Исследование возможностей алгоритмов глубокого обучения для защиты от фишинговых атак

С.П. Корнюхина, О.Р. Лапонина

Аннотация **– Фишинг является одной из наиболее распространенных угроз в интернете, и именно поэтому разработка эффективных методов защиты является крайне важной задачей. В данной статье рассмотрены работы, применяющие возможности алгоритмов машинного и глубокого обучения в целях защиты от фишинговых атак, а также разработаны критерии сравнения и проведен сравнительный анализ решений.**
Сравнение систем защиты от фишинговых атак выполнено по следующим критериям: тип анализируемых элементов (HTML, URL, CSS); способы предварительной обработки датасета (нормализация и отбор признаков); необходимый объем выборок; алгоритмы ML/DL, используемые для определения фишинговых атак; количество ошибок 1 и 2 рода, критерии качества модели.
В рассмотренных работах наиболее часто изучаются CNN (Convolutional Neural Network) и LSTM (Long Short-Term Memory), как отдельно, так и в комбинации друг с другом. Также часто исследуются алгоритмы SVM (Support Vector Machine) и DT (Decision Tree), которые применяются для задач классификации.

Ключевые слова **- фишинг, фишинговая атака, машинное обучение, глубокое обучение, нейронные сети, ML, DL, CNN, LSTM, SVM, DT, DNN.**

I. Введение

Веб-приложения собирают и обрабатывают огромное количество данных пользователей, которые могут содержать конфиденциальную или личную информацию, такую как имена, адреса электронной почты, пароли, данные на кредитных картах. В связи с растущей популярностью различных электронных услуг (например, медицинские услуги, услуги онлайн-банков) количество собираемой частной информации постоянно растет. Злоумышленники заинтересованы в доступе к этим данным.

Развитие интернета сделало жизнь проще во многих отношениях. Примерно 62,5% мирового населения используют интернет, за последние 10 лет число пользователей выросло более чем в два раза, а в начале 2022 года численность интернет-аудитории достигла 4,95 млрд пользователей. Аудитория социальных сетей тоже выросла более чем на 10% и насчитывает 4,62 млрд человек, что составляет 58,4% от общей численности населения мира.

Электронная почта - самая большая уязвимость любой организации и является точкой входа для 91% кибератак. Одно вредоносное электронное письмо может нанести значительные финансовые потери и ущерб предприятию. Обычно такие преступления основаны на принципах социальной инженерии: преступники выдают себя за доверенное лицо с целью получения доступа к личным данным. Понимание принципов этих постоянно развивающихся атак и определение используемых тактик является ключом к тому, чтобы оставаться на шаг впереди киберпреступников.

Одной из серьезных угроз в приложениях электронной почты и социальных сетях являются фишинговые атаки. Фишинг — это метод кибератаки, при котором мошенники связываются с жертвой по электронной почте или в социальных сетях, зачастую выдавая себя за реально существующую организацию (например, банк, университет и т.д.), чтобы побудить людей предоставить конфиденциальные данные (логины, пароли, данные банковских и кредитных карт и другие личные данные). Жертву обманом побуждают перейти по вредоносной ссылке или ввести личную информацию, что может привести к установке вредоносного ПО или раскрытию конфиденциальной информации. Фишинговые атаки на крупные компании являются высокотехнологичными и комплексными атаками и часто используются для проникновения в корпоративные сети в рамках более крупной атаки.

Фишинг является серьезной проблемой и в социальных сетях. Несмотря на значительное внимание, которое уделялось фишингу на протяжении многих лет, окончательного решения так и не было найдено. Так как фишинг в значительной степени полагается на методы социальной инженерии (методы использования психологических механизмов для манипуляции), для защиты от него пользователи должны быть предельно осторожными в интернете и просвещенными в сфере интернет-мошенничества. Однако, как показали исследователи в [2], лица с высоким уровнем образования наиболее подвержены фишинговым атакам. Кроме того, они же являются наиболее вероятными целями такого мошенничества (кражи личных данных и данных кредитных карт), так как с большей вероятностью являются обладателями денежных средств на счетах. Возможное объяснение таких высоких показателей среди этой группы заключается в том, что более образованные пользователи увереннее в своей способности распознавать угрозу безопасности и,

Статья получена 24 апреля 2023.
С.П. Корнюхина - МГУ имени М.В. Ломоносова (email: kornyukhina.sofya@gmail.com)
О.Р. Лапонина - МГУ имени М.В. Ломоносова (email: laponina@oit.cmc.msu.ru)

следовательно, невнимательны и более восприимчивы к новым формам фишинговых атак.

Существуют различные способы противодействия фишингу: как простые (черные списки), так и сложные, основанные на алгоритмах машинного обучения. Также от фишинга могут уберечь такие технологии, как многофакторная аутентификация и антивирусное программное обеспечение.

II. Цели работы

Целью данной работы является определение типов фишинговых атак, анализ принципов их выполнения, разработка критериев для сравнения методов обнаружения такого вида атак, в том числе алгоритмов глубокого обучения (deep learning), особенностей наиболее часто используемых наборов данных для обучения алгоритмов обнаружения.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать возможные технологии выполнения фишинговых атак
2. Проанализировать возможные технологии защиты от фишинговых атак
3. Проанализировать существующие наборы данных в открытом доступе для обнаружения фишинговых атак
4. Разработать критерии сравнения алгоритмов глубокого обучения для обнаружения фишинговых атак
5. Сравнить наиболее часто используемые алгоритмы глубокого обучения

III. Анализ проблематики фишинговых атак

A. Классификация фишинговых атак

Как уже упоминалось ранее, фишинг — это сочетание методов социальной инженерии и технических приемов, предназначенных для того, чтобы убедить жертву предоставить персональную информацию, как правило, для получения финансовой выгоды злоумышленником.

Существует несколько технологий, которые могут быть использованы для осуществления фишинговых атак:

1. Подделка электронной почты (маскировка): злоумышленник может отправить электронное письмо, представляясь знакомым отправителем, который просит предоставить конфиденциальную информацию.
2. Фишинговые сайты: злоумышленник может создать веб-сайт, который выглядит так же, как официальный веб-сайт компании или сервиса, чтобы пользователи предоставляли свои личные данные.
3. Социальные сети: злоумышленник может использовать социальные сети для получения доступа к личным данным пользователей и использовать их для подделки электронных писем или создания фишинговых сайтов.
4. Вредоносные программы: злоумышленник может отправить вредоносное программное обеспечение по электронной почте, которое установит на компьютере пользователя, чтобы собирать конфиденциальную информацию.
5. Смс-фишинг: злоумышленники могут отправлять текстовые сообщения на мобильные устройства, в которых пользователей просят перейти по ссылке или ввести личную информацию.

Эти технологии можно поделить на два класса:

1. фишинг с помощью вредоносного ПО;
2. фишинг с помощью введения в заблуждение.

