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

УТВЕРЖДАЮ

Директор по образовательной деятельности

_____ С.Т. Князев

«14» декабря 2021 г.

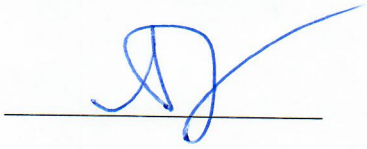
# Искусственный интеллект для информационной безопасности

Учебно-методические материалы по направлению подготовки
**09.04.01 Информатика и вычислительная техника**
Образовательная программа «Инженерия искусственного интеллекта»

**Екатеринбург**

**2021**

**РАЗРАБОТЧИКИ УЧЕБНО-МЕТОДИЧЕСКИХ МАТЕРИАЛОВ**

Доцент, канд.техн.наук  Созыкин Андрей Владимирович

**Основы информационной безопасности**

## Что такое информация?

Понятие информации можно определить как со стороны чистой математики (где это мера, обратная неопределенности), так и со стороны законодательства (в частности, определение из ФЗ-149 «Об информации, информационных технологиях и о защите информации».

На практике удобно использовать некий синтез определений, который отражает нематериальную природу информации (не зависит от формы представления), а также ее ценность для снижения неопределенности, которая еще и падает со временем.

Ссылки: http://www.consultant.ru/document/cons_doc_LAW_61798/

## Среда обитания информации – информационная сфера

Рассмотреть модель информационной сферы в изложении Доктрины обеспечения ИБ Российской Федерации 2000-го года.

Сделать акцент на том, что, когда мы говорим «информационная безопасность», мы говорим о безопасности информационной сферы в целом – не выделяя какой-то конкретной ее части.

Ссылки: https://base.garant.ru/182535/

## Что такое безопасность?

После перейти к понятию безопасность в наиболее широкой трактовке.

Акцентировать внимание, что свойство безопасности симметрично: как среда не должна навредить системе, так и наоборот.

## Пример информационной сферы

Разобрать абстрактные понятия информационной сферы на жизненном примере: некоторое предприятие.

Обратить внимание, что даже в этом примере в границы информационной сферы попадают субъекты, не относящиеся к самому предприятию: это его контрагенты.

## Иерархия понятий

Далее обсудить, что мы будем говорить в первую очередь о компьютерной безопасности.

Акцентировать внимание на отличии этого термина от термина «Информационная безопасность» - компьютерная безопасность сосредоточена только на информационной инфраструктуре и той информации, которая представлена носителями в информационной инфраструктуре.

При этом далее мы будем использовать термины ИБ и КБ в качестве синонимов.

Также здесь нужно обратить понятие на термин «компьютерная система», под которым мы будем понимать часть автоматизированной системы, включающей только средства автоматизации.

## Что такое компьютерная система с точки зрения КБ?

Вновь вернемся к понятию компьютерной системы. Согласно введенному определению это набор инструментов для обработки информации и сама информация, которая может либо кратковременно попадать в систему, либо храниться в ней длительное время, что определяется процессами обработки информации.

Сама информация при этом обладает рядом свойств, называемых свойствами безопасности информации: конфиденциальность, целостность и доступность (пояснить по каждому свойству, что за ними скрывается). Это не исчерпывающий список свойств безопасности информации, но указанные 3 свойства наиболее часто рассматриваются в вопросах ИБ.

Нарушение состояния безопасности информации – это нарушение одного из свойств.

## Составные части атаки

Ввести несколько формальных определений, заодно упомянуть базовые стандарты в области ИБ:
ГОСТ Р 50922-2006 «Защита информации. Основные термины и определения»
ГОСТ Р 53113.1-2008 «Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Часть 1. Общие положения».
Перейти к визуализации понятий.
Пояснить, что угрозы и источники угроз – это объективные свойства среды, которые никак не связаны с защищаемой компьютерной системой.
А вот уязвимость и актив – это базовые элементы (с точки зрения КБ) компьютерной системы.

**Отличие ущерба и потерь**

Ущерб выражается в терминах безопасности информации, т.е. нарушение свойства целостности, доступности или конфиденциальности.
А вот потеря – это уже некоторые негативные последствия за рамками компьютерной системы: реальные денежные потери, срыв планов производства, экологические бедствия, травмы и гибель людей.
Отметить, что атака, приводящая к потерям, может содержать множество шагов (реализаций различных угроз). Каждый такой шаг будет наносить ущерб активам КС, но потери возникнут только в конце цепочки атаки.

**Появление уязвимостей в системе**

Разобрать понятия и разные типы уязвимостей по ГОСТ Р 56546-2015 «Защита информации. Уязвимости информационных систем. Классификация уязвимостей информационных систем»
Также упомянуть понятие многофакторной уязвимости (сразу несколько вариантов) и сказать, что в вопросах ИБ предполагают, что уязвимости присутствуют всегда – вопрос лишь времени (вероятности) их обнаружения источником угрозы.

**Прочие важные понятия**

Проговорить прочие важные понятия, которые вводятся с помощью базовых понятий:
Риск – некая мера, объединяющая последствия реализации угрозы и вероятность их наступления.
Модель угроз – попытка описания возможных угроз системы и определение наиболее актуальных из них.
Сценарий атаки – описание действий источника угрозы по реализации угрозы (с использованием уязвимости и нанесением ущерба активу).
Понятие атаки – когда мы наблюдаем в реальности реализацию угрозы посредством уязвимости.
Инцидент – по сути, факт успешной атаки.

**Уровни обеспечения информационной безопасности**

4 уровня обеспечения ИБ: законодательство – создание правового поля, формирование отношения к сфере ИБ.
Административный – цели и ресурсы на обеспечение ИБ со стороны высшего руководства предприятием.
Процедурный – все процессы обеспечения ИБ, упомянуть афоризм Брюса Шнайера и акцентировать внимание на том, что состояние безопасности не является устойчивым – оно требует постоянной поддержки, поэтому нужна процедурная составляющая.
Программно-технический уровень – то, где сосредоточены все программно-аппаратные средства обеспечения ИБ.

**Законодательный уровень**

Рассматриваем каждый уровень чуть более подробно.

На законодательном уровне регулируются информационные отношения в информационной сфере (всех уровней – даже внутри организации мы ориентируемся на действующее законодательство и, например, не можем объявить коммерческой тайной информацию, для которой явно установлен запрет на отнесение к коммерческой тайне в одноименном законе).

Также отметить, что только на законодательном уровне может быть введено понятие компьютерного преступление, а без него невозможно полноценно обеспечить ИБ на любом уровне.

**Общая структура нормативно-правовых актов РФ**

Описать иерархию НПА Российской Федерации.

Напомнить понятие юридической силы и назначения отдельных видов НПА.

**Административный уровень**

Пояснить, что административный уровень – это придание «законности» мероприятий по обеспечение ИБ на отдельно взятом предприятии. Пояснить примером, что подразделения по обеспечению ИБ могут, например, накладывать запреты на определенные действия смежных подразделений и даже вышестоящего руководства. Чтобы у них была возможность это делать – нужно указание со стороны высшего руководства компании.

Кроме того, обеспечение ИБ – это затратное мероприятие, соответственно, руководство должна поставить цели и выделить средства для их достижения.

**Политика безопасности**

Основной документ, отражающий волю высшего руководства по вопросам ИБ – это политика [информационной] безопасности.

