Отзывы

Будущие меры по обеспечению безопасности и защиты должны определяться способностью защищающихся развертывать средства машинного обучения для быстрого обнаружения и прекращения деятельности злоумышленников в интернете в любых маштабах. Чио и Фримэн написали по этой теме исчерпывающе полную книгу, включающую самые последние достижения научной мысли, а также трудные для изучения практические методики развертывания средств машинного обучения с целью обеспечения защиты людей в этой сфере деятельности.

– Алекс Стамос,
руководитель службы безопасности Facebook

Превосходное практическое руководство для всех, кто намерен освоить использование технологии машинного обучения для обеспечения безопасности компьютерных систем, от выявления аномалий до защиты конечных пользователей.

– Дэн Боне,
профессор ИТ, Стэнфордский университет

Если вы хотите знать, какое место занимает машинное обучение в области обеспечения безопасности, то книга Чио и Фримэна даст вам четкое представление об этом.

– Нвокеди С. Идика,
доктор наук, инженер ПО, Google, Security & Privacy Organization

Содержание

Предисловие

Машинное обучение завоевывает мир. Коммуникации и связь, финансовая сфера, транспорт, производство товаров и даже сельское хозяйство[1] – практически каждая отрасль технологии изменилась под влиянием машинного обучения или изменится в ближайшем будущем.

Обеспечение компьютерной безопасности также является важнейшей проблемой для всего мира. Поскольку мы становимся все более зависимыми от компьютеров в работе, развлечениях и вообще в обычной жизни, в равных пропорциях возрастает и значимость наличия брешей и лазеек в компьютерных системах, привлекающих нездоровое внимание постоянно увеличивающегося круга атакующих злоумышленников, которые надеются такими способами получить деньги или просто причинить ущерб. Более того, поскольку системы становятся все более сложными и взаимосвязанными, все труднее обеспечить отсутствие в них ошибок и непредвиденных лазеек, которые открывают доступ атакующим. Уже после сдачи книги в печать мы узнали о том, что в настоящее время слишком много микропроцессоров (если не каждый) используется без надлежащей защиты[2].

Машинное обучение предлагает (потенциальные) решения в любой области деятельности, поэтому вполне естественно, что эта технология применима и для компьютерной безопасности, т. е. для области, которая по своей сути является источником полезных и надежных наборов данных, на основе которых, собственно, и развивается технология машинного обучения. В самом деле, во всех сообщениях об угрозах для безопасности, которые появляются в новостях, обнаруживается аналогичное количество заявлений о том, что искусственный интеллект может «совершить революцию» в области методик обеспечения безопасности. Надежды на полное уничтожение наиболее значимых преимуществ атакующих злоумышленников привели к тому, что машинное обучение было широко разрекламировано как технология, которая позволит наконец завершить длительную игру в кошки-мышки между атакующими и защищающимися. При посещении экспозиционных залов самых крупных конференций по безопасности обнаруживается следующая тенденция: все большее количество компаний начинает использовать машинное обучение для решения проблем безопасности.

Быстро растущая заинтересованность в объединении этих двух областей порождает еще и атмосферу цинизма, в которой отвергается сама идея объединения – считается, что вокруг нее создан нездоровый ажиотаж. Как найти разумный баланс? Каков реальный потенциал применения методов искусственного интеллекта в сфере обеспечения безопасности? Как отличить рекламную шумиху от действительно многообещающих технологий? Что должен взять на вооружение конкретный пользователь для решения своих проблем безопасности? Мы ре-

---

[1] Monsanto. How Machine Learning is Changing Modern Agriculture. Modern Agriculture. September 13, 2017. **https://modernag.org/innovation/machine-learning-changing-modern-agriculture/**.

[2] Meltdown and Spectre. Graz University of Technology, accessed January 23, 2018. **https://spectreattack.com/**.

шили, что наилучшим вариантом ответов на все эти вопросы является глубокое изучение научных основ и методик, понимание главных концепций, огромный объем практического тестирования и экспериментирования, чтобы полученные результаты говорили сами за себя. Но все это требует большого объема актуальных знаний как в области датологии (науки о данных), так и в области обеспечения компьютерной безопасности. В процессе нашей работы по созданию систем безопасности, руководства группами, противодействующими злоупотреблениям, а также при участии в конференциях мы встретили некоторых людей, которые обладают таким объемом знаний. Кроме того, многие хорошо знают одну из этих областей и намерены изучать другую.

В результате появилась на свет данная книга.

О чем эта книга

Эта книга была написана, чтобы предоставить рабочую платформу для обсуждения неизбежного объединения двух широко распространенных концепций: машинного обучения и безопасности. Существует некоторое количество литературы, объединяющей эти две темы (а также многочисленные рабочие группы и семинары конференций: CCS AISec (**http://ai-sec.net**), AAAI AICS (**http://www-personal.umich.edu/~arunesh/AICS2018/**), NIPS Machine Deception (**https://www.machinedeception.com/**)), но большинство опубликованных работ является научными или теоретическими. Нам не удалось найти ни одного руководства, в котором были бы представлены конкретные работающие примеры с исходным кодом, способные помочь специалистам по обеспечению безопасности освоить науку о данных, а специалистам по машинному обучению в полной мере овладеть современными методиками решения задач обеспечения безопасности.

Исследуя широкий спектр тем в области обеспечения безопасности, мы представили примеры возможного практического применения технологии машинного обучения для улучшения или замены основанных на правилах или эвристических решениях таких задач, как обнаружение вторжения, классификация вредоносных программ и анализ сетевой среды. В дополнение к изучению основных алгоритмов и методик машинного обучения особое внимание мы уделили трудным задачам по созданию работоспособных, надежных, масштабируемых систем извлечения и анализа данных в сфере обеспечения безопасности. С помощью реально работающих примеров и подробных обсуждений с разъяснениями мы демонстрируем, как воспринимать и интерпретировать данные в конкурирующей среде и как обнаружить и выделить важные сигналы, которые могут быть незаметными в общем «шумовом» фоне.

Для кого предназначена эта книга

Если вы работаете в сфере обеспечения безопасности и намерены использовать машинное обучение для усовершенствования контролируемых вами систем, то эта книга для вас. Если вы специалист по машинному обучению и намерены использовать эту технологию для решения задач по обеспечению безопасности, то эта книга будет полезной и для вас.

Предполагается, что читатель обладает основами знаний о математической статистике, но большинство более сложных математических выкладок при первом чтении можно пропустить без ущерба для понимания концепций и принципов. Кроме того, предполагается знание какого-либо языка программирования. Примеры в книге написаны на языке программирования Python, также приводятся ссылки на пакеты Python, необходимые для реализации рассматриваемых концепций, но вы можете самостоятельно реализовать эти концепции, используя библиотеки с открытым исходным кодом на языках Java, Scala, C++, Ruby и многих других языках программирования.

Условные обозначения и соглашения, принятые в книге

В книге используются следующие типографские соглашения.

Курсив используется для смыслового выделения важных положений, новых терминов, URL-адресов и адресов электронной почты в интернете, имен команд и утилит, а также имен и расширений файлов и каталогов.

`Моноширинный шрифт` используется для листингов программ, а также в обычном тексте для обозначения имен переменных, функций, типов, объектов, баз данных, переменных среды, операторов, ключевых слов и других программных конструкций и элементов исходного кода. Также применяется для команд, выполняемых в командной строке, и для вывода результатов их выполнения.

`Моноширинный полужирный шрифт` используется для обозначения команд или фрагментов текста, которые пользователь должен ввести дословно без изменений. Также применяется для выделения важных фрагментов в выводимых результатах.

`Моноширинный курсив` используется для обозначения в исходном коде или в командах шаблонных меток-заполнителей, которые должны быть заменены соответствующими контексту реальными значениями.

