе данных Spam assassin

В таблице 10 и на рисунке 22 показан результат применения той же архитектуры модели к Набор данных SpamAssassin получил точность обучения 97,62%, коэффициент полезного действия 0,017 при 16 ложноположительных результатах и 17 ложноотрицательных результатах.

В таблице 11 и на рисунке 23 приведены результаты для набора данных Enron с 21 ложноположительным результатом, 26 ложноотрицательными результатами с точностью тестирования 97,25% и коэффициентом полезного действия 0,017.

**Совершенствование алгоритма**

Теперь используем BERT, модель глубокого обучения, для динамического внедрения. Мы разметили каждый набор данных с помощью

токенизатора BERT. В архитектуру BERT был добавлен слой отсева с функцией активации RELU и слой классификатора из одного плотного (полностью связанного) нейрона с функцией активации SoftMax. Модель была скомпилирована с использованием ADAM в качестве оптимизатора и двоичной перекрестной энтропии в качестве функции потерь.

Модель была обучена с использованием 75% данных, производительность была оценена путем тестирования с использованием оставшихся 25%, поскольку для модели глубокого обучения требуется больше данных для обучения, чем для машинного обучения. На рисунке показана полученная матрица путаницы для тренировочного набора данных.

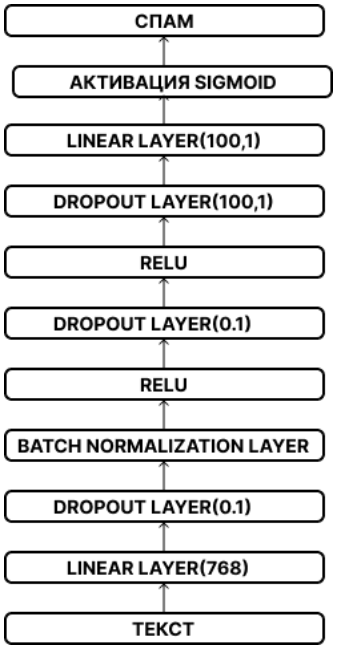



**Рисунок 23 –** Архитектура алгоритма


| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.99 | 0.99 | 1206 |
| 1 | 0.92 | 0.94 | 0.93 | 187 |
| accuracy | | | 0.98 | 1393 |
| macro avg | 0.96 | 0.96 | 0.96 | 1393 |
| weighted avg | 0.98 | 0.98 | 0.98 | 1393 |

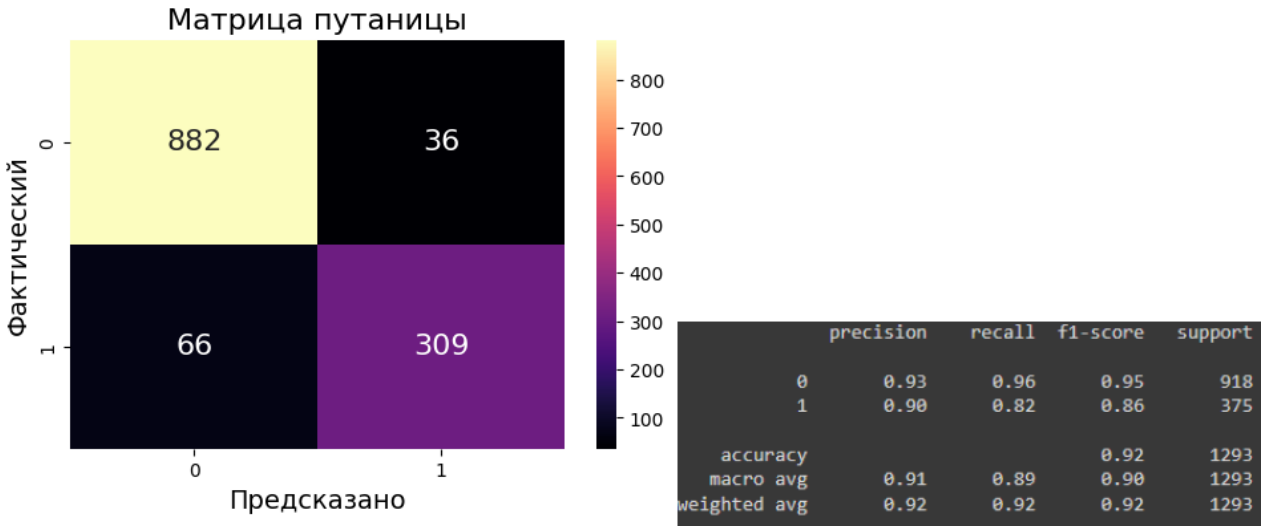
**Рисунок 24 –** Матрица путаницы BERT для тренировочного набора данных.


| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 1.00 | 0.98 | 604 |
| 1 | 0.99 | 0.78 | 0.87 | 120 |
| accuracy | | | 0.96 | 724 |
| macro avg | 0.97 | 0.89 | 0.93 | 724 |
| weighted avg | 0.96 | 0.96 | 0.96 | 724 |

**Рисунок 25 –** Матрица путаницы BERT для набора Linq


| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.84 | 0.97 | 0.90 | 560 |
| 1 | 0.91 | 0.62 | 0.74 | 280 |
| accuracy | | | 0.85 | 840 |
| macro avg | 0.87 | 0.80 | 0.82 | 840 |
| weighted avg | 0.86 | 0.85 | 0.85 | 840 |

**Рисунок 26 –** Матрица путаницы BERT для набора Spam Assassin




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.96 | 0.95 | 918 |
| 1 | 0.90 | 0.82 | 0.86 | 375 |
| accuracy | | | 0.92 | 1293 |
| macro avg | 0.91 | 0.89 | 0.90 | 1293 |
| weighted avg | 0.92 | 0.92 | 0.92 | 1293 |

**Рисунок 27 –** Матрица путаницы BERT для набора Enron.

Таблица 12 – Сравнение матрицы путаницы всех моделей встраивания и комбинирования классификаций.

| Оптимизатор | Без классификатора | С классификатором + Glove | Классификатор + HC + BERT |
|---|---|---|---|
| Тренировочный набор данных | 0.000 | 0.004 | 0.006 |
| Linq dataset | 0.000 | 0.005 | 0.002 |
| spam Assasin | 0.001 | 0.017 | 0.029 |
| Enron dataset | 0.016 | 0.017 | 0.039 |

В ходе экспериментов по классификации были рассмотрены статическое внедрение с помощью классификатора машинного обучения, статическое внедрение с помощью классификатора глубокого обучения и динамическое внедрение спомощью классификатора глубокого обучения. Первый из них дал наилучшие результаты. LTMS неизменно давал лучшие результаты

классификации с наименьшим коэффициентом полезного действия (см. таблицу 11).

## Заключение

В этой работе рассмотрены различные способы защиты от спама и атак по электронной почте с использованием машинного обучения и нейронных сетей: как статические методы встраивания, Word2Vec, GloVe, так и динамический BERT.

Высокоуровневые представления слов, которые не отражают истинного значения контекста, в котором используются слова, могут быть получены с помощью статических методов, таких как Word2Vec и GloVe. Напротив, BERT — с его двунаправленной архитектурой, помогающей улавливать контекстуальный смысл, — предлагает гораздо лучшее лингвистическое и семантическое векторное представление.

Было установлено, что среди методов машинного обучения SVM показал наилучшие результаты в точной классификации спама. Глубокие нейронные сети продемонстрировали меньшую эффективность по сравнению с классическими методами машинного обучения.

Была разработана гибридная модель, которая объединит преимущества машинного обучения с методами глубокого обучения для повышения точности классификации. Это позволяет повысить как точность классификации, так и обобщение скрытых представлений в данных, независимо полученных с помощью гибридной модели. Модель была протестирована на различных наборах данных: SMS-спам, Ling-спам, Spam Assassin и Enron.

Вот основные выводы этой работы:

- Статические методы встраивания слов, такие как Word2Vec и GloVe, ограничены в своей способности учитывать контекст слов.

- Таким образом, BERT позволил улучшить лингвистическое и семантическое представление слов, поскольку этот метод может охватывать двунаправленную модель.

- SVM продемонстрировал превосходную производительность по сравнению с различными подходами машинного обучения в плане точности категоризации спама.

- Разработанная гибридная модель, использующая методы машинного и глубокого обучения, отличается высокой эффективностью.

- Модель была успешно применена к многочисленным наборам данных и подтвердила высокую гибкость и практическую ценность.

В работе также рассмотрены индивидуальные преимущества каждого метода распознавания, что позволяет сделать конкретный вывод о высоком уровне точности современных методов фильтрации спама.