B. Жизненный цикл фишинговой атаки

Предполагаем, что атака осуществляется через электронную почту. В таком случае жизненный цикл фишинговой атаки можно разбить на несколько этапов:

1) Планирование: начальный этап, на нем злоумышленник определяет цель атаки, выбирает уязвимости и метод атаки, а также решает, от чьего имени будут рассылаться фишинговые сообщения, как получить адреса электронной почты клиентов этого бизнеса;
2) Подготовка: как только злоумышленники определили, каким бизнесом притворяться и кто будет жертвами атаки, создаются методы доставки сообщения и сбора данных. Чаще всего для рассылки используются адреса электронной почты, а для сбора - веб-страницы фишинговые сайты);
3) Атака: на этом шаге мошенники рассылают фишинговые сообщение или ссылки на фишинговые сайты жертвам. Целью является привлечение пользователя на фишинговый сайт и получение от них конфиденциальной информации.
4) Сбор данных: мошенники собирают информацию, которую жертвы вводят на фишинговых страницах, а жертвы атаки начинают осознавать, что они попали в ловушку фишинговой атаки, и предпринимают меры для защиты своих конфиденциальных данных. Обнаружение может произойти со стороны жертвы, а также со стороны компании, которая уведомляет своих клиентов о возможных атаках. Также жертвы могут предпринять действия для защиты своих данных: изменение паролей, блокирование кредитных карт, обращение в службу поддержки компании, уведомление правоохранительных органов.
5) Выполнение действий от имени жертвы и/или нанесение ущерба жертве: злоумышленники используют собранную ими информацию для совершения незаконных покупок или иного мошенничества.

Таким образом, мошенники рассылают письма и пытаются заставить жертв передать личную информацию, которая будет использоваться для кражи персональных данных. Электронное письмо направляет жертву на веб-сайт, где её просят обновить личную информацию (логины, пароли, номера банковских счетов, данные документов). Далее эти данные используются преступниками для совершения мошенничества.

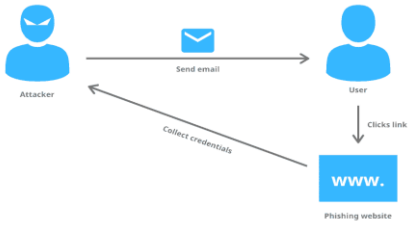

Рис. 1. Сценарий работы фишинговой атаки

C. Механизмы обнаружения фишинговой атаки

Существуют способы, с помощью которых можно избежать фишинговых атак:

1) Обучение: важно обучать пользователей основам защиты данных, чтобы они могли лучше распознать фишинговые атаки. Обучение может включать такие темы, как: распознавание поддельных электронных писем, идентификация фишинговых веб-сайтов и методов обмана пользователей. От этого зависит эффективность многих других технических методов защиты от фишинга.
2) Антивирусное программное обеспечение: этот тип программного обеспечения помогает защитить компьютер от вредоносных программ, включая те, которые могут использоваться для получения доступа к вашей личной информации. Антивирусные программы могут обнаруживать и блокировать вирусы, трояны, шпионское и рекламное ПО, а также другие типы вредоносных программ. Они также могут обеспечить защиту от атак через уязвимости в операционной системе и других программах, блокируя доступ к скрытым ресурсам компьютера или предотвращая выполнение нежелательных действий.

Следующие механизмы основаны на различных эвристиках:

3) Фильтры почты: Функции фильтрации почты позволяют определять, какие сообщения наиболее вероятно являются фишинговыми. Это могут быть сообщения, содержащие подозрительные вложения или ссылки. Также можно использовать облачные сервисы, такие как Google G Suite или Microsoft Office 365, которые предоставляют функции фильтрации почты, чтобы избежать получения спама и фишинговых сообщений.
4) Блокировка веб-сайтов: существует ряд методов, которые можно использовать для блокировки веб-сайтов, которые являются фишинговыми. Некоторые браузеры, такие как Google Chrome или Mozilla Firefox, могут использовать списки заблокированных сайтов для предотвращения доступа к веб-сайтам, известным как фишинговые. Другой метод - использование DNS-фильтров. Они могут блокировать доступ к веб-сайтам, даже если их IP-адреса изменятся.

Самые известные способы защиты от фишинга - черные списки, белые списки и различные эвристики. Распространённые виды эвристик приведены на рис. 2.

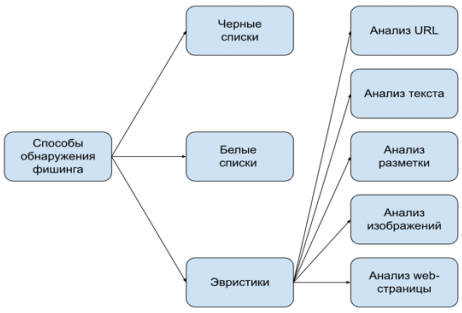

Рис. 2. Механизмы обнаружения фишинговых атак

Черные списки обеспечивают малое количество ошибок первого рода. Подозрительный URL-адрес сопоставляется со списком известных фишинговых сайтов. При совпадении переход на сайт блокируется, и система выдаёт предупреждение. Многие фишинговые страницы недолговечны, большая часть ущерба наносится в течение определенного малого промежутка времени, поэтому эффективность такого механизма зависит от частоты обновления списка, что может быть достаточно сложно. Кроме того, такие методы, как запутывание URL-адресов и маршрутизация через альтернативное доменное имя, могут сделать этот метод неэффективным.

Механизм белых списков подразумевает ведение списка разрешенных сайтов и используется намного реже, так как сильно ограничивает возможности пользователей в сети.

Эвристики так же основаны на некоторых применяемых правилах, а большинство из них субъективны и дают большое количество ложных срабатываний.

В последнее время для определения фишинга используются методы машинного и глубокого обучения, более эффективные, чем предыдущие методы. Этот метод позволяет обнаружить атаку на ранней стадии с использованием алгоритмов машинного обучения, которые способны определить, является ли страница фишинговой на основе ранее проанализированных характеристик других страниц: по совокупности различных характеристик определяется уровень доверия. Это решение не ограничивается анализом URL-адресов, также можно анализировать содержимое электронных писем, используемых для контакта с жертвой, и веб-страницы. Задача определения уровня доверия к интернет-ресурсу сложно поддается формализации и алгоритмически сложно реализуется. Несмотря на это, некоторые алгоритмы могут применяться в комплексе различных методов. Основная проблема заключается в том, что методы требуют знаний алгоритмов машинного обучения и значительного периода времени, чтобы выбрать характеристики, определяющие, является ли страница фишинговой или нет.

Глубокое обучение (deep learning) — это тип машинного обучения, который использует искусственные нейронные сети с несколькими скрытыми слоями для автоматического извлечения признаков из большого объема данных. Нейронные сети могут обрабатывать различные типы данных, такие как изображения, звук и тексты, и показывают высокую точность в решении сложных задач, таких как распознавание речи, обработка естественного языка и компьютерное зрение.

Применение глубокого обучения в защите от фишинга возможно в нескольких сценариях.

Во-первых, глубокое обучение может использоваться для определения фишинговых писем. Нейронная сеть может быть обучена распознавать определенные характеристики фишинговых писем, такие как неожиданный отправитель, ссылки на нежелательные сайты и запросы на предоставление персональной информации. С помощью такого обучения можно обнаружить мошеннические письма и блокировать их доставку.

Во-вторых, глубокое обучение может использоваться для отслеживания активности и взаимодействия пользователей с фишинговыми сайтами. Например, нейронная сеть может быть обучена распознавать определенные характеристики фишинговых сайтов, такие как схожесть дизайна с оригинальными сайтами и наличие определенных элементов, таких как формы для ввода паролей и других личных данных. Если такой сайт обнаружен, то можно предпринять действия по его блокировке и уведомлению пользователей об опасности.

Таким образом, машинное и глубокое обучение может быть эффективным инструментом для защиты от фишинга, позволяя автоматически обнаруживать и блокировать мошеннические письма и сайты.

Существует множество работ, посвященных применению различных алгоритмов машинного и глубокого обучения в контексте задачи обнаружения фишинга.