Эта пиктограмма обозначает совет, рекомендацию или примечание общего характера.

Эта пиктограмма обозначает предупреждение или особое внимание к потенциально опасным объектам.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте **www.dmkpress.com**, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу **dmkpress@gmail.com**, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу **http://dmkpress.com/authors/publish_book/** или напишите в издательство по адресу **dmkpress@gmail.com**.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте **www.dmkpress.com** или **www.дмк.рф** на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг — возможно, ошибку в тексте или в коде, — мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу **dmkpress@gmail.com**, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и O'Reilly очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты **dmkpress@gmail.com** со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Благодарности

Авторы благодарят Хайрама Андерсона (Hyrum Anderson), Джейсона Крэйга (Jason Craig), Нвокеди Идика (Nwokedi Idika), Джесса Мэйлза (Jess Males), Энди Орэма (Andy Oram), Алекса Пинто (Alex Pinto) и Джошуа Сакса (Joshua Saxe) за подробные технические рецензии и отзывы на предварительные черновые версии этой книги. Мы также благодарим Вирджинию Уилсон (Virginia Wilson), Кристен Браун (Kristen Brown) и весь коллектив издательства O'Reilly, оказавший помощь в процессе превращения нашего проекта из концепции в реальность.

Кларенс благодарен Кристине Чжоу (Christina Zhou) за понимание и терпение в течение всех бесконечных ночей и выходных, которые были посвящены написанию этой книги, Ик Лун Ли (Yik Lun Lee) за корректуру черновых версий и обнаружение ошибок в коде, Джерроду Оверсону (Jarrod Overson), который заставил поверить в то, что я смогу сделать это, а также чихуа-хуа Дэйзи, которая всегда была на моей стороне в самые трудные времена. Спасибо Энто Джозефу (Anto Joseph) за обучение науке обеспечения безопасности и всем прочим хакерам, исследователям и участникам учебных курсов, которые так или иначе повлияли на создание этой книги, моим коллегам в Shape Security, которые сделали меня более опытным инженером, а также докладчикам и участникам конференции Data Mining for Cyber Security за то, что они являются частью сообщества, выполняющего исследования в этой области. Но самая большая благодарность – моей семье в Сингапуре за поддержку вне зависимости от того, где я нахожусь, за то, что позволили осуществиться моим мечтам и поощряли мои стремления.

Дэвид благодарен Дипаку Агарвалу (Deepak Agarwal), который убедил его заняться написанием этой книги, Дэну Боне (Dan Boneh) за обучение образу мышления, свойственному специалисту в сфере обеспечения безопасности, а также Винсенту Сильвиера (Vicente Silveira) и коллегам по LinkedIn и Facebook за то, что показали мне, чем в действительности является безопасность в реальном мире. Спасибо Грейс Тан (Grace Tang) за отзыв о разделах, посвященных машинному обучению, и за «случайного пингвина» (occasional penguin). И самая большая благодарность – Торри (Torrey), Илоди (Elodie) и Фебу (Phoebe), которые смирились с моим отсутствием многими поздними вечерами и несколько необычным поведением из-за необходимости завершения работы над этой книгой. У меня никогда не было повода для сомнений в их поддержке.

Глава 1

Машинное обучение и безопасность

В начале был спам.

Как только ученые соединили достаточное количество компьютеров через интернет для создания сети обмена информацией, приносящей реальную пользу, другие люди обнаружили, что это средство свободной передачи и широкого распространения данных представляет собой превосходный способ рекламирования сделанной наспех продукции, похищения секретных учетных и регистрационных данных и распространения компьютерных вирусов.

За следующие 40 лет в отрасль компьютерной и сетевой безопасности было включено значительное количество подобластей и средств противостояния потенциальным угрозам: обнаружение вторжения, защита веб-приложений, анализ вредоносных программ, защита социальных сетей, противодействие постоянно существующим опасностям, практическое применение криптографии и многое другое. Но даже в наши дни проблема спама остается главной для всех, кто имеет дело с электронной почтой или с системой обмена сообщениями. Возможно, вездесущий спам стал едва ли не основной проблемой в сфере обеспечения компьютерной безопасности, проблемой, которая напрямую воздействует на нашу жизнь.

Машинное обучение было изобретено не борцами со спамом, но технические специалисты, сведущие в математической статистике, быстро адаптировали эту технологию, потому что увидели в ней потенциальное средство борьбы с нарастающим потоком злоупотреблений. Провайдеры электронной почты и прочих сервисов интернета (IPS) получили доступ к огромному объему содержимого e-mail-сообщений, метаданных, а также возможность наблюдать за поведением пользователя. С использованием данных сообщений электронной почты можно сформировать основанные на содержимом (контенте) модели для создания обобщенных методик распознавания спама. Метаданные и характерные репутационные объекты можно извлекать из e-mail-сообщений, для того чтобы предварительно определить вероятность того, что электронное письмо является спамом даже без обращения к его содержимому. После создания цикла обратной связи с поведением пользователя вся система в целом может сформировать «коллективный разум», который со временем будет совершенствоваться с помощью самих пользователей.

Таким образом, фильтры электронной почты постепенно развивались, чтобы успешно противодействовать разнообразным хитроумным способам, которые

непрерывно изобретают авторы спама. Даже если 85 % всех сообщений электронной почты, пересылаемых в наши дни, являются спамом (если верить данным одной из исследовательских групп[1]), самые эффективные современные фильтры блокируют более 99.9 % всего спама[2]. Поэтому пользователи наиболее известных сервисов электронной почты чрезвычайно редко обнаруживают неотфильтрованный и нераспознанный спам в своих почтовых ящиках для входящих писем. Это говорит о значительном превосходстве современных методик над упрощенными методиками фильтрации спама, разработанными на начальном этапе использования интернета и использующими простую фильтрацию слов и репутационные характеристики метаданных e-mail-сообщений (**http://www.paulgraham.com/spam.html**) для достижения весьма скромных результатов.

Основным уроком, который извлекли и исследователи-теоретики, и инженеры-практики из этого противостояния, является важность использования данных, для того чтобы одержать победу над противниками-злоумышленниками, а также улучшение качества взаимодействия пользователей (т. е. фактически всех нас) с технологическими достижениями. В действительности история борьбы со спамом представляет собой характерный пример использования данных и технологии машинного обучения в любой области обеспечения компьютерной безопасности. В наши дни почти все организации в значительной мере полагаются на технологические достижения, но почти каждый элемент любой технологии имеет свои уязвимые места. Руководствуясь теми же основными побуждениями, что и спамеры из 1980-х (нерегулируемый свободный и бесплатный доступ к аудитории, располагающей некоторыми средствами и частной закрытой информацией), злоумышленники способны сделать потенциально опасными почти все аспекты нашей современной жизни. Действительно, сама сущность противостояния атакующей и защищающейся сторон одинакова во всех областях компьютерной безопасности, как и в борьбе со спамом: имеющий конкретный стимул злоумышленник постоянно пытается некорректно использовать компьютерную систему. При этом стороны все время пытаются залатать или воспользоваться «дырами» в проектном решении или в технологии, прежде чем другая сторона обнаружит эти действия. Формулировка проблемы остается неизменной.

Компьютерные системы и веб-сервисы постепенно централизуются, поэтому многие приложения предназначены для обслуживания миллионов и даже миллиардов пользователей. Объекты, которые становятся «повелителями информации», являются самой крупной мишенью для некорректного использования, но в то же время их положение весьма удачно для улучшения защиты данных и пользователей. С учетом появления мощных аппаратных средств обработки данных и разработки более эффективных алгоритмов анализа данных и машинного обучения именно сейчас наступило наилучшее время для применения потенциальных преимуществ машинного обучения в сфере обеспечения безопасности.

