Добрышин Михаил Михайлович
Академии ФСО России, сотрудник к.т.н., г. Орёл
Кирикова Юлия Андреевна
Академии ФСО России, сотрудник, г. Орёл

ПРИМЕНЕНИЕ МЕТОДОВ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА ДЛЯ АНАЛИЗА НЕСТРУКТУРИРОВАННЫХ ДАННЫХ ОПИСЫВАЮЩИХ ТЕХНИКИ ИЗВЕСТНЫХ КОМПЬЮТЕРНЫХ АТАК

Применение средств машинного обучения позволяет не только реализовывать различные рутинные действия, но и автоматизировать процесс обработки большого массива данных генерируемого современным обществом. В статье сформулированы основные сложности обработки неструктурированных данных содержащихся в статьях, отчетах и постах в мессенджерах, для выявления описаний новых способов эксплуатации известных уязвимостей.

Ключевые слова: обработка естественного языка, машинное обучение, уязвимости, информационная безопасность.

Mikhail Mikhailovich Dobryshin
Federal Security Service Academy of Russia, PhD in Engineering, Oryol
Yulia Andreevna Kirikova
Federal Security Service Academy of Russia, employee, Oryol

APPLICATION OF NATURAL LANGUAGE PROCESSING METHODS TO ANALYZE UNSTRUCTURED DATA DESCRIBING TECHNIQUES OF KNOWN COMPUTER ATTACKS

The use of machine learning tools allows not only to implement various routine actions, but also to automate the process of processing a large amount of data generated by modern society. The article outlines the main difficulties of processing unstructured data contained in articles, reports, and posts in messengers to identify descriptions of new ways to exploit known vulnerabilities.

Keywords: natural language processing, machine learning, vulnerabilities, information security.

Текстовые данные составляют значительную долю всей информации, генерируемой человечеством. В настоящее время человечество документирует значительную часть своих действий, от личных страничек в социальных сетях, до ведения онлайн баз знаний о различных критических сферах деятельности человечества. Доступ к ресурсам мирового информационного пространства позволяет авторам (исполнителям) публиковать результаты своих исследований (научные статьи, отчеты и др.).

Однако с ростом количества публикаций, осведомленность «читателей» не увеличивается, а в отдельных случаях и снижается. Данное противоречие обусловлено наличием «информационного шума. Вследствие чего возникает задача по фильтрации и извлечению важных данных и устранению (удалению) информационного шума.

В качестве примера будет рассмотрена задача по выявлению описания новых (ранее не опубликованных) тактик и техник реализации компьютерных атак, эксплуатирующих известные уязвимости. Сформулированный подход применим и для других отраслей деятельности критичных ко времени обмена знаниями, например, финансового сектора или медицины.

При описании проблемной ситуации следует определить основные ресурсы (ограничения) и допущения. В качестве ограничений выступают конечные вычислительные ресурсы организации и временные ограничения. А в качестве допущений следует считать, что к обрабатываемой информации имеется полное доверие (доверие к источнику, автору и т.д.).

Проблемная ситуация: согласно требований регулятора (ФСТЭК России) в области информационной безопасности (ИБ), уязвимости критического уровня должны быть устранены в течение 24 часов с момента опубликования сведений о них (данное требование обусловлено потенциальным ущербом от успешной атаки на защищаемый ресурс) [1].

Методики расчета критичности уязвимостей включают различные факторы (в том числе статические – базовые и динамические – контентные), часть из которых способны повышать рассчитываемый уровень до критического,

основываясь на том, что нарушитель ИБ разработал и реализовал новый способ реализации атаки (эксплойт). Безусловно, этот факт требует внимания и незамедлительной реакции в системе безопасности.

Однако с момента реализации атаки до официального подтверждения этого факта доверенными вендорами проходит от трех до шести месяцев, что приводит к повторным успешным атакам, многократному увеличению ущерба и превышению сроков на устранение уязвимости.