Акцентировать внимание, что политика – это свод принципов. Постановка целей и общих установок по их достижению (например, без физических наказаний в отношении сотрудников J ). Политика ни в коем случае не должна быть подробной инструкцией. Хорошая политика занимает буквально несколько страниц будучи напечатанной.

**Процедурный уровень обеспечения ИБ**

Процедурный уровень – это процессы, которые позволяют достигать и сохранять свойство безопасности на длительных промежутках времени.

Для этого реализуется ряд процессов, которые делят по уровням их детализации и скорости протекания.

Стратегические процессы рассматривают вопросы ИБ верхнеуровнево и цикл их выполнения тоже длительный. Как правило, не менее года. Здесь мало связи с реальными компьютерными системами, вопросы в больше степени касаются персонала, документации и пр.

Тактические процессы реализуются в более быстром темпе, и они уже в большей степени связаны с защищаемой компьютерной системой. Сюда попадают различные манипуляции с компонентами компьютерной системы и системы защиты: обновление, изменение политик и пр.

**Программно-технический уровень, стратегии защиты**

Программно-технический уровень по большей части представлен средствами обеспечения ИБ, работающими в автоматическом режиме (в соответствии с заданной конфигурацией).

При этом программно-технические средства (ПТС) реализуют различные стратегии защиты:

· Превентивная – когда усилия направлены на то, чтобы не дать злоумышленнику использовать уязвимость и совершить за счет этого атаку. Обычно это достигается путем запрета выполнения каких-то действий в системе (например, ограничение допустимого сетевого трафика)

· Активная – на этой стадии злоумышленник уже реализует свою атаку и задача ПТС этой стадии: обнаружить признаки атаки и пресечь ее дальнейшее выполнение (напомнить, что реальная атака всегда будет представлена цепочкой, лишь в конце которой бизнесу наносятся потери)
· Реактивная – это стратегия, направленная на устранение последствий реализации атаки. Например, если актив был искажен или уничтожен – мы восстанавливаем его из резервной копии.

**Программно-технический уровень обеспечения ИБ**

Сам программно-технический уровень тесно связан с процедурным (акцентировать внимание на том, что один без другого бессмысленен).
Поэтому можно говорить о еще одном уровне процессов: оперативных процессах обеспечения ИБ. Большая часть из них реализуется автоматически с помощью ПТС обеспечения ИБ, но часть требует периодического участия человека.

**Модель компьютерной системы**

Для решения любой задачи необходимо построение адекватной модели объекта из реального мира. Для компьютерной системы наиболее общей является модель, в которой система представлена набором состояний, а также преобразованием перехода (в общем случае W – не функция, так как КС может перейти в каждый момент времени во множество новых состояний, конкретизируется выбор состояния на основе дополнительных данных: действий пользователя, сигналов от других систем и пр.), меняющей состояние системы с течением времени.

**Модель безопасности компьютерной системы**

Для модели безопасности дополнительно вводятся еще два элемента модели:
Условие перехода – булева функция, определяющая допустимые значения для функции переходов W (отражение ограничений правил ИБ в реальной системе).
g – критерий безопасности, который умеет по текущему состоянию определять является оно безопасным или нет.
Все состояния, соответственно, делятся на 2 класса: безопасные и небезопасные.
Такая модель уже позволяет исследовать вопросы защищенности КС.

**Основная теорема безопасности**

Моделируя реальную систему всегда очень сложно описать множество состояний и функцию переходов, чтобы полученная модель не была переусложнена, но при этом решала свою задачу – позволяла делать какие-то предсказания относительно эволюции реальной системы.
В частности, моделирование безопасности КС производится с целью определения функции ограничений f и критерия g, таких, чтобы для данной модели системы не был бы возможен переход в небезопасное состояние.
Если при этом критерий g соответствует реальному критерию безопасности, то функция f будет соответствовать реальным правилам безопасности, реализация которых позволит сделать моделируемую систему абсолютно защищенной (конечно, пока мы остаемся в пределах упрощений модели).
Таким образом, цель моделирования безопасности – нахождение f,g и $q_0$ таких, что, если в начальный момент времени (в состоянии $q_0$) система находится в безопасном состоянии (т.е. $g(q_0) = 1$) и переходит в новые состояния только с соблюдением правил безопасности (т.е. $f(q_t, q_{t+1})$ = истина), то она будет оставаться в безопасном состоянии в любой момент времени.
Можно дополнительно упомянуть частные примеры, например, модель Бела Лападулы, акцентировав внимание, что многое определяется понятием критерия безопасности.

**Понятие субъекта**

Переходим к рассмотрению типовых «кирпичиков» для формальных моделей безопасности КС.

Субъект – это активная сущность системы. То, что меняет систему во времени. Без субъектов система была бы статична, информация в ней не менялась бы никогда.

Субъект – это аватар пользователя. Естественно, человек не может выполнять действия в КС напрямую, ему необходимы посредники в виде пакетов прикладного ПО. Чем субъект будет являться в реальной системе – зависит от этой системы. Далее мы посмотрим на несколько примеров.

**Понятие объекта**

Следующий фундаментальный компонент модели – объект. Объект – это пассивная сущность. Он выступает в роли хранилища информации. В КС манипуляция с информацией происходит посредством объектов. Сами объекты имеют некоторую структуру, определяемую конкретной КС, физическое воплощение объекта, как и субъекта, определяется типом конкретной моделируемой системы.

Важно, что информационное содержимое объекта меняется с течением времени.

**Поток и доступ**

Для того, чтобы описать изменения информации в объектах моделируемой КС, необходимо ввести 2 дополнительных понятия.

Для начала примем, что перенос информации в системе осуществляют только субъекты, а хранение информации может осуществляться только объектами. Т.е. субъект не может перенести информацию от себя к объекту – он будет инициировать перенос от одного объекта к другому. Если при этом необходимо моделировать ситуацию, когда у субъекта есть собственная память, мы просто предполагаем наличие дополнительного объекта, связанного с субъектом, в котором эта информация может хранится.

Перенос информации от одного объекта (источника) к другому (получателю) будем называть *потоком* информации. Также примем, что один из объектов может отсутствовать – это случай передачи информации из/в смежную систему.

Инициацию потока субъектом s от объекта $o_i$ к объекту $o_j$ будем называть доступом субъекта s к объекту $o_i$ и объекту $o_j$.

Ссылки: https://elar.urfu.ru/bitstream/10995/1778/5/1335332_schoolbook.pdf

Н.А. Гайдамакин, «Учебно-методический комплекс. Теоретические основы компьютерной безопасности», раздел 1.3.

**Примечание:** в упомянутом пособии рассматривается субъектно-объектная модель целостности КС, поэтому некоторые определения в ней отличаются от приведенных здесь. Мы рассматриваем более общий случай.

**Примеры**

Рассмотрим несколько примеров применения субъектно-объектной модели к реальным КС.

Процессор. В этом случае субъект – это текущий поток исполнения процессора, а объектами являются регистры процессора, ячейки памяти, порты ввода/вывода, программные прерывания и пр. Процессор при этом контролирует доступ субъекта – потока исполнения к этим ресурсам и, в зависимости от состояния служебных регистров, может блокировать большинство вариантов доступа.

Операционная система. Наиболее каноничный пример: в качестве субъектов выступают процессы (потоки, нити), действующие от имени пользователя ОС, объектами являются все объекты ОС: файлы, сокеты, сигналы, события, каналы и пр.