В этой книге мы показываем практическое применение методик машинного обучения и анализа данных в различных проблемных областях обеспечения

---

[1] К сожалению, информация по ссылке в оригинале **http://bit.ly/2EKGDDZ** недоступна (Page not found). – *Прим. перев.*

[2] К сожалению, информация по ссылке в оригинале **http://bit.ly/2DbwD66** недоступна (Page not found). – *Прим. перев.*

безопасности и защиты от некорректного использования. Подробно излагаются методы определения применимости разнообразных методик машинного обучения в разных ситуациях. Особое внимание сосредоточено на руководящих принципах, которые помогут использовать данные для улучшения защиты. Нашей целью не являются решения для каждой конкретной проблемы безопасности, с которой можно встретиться в реальной практике, – мы хотим предоставить читателю рабочую среду, основу для работы с данными и для решения задач защиты, а также комплект инструментов, из которого вы можете выбрать наиболее подходящее средство (метод) для решения конкретной поставленной задачи.

Следующая часть этой главы формирует контекст для всей книги в целом: рассматривается, каким опасностям подвергаются современные компьютерные и сетевые системы, что такое машинное обучение и как его можно применить для борьбы с вышеупомянутыми опасностями. В конце главы подробно описываются методы борьбы со спамом, которые представляют собой конкретный пример применения машинного обучения в сфере обеспечения безопасности, который можно обобщить для последующего применения практически в любой области.

Общий обзор потенциальных киберугроз

Общая картина, характеризующая нарушителей и злоумышленников в области компьютерной безопасности, изменяется со временем, но общие категории угроз и опасностей остаются неизменными. Для того чтобы сорвать планы атакующих злоумышленников, проводятся специальные исследования, поэтому всегда важно правильно классифицировать различные типы реально существующих атак. На рис. 1.1 изображено дерево систематической классификации (таксономии) существующих киберугроз (Cyber Threat Taxonomy)[1], на котором можно видеть связи между конкретными типами киберугроз и категориями, которые в некоторых случаях могут быть достаточно сложными.

Начнем с определения основных типов киберугроз, которые будут рассматриваться в следующих главах книги:

- вредоносное программное обеспечение (ПО) (malware) или вирус (virus) – программное обеспечение, специально предназначенное для нанесения ущерба или получения несанкционированного доступа к компьютерным системам (malware – malicious software);
- червь (worm) – автономная вредоносная программа, способная размножаться и копировать себя на другие компьютерные системы;
- троянская программа (trojan) – вредоносная программа, выдающая себя за одну из обычных программ, чтобы избежать обнаружения;
- программа-шпион (spyware) – вредоносная программа, установленная на компьютерной системе без разрешения и даже без ведома оператора/пользователя для шпионажа и сбора информации. К этой категории также относятся кейлоггеры;

[1] Источник: European CSIRT Network project's Security Incidents Taxonomy (**https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends**).

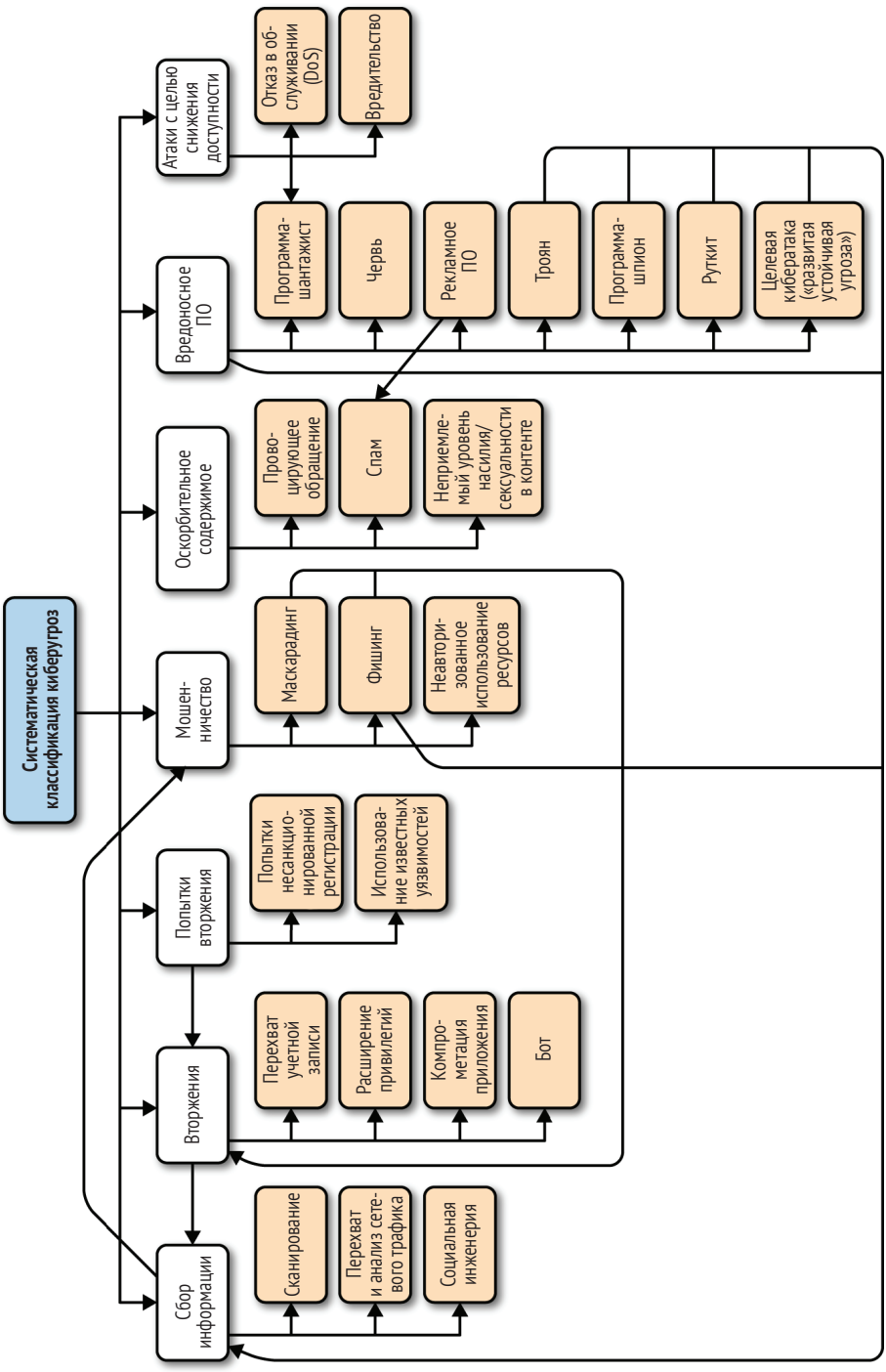

Рис. 1.1 ❖ Дерево систематической классификации (таксономии) существующих киберугроз (Cyber Threat Taxonomy)