Вместе с этим, в материалах отчетов о расследовании инцидентов безопасности, специализированные организации публикуют необходимую информацию в течение двух-пяти дней.

С целью повышения охвата обрабатываемых неструктурированных источников данных представленных естественным языком (статьи, отчеты, посты в месенджерах) и снижения времени реагирования предлагается применять инструменты машинного обучения для выявления новых тактик реализации атак эксплуатирующих известные уязвимости.

При обработке естественного языка с помощью программных средств (ПС) существует ряд сложностей [2-4]:

– Полисемия и омонимия – некоторые слова имеют несколько значений в зависимости от контекста, что может затруднить понимание их истинного смысла.

– Неполнота и неоднозначность – естественный язык часто использует сокращения, неполные предложения, а также выражения с нечетким значением. Это может приводить к неоднозначности и усложнять задачу понимания текста.

– Сложность грамматики – грамматика языка может быть сложной и изменчивой. Существуют различные грамматические правила, исключения, а также идиомы, что усложняет задачу автоматического анализа текста.

– Многоязычность – системы для обработки текстов должны работать с различными языками, что увеличивает сложность задачи. Различные языки имеют разные грамматические структуры, лексику и особенности культурного контекста.

– Необходимость контекста – значение текста часто зависит от контекста, в котором он используется. Отсутствие контекста или недостаточное его участие может привести к неправильному пониманию смысла.

Также при использовании русского языка при обработке естественного языка сопряжено с рядом других уникальных проблем и сложностей, представленных в списке ниже:

– Сложность грамматики – русский язык имеет сложную грамматическую структуру с различными падежами, временами и спряжениями. Это может усложнить задачу синтаксического и морфологического анализа;

– Флективность – русский язык является флективным, что означает, что слова могут изменяться по числу, роду, падежу и времени. Это требует более сложных методов анализа и обработки морфологии;

– Богатство словоизменительных форм – в русском языке существует большое количество словоизменительных форм для каждого слова, что усложняет задачу лемматизации и стемминга;

– Словообразование – русский язык богат различными методами словообразования, такими как аффиксация, суффиксация и префиксация. Это может приводить к образованию новых слов и сложным проблемам в разрешении омонимии;

Для достижения заявленной цели проведен анализ инструментов обработки естественного языка средствами машинного обучения (Natural Language Processing, NLP) – область компьютерной науки, которая занимается распознаванием, анализом (в том числе пониманием и толкованием содержащейся в тексте информации) и созданием текстов подобно тому, как это делает человек [4-6].

Целью применения NLP является создание систем, способных понимать, интерпретировать и генерировать естественный язык. Основные этапы обработки текста NLP:

Токенизация – это разбиение текста на отдельные слова или фразы.

Частеречная разметка. Это определение частей речи для каждого слова в тексте.

Лемматизация и стемминг. Это приведение слов к их базовым формам (леммам) или обрезание их до основы (стемминг) для уменьшения размерности словаря.

Удаление стоп-слов. Это процесс удаление часто встречающихся, но малоинформативных слов (стоп-слов) из текста.

Извлечение ключевых слов. Это выделение наиболее важных слов или фраз в тексте.

Извлечение информации. Это поиск и извлечение структурированной информации из текста, такой как именованные сущности (имена людей, места, даты и т. д.).

Синтаксический анализ – это анализ синтаксической структуры предложений для понимания их грамматической структуры и зависимостей между словами.

Семантический анализ. Понимание значения текста на более высоком уровне, включая анализ смысла, контекста и общей семантики.

Помимо перечисленных задач, при разработке ПС, обрабатывающего информацию, содержащуюся в текстовых документах, (например, в формате PDF – один из самых распространенных) необходимо реализовать задачу *оптического распознавания символов* (*OCR*) – принцип работы технологии заключается в сканировании изображения символов на страницах PDF-файла, и пытаются распознать эти символы как текст.