Переходя к более сложным и масштабным системам можно обратить внимание, что границы между субъектами и объектами размываются. В частности, если мы говорим о локальной вычислительной сети, то информационный обмен тут осуществляют сетевые

сервисы (например, браузер и web-сервер), которые могут периодически меняться ролями, в зависимости от текущего направления информационного потока.
На совсем абстрактном уровне – облаке субъекты укрупняются до законченных сервисов (например, облачная система бухгалтерского учета), объектами доступа становятся элементы облачной инфраструктуры, обеспечивающие исполнение программного кода сервиса, хранение его данных и организацию информационных потоков между другими объектами.

**От общего к частному**

Теперь попробуем перейти от общих задач ИБ к частным техническим задачам, решаемым в современных КС.
Для этого вновь пройдем с наиболее абстрактного уровня – информационной сферы в целом – до конкретного уровня информационной инфраструктуры (компьютерной системы).
Безопасность в информационной сфере в наиболее общем виде мы рассмотрели. Если же спуститься ниже – на уровень отдельного предприятия, то общая безопасность в информационной сфере превращается в безопасность бизнес-процессов. Именно от бизнес-процесса зависит и безопасность информации, и безопасность субъектов отношений.
Рассматривая безопасность бизнес-процесса с учетом определения термина «безопасность», приходим к выводу, что она достигается в случае четкой регламентированности процесса, когда каждый участник знает, что нужно делать, как нужно делать и выполняет это в соответствии с предписанием инструкции.
Соответственно, на самом нижнем уровне мы должны обеспечить взаимодействие субъектов и объектов отражающее регламент бизнес-процесса и роли его участников. Причем некорректные действия должны быть недоступны – тогда можно будет гарантировать безопасность на уровне бизнес-процесса.

**Проблема перехода**

Но, при переходе на уровень компьютерной системы, мы полностью видоизменяем объект, безопасность которого должна быть обеспечена.
Для примера рассматриваем примитивный процесс: Иванов составляет договоры и отправляет их на согласование Петрову.
С точки зрения компьютерной системы это один процесс формирует файл с договором, который затем преобразуется в электронное письмо и доставляется до почтовой программы.
И вот тут возникает проблема:

1. Отсутствует гарантия, что процесс, который создал файл с договором был действительно инициирован Ивановым (и весь сеанс работы с ним внешняя информация поступала от Иванова).
2. Далее в компьютерной системе идет пересылка контейнеров (файл, электронное письмо), но у нас абсолютно нет информации, что же внутри этих контейнеров – возможно, там совсем уже не договор, который должен был быть направлен Петрову
3. И, наконец, у нас нет гарантии, что почтовое приложение запущено Петровым и он получает из него информацию. Более того, зная техническую природу передачи данных, мы не можем гарантировать, что полученное [возможно] Петровым письмо, было получено только им. Не исключена ситуация, что это письмо было получено еще какими-то субъектами.

И вот из перечисленных проблем и можно сформировать основные задачи ИБ, решаемые в современных КС.

**Связь человека и субъекта КС**

Также упомянуть о проблеме вредоносного ПО – которое формально действует от имени легитимного пользователя, но не в его интересах.
Основные методы борьбы с ним: контроль целостности (белый список) и обнаружение вредоносного ПО (черный список). Рассказать о понятиях идентификации и аутентификации. Хорошим источником информации является ГОСТ, ссылка на который приведена ниже.
Ссылки: ГОСТ Р 58833-2020 Защита информации. Идентификация и аутентификация. Общие положения.

**Ограничение доступа**

На этапе доступа субъекта к объекту есть развитые в современных КС решения по авторизации, которые опираются на политику управления доступом. Тут также рекомендуется опираться на ГОСТ Р 58833-2020.
Ссылки: ГОСТ Р 58833-2020 Защита информации. Идентификация и аутентификация. Общие положения.

**Контроль информационных потоков**

Больше всего задач решается на этапе контроля информационных потоков. Так как потоков множество на разных уровнях архитектуры КС (внутри одного хоста, в масштабах локальной вычислительной сети, распределенной сети предприятия и даже всего мира – сети Интернет).
При этом задача решается в двух направлениях: не допустить входящего потока с опасной или вредной информацией (межсетевое экранирование, обнаружение вторжений, защита от нежелательной корреспонденции), а также не допустить исходящего потока критичной информации в сторону нелегитимного получателя (средства предотвращения утечек информации).

**Обеспечение конфиденциальной передачи**

При защите информационного потока от прослушивания (т.е. избегание ситуации, когда поток имеет несколько адресатов, часть из которых нелегитимна) также есть две стратегии:

1. Сокрытие самого факта передачи информации (стеганография)
2. Криптографическое преобразование передаваемой информации, чтобы нелегитимный получатель не мог извлечь из соответствующих контейнеров данных осмысленную информацию.

**Выявление инцидентов ИБ**

Но, как мы говорили ранее, помимо мер, направленных на противодействие реализации угрозы ИБ, нужно также обеспечить оперативное выявление признака реализации угрозы для последующей нейтрализации цепи атаки. Или, говоря в терминах модели, научиться выявлять переход в небезопасное состояние.
Основным методом для наблюдения за изменениями КС, важными с точки зрения ИБ, является регистрация и учет событий безопасности. При этом событие – это информация, на основе которой можно сделать вывод о том, в какое состояние перешла системы. Специальные средства автоматического анализа событий ИБ позволяют выявить по ним признаки инцидента (переход в небезопасное состояние).

**Если вовремя не выявили…**

И, если ничего не помогло, должны вступать в дело средства восстановления пострадавших от атаки активов.
Необходимо сделать ремарку, что восстановить нарушено свойство конфиденциальности невозможно – тут больше речь про целостность и доступность.
Если речь идет о целостности, то восстановить целостность актива поможет резервная копия. Если же речь о доступности – то современные отказоустойчивые архитектуры

позволят максимально быстро восстановить нормальный режим функционирования актива.

## Применение методов ИИ в ИБ

Естественно, что для всех перечисленных задач актуально применение современных алгоритмов машинного обучения и искусственного интеллекта.

В частности: классификация позволяет определять пользователя, например, по его клавиатурному почерку (непрерывная аутентификация), классификация информации позволяет «заглянуть» внутрь контейнера и более осознанно применять политики разграничения доступа, классификация сетевых узлов (по профилю их сетевой активности) позволяет выявить вредоносные сетевые узлы.

Также находят свое применение методы выявления аномалий: это и аномалии в потоке событий ИБ (что может говорить о признаках инцидента), и аномальное поведение пользователей и процессов (что может сигнализировать о внедрении вредоносного ПО) и аномалии в сетевой активности, что позволяет выявлять, кроме всего прочего, скрытые каналы утечки информации.

В последнее время также все чаще реальных операторов и диспетчеров заменяют программными алгоритмами, построенными на основе техник обучения в подкреплением – в вопросах ИБ также есть ряд задач, где нужна скорость реакции и принятия решений значительно выше возможностей человека. Например, при реагировании на обнаруженный инцидент ИБ.

## Примеры применения ИИ в ИБ

Обратить внимание, что технологии при этом могут использовать не только во благо, но и во вред.