- рекламное ПО (adware) – вредоносная программа, которая вводит непредусмотренные рекламные материалы (например, всплывающие окна, баннеры, видеоклипы) в подсистему пользовательского интерфейса, чаще всего появляющиеся при просмотре пользователем веб-контента;
- программа-шантажист (ransomware) – вредоносная программа, специально предназначенная для ограничения функциональных возможностей компьютерных систем до тех пор, пока не будет выплачена определенная денежная сумма (выкуп);
- руткит (rootkit) – комплект ПО низкого уровня (чаще всего), специально предназначенного для получения доступа или полного захвата управления компьютерной системой (root обозначает самый высокий уровень доступа и управления системой);
- бэкдор, или «черный ход» (backdoor) – преднамеренно созданная или оставленная лазейка («дыра»), размещенная на периметре защиты системы и позволяющая в будущем получить доступ в обход подсистемы внешней защиты;
- бот (bot) – вариант вредоносной программы, позволяющий атакующему в удаленном режиме перехватить управление компьютерными системами, превращая их в «зомби»;
- ботнет, сеть ботов (botnet) – крупная сеть ботов;
- эксплойт (exploit) – фрагмент кода или программа, использующая конкретные уязвимости в других прикладных программах или программных средах;
- сканирование (scanning): при этом типе атаки на компьютерные системы отправляются разнообразные запросы, часто в режиме простого перебора (грубой силы), с целью обнаружения слабых мест и уязвимостей, а также для сбора информации;
- перехват и анализ сетевого трафика (sniffing) – незаметное наблюдение и фиксация сетевого трафика и внутреннего трафика на сервере без ведома сетевых операторов;
- кейлоггер (keylogger) – деталь аппаратуры или фрагмент ПО (чаще всего скрытые от пользователя), которые фиксируют все нажатия клавиш на клавиатуре или действия на другом устройстве ввода;
- спам (spam) – незапрашиваемые сообщения, рассылаемые в крупных масштабах, чаще в рекламных целях. Обычно используется электронная почта, но спам также может распространяться в смс-сообщениях или через провайдера системы обмена сообщениями (например, WhatsApp);
- атака во время процедуры регистрации (login attack) – многочисленные, обычно автоматизированные попытки подобрать учетные данные для систем аутентификации, реализованные в форме простого перебора (грубой силы) или использующие похищенные/незаконно приобретенные учетные данные;
- захват учетной записи (account takeover – ATO) – получение доступа к чужой учетной записи, как правило, с целью нарушения коммерческой деятельности, кражи личных данных, похищения денежных средств и т. п. Обычно перехват учетной записи является целью атаки во время процедуры регистрации, но также может иметь меньший масштаб и более высокую целенаправленность (например, шпионское ПО, социальная инженерия);

- фишинг (phishing) или маскарадинг (masquerading) – установление связи от имени человека или организации, заслуживающих доверия. Цель: убедить объект фишинга предоставить личную информацию или передать права владения материальными ценностями;
- направленный, или целевой, фишинг (spear phishing) – фишинг, целью которого является конкретный пользователь, с использованием информации об этом пользователе, собранной из различных внешних источников;
- социальная инженерия (social engineering) – получение информации от людей с применением нетехнических методов, таких как ложная информация, обман, подкуп, шантаж и т. п.;
- провоцирующее обращение (incendiary speech) – унижающее, дискредитирующее или другое подобное враждебное обращение, адресованное отдельному лицу или группе лиц;
- атака типа «отказ в обслуживании», или DoS-атака, и распределенная DoS-атака – атаки, направленные на снижение доступности систем и выполняемые с помощью многочисленных некорректных запросов и/или запросов, содержащих большие объемы данных. Зачастую такие атаки также нарушают целостность и надежность систем;
- целевая кибератака («развитая устойчивая угроза») (advanced persistent threat – APT) – целенаправленная атака на сеть или на хост, при которой скрывающийся нарушитель остается необнаруженным в течение долгого времени и постоянно похищает и отслеживает передаваемые данные;
- уязвимость нулевого дня (zero-day vulnerability) – уязвимость или ошибка в ПО или в компьютерной системе, которая неизвестна производителю (поставщику), позволяющая воспользоваться ею (атака «нулевого дня»), прежде чем у производителя (поставщика) появится возможность устранить эту проблему.

Экономическая подоплека кибератак

По каким причинам предпринимаются кибератаки? Преступления в интернете становятся все более коммерциализированными по сравнению с начальным этапом распространения этой технологии. Переход от кибератак, мотивацией которых являлось зарабатывание определенной репутации (дешевая популярность, особенно в молодежной среде, известность и даже просто возможность совершать подобные проделки), к кибератакам с целью получения денег (прямые хищения денег, реклама, продажа личной секретной информации) стал весьма привлекательным процессом, особенно с точки зрения злоумышленников. В наши дни главная побуждающая причина кибератак – получение крупных денежных сумм. Атаки на финансовые организации или каналы (онлайновые платежные платформы, учетные записи, содержащие данные о кредитных и дебетовых картах, кошельки биткойнов и т. п.) могут открыть атакующим злоумышленникам прямой доступ к денежным средствам. Но с увеличением денежного объема, вовлеченного в онлайновый оборот, финансовые организации все чаще применяют усовершенствованные механизмы защиты, усложняющие жизнь злоумышленников. Из-за соблазна найти более короткий и легкий путь в сферу финансовой деятельности «рынок», предлагающий использование уязвимостей в системах защиты

финансовых организаций и каналов платежей, также представляет собой многочисленное и оживленное сообщество. Злоумышленники постоянно ищут объекты с более слабой защитой, неправильно эксплуатируемые системы с уязвимостями из-за ошибок при проектировании, а также обращаются к более изощренным методикам, которые в конечном итоге позволяют получить в свое распоряжение некоторую денежную сумму.

Рынок услуг взломщиков

Всем известно о существовании рынков darknet и не вполне законных форумов хакеров и взломщиков. До появления организованных подпольных сообществ, занимающихся незаконной деятельностью, только самые умелые хакеры способны были принять участие в организации кибератак и взломе учетных записей и компьютерных систем. Но с ростом коммерциализации хакерской деятельности и при массовом применении компьютеров во всех сферах жизни даже малоопытные «хакеры»[1] смогли внедриться в экосистему кибератак, получая в свое распоряжение (прибретая) информацию об уязвимостях и удобные для любого пользователя хакерские скрипты, программы и инструментальные средства для осуществления собственных кибератак.

На рынке уязвимостей нулевого дня существуют и практически законные, и абсолютно незаконные варианты. Торговля информацией об уязвимостях и методиками их использования может стать реальным источником дохода как для исследователей в области защиты и обеспечения безопасности, так и для хакеров. Но большинство самых «элитных» хакеров не склонно пользоваться уязвимостями «нулевого дня» и участвовать в организации массовых атак. Слишком велик риск, а кроме того, процесс обналичивания слишком долговременный и неопределенный. Создание ПО, которое дает возможность любому неопытному скрипт-кидди (script-kiddy) совершить попытку реального взлома, продажа информации об уязвимостях на свободных рынках, а в некоторых случаях даже небольшие компании, предоставляющие консультации и сервисы по взлому, – все это позволяет поверить в то, что существует быстрый и легкий путь к финансовому благополучию. Как во времена знаменитой золотой лихорадки в Калифорнии в конце 1840-х годов, продавцы, проявляющие чрезвычайную учтивость и любезность к растущей армии охотников за богатством, гораздо чаще получают неожиданные прибыли, чем сами охотники.

Косвенная монетизация

Процесс монетизации (или обналичивания денежных средств), предпринимаемый злоумышленниками, связан с различными типами компьютерных атак и настолько многообразен, что заслуживает подробнейшего изучения. Здесь мы не будем углубляться в исследования подобного рода, но рассмотрим несколько примеров, демонстрирующих, как может осуществляться косвенная монетизация.

Распространение вредоносного ПО было коммерциализировано способом, похожим на развитие облачных вычислений (cloud computing) и развертывание

[1] Вообще говоря, они не заслуживают звания «хакер» в изначальном, не столь криминализированном значении этого термина. – *Прим. перев.*

провайдеров инфраструктуры как сервиса (IaaS – Infrastructure-as-a-Service). Рыночные отношения типа «плата за установку» (pay-per-install – PPI) при распространении вредоносного ПО представляют собой вполне сложившуюся сложную экосистему, предоставляющую мощные каналы распространения, доступные и авторам, и потребителям[1]. Аренда ботнет основана на том же принципе, что и облачная инфраструктура, предоставляемая по запросу, с почасовой оплатой выделяемых ресурсов за приемлемую цену. Развертывание вредоносного ПО на удаленных серверах также может быть прибыльным с финансовой точки зрения, с разнообразными специфическими способами монетизации. Целевые атаки на конкретные объекты иногда связаны с получением какой-либо финансовой выгоды, а распространение программ-шантажистов может быть достаточно эффективным способом вымогательства денежных средств у обширной группы жертв.