После этого полученный текст может быть извлечен и использован. Данный метод широко применяется для преобразования отсканированных документов, рукописных текстов или изображений с текстом в редактируемый формат. Программными решениями являются: *Tesseract, Abbyy FineReader, Adobe Acrobat* и другие.

Преимущества *OCR*-технологии заключаются в извлечении текста из изображений, сохраняя при этом его форматирование.

Недостатки: точность распознавания может зависеть от качества изображения и самого текста, что может привести к ошибкам распознавания.

Для решаемой задачи и разрабатываемого ПС возможно использовать универсальный язык программирования *Python*, который позволяет компилировать, подключать библиотеки и модули. Библиотеками *Python* для извлечения текста из PDF-файла являются:

– *PyPDF2* (используется для считывания файла);

– *Pdfpminer.six* (для выполнения анализа структуры и извлечения текста из PDF-файла).

Для дальнейшей обработки полученного текста используется библиотека на языке *Python Natural Language Toolkit(NLTK)*, которая будет эксплуатировать и реализовывать следующие функции в ПС:

– Токенизация *NLTK* – первый шаг при обработке текста, текст разбивается на отдельные слова, фразы, предложения или другие единицы (в данном ПС токенами являются отдельные слова и предложения);

– Удаление стоп-слов – удаление наиболее часто встречающиеся и малозначимые слова, которые не несут смысловой нагрузки, для чего используется список стоп-слов из *NLTK* (для каждого языка этот список уникален);

– Лемматизация и стемминг – преобразование слов к их базовым формам (нормальной форме), учитывая при этом морфологический анализ слов, а стемминг удаляет аффиксы, сохраняя основы слов.

Для получения данных о новых структурированных техниках в разработанном ПС реализован поиск совпадений с помощью сопоставления выделенных в обработанном тексте токенов с описанием известных уязвимостей и структурированных техник реализации КА.

Для поиска совпадений используется два способа.

Первым способом является поиск косинусного сходства *TF-IDF* – метод анализа текста, при котором для оценки сходства документов используется

косинусное сходство между векторами, представленными в виде Term Frequency-Inverse Document Frequency (TF-IDF).

Вторым способом является исследование текста на наличие в нем токенов в виде отдельных слов, приведенных к нормальной форме относящихся к описанию известных структурированных техник, при этом коэффициент сходства должен превышать заданное значение, пользователем.

Сведения об известных уязвимостях и техник реализации атак скачиваются с внешних источников данных и хранятся в *json*-файлах. Структура *json*-файла состоит из пар «ключ-значение» и может быть представлена в несколько уровней вложенности.

Проведенный анализ известных решений, их достоинств и недостатков позволил определить перечень библиотек, применение которых позволит реализовать заявленные функции и решить сформулированную задачу [7]:

– *Natural Language Toolkit (NLTK)* – служит для обработки естественного языка *(NLP)* и предоставляет множество инструментов анализа текста, включая токенизацию, лемматизацию, стемминг, анализ синтаксиса и многое другое;

– *Nltk.tokenize.word_tokenize* – метод *NLTK* используется для разделения текста на отдельные слова или токены и преобразует текст в список слов или токенов для последующей обработки;

– *Nltk.corpus.stopwords* – модуль *NLTK* содержит список стоп-слов на разных языках;

– *Nltk.stem.PorterStemmer, nltk.stem.WordNetLemmatizer* – используются для стемминга и лемматизации слов;

– *Sklearn.feature_extraction.text.TfidfVectorizer* – класс из библиотеки *scikit-learn* используемый для преобразования текстовых данных в матрицу *TF-IDF*;

– *Sklearn.metrics.pairwise.cosine_similarity* – метод из *scikit-learn* вычисляет косинусное сходство между векторами;

– *PyPDF2.PdfReader, pypdf.PdfReader* – классы позволяют читать *PDF*-файлы в *Python*.