Сбор информации (crawlers, scrappers): Озвучить варианты использования ИИ алгоритмов для обработки больших информационных массивов, которые представлены преимущественно на естественном языке.

Опять акцентировать внимание на том, что задачи очень схожи, несмотря на то, что в одном случае результат – это повышение защищенности, а в другом – совсем наоборот.

Моделирование (генерация) информации: Акцентировать внимание на генеративном ИИ. Эти технологии позволяют создавать сложные эмуляторы жертв для того, чтобы потом ловить на них нарушителей. Кроме того, генеративный ИИ может применяться для кодирования информации или как замена датчику случайных чисел.

Анализ трафика: Одна из самых популярных областей применения технологий ИИ – это анализ трафика. Количество и разнообразие трафика давно уже создало проблему понимания структуры и легитимности информационных потоков даже в небольших сетях передачи данных. При этом технологии ИИ могут решить многие задачи анализа трафика: автоматически строить правила глубокой инспекции пакетов, выявлять фрагменты атак или вредоносного кода в трафике, выявлять признаки ботнета в сети и пр. И опять все методы могут быть использованы и во вред – для сбора детальной информации об объекте атаки.

Анализ и защита кода: Тема безопасной разработки ПО крайне актуальна уже на протяжении последних 10 лет. Но в этой отрасли до сих пор множество проблем с выполнением стандартных процедур безопасной разработки. Технологии ИИ позволяют значительно снизить трудозатраты на различные виды проверок ПО, а где-то даже и автоматически корректировать написанный код, если он содержит ошибки и потенциальные уязвимости.

И снова злоумышленник может использовать инструмент во вред – то, что помогает найти слабые места в коде для их устранения, точно также поможет найти эти же места, но уже для использования их при проведении атаки.

Поведенческий анализ: Человек остается самым слабым звеном информационной системы, поэтому анализ поведения пользователя КС может говорить о многом. Как для нужд защиты, так и нападения.
Базы знаний, Threat Intelligence: Сфера ИБ долгое время оставалась и остается до сих пор не очень сильно автоматизированной сферой. По-прежнему множество ценной информации представлено только в человекочитаемом виде. И здесь также может помочь ИИ – методы анализа естественных языков позволяют в автоматическом режиме вобрать в систему обеспечения ИБ содержимое баз знаний.
Вопросы к экзамену по модулю «Основы ИБ»

| № | Вопрос |
|---|---|
| 1 | Выберите свойства информации, защиту которых должна обеспечить информационная безопасность:<br>1. Целостность, доступность, понятность.<br>2. Непротиворечивость, наглядность.<br>**3. Целостность, доступность, конфиденциальность.**<br>4. Конфиденциальность, малая емкость. |
| 2 | Выберите субъекты, защиту которых обеспечивает ИБ:<br>**1. Файлы**<br>**2. Базы данных**<br>**3. Данные о сотрудниках**<br>4. **Каналы передачи информации** |
| 3 | Выберите элементы, относящиеся к атаке на защищаемую информацию<br>**1. Источник угрозы**<br>**2. Уязвимость**<br>**3. Актив**<br>4. ФСТЭК |
| 4 | Что может являться причиной уязвимости в системе<br>**1. Ошибки в коде (программном обеспечении)**<br>**2. Ошибки в настройках оборудования и программного обеспечения**<br>3. Использование open-source ПО<br>**4. Неправильная организация работы с информацией** |
| 5 | Какие документы регламентируют сферу ИБ:<br>**1. Федеральные законы**<br>**2. Приказы ФСБ**<br>**3. Постановления правительства РФ**<br>4. Документация вендоров |

| | |
|---|---|
| 6 | Какие информационные системы используются на программно-техническом уровне обеспечения ИБ<br>**1. SIEM**<br>**2. Антивирусные программы**<br>**3. NG-FW**<br>4. CRM |
| 7 | Стеганография это<br>**1. Методы для сокрытия факта передачи информации**<br>2. Раздел высшей математики<br>3. Набор библиотек для криптографии<br>4. Защитная технология в блокчейне |
| 8 | Какие методы применяются для упреждения инцидентов<br>**1. Обновление антивирусных баз**<br>**2. Анализ данных об уязвимостях используемого ПО**<br>3. Резервирование данных<br>**4. Авторизация** |
| 9 | Выберите методы выявления инцидента:<br>1. Обновление антивирусных баз данных<br>**2. Анализ данных ИС ИБ (например, на основе правил)**<br>3. Резервное копирование данных<br>4. Организация VPN |
| 10 | Выберите методы, позволяющие минимизировать расходы на устранение инцидента:<br>1. Авторизация<br>**2. Резервное копирование данных**<br>**3. Резервирование архитектуры системы**<br>4. Обновление антивирусов |

**Сетевые технологии.**

Модель OSI. https://docs.cntd.ru/document/1200007766

Модель OSI в реальности. Или Department of Defense модель TCP/IP.

Модель OSI это идеальная модель, но избыточная. Была обобщена к модели TCP/IP. Начинать рассказ следует с 4 и 1 уровня – то, что студенты «ощущают» глазами. Далее, подниматься от 4 к 1 на конкретных примерах. Коммутатор, маршрутизатор, тип управления потоками данных и порты приложений.

https://youtu.be/BJSITWkSDQg

Network Access Layer. Погружение в работу канального уровня. Что такое MAC-адрес? Где он в оборудовании и в фрейме?

https://youtu.be/9eH16Fxeb9o?t=545

Network Access Layer. Развитие соединений устройств.
Internet. Что такое IP-адрес? Потребность в сетях, локальные и глобальные сети. Route table.
https://youtu.be/zeArDrC2xPg
https://docs.microsoft.com/en-us/troubleshoot/windows-client/networking/tcpip-addressing-and-subnetting
Internet. Как связаны IP и MAC? Устройство ARP-таблиц. Процесс ARP и RARP-запроса.
https://youtu.be/cn8Zxh9bPio
9. Internet. Визуализация Ucast, Bcast, Mcast. Устройство DHCP.
https://youtu.be/e6-TaH5bkjo
https://youtu.be/4pkDL1pgCgQ
https://youtu.be/S43CFcpOZSI
Transport. Различие управления пакетами при UDP и TCP. Устройство TCP. Потоки данных.
https://youtube.com/playlist?list=PLW8bTPfXNGdAZIKv-y9v_XLXtEqrPtntm
Application. Что такое порт, как он используется, стандартные протоколы по RFC 1340. Зачем нужен DNS? Устройство DNS LOOKUP.
https://youtu.be/72snZctFFtA
Application. Устройство SMTP. Различие POP3 и IMAP.
https://youtu.be/PJo5yOtu7o8
https://youtu.be/SBaARws0hy4
https://www.javatpoint.com/simple-mail-transfer-protocol
https://medium.com/@jonathansychan/smtp-simple-mail-transfer-protocol-ed443b1f51d7
https://medium.com/@imranshaikh2124/how-smtp-works-3c5aa6e81779
https://postmarkapp.com/guides/everything-you-need-to-know-about-smtp
Мониторинг. Варианты мониторинга сети. Устройство SNMP. Устройство Syslog.
http://math.gsu.by/wp-content/uploads/courses/networks/r7.3.html
https://www.securitylab.ru/analytics/301808.php
https://youtu.be/2IXP0TkwNJU
https://youtu.be/Lq7j-QipNrI?t=17
https://youtu.be/BMVHHX02T4Q?t=17
Основы анализа трафика. Краткий обзор инструментов для анализа. Пакетный и потоковый подход.
https://www.tcpdump.org/
https://www.wireshark.org/
https://www.ntop.org/
https://zeek.org/
https://github.com/phaag/nfdump
IDS vs IPS. Основное отличие и принцип работы.
https://youtu.be/rvKQtRklwQ4
https://youtu.be/_gHMkEKGwBM?t=56
DLP. Что такое DLP и зачем это необходимо.
https://youtu.be/S-5oLGrTHcU
https://youtu.be/jf-GSmiQZgw
https://youtu.be/-Jpec7tOQqM?t=262
(NextGen) SIEM. Событийный подход к данным безопасности. Инциденты. Расследование атак.
https://youtu.be/GbFtSDnPZBQ?t=75
https://youtu.be/sgMxl2dmkOQ