Шпионское ПО может способствовать похищению личной секретной информации, которую затем можно выгодно продать оптом на тех же онлайновых рынках ПО для шпионажа. Рекламное ПО и средства распространения спама можно использовать как дешевый способ рекламирования не вполне легальных фармацевтических товаров и финансовых инструментов. Онлайновые учетные записи часто перехватываются с целью похищения ценностей, хранящихся в особой форме, как, например, подарочные (призовые) карты, бонусные баллы за лояльность, открытые кредиты в магазинах или премиальный возврат денег при покупках. Похищенные номера кредитных карт, номеров социальной страховки, учетных записей электронной почты, номеров телефонов, адресов и прочая личная секретная информация может быть продана в режиме онлайн преступникам, намеревающимся заняться воровством, подделками, мошенничествами и прочими подобными деяниями. Но процесс монетизации (обналичивания), в особенности если злоумышленник располагает номером кредитной карты жертвы, может стать долгим и сложным. Из-за возможности легкого похищения такой информации компании, предоставляющие кредитные карты, а также компании, обслуживающие учетные записи для спецхранения ценностей, часто применяют хитроумные технические методики для предотвращения монетизации, предпринимаемой злоумышленниками. Например, если возникает подозрение, что учетные записи скомпрометированы, то они объявляются некорректными и неработающими, а для карт с премиальным возвратом денег требуются дополнительные процедуры аутентификации.

Подведем итоги

Побудительные мотивы киберпреступников сложны, а способы монетизации извилисты. Тем не менее финансовые выгоды от атак в интернете могут стать мощным стимулом для технически подготовленных людей, особенно из не очень богатых стран и сообществ. Пока компьютерные атаки способны создавать обширную криминальную сферу деятельности для злоумышленников, такие атаки будут продолжаться.

---

[1] *Juan Caballero et al*. Measuring Pay-per-Install: The Commoditization of Malware Distribution. Proceedings of the 20th USENIX Conference on Security (2011).

Что такое машинное обучение

В самом начале «технологической эры» ученые мечтали о том, чтобы научить компьютеры рассуждать логически и принимать «разумные» решения точно так же, как это делает человек, выводя общие правила и выделяя концепции из больших объемов сложной информации без точно определенных инструкций.

Машинное обучение (machine learning) связано только с одной из этих перспектив, а именно с алгоритмами и процессами, которые являются «обучающими» в смысле обеспечения возможности обобщать данные и практические сведения, полученные в прошлом, для того чтобы предсказывать будущие результаты. По своей сущности машинное обучение представляет собой набор математических методов, реализованных в компьютерных системах, обеспечивающих процесс извлечения информации, обнаружение шаблонов и формирование выводов из данных.

На самом высоком (обобщенном) уровне при машинном обучении с учителем (supervised machine learning)[1] применяется методика Байеса (Bayes) для выявления знаний, использующая известные вероятности наступления ранее наблюдаемых событий для определения вероятностей новых событий. Машинное обучение без учителя (unsupervised machine learning)[2] выделяет абстрактные признаки из наборов непомеченных данных и применяет эти признаки к новым данным. Обе группы методик можно применить к задачам классификации (classification), т. е. распределения наблюдений по категориям, или регрессии (regression), т. е. прогнозирование числовых характеристик наблюдения.

Предположим, что необходимо выполнить классификацию группы животных, разделяя их на млекопитающих и рептилий. По методике обучения с учителем берется группа животных, для которых явно указана их категория (например, четко определено, что собака и слон – млекопитающие, а аллигатор и игуана – рептилии). Затем мы пытаемся извлечь какие-либо характерные признаки из каждого элемента этих помеченных данных и найти сходство в этих признаках, позволяющее различать животных, принадлежащих к разным классам. Например, очевидно, что собака и слон порождают живое потомство, в отличие от аллигатора и игуаны. Бинарное свойство «порождает живое потомство» называют характеристикой или признаком (feature), т. е. полезной абстракцией для наблюдаемых признаков, которые позволяют сравнивать различные наблюдения. После окончательного определения набора характеристик, которые могут помочь отличить млекопитающих от рептилий в помеченных данных, можно начать выполнение алгоритма обучения на наборе помеченных данных, затем применить то, чему научился алгоритм, к новым, ранее не называемым животным. Если применить этот алгоритм к сурикату, то после обучения должна быть выполнена классификация, относящая это животное либо к млекопитающим, либо к рептилиям. Получив набор характеристик из данных об этом новом животном, алгоритм знает, что сурикат не откладывает яиц, не имеет чешуи и теплокровный. На основании

---

[1] Также используются термины «контролируемое обучение» и «управляемое обучение». – *Прим. перев.*

[2] Также используются термины «неконтролируемое обучение» и «самообучение». – *Прим. перев.*

предыдущих наблюдений делается вывод: сурикат принадлежит к категории млекопитающих, и этот вывод абсолютно верный.

По методике обучения без учителя предпосылка та же самая, но алгоритм не получает в свое распоряжение начальный набор помеченных данных о животных. Вместо этого алгоритм должен группировать различные наборы элементов данных таким способом, чтобы в результате получить бинарную классификацию. Определив по наблюдениям, что большинство животных, которые не имеют чешуи, порождают живое потомство и являются теплокровными, а большинство животных, которые имеют чешую, откладывают яйца и являются холоднокровными, алгоритм способен распределять по двум категориям животных из предоставленной группы и предсказывать основные характеристики так же, как и в случае обучения с учителем.

Алгоритмы машинного обучения основаны на математике и статистике, а алгоритмы, выявляющие шаблоны, корреляции и аномалии в данных, имеют различные степени сложности. В следующих главах будут более подробно рассматриваться механизмы некоторых наиболее часто применяемых алгоритмов машинного обучения, используемых в этой книге. Книга не поможет вам в полной мере освоить машинное обучение, в ней не излагаются подробно математические и теоретические основы этой технологии. Авторы попытались привить читателю разумное отношение к машинному обучению и практические навыки для проектирования и реализации «интеллектуальных» динамически адаптируемых систем в контексте обеспечения безопасности.

Чем не является машинное обучение

Искусственный интеллект (artificial intelligence – AI) – широко распространенный, но весьма неопределенный термин, который в общем обозначает алгоритмические решения сложных задач, которые обычно решаются людьми. Как показано на рис. 1.2, машинное обучение является основным конструктивным блоком (и даже ядром) искусственного интеллекта. Например, автомобили с автоматическим управлением обязательно должны классифицировать наблюдаемые образы-объекты как людей, автомобили, деревья и т. п., а кроме того, непременно должны прогнозировать положение и скорость других машин. Они также должны определять угол поворота колес для изменения направления движения. Такие задачи классификации и прогнозирования решаются с использованием машинного обучения, поэтому система автоматического управления автомобилем представляет собой некоторую форму искусственного интеллекта. Существуют и другие элементы механизма принятия решения в интеллектуальной системе автоматического управления, которые жестко закодированы в виде набора правил, следовательно, их нельзя считать методами машинного обучения. Машинное обучение помогает создавать искусственный интеллект, но это не единственный метод формирования ИИ.

Глубокое обучение (deep learning) – это еще один широко распространенный термин, который, вообще говоря, тесно связан с машинным обучением. Глубокое обучение представляет собой строго ограниченное подмножество методик машинного обучения, применяемых к особому классу многоуровневых моделей, в которых используются уровни упрощенных статистических компонентов для изучения представлений данных. Нейронная сеть (neural network) – более общий

термин для обозначения этого типа многоуровневой статистической архитектуры обучения, которая может быть или не быть «глубокой» (т. е. иметь или не иметь много уровней). Эта тема превосходно раскрыта в книге Deep Learning Иэна Гудфеллоу (Ian Goodfellow), Йошуа Бенджио (Yoshua Bengio) и Аарона Курвиля (Aaron Courville), выпущенной издательством MIT Press.