Так в работе [3] обсуждается задача классификации электронных писем как фишинговых или безопасных с помощью алгоритмов машинного обучения и глубокого обучения. Набор данных был предварительно обработан и преобразован с использованием регулярных выражений (RE) и NLP. Использовались алгоритмы обучения с учителем и DL, которые требуют набора компоновок для сортировки тестового набора. Для обнаружения фишинговых атак используются алгоритмы SVM, NB и LSTM.

В [4] предложено отслеживать нормальное или ненормальное поведение программного обеспечения. Основной идеей работы было предложение многоступенчатой системы обнаружения вредоносных программ, использующей комбинацию алгоритмов машинного и глубокого обучения для повышения точности классификации. Если по какой-то алгоритмами машинного обучения программное обеспечение оценивалось как подозрительное, то оно проходило следующий этап оценки с использованием глубокого обучения.

В [5] исследователями предложено решение с использованием нейронной сети eXpose, в которой в качестве входных данных принимаются необработанные короткие строки символов, извлекаются признаки и выполняется классификация, используя сверточные нейронные сети на уровне символов.

Авторы работы [6] предложили модель нейронной сети для классификации URL-адресов на безопасные и фишинговые. Топология состоит из трех слоев линейных сетей.

В [7] проанализирована производительность логистической регрессии с использованием биграмм, моделей CNN и CNN-LSTM для обнаружения фишинговых URL-адресов. Сделан вывод, что методы глубокого обучения, такие как CNN и LSTM, предпочтительнее методов машинного обучения, поскольку они сами могут получить оптимальное представление функций, взяв необработанные URL-адреса в качестве входных данных.

В статье [8] были изучены 16 систем классификации, основанных на семантических характеристиках URL. Также были собраны и проанализированы десять характеристик, которые отличают безопасные веб-сайты от фишинговых веб-сайтов с использованием семантических признаков. По результатам сравнения GradientBoostingClassifier и RandomForestClassifier показали наибольшую точность. Исследователи отметили, что одним из возможных ограничений является задача отбора признаков.

В [9] также рассматривают обнаружение вредоносных URL-адресов как проблему двоичной классификации и изучают производительность известных классификаторов (наивного Байеса, метода опорных векторов, многослойного персептрона, деревьев решений, случайного леса и k-ближайших соседей).

В [10] для эффективного обнаружения фишинговых атак была разработана новая система обнаружения фишинговых веб-сайтов с использованием рекуррентных нейронных сетей LSTM.

Авторы [11] сосредоточили внимание на методах извлечения семантических признаков с помощью word2vec для улучшения описания особенностей фишинговых сайтов, а затем объединили эти признаки с другими статистическими характеристиками для создания более надежной модели обнаружения фишинга. Результаты экспериментов с фактическими наборами данных показали, что комбинации признаков улучшают эффективность обнаружения фишинга.

В исследовании [12] были проверены веб-сайты и проведено сравнение алгоритмов нейронных сетей, машины опорных векторов, дерева решений и многоуровневых автоэнкодеров в качестве методов классификации.

Авторы [13] сравнивают метод случайного леса и рекуррентные нейронные сети в рамках задачи классификации URL-адресов. Нейронные сети показали лучшую эффективность, поэтому авторы пришли к выводу о предпочтительности выбора именно этой группы методов для данной задачи.

Подход [14] основан на нейронных сетях на уровне символов. В частности, строки URL и DNS преобразуются в векторную форму с использованием методов обработки естественного языка. Далее используется CNN для извлечения свойств и обучения модели классификации.

В работах [15], [16] особенность подхода заключается в том, что модели работают непосредственно с трафиком в DNS. В частности, в [15] представлена разработанная система DNS-фильтрации и система извлечения данных из сети (D-FENS). Для задачи классификации в реальном времени были использованы CNN и LSTM. Особенность метода заключается в том, что система работает непосредственно с трафиком в DNS.

В [17] исследованы недостатки современных систем обнаружения фишинговых атак, проведено сравнение

алгоритмов машинного обучения при определении URL-адреса фишинг-сайта. Результаты также показали, что использование RFE (Recursive Feature Elimination) для исключения свойств не только повышает производительность во время выполнения, но и повышает точность модели за счет устранения избыточных функций.

В [18] исследователи сформировали характерный профиль мошенников. Для этого на основе полученных исходных данных сравниваются внутренние данные людей (возраст, пол, род занятий и т. д.) с данными посещенных сайтов (бизнес, искусство, социальные сети и т. д.). А после проведения анализа применили DNN для получения профилей.

Авторы [19] предложили свою модификацию DNN с помощью Bat Algorithm (алгоритма летучих мышей), чем улучшили эффективность модели.

IV. Сравнение моделей

Проведем сравнение вышеописанных систем защиты от фишинговых атак по следующим критериям (см. сравнительную таблицу в разделе V):

1) Архитектура системы
   a) Тип анализируемых элементов (HTML, URL, CSS, …)
   b) Свойства датасетов, необходимый объем выборок
   c) Способы предварительной обработки датасета
2) Свойства ML и DL алгоритмов
   a) Алгоритмы ML/DL, используемые для определения фишинговых атак
   b) Количество ошибок 1 и 2 рода, критерии качества модели

Разберем подробнее каждый из критериев.

1) Тип анализируемых элементов можно разделить на несколько групп:
   a) основанные на анализе URL-адресов (URL, DNS);
   b) основанные на анализе электронной почты (.eml);
   c) основанные на анализе контента веб-сайтов (HTML);
   d) основанные на поведении ПО (логи ПО);
   e) основанные на профилях пользователей.

Большое внимание исследователи уделяют анализу URL-адресов, так как именно они являются основным инструментом для проведения фишинговых атак. Однако для эффективного обнаружения фишинга необходимо использовать комплексный подход, включающий в себя анализ данных разных типов и источников. Например, учитывать также и HTML-контент, расположенный по данному URL-адресу. Исследования в этой области позволяют разработать более точные и эффективные методы обнаружения фишинга.

2) Свойства датасетов, необходимый объем выборок

Для машинного обучения, который использует более простые модели, часто достаточно использования небольших датасетов, содержащих от сотен до нескольких тысяч образцов. Для глубокого обучения часто требуется использование гораздо больших датасетов. Это связано с тем, что такие модели имеют большое число параметров и требуют большого объема данных для оптимизации этих параметров. Объем датасета для глубокого обучения может составлять несколько миллионов или даже миллиардов образцов. В обоих случаях деление датасета на тренировочную и тестовую выборки необходимо для оценки точности модели и избежания ее переобучения на тренировочных данных. При правильном разделении данных модель будет более точной и устойчивой к внешним воздействиям.

3) Способы предварительной обработки датасета

Качество предиктивной модели напрямую зависит от собранных данных. Прежде чем обучать модель на данных, их нужно обработать.

Первый этап обработки – очистка данных. Необходимо исследовать их на наличие пропусков и выбросов. Нет универсального решения проблемы отсутствующих данных, существует несколько способов смягчения данной проблемы. Для каждого конкретного набора данных нужно подбирать наиболее подходящие методы или их комбинации. Одним из основных методов является отбрасывание записей/признаков, содержащих пропуски. Данный метод можно использовать только в том случае, если недостающие данные являются неинформативными. В противном случае, если модель пропустит такие данные, её работа может ухудшиться.

Пропуски числовых признаков можно заполнить стандартными значениями, полученными из остальных записей, например, константой, медианой, средним значением и др. Для категориальных признаков допустимо использование наиболее часто встречающегося значения. Иногда используется некоторое базовое значение, которое не может принимать данный признак. Таким образом сохранятся данные о пропущенных значениях, что в некоторых задачах может оказаться ценной информацией.