Выявление аномалий в сетевом трафике. Ошибки в передаче. Аномальное поведение систем.
https://izv.etu.ru/assets/files/izvestiya-4_2021-36-41.pdf
https://www.ussc.ru/news/novosti/na-chto-sposoben-ii-v-sfere-ib/
http://uzulo.su/prav-inf/pdf-jpg/pi-2020-1-st7-s67-75.pdf
Сетевые атаки. Некоторые виды атак, которые будут рассмотрены далее.
https://www.smart-soft.ru/blog/tipy_setevyh_atak_i_sposoby_borjby_s_nimi/
http://teacherbox.ru/kompseti/pm02/2-1-25-vidyi-i-tipyi-setevyih-atak/.html
Сетевые атаки (Malware). Виды Malware.
https://www.wallarm.com/what/malware-types-and-detection
Сетевые атаки (Phishing). Принцип атаки.
https://www.wallarm.com/what/types-of-phishing-attacks-and-business-impact
Сетевые атаки (SQL Injection Attack). Принцип атаки.
https://www.acunetix.com/blog/articles/exploiting-sql-injection-example/
https://www.wallarm.com/what/types-of-phishing-attacks-and-business-impact
Сетевые атаки (Cross-Site Scripting). Принцип атаки.
https://www.wallarm.com/what/what-is-xss-cross-site-scripting
Сетевые атаки (Denial of Service). Принцип атаки.
https://docs.aws.amazon.com/whitepapers/latest/aws-best-practices-ddos-resiliency/introduction-denial-of-service-attacks.html
https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/
https://www.coursera.org/lecture/it-security/denial-of-service-lH7Y7
Сетевые атаки (MitM). Принцип атаки.
https://www.varonis.com/blog/man-in-the-middle-attack/
https://www.wallarm.com/what/what-is-mitm-man-in-the-middle-attack
https://www.fortinet.com/resources/cyberglossary/man-in-the-middle-attack
https://www.apriorit.com/dev-blog/723-qa-mitm-tools-for-penetration-testing-and-cybersecurity-enhancement
Сетевые атаки (ARP spoofing). Принцип атаки.
Сетевые атаки (DNS spoofing). Принцип атаки.
Вопросы к экзамену по модулю «Сетевые технологии»

| № | Вопрос |
|---|---|
| 1 | До какого уровня модели TCP/IP коммутатор разбирает фрейм?<br>1. Приложения<br>2. Транспортного<br>3. Сети Интернет<br>4. **Доступа к сети** |
| 2 | Какой уровень OSI не включает уровень приложений TCP/IP?<br>1. Прикладной<br>2. Сеансовый<br>**3.** Представления<br>**4.** **Канальный** |

| | |
|---|---|
| 3 | Какие устройства из перечисленных имеют MAC-адрес?<br>1. Сетевая карта конечного устройства<br>2. Коммутатор<br>3. Маршрутизатор<br>**4. Все из перечисленного** |
| 4 | Что из перечисленного работает по типу общей шины?<br>**1. Концентратор**<br>2. Коммутатор<br>3. Маршрутизатор<br>4. Wi-fi роутер |
| 5 | Какая фраза не относится к частному IP-адресу?<br>1. Назначает администратор сети<br>2. Используется для связи в одной сети<br>**3. Маршрутизируемый**<br>4. Бесплатный |
| 6 | Что транслирует компьютер, чтобы узнать неизвестный MAC-адрес?<br>1. Ничего из перечисленного<br>2. ICMP-запрос<br>**3. ARP-запрос**<br>4. ICMP-ответ<br>5. ARP-ответ |
| 7 | Что получает DHCP-клиент?<br>1. MAC-адрес<br>**2. Настройки сети**<br>3. DHCPDiscover<br>4. Ничего из перечисленного |
| 8 | Какие выражения верны?<br>1. TCP имеет 8-битный заголовок.<br>2. UDP имеет гарантированную передачу.<br>**3. В UDP нет механизма проверки ошибок.**<br>**4. TCP — это безопасный протокол, UDP не является безопасным.**<br>**5. TCP медленнее, чем UDP.** |
| 9 | Порт дает понимание:<br>1. Какой IP-адрес использовать<br>2. Тип протокола: UDP или TCP<br>**3. Какому приложению направлен пакет** |
| 10 | Какое из устройств не блокирует сетевые пакеты:<br>**1. IDS**<br>2. IPS<br>3. Firewall |

| | |
|---|---|
| 11 | Атака, нацеленная на доступность:<br>1. Phishing<br>2. MitM<br>3. XSS<br>**4. DoS** |
| 12 | Подмена доменного имени сайта это:<br>1. ARP spoofing<br>2. Cross-Site Scripting<br>3. Malware<br>4. **DNS spoofing** |

## Генетические алгоритмы и информационная безопасность

Определение. Генетический алгоритм — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе.

Генетические алгоритмы реализуют упрощенный вариант дарвиновской эволюции. Имитируя процессы естественного отбора и воспроизводства, генетические алгоритмы могут находить высококачественные решения задач, включающих поиск, оптимизацию и обучение. В то же время аналогия с естественным отбором позволяет этим алгоритмам преодолевать некоторые препятствия, встающие на пути традиционных алгоритмов поиска и оптимизации, особенно в задачах с большим числом параметров и сложными математическими представлениями.

Цель генетических алгоритмов – найти оптимальное решение некоторой задачи. Если дарвиновская эволюция развивает популяцию отдельных особей, то генетические алгоритмы развивают популяцию потенциальных решений данной задачи, называемых индивидуумами. Эти решения итеративно оцениваются и используются для создания нового поколения решений. Те, что лучше проявили себя при решении задачи, имеют больше шансов пройти отбор и передать свои качества следующему поколению. Так постепенно потенциальные решения совершенствуются в решении поставленной задачи.

В случае генетических алгоритмов каждому индивидууму соответствует хромосома, представляющая набор генов. Например, хромосому можно представить двоичной строкой, в которой каждый бит соответствует одному гену.

### Определение популяции

В любой момент времени генетический алгоритм хранит популяцию индивидуумов – набор потенциальных решений поставленной задачи. Поскольку каждый индивидуум представлен некоторой хромосомой, эту популяцию можно рассматривать как коллекцию хромосом. Популяция всегда представляет текущее поколение и эволюционирует со временем, когда текущее поколение заменяется новым.