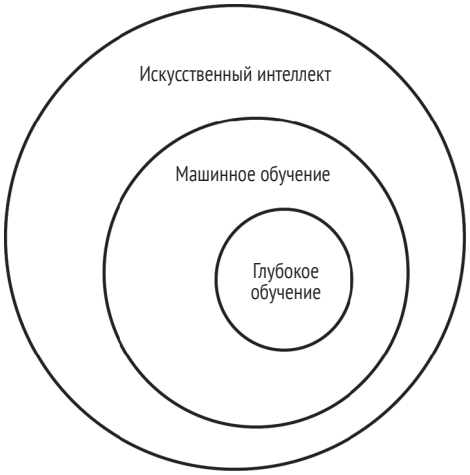

Рис. 1.2 ❖ Взаимосвязь искусственного интеллекта с машинным обучением и глубоким обучением

Статистический анализ является важнейшей частью машинного обучения: результаты выполнения алгоритмов машинного обучения часто представлены в форме вероятностей и доверительных интервалов. Некоторые статистические методы будут кратко описаны при рассмотрении темы выявления аномалий, но мы оставили за пределами данной книги многие вопросы, касающиеся методов проверки экспериментальных и статистических предположений. Эти вопросы в полной мере рассматриваются в книге Probability & Statistics for Engineers & Scientists, Роналд Уолпол (Ronald Walpol) и др. (издательство Prentice Hall).

Что такое искусственный интеллект

Определение искусственного интеллекта представляет собой несколько более спорную тему, по сравнению с определением машинного обучения. Машинное обучение – это комплект алгоритмов статистического обучения, способных создать обобщенные абстракции (модели) посредством наблюдения и тщательного анализа наборов данных. Системы искусственного интеллекта были определены менее точно как механизмы управляемого машиной принятия решений, которые могут достигать уровня интеллекта, сравнимого с человеческим. Но в какой степени этот интеллект должен быть «сравнимым» с человеческим разумом, чтобы можно было считать его искусственным интеллектом? Вполне объяснимо, что разнообразие предположений и трактований этого термина чрезвычайно затрудняет определение границ данной области, которое стало бы общепринятым.

Другие варианты использования машинного обучения

Следует отметить, что нет препятствий для использования преимуществ машинного обучения злоумышленниками, чтобы избежать обнаружения и обойти систему защиты. Защищающаяся сторона имеет возможность извлекать полезный опыт из атак и соответствующим образом принимать контрмеры, но и атакующая сторона также изучает внутренние механизмы систем защиты, чтобы извлечь личную выгоду. Распространители спама освоили практическое применение полиморфизма (т. е. изменение внешнего вида содержимого без изменения его смысла), для того чтобы не позволить распознать откровенно их рекламный контент, или для прощупывания фильтров спама с выполнением A/B-тестирования содержимого электронной почты и изучением процентного соотношения успешных и отфильтрованных вариантов. Обе стороны используют машинное обучение в своих «кампаниях фаззинг-тестирования, направленных на ускорение процесса поиска уязвимостей в программном обеспечении» (**http://www.vdiscover.org/OS-fuzzing.html**). Злоумышленники могут воспользоваться машинным обучением даже для исследования ваших личных данных и интересов в соцсетях, чтобы сформировать идеальное фишинговое сообщение персонально для вас.

Наконец, использование динамических и адаптивных методов в области обеспечения безопасности всегда связано с определенной долей риска. Особенно в тех случаях, когда обоснованность прогнозов машинного обучения часто становится недостаточной, атакующая сторона узнает о различных алгоритмах, ставших причиной ошибочных предсказаний или неправильного обучения[1]. В этой постоянно расширяющейся области обучения, называемой «состязательным машинным обучением» (adversarial machine learning), атакующие с той или иной степенью доступа к системе машинного обучения могут выполнять некоторый набор атак для достижения своих целей. В главе 8 подробно рассматривается эта тема и читателю предоставляется более полное описание задач и решений в этой области.

Алгоритмы машинного обучения во многих случаях не предназначены специально для обеспечения безопасности, поэтому зачастую уязвимы для попыток вмешательства в их работу, предпринимаемых мотивированным злоумышленником. Поэтому важно иметь полное представление о подобных моделях угроз при проектировании и создании систем машинного обучения, специализированных для обеспечения безопасности.

Практические варианты использования машинного обучения для обеспечения безопасности

В этой книге рассматриваются различные приложения, предназначенные для обеспечения компьютерной безопасности, для которых машинное обучение уже продемонстрировало многообещающие результаты. Практическое применение машинного обучения и науки о данных (даталогии) для решения задач – дело непростое. Несмотря на то что удобные в использовании программные библиотеки снижают уровень сложности, для разработчиков сохраняется необходимость постоянно принимать множество самостоятельных решений.

---

[1] *Ling Huang et al.* Adversarial Machine Learning. Proceedings of the 4th ACM Workshop on Artiicial Intelligence and Security (2011): 43–58.

Рассматривая различные примеры в текущей главе, мы исследуем наиболее часто возникающие в реальной практике проблемы при проектировании систем машинного обучения, вне зависимости от того, предназначены они для обеспечения безопасности или нет. Приложения, описываемые в данной книге, не новы, поэтому можно обнаружить обсуждаемые здесь методики даталогии во многих компьютерных системах, с которыми вы, возможно, ежедневно имеете дело.

Практические варианты использования машинного обучения в обеспечении безопасности можно классифицировать по двум основным категориям: распознавание шаблонов (pattern recognition) и выявление аномалий (anomaly detection). Границу между распознаванием шаблонов и выявлением аномалий не всегда можно четко определить, но для каждой конкретной задачи поставлена точно сформулированная цель. При распознавании шаблонов мы пытаемся обнаружить явные или неявные характеристики, скрытые в данных. Эти характеристики, выделенные и объединенные в наборы признаков, могут использоваться для обучения алгоритма распознаванию других форм данных с точно таким же набором характеристик. Выявление аномалий – это получение знаний при противоположном подходе к той же задаче. Вместо изучения характерных шаблонов, существующих в конкретных наборах данных, главной целью становится определение понятия нормальности, которое описывает большую часть (например, более 95 %) данных в исследуемом наборе. После этого любые отклонения от установленной нормальности будут определяться как аномалии.

Распространено ошибочное мнение о том, что выявление аномалий является процессом распознавания набора «нормальных» шаблонов и установления их отличий от набора «ненормальных» шаблонов. Шаблоны, извлекаемые по методике распознавания шаблонов, непременно должны быть производными от исследуемых данных, используемых для предварительной подготовки (тренировки) алгоритма. С другой стороны, при использовании методики выявления аномалий возможно существование бесконечного количества аномальных шаблонов с характеристиками, соответствующими заявленным описаниям промахов (выбросов), даже тех, которые являются производными от гипотетических данных, реально не существующих в тренировочных или тестовых наборах данных.

Выявление спама, возможно, представляет собой классический пример распознавания шаблонов, поскольку спам обычно обладает вполне предсказуемым набором характеристик, и алгоритм можно подготовить для распознавания этих характеристик как шаблона, по которому классифицируются сообщения электронной почты. Кроме того, также допустима трактовка выявления спама как задачи выявления аномалий. Если есть возможность вывода набора признаков, которые описывают обычный сетевой трафик достаточно подробно и точно, для того чтобы определить существенные отклонения от нормы как спам, то задача решена успешно. Но в действительности задача выявления спама не вполне соответствует парадигме выявления аномалий, поскольку не составляет никакого труда убедиться в том, что в большинстве контекстов легче найти одинаковые свойства, присущие спам-сообщениям, чем в более обширном наборе данных обычного трафика.