В данных могут присутствовать значения, являющиеся выбросами. Эти значения сильно влияют на модель. Для проверки признаков на наличие таких значений полезно визуализировать данные. Также можно применить и другие методы: кластеризацию и т.п.

Выбросами принято считать значения, не попадающие в интервал:

$$[Q_1 - 1{,}5 * IQR; Q_3 + 1{,}5 * IQR\,], \qquad (1)$$

где $IQR = Q_3 - Q_1$ — интерквантильный размах, Q_1 — первый квартиль, Q_3 — третий квартиль.

Самый простой способ справиться с выбросами — изменить все значения выше верхнего порога и ниже нижнего порога этими пороговыми значениями.

Следующие два этапа обработки данных — нормализация и отбор признаков. Последовательность их применения может варьироваться в зависимости от данных и методов.

a) Нормализация

Нормализация — это приведение всех значений признака к новому диапазону. Значения различных числовых признаков могут отличаться на несколько порядков, что сильно влияет на работу модели. После нормализации значения признаков будут находиться в узком (и, зачастую, едином для всех признаков) диапазоне, например, от 0 до 1 или от -1 до +1.

Самым распространённым способом нормализации является нормализация по формуле:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} * (I_{max} - I_{min}) + I_{min}, \quad (2)$$

где x_{max}, x_{min} — максимальное и минимальное значения признака, где I_{max}, I_{min} — максимальное и минимальное значения интервала, к которому мы преобразуем значения признака.

Если у данных нормальное распределение, то часто используется z-нормализация. Новые значения вычисляются следующим образом:

$$x_{new} = \frac{x - M[X]}{\sigma[X]}, \quad (3)$$

где $M[X]$ — математическое ожидание признака, а $\sigma[X]$ — среднеквадратическое отклонение признака.

b) Отбор признаков

Отбор признаков — это процедура отбрасывания переменных из выборки. Некоторые модели чувствительны к величине входного вектора, большое число признаков замедляет работу модели и в некоторых случаях может привести к переобучению (например, в нейросетях).

4) Алгоритмы ML/DL, используемые для определения фишинговых атак

Применение алгоритмов машинного и глубокого обучения позволяет улучшить качество определения фишинговых атак и увеличить эффективность мер по их предотвращению. В рассмотренных нами работах наиболее часто изучаются CNN (Convolutional Neural Network) и LSTM (Long Short-Term Memory), как отдельно, так и в комбинации друг с другом. Также часто исследуются алгоритмы SVM (Support Vector Machine) и DT (Decision Tree), которые применяются для задач классификации.

- Support Vector Machine (SVM)

Метод опорных векторов – это один из линейных классификаторов. Его задачей является поиск такой гиперплоскости в пространстве признаков, что она разделяет объекты с разными метками и расстояние от гиперплоскости до ближайшего объекта обучающей выборки максимально.

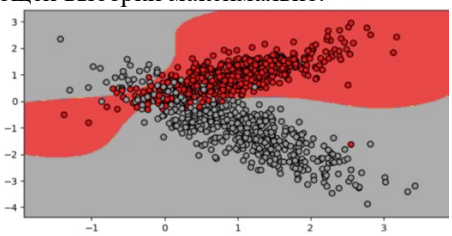

Рис. 3. Метод опорных векторов

- Decision Tree (DT)

Решающие деревья – алгоритм машинного обучения, использующийся для анализа данных и предсказательной аналитики. Представляет собой древовидную структуру, т.е. содержит “листья” и “ветки” (Рис. 4). Получая на вход данные, мы проходимся по ребрам деревьев. Каждой вершине v дерева T ставится в соответствие предикат, касающийся значения одного из признаков. В зависимости от ответа осуществляется переход к вершине следующего уровня. Листьям соответствуют метки, указывающие на отнесение распознаваемого объекта к одному из классов.

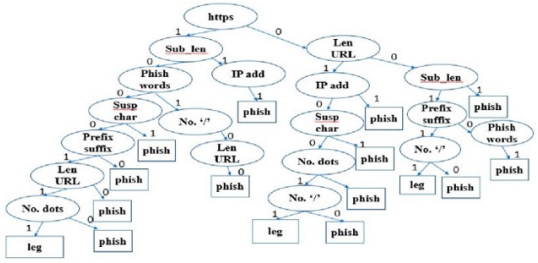

Рис. 4. Решающее дерево

- Convolutional Neural Networks (CNN)

Основной элемент CNN – это сверточный слой (Convolutional Layer), который используется для обработки входных данных и извлечения признаков. Свертка – это математическая операция, которая позволяет получить новое изображение путем перемещения ядра свертки (весов) по исходному изображению. Веса ядра свертки устанавливаются в процессе обучения.

- Long Short-Term Memory (LSTM)

LSTM – это рекуррентная нейронная сеть (RNN). LSTM используется для обработки естественных языков, анализа временных рядов и других задач, где необходимо учитывать контекст.

Основным элементом LSTM является блок памяти. Блок памяти состоит из трех гейтов – входного (input gate), забывания (forget gate) и выходного (output gate). Каждый гейт использует сигмоидную функцию, различающую значимые и незначимые элементы последовательности данных.

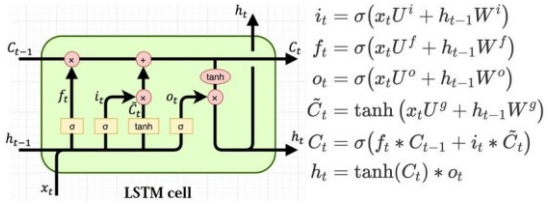

Рис. 5. Блок LSTM

5) Количество ошибок 1 и 2 рода, критерии качества модели

Одним из критериев является время обучения модели (Training Time). Однако эта метрика не несет информации о том, дает ли модель правильные прогнозы.

Модель тестируется на тестовом наборе данных, чтобы проверить достоверность прогнозов. При обучении с учителем существует ответ, с которым можно сравнить результат работы модели.

В задаче классификации пространство ответов имеет фиксированный набор значений. В задаче обнаружения фишинга таких значений два (1 – фишинг и 0 – безопасность). В таком случае задача называется задачей бинарной классификации, для которой существует четыре результата:

		Predicted Label	
		Positive	Negative
Actual Label	Positive	TRUE POSITIVE TP	FALSE NEGATIVE FN
	Negative	FALSE POSITIVE FP	TRUE NEGATIVE TN

Рис. 6. Матрица ошибок

Матрица ошибок представляет из себя таблицу, где по строкам отложены фактические значения класса, а по столбцам — прогнозируемые. По главной диагонали отложены правильные прогнозы True Positive (TP), либо правильные прогнозы нуля — True Negative (TN). По обратной диагонали лежат ошибки прогнозирования. FP (False Positive) — спрогнозирован класс 1, хотя фактический класс был 0. FN (False Negative) — когда спрогнозирован класс 0, хотя фактический класс был 1.
Accuracy (ACC) показывает долю верно определенных наблюдений (достоверность):

$$Accuracy = \frac{TP+TN}{TN+FP+FN+TP} \quad (4)$$

Precision (PPV) показывает точность прогноза:

$$Precision = \frac{TP}{FP+TP} \quad (5)$$

Specificity (специфичность) показывает насколько хорошо модель отделяет негативные примеры от положительных:

$$Specificity = \frac{TN}{FP+TN} \quad (6)$$

Sensitivity (чувствительность) измеряет, насколько хорошо модель находит положительные примеры из общего числа положительных примеров:

$$Sensitivity = \frac{TP}{TP+FN} \quad (7)$$

Error rate (ошибка) измеряет, насколько часто модель допускает ошибки:

$$Error\ rate = \frac{FP+FN}{TP+TN+FN+FP} \quad (8)$$

Recall (полнота) показывает, сколько наблюдений с фактическим классом 1 смогли найти с помощью модели:

$$Recall = \frac{TP}{FN+TP} \quad (9)$$

F-мера (F-measure, F-score, среднее гармоническое) позволяет совместить точность и полноту для оценки:

$$F = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (10)$$

TPR полностью совпадает с полнотой, и показывает долю верно предсказанных классов у объектов, относящихся к положительному классу.

$$TPR = \frac{TP}{FP+TP} \quad (11)$$

FPR — это доля неправильно предсказанных классов среди объектов отрицательного класса.

$$FPR = \frac{FP}{TN+FP} \quad (12)$$

ROC-кривая – это графическое представление, используемое для оценки качества бинарной классификации, которое позволяет оценить, насколько точно классификатор различает две категории объектов.