### Создание начальной популяции

Начальная популяция состоит из случайным образом выбранных потенциальных решений (индивидуумов). Поскольку в генетических алгоритмах индивидуумы представлены хромосомами, начальная популяция – это, по сути дела, набор хромосом. Формат хромосом должен соответствовать принятым для решаемой задачи правилам, например это могут быть двоичные строки определенной длины.

Если имеется априорная информация, то для увеличения скорости сходимости поиска она может быть использована при формировании начальной популяции

### Функция приспособленности

Разбор определения. Для каждого индивидуума вычисляется функция приспособленности. Это делается один раз для начальной популяции, а затем для каждого нового поколения после применения операторов отбора, скрещивания и мутации. Поскольку приспособленность любого индивидуума не зависит от всех остальных, эти вычисления можно производить параллельно. Так как на этапе отбора, следующем за вычислением приспособленности, более приспособленные индивидуумы обычно считаются лучшими решениями, генетические алгоритмы естественно «заточены» под нахождение максимумов функции приспособленности. Если в какой-то задаче нужен минимум, то при вычислении приспособленности следует инвертировать найденное значение, например умножив его на –1.

**Отбор. Правило рулетки**

После того как вычислены приспособленности всех индивидуумов в популяции, начинается процесс отбора, который определяет, какие индивидуумы будут оставлены для воспроизводства, т. е. создания потомков, образующих следующее поколение. Процесс отбора основан на оценке приспособленности индивидуумов. Те, чья оценка выше, имеют больше шансов передать свой генетический материал следующему поколению. Плохо приспособленные индивидуумы все равно могут быть отобраны, но с меньшей вероятностью. Таким образом, их генетический материал не полностью исключен.

Рассмотрим некоторые методы реализации отбора. Правило рулетки.

Метод отбора по правилу рулетки, или отбор пропорционально приспособленности (fitness proportionate selection – FPS), устроен так, что вероятность отбора индивидуума прямо пропорциональна его приспособленности. Тут можно провести аналогию с вращением колеса рулетки, где каждому индивидууму соответствует сектор, стоимость которого равна приспособленности индивидуума. Шансы, что шарик остановится в секторе индивидуума, пропорциональны размеру этого сектора. Пусть, например, имеется популяция из шести индивидуумов с такими значениями приспособленности, как в таблице ниже. По этим значениям вычисляются доли, занимаемые секторами каждого индивидуума.

После каждого запуска рулетки отбор индивидуума из популяции производится в точке отбора. Затем рулетка запускается еще раз для выбора следующего индивидуума, и так до тех пор, пока не наберется достаточно индивидуумов для образования следующего поколения. В результате один и тот же индивидуум может быть выбран несколько раз.

**Отбор. Стохастическая универсальная выборка**

Стохастическая универсальная выборка (stochastic universal sampling – SUS) – немного модифицированный вариант правила рулетки. Используется та же рулетка с такими же секторами, но вместо одной точки отбора и многократного запуска рулетки мы вращаем колесо только один раз, а отбор индивидуумов производим в нескольких точках, равномерно расставленных по окружности. Тем самым все индивидуумы выбираются одновременно, как показано на рисунке.

Этот метод отбора не дает индивидуумам с особенно высокой приспособленностью заполнить все следующее поколение в результате повторного выбора. Поэтому более слабым индивидуумам предоставляется шанс, а несправедливость чистого правила рулетки в какой-то мере сглаживается.

**Отбор. Ранжированный отбор**

Метод ранжированного отбора похож на правило рулетки, но значения приспособленности используются не для вычисления вероятностей выбора, а просто для сортировки индивидуумов. После сортировки каждому индивидууму назначается ранг, соответствующий его позиции в списке, а вероятности секторов рулетки вычисляются на основе этих рангов. Возьмем ту же самую популяцию из шести индивидуумов, что и

раньше. И добавим в таблицу столбец с рангом индивидуума. Поскольку размер популяции равен 6, наивысший возможный ранг тоже равен 6, следующий по порядку – 5 и т. д. Каждому индивидууму сопоставляется сектор рулетки, вычисленный по этим рангам, а не по значениям функции приспособленности.

Ранжированный отбор полезен, когда есть несколько индивидуумов, гораздо лучше приспособленных, чем все остальные. Использование ранга вместо самой приспособленности мешает этим индивидуумам захватить всю популяцию в следующем поколении, поскольку ранжирование сглаживает значительные различия. Кроме того, когда все индивидуумы обладают почти одинаковой приспособленностью, ранжированный отбор позволяет разделить их, отдавая преимущество лучшим, даже когда различия малы.

**Отбор. Турнирный отбор**

В каждом раунде турнирного отбора из популяции выбираются два или более индивидуумов, и тот, у кого приспособленность больше, выигрывает и отбирается в следующее поколение. Рассмотрим тех же индивидуумов с такими же приспособленностями, что и ранее. На рисунке показан результат случайного выбора трех из них (A, B и F) с последующим объявлением F победителем, поскольку у него приспособленность максимальная из трех.

Количество индивидуумов, участвующих в каждом раунде турнирного отбора (в нашем примере – три), называется размером турнира. Чем больше размер турнира, тем выше шансы, что в раундах будут участвовать лучшие индивидуумы, и тем меньше шансов у слабых участников победить в турнире и отобраться. У этого метода отбора есть интересная особенность: если мы умеем сравнивать любых двух индивидуумов и определять, какой из них лучше, то сами значения функции приспособленности и не нужны.

**Скрещивание**

Для создания пары новых индивидуумов родители обычно выбираются из текущего поколения, а части их хромосом меняются местами (скрещиваются), в результате чего создаются две новые хромосомы, представляющие потомков. Эта операция называется скрещиванием, или рекомбинацией.

Как правило, оператор скрещивания применяется не всегда, а с некоторой (высокой) вероятностью. Если скрещивание не применяется, то копии обоих родителей переходят в следующее поколение без изменения.

Простейший способ – одноточечное скрещивание.

В этом случае позиция в хромосомах обоих родителей выбирается случайным образом. Эта позиция называется точкой скрещивания, или точкой разреза. Гены одной хромосомы, расположенные справа от этой точки, обмениваются с точно так же расположенными генами другой хромосомы. В результате мы получаем двух потомков, несущих генетическую информацию обоих родителей.

**Двухточечное и k-точечное скрещивание**

При двухточечном скрещивании случайным образом выбираются по две точки скрещивания в каждой хромосоме. Гены одной хромосомы, расположенные между этими точками, обмениваются с точно так же расположенными генами другой хромосомы. Метод двухточечного скрещивания можно реализовать с помощью двух одноточечных скрещиваний с разными точками скрещивания. Его обобщением является метод k-точечного скрещивания, где k – целое положительное число.

**Равномерное скрещивание**

При равномерном скрещивании каждый ген обоих родителей определяется независимо путем случайного выбора с равномерным распределением. Когда выбирается 50 % генов, оба родителя имеют одинаковые шансы повлиять на потомков. Заметим, что в

этом примере гены обоих потомков меняются местами, но в принципе потомков можно создавать и независимо. Поскольку в этом методе не производится обмен целых участков хромосом, потенциально он может повысить разнообразие потомков.
Существуют и другие методы, но разобрать все в рамках лекции не успеть. Более того, можно придумывать свои методы скрещивания.