Json – встроенная библиотека *Python* для работы с форматом данных JSON, позволяет кодировать и декодировать данные JSON, предоставляя удобный интерфейс для работы с данными в этом формате (в разработанном ПС реализовано извлечение значений «*value*», «*name*» и «*description*» из *json*-файла в которых хранятся описания об известных уязвимостях и техник реализации атак);

По результатам анализа всего документа сопоставленные пары: техника реализации КА – уязвимость, упорядочиваются и сопоставляются с номерами известных техник реализации атак (например, *MITRE ATT&CK*) и уязвимостей (например, *CVE*) соответственно. После чего создается отчет в формальном (номерном) и текстовом описании реализации КА с указанием используемых уязвимостей.

Разработанное ПС зарегистрировано в Роспатенте, а логика работы в составе других решений описана в патенте РФ на изобретение [8].

Разработанное ПС на основе применения элементов машинного обучения позволяет выявлять сведения о фактах эксплуатации известных уязвимостей при реализации новых тактик и техник реализации атак, что позволяет организации своевременно предпринять меры по недопущению нанесения ущерба [9].

К вопросу этичности применения искусственного интеллекта и машинного обучения в частности: разработанное ПС не заменяет аналитиков работающих в организации, а позволяет расширить объем обрабатываемых данных, что, несомненно, способствует снижению финансовых рисков и не исключает деятельность человека.

Возможности использования в других областях деятельности общества: в статье, большое внимание уделено методам, механизмам и средствам обработки естественного языка для того, чтобы было понимание общих процессов, протекающих в разработанном ПС. Так замена стоп-слов и использование для сравнения баз данных с набором слов из других областей деятельности позволяет считать данное средство универсальным.

Список литературы:

1. Белов А. С., Добрышин М. М., Громов Ю. Ю., Душкин А. В. / Квалиметрический анализ защищенных инфотелекоммуникационных систем / Учебное пособие для вузов. под науч. ред. А. В. Душкина // М. : Горячая линия – Телеком, 2025. – 156 с.

2. Контент-анализ СМИ: проблемы и опыт применения / Под ред. В. А. Мансурова. – М.: Институт социологии РАН, 2010. – 324 с.

3. Сапин А. С. Построение нейросетевых моделей морфологического и морфемного анализа текста. Труды ИСП РАН, том 33, вып. 4, – 2021 г., – С. 117-130.

4. Большакова Е. И., Воронцов К. В. и др. Автоматическая обработка текстов на естественном языке и анализ данных: учебное пособие. Изд-во НИУ ВШЭ, – 2017 г., 269 с.

5. Рассел, Стюарт, Норвиг, Питер. Искусственный интеллект: современный подход, 4-е издание, том 3. Обучение, восприятие и действие: Пер. с англ. – СПб.: ООО Диалектика, – 2022. – 640 с.

6. Барретт С. Ф. Arduino: искусственный интеллект и машинное обучение / пер. с англ. Ю. В. Ревича. – М.: ДМК Пресс, 2024. – 242 с.: ил.

7. Python [Электронный ресурс] / Python // www.python.org – Электрон. дан. – 2001-2024. Режим доступа: https://python.org/about – Дата обращения: 20.01.2026.

8. Добрышин М. М., Шугуров Д. Е., Кирикова Ю. А., Погодин Н. В. / Программа автоматического обновления и формирования техник реализации компьютерных атак для системы обеспечения информационной безопасности / Свидетельство о государственной регистрации программы для ЭВМ № 2023688 299 от 14.12.2023 Бюл. № 1.

9. Добрышин М. М., Белов А. С., Шугуров Д. Е., Кирикова Ю. А., и др. Система автоматического обновления и формирования техник реализации компьютерных атак для системы обеспечения информационной безопасности / Патент РФ на изобретение № 2809929 от 19.12.2023 Бюл. № 35 Заявка

2023118422, 12.07.2023. Патентообладатель: Академия ФСО России. G06F 21/50 (2013.01), G06F 16/22 (2019.01).