Основной параметр ROC-кривой – это False Positive Rate (FPR) и True Positive Rate (TPR). TPR представляет собой число модели оценок, которые положительны и правильны, FPR – это число модели оценок, которые ложно положительны.
ROC-кривая представляет собой график TPR по оси Y и FPR по оси X. Каждая точка в ROC-кривой соответствует выбору определенного порога классификации. Чем ближе ROC-кривая расположена к левому верхнему углу графика, тем лучше работает модель.

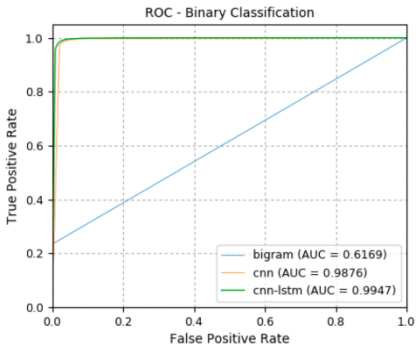

Рис. 7. ROC-кривая из исследования [7]

ROC-кривая описывает баланс между TPR и FPR, что позволяет оценить качество классификатора в зависимости от порога классификации. AUC (Area Under Curve) ROC-кривой отражает «качество» классификатора - чем выше AUC, тем «лучше» классификатор. AUC ROC-кривой находится в диапазоне от 0 до 1, где 1 представляет идеальную оценку.