**Мутация**

Цель оператора мутации – периодически случайным образом обновлять популяцию, т. е. вносить новые сочетания генов в хромосомы, стимулируя тем самым поиск в неисследованных областях пространства решений. Операция мутации вероятностная, обычно она выполняется изредка, с очень низкой вероятностью, поскольку может ухудшить качество индивидуума, к которому применена. В некоторых вариантах генетических алгоритмов вероятность мутации постепенно увеличивается, чтобы предотвратить стагнацию и повысить разнообразие популяции. С другой стороны, если частота мутации слишком велика, то генетический алгоритм выродится в случайный поиск.
Мутация может проявляться как случайное изменение гена. Мутации реализуются с помощью внесения случайных изменений в значения хромосом, например инвертирования одного бита в двоичной строке (отображено на слайде).
Другой пример – мутация обменом.
Мутация обращением – изменение порядка генов на противоположный
Мутация перетасовкой - в этом случае выбирается случайная последовательность генов, и порядок генов в ней изменяется случайным образом (тасуется).
Но не забывайте, что вы всегда можете придумать свой собственный метод, отвечающий специфике конкретной задачи.

**Критерии останова**

Может существовать несколько условий, при выполнении которых процесс останавливается. Сначала отметим два самых распространенных:
· достигнуто максимальное количество поколений;
· достигнуто целевое значение функции приспособленности.
Перечислим также другие возможные условия:
· с момента начала прошло заранее определенное время;
· превышен некоторый лимит затрат, например процессорного времени или памяти;
· наилучшее решение заняло часть популяции, большую заранее заданного порога;
· на протяжении нескольких последних поколений не наблюдается заметных улучшений. Это можно реализовать путем запоминания наилучшей приспособленности, достигнутой в каждом поколении, и сравнения наилучшего текущего значения со значениями в нескольких предыдущих поколениях. Если разница меньше заранее заданного порога, то алгоритм можно останавливать.

**Теоретическая основа – понятие схемы**

Пусть используется простой генетический алгоритм, то есть ГА с одноточечным кроссинговером и одноточечной мутацией. Будем считать, что особью в популяции является бинарная строка длины l. Если это не так, всегда можно закодировать ее нужным образом. Разбираем определение схемы, порядка схемы и определяющей длины схемы. Подкрепляем все примерами со слайда. Каждая хромосома в популяции соответствует нескольким схемам – точно так же, как заданная строка соответствует разным регулярным выражениям.
Подробнее - http://www.machinelearning.ru/wiki/index.php?title=%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D1%81%D1%85%D0%B5%D0%BC%D1%8B

**Теоретическая основа – теорема Холланда о схемах**

Суть теоремы. Теорема о схемах утверждает, что частота схем низкого порядка с малым определяющим расстоянием и приспособленностью выше средней экспоненциально возрастает в последующих поколениях. Иными словами, генетический алгоритм увеличивает частоту в популяции небольших и простых структурных элементов, представляющих атрибуты, благодаря которым решение становится лучше.

Подробнее - http://www.machinelearning.ru/wiki/index.php?title=%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D1%81%D1%85%D0%B5%D0%BC%D1%8B

**Отличия ГА от традиционных алгоритмов**

Популяция решений. Целью генетического поиска является популяция потенциальных решений (индивидуумов), а не единственное решение. В любой точке поиска алгоритм сохраняет множество индивидуумов, образующих текущее поколение. На каждой итерации генетического алгоритма создается следующее поколение индивидуумов. С другой стороны, в большинстве других алгоритмов поиска хранится единственное решение, которое итеративно улучшается. Например, алгоритм градиентного спуска итеративно сдвигает текущее решение в направлении наискорейшего спуска, которое определяется антиградиентом заданной функции.

Генетическое представление. Генетические алгоритмы работают не с самими потенциальными решениями, а с их кодированными представлениями, которые часто называют хромосомами. Простым примером хромосомы является двоичная строка фиксированной длины. Хромосомы позволяют определить генетические операции скрещивания и мутации. Скрещивание реализуется обменом частей родительских хромосом, а мутация – изменением частей хромосом. Побочный эффект генетического представления – отделение поиска от исходной предметной области. Генетические алгоритмы не знают, что именно представляют хромосомы, и не пытаются их интерпретировать.

Функция приспособленности. Функция приспособленности представляет проблему, которую мы пытаемся решить. Цель генетического алгоритма – найти индивидуумов, для которых оценка, вычисляемая функцией приспособленности, максимальна. В отличие от традиционных алгоритмов поиска, генетические алгоритмы анализируют только значение, возвращенное функцией приспособленности, их не интересует ни производная, ни какая-либо другая информация. Поэтому они могут работать с функциями, которые трудно или невозможно продифференцировать.

Вероятностное поведение. Многие традиционные алгоритмы по природе своей детерминированы, тогда как правила, применяемые генетическими алгоритмами для перехода от предыдущего поколения к следующему, вероятностные. Например, вероятность отбора индивидуума для создания следующего поколения тем выше, чем больше значение функции приспособленности, но элемент случайности все равно присутствует. Слабо приспособленные индивидуумы могут быть отобраны, хотя вероятность этого ниже. Мутации тоже имеют вероятностный характер, обычно их вероятность мала, а изменению подвергаются случайные позиции в хромосоме. Случайность присутствует и в операторе скрещивания. В некоторых генетических алгоритмах скрещивание происходит лишь с некоторой вероятностью. Если скрещивания не было, то оба родителя дублируются в следующем поколении вообще без изменений. Несмотря на вероятностную природу процесса, поиск, основанный на генетическом алгоритме, нельзя назвать случайным; случайность используется, чтобы направить поиск в сторону тех областей пространства поиска, где выше шансы улучшить результаты. Теперь рассмотрим преимущества генетических алгоритмов.

**Преимущества генетических алгоритмов**

Особенности генетических алгоритмов, рассмотренные в предыдущих разделах, определяют их преимущества по сравнению с традиционными алгоритмами поиска.

Глобальная оптимизация. Во многих задачах оптимизации имеются точки локального максимума и минимума, которые представляют решения, лучшие, чем те, что находятся поблизости, но необязательно лучшие в глобальном смысле. Большинство традиционных алгоритмов поиска и оптимизации, а особенно те, что основаны на вычислении градиента, могут застревать в локальном максимуме, вместо того чтобы найти глобальный. Это связано с тем, что в окрестности локального максимума всякое небольшое изменение решения ухудшает оценку. Генетические алгоритмы менее подвержены этой напасти и имеют больше шансов отыскать глобальный максимум. Объясняется это тем, что используется популяция потенциальных решений, а не единственное решение, а операции скрещивания и мутации зачастую порождают решения, далеко отстоящие от ранее рассмотренных. Это остается справедливым при условии, что мы поддерживаем разнообразие популяции и избегаем преждевременной сходимости, о чем поговорим далее.

Применимость к сложным задачам. Поскольку генетическим алгоритмам нужно знать только значение функции приспособленности каждого индивидуума, а все остальные ее свойства, в частности производные, несущественны, их можно применять к задачам со сложным математическим представлением, включающим функции, которые трудно или невозможно продифференцировать. К сложным случаям, когда достоинства генетических алгоритмов раскрываются во всем блеске, относятся также задачи с большим числом параметров или со смешанными параметрами, например непрерывными и дискретными.