## Список использованных источников


1. An Anti-Spam Detection Model for Emails of Multi-Natural Language. [Сайт] – URL: https://doi.org/10.35741/issn.0258-2724.54.3.6
2. Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges. [Сайт] – URL: https://doi.org/10.1155/2022/1862888
3. Email Spam Detection using Machine Learning Techniques. [Сайт] – URL: https://doi.org/10.17148/IARJSET.2021.8632
4. A Systematic Review on Deep-Learning-Based Phishing Email Detection. [Сайт] – URL: https://doi.org/10.3390/electronics12214545
5. Phishing email detection using deep learning algorithms. [Сайт] – URL: https://doi.org/10.53730/ijhs.v6nS3.7944
6. Improving spam email classification accuracy using ensemble techniques: a stacking approach. [Сайт] – URL: https://doi.org/10.1007/s10207-023-00756-1
7. Evaluating the Effectiveness of Machine Learning Methods for Spam Detection. [Сайт] – URL: https://doi.org/10.1016/j.procs.2021.06.056
8. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ АЛГОРИТМОВ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ ЗАЩИТЫ ОТ ФИШИНГОВЫХ АТАК. [Сайт] – URL: https://cyberleninka.ru/article/n/issledovanie-vozmozhnostey-algoritmov-glubokogo-obucheniya-dlya-zaschity-ot-fishingovyh-atak?ysclid=lwbwysq0wt446377668
9. МЕТОД ВЫЯВЛЕНИЯ АНОМАЛИЙ СЕТЕВОГО ТРАФИКА, ОСНОВАННЫЙ НА ЕГО САМОПОДОБНОЙ СТРУКТУРЕ. [Сайт] – URL: https://bit.mephi.ru/index.php/bit/article/view/1185/1138
10. Современные методы защиты от нежелательных почтовых рассылок. [Сайт] – URL: https://cyberleninka.ru/article/n/sovremennye-metody-zaschity-ot-nezhelatelnyh-pochtovyh-rassylok/viewer
11. Unsupervised feature learning for spam email filtering. [Сайт] – URL: https://sci-hub.ru/https://doi.org/10.1016/j.compeleceng.2019.01.004

12. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks / Barushka, A., Hajek, P. — Applied Intelligence. — 2018. — № 48. — С. 3538-3556. [Сайт] – URL: https://doi.org/10.1007/s10489-018-1161-y
13. An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks / Faris Hossam, M. A. Ala', A. H. Ali [и др.]. // Information Fusion. — 2019. — № 48. — С. 67-83. [Сайт] – URL: https://doi.org/10.1016/j.inffus.2018.08.002
14. Deep learning to filter SMS Spam / Kumar,Roy Pradeep, P. S. Jyoti, Banerjee Snehasish. / Future Generation Computer Systems. — 2020. — № 102. — С. 524–533. [Сайт] – URL: https://doi.org/10.1016/j.future.2019.09.001
15. Twitter spam detection: Survey of new approaches and comparative study / Wu Tingmin, Wen Sheng, Xiang Yang, Zhou Wanlei./ Computers & Security. — 2018. — № 76. — С. 265-284. [Сайт] – URL: https://doi.org/10.1016/j.cose.2017.11.013
16. METHOD OF SPAM DETECTION BASED ON ARTIFICIAL IMMUNE SYSTEMS / M. P. Malykhina, V. A. Chastikova, A. A. Biktimirov./ VESTNIK OF ASTRAKHAN STATE TECHNICAL UNIVERSITY. — 2018. — №3. [Сайт] – URL: https://doi.org/10.24143/2072-9502-2018-3-38-48
17. АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ РАСПОЗНАВАНИЯ СПАМА / В. В. Шелепов./ Journal Neurocomputers. — 2022. — № 5. — С. 5-18. [Сайт] – URL: https://doi.org/10.18127/j19998554-202205-01
18. Коротких Д.С. Способ и система для определения параметра ошибки прогноза спама: патент РФ 2778381 С2: заявл. 06.10.2020; опубл. 18.08.2022, Бюл. № 23. - Москва : Федеральная служба по интеллектуальной собственности, 2022.

19. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ФИЛЬТРАЦИИ СПАМА НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В СООБЩЕНИЯХ РАЗЛИЧНОЙ ПРИРОДЫ / И. А. ЧИЖОВА, Е. И. КУБЛИК, М. С. ЧИПЧАГОВ, А. И. ЛАБИНЦЕВ./ Journal Neurocomputers. — 2022. — № 5. — С. 5-18. [Сайт] – URL: https://doi.org/10.18127/j19998554-202205-01

20. ОБЗОР АКТУАЛЬНЫХ ПРОБЛЕМ ОСНОВНЫХ МЕТОДОВ ФИЛЬТРАЦИИ СПАМА И АНАЛИЗ ИХ ЭФФЕКТИВНОСТИ / В. А. ЧАСТИКОВА, К. В. КОЗАЧЁК. / Вестник АГУ. — 2021. — № 286. [Сайт] – URL: https://doi.org/10.53598/2410-3225-2021-3-286-98-106

21. ОБНАРУЖЕНИЕ СПАМА В СМС-СООБЩЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ WORD EMBEDDING И TERM FREQUENCY- INVERSE DOCUMENT FREQUENCY (TF-IDF) / М. М. АББАСИ, А. П. БЕЛЬТЮКОВ, Х. ЛАЛ [и др.]./ XXI век: итоги прошлого и проблемы настоящего плюс. — 2020. — № 2. — С. 143-148. [Сайт] – URL: https://doi.org/10.46548/21vek-2020-0950-0026

22. Graham, P. (2002). "A Plan for Spam." Retrieved from [Сайт] – URL: http://www.paulgraham.com/spam.html

23. Carrillo-Ramos, A., Feregrino-Uribe, C., & Villa-Ramírez, M. S. (2019). "Review of Techniques for Email Spam Filtering." IEEE Access, 7, 15322-15333. [Сайт] – URL: https://doi.org/10.1109/ACCESS.2019.2893865

24. Liao, Y., Li, J., & Fu, X. (2020). "Spam Filtering Based on Machine Learning: A Review." Journal of Physics: Conference Series, 1529(4), 042041. [Сайт] – URL: https://doi.org/10.1088/1742-6596/1529/4/042041

25. Sakkis, G., Androutsopoulos, I., Paliouras, G., & Karkaletsis, V. (2003). "A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists." Information Retrieval, 6(1), 49-73. [Сайт] – URL: https://10.1023/A:1022840414903

26. Cormack, G. V., & Lynam, T. R. (2007). "TREC: Experiment and Evaluation in Information Retrieval." MIT Press. ISBN: 9780262033582
27. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). "A Bayesian Approach to Filtering Junk E-Mail." AAAI Workshop on Learning for Text Categorization, 62-66.
28. Усовершенствованный метод фильтрации нежелательного трафика. [Сайт] – URL: https://cyberleninka.ru/article/n/usovershenstvovannyy-metod-filtratsii-nezhelatelnogo-trafika/viewer
29. Вероятностный метод идентификации спама. [Сайт] – URL: https://cyberleninka.ru/article/n/veroyatnostnyy-metod-identifikatsii-spama/viewer
30. A weighted feature enhanced Hidden Markov Model for spam SMS filtering. [Сайт] – URL: https://doi.org/10.1016/j.neucom.2021.02.075
31. Development of a Machine Learning Model for Image-based Email Spam Detection. [Сайт] – URL: https://doi.org/10.46792/fuoyejet.v6i4.718
32. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. [Сайт] – URL: https://www.researchgate.net/publication/340475800_A_Survey_of_Convolutional_Neural_Networks_Analysis_Applications_and_Prospects (дата обращения )
33. Brain-inspired learning in artificial neural networks: a review. [Сайт] – URL: https://www.researchgate.net/publication/370937701_Brain-inspired_learning_in_artificial_neural_networks_a_review
34. EMAIL SPAM DETECTION USING HIERARCHICAL ATTENTION HYBRID DEEP LEARNING METHOD. [Сайт] – URL: https://arxiv.org/pdf/2204.07390
35. Идентификация текстового спама методом генетических карт. [Сайт] – URL: https://cyberleninka.ru/article/n/identifikatsiya-tekstovogo-spama-metodom-geneticheskih-kart/viewer

36. Вероятностный метод идентификации спама. [Сайт] – URL: https://cyberleninka.ru/article/n/veroyatnostnyy-metod-identifikatsii-spama