V. РЕЗУЛЬТИРУЮЩЕЕ СРАВНЕНИЕ РАССМОТРЕННЫХ АЛГОРИТМОВ

Таблица 1. Сравнение рассмотренных алгоритмов

<table>
<tr><th rowspan="2">№</th><th colspan="3">Архитектура системы</th><th rowspan="2">Алгоритмы и критерии качества</th></tr>
<tr><th>Тип анализируемых элементов</th><th>Свойства, объем выборок</th><th>Обработка датасета</th></tr>
<tr><td>[3]</td><td>.eml</td><td>5000 электронных писем, 31% из которых спам.</td><td>Извлечение свойств из текста, нормализация, удаление выбросов.</td><td><table><tr><th>Алгоритм</th><th>Точность</th></tr><tr><td>SVM</td><td>99,62%</td></tr><tr><td>NB</td><td>97%</td></tr><tr><td>LSTM</td><td>98%</td></tr></table></td></tr>
<tr><td>[4]</td><td>Логи ПО</td><td>100 вредоносных программ и 400 безопасных. Проведены эксперименты на 20 виртуальных машинах Ubuntu 14.04 (32-разрядной версии) и сгенерировано 100 тыс. экземпляров для датасета.</td><td>Разбиение на n-граммы.</td><td><table><tr><th>Алгоритм</th><th>Точность</th><th>Достоверность</th></tr><tr><td>LSTM</td><td>99,14%</td><td>94,36%,</td></tr></table></td></tr>
<tr><td>[5]</td><td>URL</td><td>19 млн уникальных URL-адресов отобрано случайным образом в течение двух месяцев с сайта VirusTotal. Тренировочные данные - 45% от датасета (безопасные – 7211705, фишинговые - 1496198), тестовые - 55% (безопасные – 9718748, фишинговые - 641228).</td><td>–</td><td><table><tr><th>Алгоритм</th><th colspan="4">AUC = 0,993</th></tr><tr><td rowspan="2">CNN</td><td>FPR</td><td>10^{-4}</td><td>10^{-3}</td><td>10^{-2}</td></tr><tr><td>TPR</td><td>77%</td><td>84%</td><td>92%</td></tr></table></td></tr>
<tr><td>[6]</td><td>URL</td><td>Тренировочные данные: фишинговые сайты собраны с PhishTank, всего 26722 экземпляров + 68172 экземпляров, легальных сайтов 26722 экземпляров. Тестовые данные: фишинговые сайты собраны с PhishTank, всего 39776 экземпляров, легальных сайтов 39776 экземпляров.</td><td>–</td><td><table><tr><th>Алгоритм</th><th>Оптимизатор</th><th>Достоверность</th><th>Время обучения, с</th></tr><tr><td rowspan="3">CNN</td><td>Adam</td><td>94,18%</td><td>32</td></tr><tr><td>AdaDelta</td><td>93,54%</td><td>31</td></tr><tr><td>SGD</td><td>88,29%</td><td>31</td></tr><tr><td rowspan="3">eXpose</td><td>Adam</td><td>90,52%</td><td>119</td></tr><tr><td>AdaDelta</td><td>91,31%</td><td>119</td></tr><tr><td>SGD</td><td>77,99%</td><td>116</td></tr></table></td></tr>
<tr><td>[7]</td><td>URL</td><td>60 тыс. URL-адресов для обучения и 56 101 URL-адреса для тестирования. Строятся матрицы для обучения (60000*2307) и тестирования (30101*2307) (набор данных 1), обучения (60000*2307) и</td><td>Для уменьшения размерности используются векторные представления слов (эмбеддинги). Тестовые данные разбиваются на 2 датасета, модель оценивается по ним.</td><td><table><tr><th>Алгоритм</th><th>Достоверность</th><th>Точность</th><th>Полнота</th><th>F-мера</th><th>AUC</th></tr><tr><td colspan="6">Датасет 1</td></tr><tr><td>CNN</td><td>97,7%</td><td>97%</td><td>98,5%</td><td>97,8%</td><td>0,9876</td></tr><tr><td>CNN-LSTM</td><td>98,2%</td><td>97,8%</td><td>98,6%</td><td>98,2%</td><td>0,9947</td></tr><tr><td>Bigram</td><td>96,2%</td><td>95,9%</td><td>96,6%</td><td>96,3%</td><td>0,6169</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td>тестирования (26000*2307) (набор данных 2).</td><td></td><td>
<table>
<tr><th colspan="6">Датасет 2</th></tr>
<tr><td>CNN</td><td>98,7%</td><td>98%</td><td>98,9%</td><td>98,5%</td><td>0,9989</td></tr>
<tr><td>CNN-LSTM</td><td>98,9%</td><td>98,8%</td><td>98,6%</td><td>98,7%</td><td>0,9992</td></tr>
</table>
</td></tr>
<tr><td>[8]</td><td>URL, HTML</td><td>8 тыс. URL и HTML-файлов.</td><td>Использование классических признаков, усредненных эмбеддингов, взвешенных эмбеддингов, совмещения усредненных эмбеддингов и классических признаков.</td><td>Лучшие показатели достигнуты при совмещении классических признаков и усредненных эмбеддингов.
<table>
<tr><th>Алгоритм</th><th>Точность</th><th>Полнота</th><th>F-мера</th><th>FPR</th><th>Ошибка</th><th>ROC</th></tr>
<tr><td>AdaBoost</td><td>99,8%</td><td>99,8%</td><td>0,998</td><td>0,3%</td><td>0,15%</td><td>0,999</td></tr>
<tr><td>Bagging</td><td>99,6%</td><td>99,6%</td><td>0,996</td><td>0,7%</td><td>0,4%</td><td>0,999</td></tr>
<tr><td>RF</td><td>99,6%</td><td>99,6%</td><td>0,996</td><td>0,8%</td><td>0,42%</td><td>1</td></tr>
<tr><td>SMO</td><td>99,6%</td><td>99,6%</td><td>0,996</td><td>0,6%</td><td>0,44%</td><td>0,995</td></tr>
</table>
</td></tr>
<tr><td>[9]</td><td>URL</td><td>2,4 миллиона URL-адресов (экземпляров).</td><td>3,2 млн признаков, номинальные преобразованы в бинарные. Применены 3 различных метода выбора признаков, в которых отдельные признаки имеют самый высокий абсолютный коэффициент корреляции Пирсона с фактическим классом URL-адреса.</td><td>
<table>
<tr><th rowspan="2">Алгоритм</th><th colspan="4">Достоверность</th></tr>
<tr><th>Датасет A</th><th>Датасет B</th><th>Датасет C</th><th>AVG</th></tr>
<tr><td>RF</td><td>98,26%</td><td>96,91%</td><td>97,91%</td><td>97,69%</td></tr>
<tr><td>MLP</td><td>97,97%</td><td>96,57%</td><td>97,31%</td><td>97,28%</td></tr>
<tr><td>C4.5</td><td>97,33%</td><td>96,78%</td><td>96,33%</td><td>96,82%</td></tr>
<tr><td>kNN</td><td>97,54%</td><td>95,23%</td><td>95,98%</td><td>96,25%</td></tr>
<tr><td>SVM</td><td>97,11%</td><td>96,01%</td><td>95,17%</td><td>96,10%</td></tr>
<tr><td>C5.0</td><td>97,40%</td><td>96,72%</td><td>93,65%</td><td>95,92%</td></tr>
<tr><td>NB</td><td>95,98%</td><td>91,36%</td><td>94,25%</td><td>93,86%</td></tr>
<tr><td></td><td>97,37%</td><td>95,65%</td><td>95,80%</td><td></td></tr>
</table>
</td></tr>
<tr><td>[10]</td><td>URL</td><td>2000 законных веб-сайтов, собранных из каталога Yahoo, 2000 фишинговых веб-сайтов, собранных из PhishTank. Случайным образом выбрано 70% для обучения, 30% для теста.</td><td>Из URL извлекаются признаки: длина домена и поддоменов, префиксы и суффиксы, IP порта, подсчитываются вхождения специальных символов и энтропия URL.</td><td>
<table>
<tr><th>Алгоритм</th><th>Достоверность</th><th>Точность</th><th>Полнота</th><th>FNR</th></tr>
<tr><td>CNN</td><td>97,42%</td><td>96,48%</td><td>97,23%</td><td>5,91%</td></tr>
<tr><td>LSTM</td><td>99,14%</td><td>98,74%</td><td>98,91%</td><td>2,12%</td></tr>
</table>
</td></tr>
<tr><td>[11]</td><td>URL, HTML</td><td>200 фишинговых сайтов с веб-сайта PhishTank и 2000 легитимных веб-страниц.</td><td>TF-IDF</td><td>
<table>
<tr><th>Алгоритм</th><th>Точность</th><th>Полнота</th></tr>
<tr><td>SVM</td><td>95,2%</td><td>90,3%</td></tr>
<tr><td>DBN</td><td>96,5%</td><td>90,7%</td></tr>
</table>
</td></tr>
<tr><td>[12]</td><td>URL</td><td>1000 вредоносных и 1000 легитимных URL-адресов.</td><td>URL-адреса преобразуются в значения ASCII, чтобы информация, содержащаяся в наборе данных, была детально изучена, а классификация выполнялась на матрицах, содержащих значения ASCII.</td><td>
<table>
<tr><th>Алгоритм</th><th>Тренировочные данные</th><th>Достоверность</th></tr>
<tr><td rowspan="3">SVM</td><td>70%</td><td>56,65%</td></tr>
<tr><td>80%</td><td>57,425%</td></tr>
<tr><td>90%</td><td>60,05%</td></tr>
<tr><td rowspan="3">DT</td><td>70%</td><td>80,65%</td></tr>
<tr><td>80%</td><td>80,825%</td></tr>
<tr><td>90%</td><td>81,3%</td></tr>
<tr><td rowspan="3">NN</td><td>70%</td><td>78,5%</td></tr>
<tr><td>80%</td><td>76,75%</td></tr>
<tr><td>90%</td><td>79,5%</td></tr>
<tr><td rowspan="3">AE</td><td>70%</td><td>84%</td></tr>
<tr><td>80%</td><td>86,5%</td></tr>
<tr><td>90%</td><td>83,5%</td></tr>
</table>
</td></tr>
<tr><td>[13]</td><td>URL</td><td>150 тыс. легитимных URL-адресов и 240 тыс. вредоносных URL-адресов.</td><td>Датасет собран из 21 признака, основанных на длинах, числах, символах, уровне риска и т.д.</td><td>
<table>
<tr><th rowspan="2">Алгоритм</th><th colspan="4">Достоверность на различных размерах тренировочного датасета</th></tr>
<tr><th>600</th><th>3000</th><th>60000</th><th>240000</th></tr>
<tr><td>RF</td><td>87,8%</td><td>91,7%</td><td>94,5%</td><td>96,4%</td></tr>
<tr><td>GRU network</td><td>89,3%</td><td>95,6%</td><td>97,6%</td><td>98,5%</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>[14]</td><td>URL, DNS</td><td>7 млн URL-адресов. Из них 1 млн вредоносных, 6 млн легитимных. Набор данных для обучения и набор тестовых данных были рандомизированы в соответствии с соотношением 9:1.</td><td>Три модели признаков:
1) Длины, частота вхождения символов и т.д.
2) Эмбеддинги на уровне символов
3) Эмбеддинги на уровне слов.</td><td>
<table>
<tr><th>Алгоритм</th><th>Достоверность</th></tr>
<tr><td>Feature Based CNN</td><td>71,8%</td></tr>
<tr><td>Character level CNN</td><td>96%</td></tr>
<tr><td>Word level CNN</td><td>84,2%</td></tr>
</table>
</td></tr>
<tr><td>[15]</td><td>DNS-трафик</td><td>Вредоносные данные (51033 (7%)) собраны с PhishTank, OpenPhish, malwaredomains.com, Agten. Легитимные (676154 (93%))– с Alexa, Agten, DMOZ</td><td>Для уменьшения размерности используются эмбеддинги.</td><td>
<table>
<tr><th>Алгоритм</th><th>AUC</th></tr>
<tr><td>CNN-LSTM</td><td>0,95</td></tr>
</table>
</td></tr>
<tr><td>[16]</td><td>DNS-трафик</td><td>80 тыс. безопасных доменных имен, случайно выбранных из доменов Alexa. 50 тыс. из 80 тыс. доменных имен были отобраны для обучения,
а остальные — для тестирования. Также были собраны данные в экспериментах с реальным трафиком, состоящие из потока DNS-трафика в реальном времени, 10 миллиардов DNS-запросов в день, собираемых от нескольких интернет-провайдеров, распределенных по всему миру.</td><td></td><td>
<table>
<tr><th rowspan="2">Алгоритм</th><th colspan="2">Round 1</th><th colspan="2">Round 2</th><th colspan="2">Round 3</th></tr>
<tr><th>FPR</th><th>Точность</th><th>FPR</th><th>Точность</th><th>FPR</th><th>Точность</th></tr>
<tr><td>Word Graph</td><td>$2{,}67 * 10^{-4}$</td><td>0,999</td><td>$1{,}33 * 10^{-4}$</td><td>0,999</td><td>$3{,}33 * 10^{-5}$</td><td>0,999</td></tr>
<tr><td>CNN</td><td>0,018</td><td>0,981</td><td>0,015</td><td>0,982</td><td>0,014</td><td>0,983</td></tr>
<tr><td>RF</td><td>1,0</td><td>0,444</td><td>1,0</td><td>0,444</td><td>1,0</td><td>0,444</td></tr>
</table>
</td></tr>
<tr><td>[17]</td><td>URL, WHOIS</td><td>3526 веб-сайтов, включающих как легитимные (1407 экземпляров), так и фишинговые (2119 экземпляров). Фишинговые сайты собраны с PhishTank.</td><td>Извлечены признаки:
• энтропии URL-адресов
• на основе гиперссылок
• сторонние</td><td>
<table>
<tr><th></th><th>RF</th><th>J48</th><th>LR</th><th>BN</th></tr>
<tr><td>Чувствительность</td><td>99,44%</td><td>99,01%</td><td>95,97%</td><td>99,21%</td></tr>
<tr><td>Специфичность</td><td>99,1%</td><td>98,95%</td><td>94,13%</td><td>98,25%</td></tr>
<tr><td>Точность</td><td>99,42%</td><td>99,3</td><td>96%</td><td>98,83%</td></tr>
<tr><td>Достоверность</td><td>99,31%</td><td>98,98%</td><td>95,22%</td><td>98,82%</td></tr>
<tr><td>Ошибка</td><td>0,69%</td><td>1,02%</td><td>4,78%</td><td>1,18%</td></tr>
<tr><th></th><th>MLP</th><th>SMO</th><th>AdaBoost Ml</th><th>SVM</th></tr>
<tr><td>Чувствительность</td><td>95,81%</td><td>94,58%</td><td>98,07%</td><td>97,14%</td></tr>
<tr><td>Специфичность</td><td>94,07%</td><td>92,2%</td><td>95,87%</td><td>94,3%</td></tr>
<tr><td>Точность</td><td>96,12%</td><td>94,76%</td><td>97,25%</td><td>96,09%</td></tr>
<tr><td>Достоверность</td><td>95,12%</td><td>93,63%</td><td>97,18%</td><td>95,94%</td></tr>
<tr><td>Ошибка</td><td>4,88%</td><td>6,37%</td><td>2,82%</td><td>4,06%</td></tr>
</table>
</td></tr>
<tr><td>[18]</td><td>Логи пользователей</td><td>Данные состоят из логов 162 людей, из них 102 молодых и 60 пожилых, 49% мужчин, 51% женщин. Логи собраны за 4 дня. Были исключены участники, чьи данные были нерелевантными для исследования.</td><td>Признаки для датасета сформированы на основе профилей пользователей (пол, возраст, образование, страна и т.д.), а также поведении в интернете. Произведена очистка датасета от выбросов различными способами.</td><td>
<table>
<tr><th>Алгоритм</th><th>Достоверность</th></tr>
<tr><td>DNN</td><td>91%</td></tr>
<tr><td>RF</td><td>93%</td></tr>
<tr><td>SVM</td><td>90%</td></tr>
<tr><td>Perceptron</td><td>90%</td></tr>
<tr><td>LR</td><td>93%</td></tr>
<tr><td>NB</td><td>95%</td></tr>
</table>
</td></tr>
<tr><td>[19]</td><td>URL, HTML</td><td>11055 экземпляров. 3793 фишинговых сайтов, 7262 легитимных страниц.</td><td>31 признак</td><td>
<table>
<tr><th>Алгоритм</th><th>Достоверность</th></tr>
<tr><td>DNN</td><td>96,9%</td></tr>
</table>
</td></tr>
</table>