Выявление вредоносного ПО и обнаружение ботнетов представляют собой другие типы приложений, которые явно попадают в категорию распознавания шаблонов, где машинное обучение становится особенно полезным при атаках с использованием полиморфизма, чтобы скрыться от обнаружения. Фаззинг (fuz-

zing) – это процесс передачи случайных входных данных в программу или какой-либо компонент программы с целью перевода приложения в непредусмотренное некорректное состояние. Чаще всего в таком процессе ставится задача достижения аварийного завершения программы или создания уязвимого режима ее работы с возможностью дальнейшего использования этой уязвимости. Плохо подготовленные («наивные») мероприятия по фаззинг-тестированию часто сводятся к простому итеративному проходу по трудно определяемому огромному пространству состояний приложения. Наиболее часто применяемое ПО для фаззинг-тестирования предоставляет средства оптимизации, которые существенно повышают эффективность этой методики по сравнению со «слепым» перебором состояний (**http://lcamtuf.coredump.cx/afl/**). В оптимизациях подобного рода также применялось и применяется машинное обучение – исследование образцов (шаблонов) ранее обнаруженных уязвимостей в похожих программах и ориентирование фаззера (fuzzer; программа фаззинг-тестирования) по маршруту аналогичного уязвимого кода или по характерным особенностям кода для потенциально более быстрого получения результатов.

При аутентификации пользователей и анализе поведения различия между распознаванием шаблонов и выявлением аномалий становятся менее очевидными. В случаях, когда модель угрозы хорошо известна, возможно, более подходящим является методика решения задачи с помощью двунаправленной трансформации данных при распознавании шаблонов (lens of pattern recognition). В других ситуациях более предпочтительно применение выявления аномалий. Во многих случаях система может применять обе методики для обеспечения наилучшего покрытия. Выявление промахов в сетевой среде – классический пример выявления аномалий, поскольку почти весь сетевой трафик строго соблюдает протоколы и обычное поведение соответствует набору шаблонов по форме или по порядку следования. В сети любая злонамеренная деятельность, которая не применяет технику маскарадинга для имитации обычного трафика, будет обнаружена с помощью алгоритмов выявления промахов. Решения других задач выявления событий, связанных с сетью, такие как выявление вредоносных URL, также могут рассматриваться с точки зрения выявления аномалий.

Управление доступом (access control) обозначает любой набор стратегий, управляющих возможностью пользователей какой-либо системы получать доступ к конкретным элементам информации. Часто используемые для защиты важной секретной информации от нежелательного раскрытия, стратегии управления доступом в большинстве случаев представляют собой первую линию защиты от проникновений и похищения информации. Технология машинного обучения постепенно нашла свое место в решениях задач управления доступом как средство облегчения жизни пользователей систем, находящихся во власти строгих и беспощадных стратегий управления доступом[1]. С помощью сочетания методики обучения без учителя и выявления аномалий такие системы могут делать логические выводы относительно информации о шаблонах доступа для конкретных пользователей или ролей в организации и принимать ответные меры при обнаружении несоответствия установленным шаблонам.

---

[1] *Evan Martin and Tao Xie.* Inferring Access-Control Policy Properties via Machine Learning. Proceedings of the 7th IEEE International Workshop on Policies for Distributed Systems and Networks (2006): 235–238.

Например, предположим, что существует система хранения записей о пациентах больницы, к которой необходимо обеспечить постоянный доступ медсестер и медтехников, но при этом они не должны устанавливать какие-либо связи и отношения между различными пациентами. С другой стороны, врачам разрешается делать запросы и объединять регистрационные записи группы пациентов для выявления аналогичных симптомов и диагнозов. Нет необходимости запрещать медсестрам и медтехникам выполнять запросы на получение записей о нескольких пациентах, потому что в редких случаях потребуется разрешение на подобные действия. Основанная на правилах строгая система управления доступом не сможет обеспечить гибкость и адаптируемость, которую способна предоставить методика машинного обучения.

В следующих главах книги будет более подробно рассматриваться выбор подобных приложений в реальной практике. Мы получим возможность обсудить нюансы, касающиеся применения машинного обучения для распознавания шаблонов (образов) и выявления аномалий в области обеспечения безопасности. В заключительной части текущей главы мы сосредоточимся на разборе примера борьбы со спамом как одной из наглядных иллюстраций применения базовых принципов, используемых в любом приложении машинного обучения для обеспечения безопасности.

Борьба со спамом: итеративный подход

Как уже было отмечено выше, пример борьбы со спамом является едва ли не самой старой задачей обеспечения компьютерной безопасности, которая к тому же успешно решается с помощью машинного обучения. В этом разделе подробно рассматривается эта тема и наглядно демонстрируется постепенное создание «интеллектуальной» системы классификации спама с использованием машинного обучения. Описываемый здесь подход является обобщенным для многих других типов задач обеспечения безопасности, включая и задачи, рассматриваемые в последующих главах. Но не следует полагать, что круг решаемых с помощью этого подхода задач ограничен лишь описанными в нашей книге, на самом деле он более широк.

Рассмотрим случай, когда предлагается решить задачу устранения угрозы распространения по электронной почте спама, мешающего работе сотрудников в некоторой организации. По определенным причинам получено распоряжение разработать собственное решение, а не использовать стороннее коммерческое ПО. Обладая правами доступа администратора на внутренних серверах электронной почты в организации, вы можете извлекать тело (содержимое) e-mail-сообщений для анализа. Все сообщения электронной почты помечаются получателями соответственно либо как «спам» (spam), либо как «не спам» (non-spam; «ham»)[1], поэтому не нужно тратить времени на фильтрацию данных[2].

---

[1] Первоначально слово «SPAM» появилось в 1936 г. как аббревиатура от «**Sp**iced h**am**» (острая ветчина) и было торговой маркой для мясных консервов компании Hormel Foods Corporation – острого колбасного фарша из свинины. Всемирную известность в применении к назойливой рекламе термин «SPAM» получил благодаря знаменитому скетчу «Спам» из известного телевизионного шоу «Летающий цирк Монти Пайтона» (1969). – *Прим. перев.*

[2] В реальной практике приходится затрачивать достаточно много времени на фильтрацию данных, чтобы сделать их доступными и действительно полезными для применяемых алгоритмов.

Человек успешно распознает спам, поэтому начнем с реализации простого решения, приближенно имитирующего процесс человеческого мышления при выполнении этой задачи. Теоретическая предпосылка заключается в наличии или отсутствии некоторых известных ключевых слов в сообщении электронной почты – это четкий признак того, что сообщение является спамом или не спамом. Например, замечено, что слово «лотерея» (lottery) весьма часто встречается в спам-сообщениях, но крайне редко появляется в обычных деловых письмах. Возможно, вы в итоге составите список таких слов и выполните классификацию, проверяя, содержится ли в тексте сообщения какое-либо слово из этого «черного списка».

Набор данных, используемый для решения поставленной задачи, взят из источника 2007 TREC Public Spam Corpus (**https://plg.uwaterloo.ca/~gvcormac/treccorpus07/**). Это слегка отфильтрованный массив настоящих сообщений электронной почты, содержащий 75 419 экземпляров, собранных на сервере электронной почты в течение трех месяцев 2007 года. Треть этого набора данных составляют образцы спама, остальные – обычные сообщения. Набор данных создан участниками конференции Text REtrieval Conference (TREC) Spam Track (**https://trec.nist.gov/data/spam.html**) в 2007 году как часть работы по расширению границ самой передовой технологии выявления спама.