VI. Заключение

Фишинг является актуальной проблемой, поскольку вредоносные программы и методы социальной инженерии, используемые при этих атаках, могут нанести значительный ущерб как частным лицам, так и организациям. Поэтому важно принимать меры по защите от них.

Традиционные методы (например, черные и белые списки) не всегда обеспечивают достаточный уровень защиты и скорости реагирования на изменения. Использование машинного и глубокого обучения может значительно ускорить процесс обнаружения и противодействия фишинговым атакам.

ML и DL системы анализируют признаки веб-страниц и электронных писем, определяя степень доверия к ресурсу, и могут работать в режиме реального времени, что позволяет своевременно обнаруживать потенциально опасные сайты и электронные письма и блокировать их еще до того, как они достигнут конечного пользователя. Такой подход повышает эффективность обнаружения атаки и безопасность пользователя и организаций.

Большинство исследователей сосредотачиваются на предложении новых признаков, оптимизации алгоритмов классификации или предложении новых архитектур. В дальнейшем необходимо уделить больше внимания разработке надлежащего метода выбора и анализа признаков, что позволит определить компактный набор функций, которые действительно эффективны при обнаружении фишинговых атак. Данный подход позволит избежать использования нерелевантных признаков, что увеличит эффективность системы и уменьшит ее стоимость. Систематический подход к выбору признаков может значительно повысить эффективность систем борьбы с фишинговыми атаками и является перспективной областью для новых исследований.

Библиография

[1] The Open Web Application Security Project (OWASP) Top 10, https://owasp.org/Top10/#welcome-to-the-owasp-top-10-2021, 2021.

[2] Mirjana Pejić Bach, Tanja Kamenjarska, Bersilav Žmuk Targets of phishing attacks: The bigger fish to fry // Procedia Computer Science. - 2022. - №204. - С. 448-455.

[3] Butt, U.A., Amin, R., Aldabbas, H. et al. Cloud-based email phishing attack using machine and deep learning algorithm // Complex Intell. Syst. - 2022.

[4] Yuan, X.: PhD Forum: Deep Learning-Based Real-Time Malware Detection with Multi-Stage Analysis. In 2017 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–2 (2017)

[5] Saxe, J., Berlin, K.: eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys (2017)

[6] Shima, K., et al.: Classification of URL bitstreams using Bag of Bytes (2018)

[7] Vazhayil, A., Vinayakumar, R., Soman, K.: Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6 (2018)

[8] Zhang, X., Zeng, Y., Jin, X.-B., Yan, Z.-W., Geng, G.-G.: Boosting the phishing detection performance by semantic analysis. In 2017 IEEE International Conference on Big Data (BigData), pp. 1063–1070 (2017)

[9] Vanhoenshoven, F., Napoles, G., Falcon, R., Vanhoof, K., Koppen, M.: Detecting malicious URLs using machine learning techniques. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8 (2016)

[10] Chen, W., Zhang, W., Su, Y.: Phishing Detection Research Based on LSTM Recurrent Neural Network, pp. 638–645. Springer, Singapore (2018)

[11] Zhang, J., Li, X.: Phishing Detection Method Based on Borderline-Smote Deep Belief Network, pp. 45–53. Springer, Cham (2017)

[12] Aksu, D., Turgut, Z., Üstebay, S., Aydin, M.A.: Phishing Analysis of Websites Using Classification Techniques, pp. 251–258. Springer, Singapore (2019)