Для оценки эффективности работы различных методик будет применяться простой процесс проверки (валидации)[1]. Основной набор данных разделяется на неперекрывающиеся подготовительный (тренировочный) и тестовый наборы, при этом тренировочный набор состоит из 70 % данных (пропорция выбрана произвольно), а на долю тестового набора остается 30 % данных. Такая методика является стандартной практикой для оценки эффективности алгоритма или модели, разрабатываемых на основе подготовительного (тренировочного) набора данных. В дальнейшем проверенный алгоритм (или модель) обобщается для работы с независимым набором данных.

Первый этап – использование инструментального пакета Natural Language Toolkit (NLTK) (**http://www.nltk.org/**) для удаления морфологических аффиксов из слов для более адаптивного процесса поиска совпадений (такой процесс называют стеммингом (stemming), т. е. определением основы слова). Например, этот подход позволяет сократить слова «congratulations» и «congrats» до одной и той же основы «congrat». Также удаляются шумовые слова (stopwords) (например, артикли the и a, глаголы-связки is и are) перед процессом извлечения токенов-образ-

[1] Этот процесс валидации, который иногда называют условной валидацией (conventional validation), является не столь строгим методом, как перекрестная валидация (cross-validation), обозначающая целый класс методов, многократно генерирующих все (или большинство) различные возможные варианты разделения набора данных (на подготовительный и тестовый наборы) и выполняющих проверку (валидацию) прогнозирующего алгоритма машинного обучения отдельно на каждом из этих наборов. Результатом перекрестной валидации является усредненная точность прогнозов по всем этим различным вариантам разделенных наборов. Перекрестная валидация оценивает точность модели лучше, чем условная валидация, поскольку позволяет избежать возможных потерь информации при обработке единственного подготовительного/тестового варианта набора данных, который может не вполне корректно определить статистические свойства данных (обычно это не становится причиной для беспокойства, если подготовительный (тренировочный) набор данных достаточно велик). В нашем примере для простоты выбрана методика условной валидации.

цов, поскольку шумовые слова обычно не несут какой-либо смысловой нагрузки. Определяется набор вспомогательных функций[1] для загрузки и предварительной обработки данных и меток, как показано в следующем коде[2]:

```
import string
import e-mail
import nltk

punctuations = list(string.punctuation)
stopwords = set(nltk.corpus.stopwords.words('english'))
stemmer = nltk.PorterStemmer()

# Объединение различных частей e-mail-сообщения в простой список строк
def flatten_to_string(parts):
    ret = []
    if type(parts) == str:
        ret.append(parts)
    elif type(parts) == list:
        for part in parts:
            ret += flatten_to_string(part)
    elif parts.get_content_type == 'text/plain':
        ret += parts.get_payload()
    return ret

# Извлечение текста темы и тела из одного e-mail-файла
def extract_e-mail_text(path):
    # Загрузка одного e-mail-сообщения из входного файла
    with open(path, errors='ignore') as f:
        msg = e-mail.message_from_file(f)
    if not msg:
        return ""

    # Чтение темы сообщения
    subject = msg['Subject']
    if not subject:
        subject = ""

    # Чтение тела сообщения
    body = ' '.join(m for m in flatten_to_string(msg.get_payload())
                    if type(m) == str)
    if not body:
        body = ""
    return subject + ' ' + body

# Обработка одного e-mail-файла для преобразования слов в основы-токены
def load(path):
    e-mail_text = extract_e-mail_text(path)
    if not e-mail_text:
```

[1] Эти вспомогательные функции определены в файле *chapter1/e-mail_read_util.py* (**https://github.com/oreilly-mlsec/book-resources/blob/master/chapter1/e-mail_read_util.py**) в репозитории кода для данной книги.

[2] Для выполнения этого кода необходимо установить Punkt Tokenizer Models и массив шумовых слов в NLTK с помощью утилиты `nltk.download()`.

```
        return []

    # Разбивка сообщения на токены
    tokens = nltk.word_tokenize(e-mail_text)

    # Удаление знаков пунктуации из токенов
    tokens = [i.strip("".join(punctuations)) for i in tokens
              if i not in punctuations]

    # Удаление шумовых слов и выделение основ токенов
    if len(tokens) > 2:
        return [stemmer.stem(w) for w in tokens if w not in stopwords]
    return []
```

Далее загружаются сообщения электронной почты и метки. В этом наборе данных каждое сообщение помещено в отдельный файл (*inmail.1*, *inmail.2*, *inmail.3*, ...), кроме того, имеется отдельный файл меток (*full/index*) в следующем формате:

```
spam   ../data/inmail.1
ham    ../data/inmail.2
spam   ../data/inmail.3
...
```

В файле меток в начале каждой строки содержится метка «spam» или «ham» для всех образцов сообщений в исследуемом наборе данных. Этот набор данных считывается, и формируется черный список слов, характеризующих спам[1]:

```
import os

DATA_DIR = 'datasets/trec07p/data/'
LABELS_FILE = 'datasets/trec07p/full/index'
TRAINING_SET_RATIO = 0.7

labels = {}
spam_words = set()
ham_words = set()

# Чтение меток
with open(LABELS_FILE) as f:
    for line in f:
        line = line.strip()
        label, key = line.split()
        labels[key.split('/')[-1]] = 1 if label.lower() == 'ham' else 0

# Разделение массива на тренировочный и тестовый наборы данных
filelist = os.listdir(DATA_DIR)
X_train = filelist[:int(len(filelist)*TRAINING_SET_RATIO)]
X_test = filelist[int(len(filelist)*TRAINING_SET_RATIO):]

for filename in X_train:
    path = os.path.join(DATA_DIR, filename)
    if filename in labels:
```

[1] Этот пример можно найти в файле примечаний Python Jupyter *chapter1/spam-fighting-blacklist.ipynb* в репозитории исходного кода для данной книги (**https://github.com/oreilly-mlsec/book-resources/blob/master/chapter1/spam-fighting-blacklist.ipynb**).

```
        label = labels[filename]
        stems = load(path)
        if not stems:
            continue
        if label == 1:
            ham_words.update(stems)
        elif label == 0:
            spam_words.update(stems)
        else:
            continue

blacklist = spam_words - ham_words
```

При внимательном изучении токенов в черном списке `blacklist` можно заметить, что многие слова не несут смысловой нагрузки (например, слова в кодировке Unicode, URL, имена файлов, символы, иностранные слова). Эту проблему можно устранить с помощью более тщательной фильтрации данных, но даже такой упрощенный результат должен в достаточно неплохой степени соответствовать целям нашего эксперимента:

```
greenback, gonorrhea, lecher, ...
```

Оценивая работу применяемой методики на 22 626 сообщениях электронной почты из тестового набора, мы обнаруживаем, что этот упрощенный алгоритм не так эффективен, как можно было надеяться. Результаты объединены в итоговом отчете в форме матрицы неточностей или несоответствий (confusion matrix) размером 2×2, где указано количество образцов, для которых были сделаны прогнозы, и предварительно присвоенные метки для каждого из четырех возможных вариантов (табл. 1.1).

Таблица 1.1

	Прогнозируемый HAM	**Прогнозируемый SPAM**
Действительный HAM	6772	714
Действительный SPAM	5835	7543
Истинно положительное срабатывание: прогнозируемый спам + действительный не спам Истинно отрицательное срабатывание: прогнозируемый не спам + действительный не спам Ложноположительное срабатывание: прогнозируемый спам + действительный не спам Ложноотрицательное срабатывание: прогнозируемый не спам + действительный спам		

Процентные соотношения полученных результатов показаны в табл. 1.2.

Таблица 1.2

	Прогнозируемый HAM	**Прогнозируемый SPAM**
Действительный HAM	32.5 %	3.4 %
Действительный SPAM	28.0 %	36.2 %
Точность классификации: 68.7 %		