[13] Zhao, J., Wang, N., Ma, Q., Cheng, Z.: Classifying Malicious URLs Using Gated Recurrent Neural Networks, pp. 385–394. Springer, Cham (2019)

[14] Jiang, J., et al.: A Deep Learning Based Online Malicious URL and DNS Detection Scheme, pp. 438–448. Springer, Cham (2018)

[15] Spaulding, J., Mohaisen, A.: Defending Internet of Things Against Malicious Domain Names using D-FENS. In 2018 IEEE/ACM Symposium on Edge Computing (SEC), pp. 387–392 (2018)

[16] Pereira, M., Coleman, S., Yu, B., DeCock, M., Nascimento, A.: Dictionary Extraction and Detection of Algorithmically Generated Domain Names in Passive DNS Traffic, pp. 295–314. Springer, Cham (2018)

[17] Rao, R.S., Pais, A.R.: Detection of phishing websites using an efficient feature-based machine learning framework. Neural Comput. Appl., 1–23 (2018)

[18] Sur, C.: DeepSeq: learning browsing log data based personalized security vulnerabilities and counter intelligent measures. J. Ambient Intell. Humaniz. Comput., 1–30 (2018)

[19] Vrban˘ci˘c, G., Fister, I., Podgorelec, V.: Swarm Intelligence Approaches for Parameter Setting of Deep Learning Neural Network. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics—WIMS ’18, pp. 1–8 (2018)

Список используемых терминов

AdaBoost – адаптивный бустинг
AdaBoostM1 – адаптивный бустинг
AE (AutoEncoder) – автокодировщик
Bagging – бэггинг
BN (Bayes Network) – байесовская сеть
C4.5 – алгоритм для построения деревьев решений
C5.0 – реализация C4.5 на языке Си
SGD (Stochastic gradient descent) - стохастический градиентный спуск
CNN (Convolutional Neural Network) – сверточная нейронная сеть
DBN (Deep Belief Network) – глубокая сеть доверия
DL (Deep Learning) – глубокое обучение
DNN (Deep Neural Network) – глубокая нейронная сеть
DT (Decision Tree) – дерево решений
FNR (False negative rate) - частотность ошибок второго рода, т.е. количество ложно отрицательных результатов
GRU network (Gated Recurrent Units network) – нейронная сеть с механизмом вентилей
J48 – реализация C4.5 на языке Java
*k*NN (k-Nearest Neighbors) – метод k-ближайших соседей
LR (Logistic Regression) – логистическая регрессия
LSTM (Long Short-Term Memory) - рекуррентные нейронные сети с долгой краткосрочной памятью
ML (Machine Learning) – машинное обучение
MLP (Multilayer Perceptron) - многослойный персептрон
NB (Naive Bayes) – наивный байесовский классификатор
NLP (Natural Language Processing) – обработка естественного языка
NN (Neural Network) – нейронная сеть
RE (Regular expression) – регулярное выражение
RF (Random Forest) – метод случайного леса
RFE (Recursive Feature Elimination) – рекурсивное исключение признаков
RNN (Recurrent Neural Network) – рекуррентная нейронная сеть
SMO (Sequential Minimal Optimization) – метод последовательной минимальной оптимизации
SVM (Support Vector Machine) – метод опорных векторов

Research of the Capabilities of Deep Learning Algorithms to Protection Against Phishing Attacks

S.P. Korniukhina, O.R. Laponina

Abstract - **Phishing is one of the most common threats on the Internet which is why the development of effective protection methods is an extremely important task. This article discusses works that use the capabilities of machine and deep learning algorithms to protect against phishing attacks, as well as developed the comparison criteria and carried out the comparative analysis of solutions. The comparison of protection systems against phishing attacks was carried out according to the following criteria: the type of analyzed elements (HTML, URL, CSS); the dataset preprocessing methods (normalization and feature selection); the required sample size; the ML/DL algorithms used to detect phishing attacks; the number of errors of the 1st and 2nd kind, the quality criteria of the model. In the works CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) are most frequently studied, both separately and in combination with each other. Also, the SVM (Support Vector Machine) and DT (Decision Tree) algorithms, which are used for classification problems, are often studied.**

Keywords - **phishing, phishing attack, machine learning, deep learning, neural networks, ML, DL, CNN, LSTM, SVM, DT, DNN.**

REFERENCES

[1] The Open Web Application Security Project (OWASP) Top 10, https://owasp.org/Top10/#welcome-to-the-owasp-top-10-2021, 2021.
[2] Mirjana Pejić Bach, Tanja Kamenjarska, Bersilav Žmuk Targets of phishing attacks: The bigger fish to fry // Procedia Computer Science. - 2022. - №204. - С. 448-455.
[3] Butt, U.A., Amin, R., Aldabbas, H. et al. Cloud-based email phishing attack using machine and deep learning algorithm // Complex Intell. Syst. - 2022.
[4] Yuan, X.: PhD Forum: Deep Learning-Based Real-Time Malware Detection with Multi-Stage Analysis. In 2017 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–2 (2017)
[5] Saxe, J., Berlin, K.: eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys (2017)
[6] Shima, K., et al.: Classification of URL bitstreams using Bag of Bytes (2018)
[7] Vazhayil, A., Vinayakumar, R., Soman, K.: Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6 (2018)
[8] Zhang, X., Zeng, Y., Jin, X.-B., Yan, Z.-W., Geng, G.-G.: Boosting the phishing detection performance by semantic analysis. In 2017 IEEE International Conference on Big Data (BigData), pp. 1063–1070 (2017)
[9] Vanhoenshoven, F., Napoles, G., Falcon, R., Vanhoof, K., Koppen, M.: Detecting malicious URLs using machine learning techniques. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8 (2016)
[10] Chen, W., Zhang, W., Su, Y.: Phishing Detection Research Based on LSTM Recurrent Neural Network, pp. 638–645. Springer, Singapore (2018)
[11] Zhang, J., Li, X.: Phishing Detection Method Based on Borderline-Smote Deep Belief Network, pp. 45–53. Springer, Cham (2017)
[12] Aksu, D., Turgut, Z., Üstebay, S., Aydin, M.A.: Phishing Analysis of Websites Using Classification Techniques, pp. 251–258. Springer, Singapore (2019)
[13] Zhao, J., Wang, N., Ma, Q., Cheng, Z.: Classifying Malicious URLs Using Gated Recurrent Neural Networks, pp. 385–394. Springer, Cham (2019)
[14] Jiang, J., et al.: A Deep Learning Based Online Malicious URL and DNS Detection Scheme, pp. 438–448. Springer, Cham (2018)
[15] Spaulding, J., Mohaisen, A.: Defending Internet of Things Against Malicious Domain Names using D-FENS. In 2018 IEEE/ACM Symposium on Edge Computing (SEC), pp. 387–392 (2018)
[16] Pereira, M., Coleman, S., Yu, B., DeCock, M., Nascimento, A.: Dictionary Extraction and Detection of Algorithmically Generated Domain Names in Passive DNS Traffic, pp. 295–314. Springer, Cham (2018)
[17] Rao, R.S., Pais, A.R.: Detection of phishing websites using an efficient feature-based machine learning framework. Neural Comput. Appl., 1–23 (2018)
[18] Sur, C.: DeepSeq: learning browsing log data based personalized security vulnerabilities and counter intelligent measures. J. Ambient Intell. Humaniz. Comput., 1–30 (2018)
[19] Vrban˘ci˘c, G., Fister, I., Podgorelec, V.: Swarm Intelligence Approaches for Parameter Setting of Deep Learning Neural Network. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics—WIMS '18, pp. 1–8 (2018)