Применимость к задачам, не имеющим математического представления. Генетические алгоритмы применимы и к задачам, вообще не имеющим математического представления. Один из таких случаев, представляющий особый интерес, – когда оценка приспособленности основана на мнении человека. Пусть, например, требуется найти наиболее привлекательную цветовую палитру для веб-сайта. Мы можем попробовать разные комбинации цветов и попросить пользователей оценить привлекательность сайта. А затем применить генетический алгоритм, чтобы найти лучшую комбинацию, используя функцию приспособленности, основанную на оценках пользователей. Алгоритм будет работать, несмотря на то что никакого математического представления нет и невозможно вычислить оценку заданной комбинации непосредственно.

Устойчивость к шуму. Для некоторых задач характерно присутствие шума. Это означает, что даже при близких истинных значениях входных параметров результаты их измерений могут довольно сильно различаться. Например, так бывает, когда данные считываются с датчиков или когда оценка основана на мнении человека. Подобное поведение может сделать непригодными многие традиционные алгоритмы поиска, но генетические алгоритмы в общем случае устойчивы к нему благодаря повторяющимся операциям сборки и оценивания индивидуумов.

Распараллеливание. Генетические алгоритмы хорошо поддаются распараллеливанию и распределенной обработке. Функция приспособленности независимо вычисляется для каждого индивидуума, а это значит, что все индивидуумы в популяции могут обрабатываться одновременно. Кроме того, операции отбора, скрещивания и мутации могут одновременно выполняться для индивидуумов и пар индивидуумов. Поэтому подход, основанный на генетических алгоритмах, естественно адаптируется к распределенным и облачным реализациям.

Непрерывное обучение. Если окружающие условия изменяются, популяция приспосабливается к ним. Так и генетические алгоритмы могут непрерывно работать в постоянно изменяющихся условиях, и мы всегда можем получить и использовать

лучшее на данный момент решение. Но это возможно, только если окружающая среда изменяется медленно по сравнению со скоростью смены поколений в генетическом алгоритме.

## Ограничения генетических алгоритмов

Чтобы получить максимум пользы от генетических алгоритмов, мы должны знать об их ограничениях и потенциальных подвохах.

Специальные определения. Пытаясь применить генетические алгоритмы к некоторой задаче, мы должны создать подходящее представление – определить функцию приспособленности и структуру хромосом, а также операторы отбора, скрещивания и мутации. Зачастую это совсем не просто и занимает много времени.

Настройка гиперпараметров. Поведение генетических алгоритмов контролируется набором гиперпараметров, например размером популяции и скоростью мутации. Точных правил для выбора значений гиперпараметров не существует. Однако так обстоит дело практически со всеми алгоритмами поиска и оптимизации. Опыт – основной пункт к подбору разумных гиперпараметров.

Большой объем счетных операций. Работа с потенциально большими популяциями и итеративный характер генетических алгоритмов обусловливают большой объем вычислений, поэтому на получение приемлемого результата может уйти много времени. Проблему можно сгладить за счет хорошего выбора гиперпараметров, распараллеливания и в некоторых случаях кеширования промежуточных результатов.

Преждевременная сходимость. Если приспособленность какого-то индивидуума гораздо больше, чем у всей остальной популяции, то не исключено, что он продублируется так много раз, что в конечном счете, кроме него, в популяции ничего не останется. В результате генетический алгоритм может застрять в локальном максимуме и не найдет глобального. Чтобы предотвратить такое развитие событий, важно поддерживать разнообразие популяции.

Отсутствие гарантированного решения. Использование генетических алгоритмов не гарантирует нахождения глобального максимума. Однако это типично для всех алгоритмов поиска и оптимизации, если только у задачи не существует аналитического решения.

## Для решения каких задач подходят ГА?

Резюмируя изложенное в предыдущих слайдах, можно сказать, что генетические алгоритмы лучше применять для решения следующих задач.

Задачи со сложным математическим представлением. Поскольку генетическим алгоритмам нужно знать только значение функции приспособленности, их можно использовать для решения задач, в которых целевую функцию трудно или невозможно продифференцировать, задач с большим количеством параметров и задач с параметрами разных типов.

Задачи, не имеющие математического представления. Генетические алгоритмы не требуют математического представления задачи, коль скоро можно получить значение оценки или существует метод сравнения двух решений.

Задачи с зашумленной окружающей средой. Генетические алгоритмы устойчивы к зашумленным данным, например прочитанным с датчика или основанным на оценках, сделанных человеком.

Задачи, в которых окружающая среда изменяется во времени. Генетические алгоритмы могут адаптироваться к медленным изменениям окружающей среды, поскольку постоянно создают новые поколения, приспосабливающиеся к изменениям.

## Фреймворк DEAP

Для работы с генетическими алгоритмами создан целый ряд фреймворков на Python, например GAFT, Pyevolve и PyGMO. Но мы остановимся на фреймворке DEAP,

поскольку он прост в использовании и предлагает широкий набор функций, поддерживает расширяемость и может похвастаться подробной документацией. DEAP (сокращение от Distributed Evolutionary Algorithms in Python – распределенные эволюционные алгоритмы на Python) поддерживает быструю разработку решений с применением генетических алгоритмов и других методов эволюционных вычислений. DEAP предлагает различные структуры данных и инструменты, необходимые для реализации самых разных решений на основе генетических алгоритмов.
Рассмотрим два основные модуля toolbox и creator. Подробнее о фреймфорке и его возможностях - https://deap.readthedocs.io/en/master/

**Примеры применения в ИБ**

Одной из задач при построении системы защиты информации является составление набора защитных мер. При этом в необходимо учитывать не только требования безопасности, но и затраты на внедрение и поддержание системы защиты информации. Обычно при составлении набора защитных мер используется метод, основанный на экспертном подходе, который сводится к предпочтениям экспертов, но так как количество возможных вариантов конфигурации системы защиты превышает мыслительные возможности, то в итоге к типовым схемам. В качестве критериев используются суммарная величина риска и стоимость системы защиты информации.
Хромосома – набор защитных мер, закодированных в форме двоичного числа.
Функция приспособленности - оценка эффективности системы защиты для заданного профиля атаки с учетом ее стоимости, которую необходимо максимизировать.
Оценка рисков выполняется с помощью специальной модели (на слайде приведен пример таблицы для расчета рисков).
Мутация – случайное инвертирование двух бит.
Начальная популяция – два случайных индивидуума
Селекция – отбор K лучших.
Детальный пример: http://itids.ugatu.su/index.php/itids/itids2018/paper/view/51

**Примеры применения в ИБ**

Формирует именно набор правил, а не одно правило, выявляющее все атаки (такое правило не достижимо).
Описание алгоритма: https://www.researchgate.net/publication/288345210_Using_Genetic_Algorithm_in_Network_Security
Задача: выявить какие параметры конфигурации ведут к тем или иным уязвимостям. Существуют алгоритмы, которые позволяют на основе уязвимых конфигураций выявлять параметры, которые ведут к тем или иным уязвимостям. Для качественной работы алгоритма нужно получить большой набор уязвимых конфигураций. Эту задачу можно решить с использованием генетических алгоритмов.
Подробнее - https://wakespace.lib.wfu.edu/bitstream/handle/10339/59317/Odell_wfu_0248M_10904.pdf
Пример шифрования изображения из статьи https://docsdrive.com/pdfs/ansinet/itj/2006/516-519.pdf
Подробнее познакомимся в рамках практического занятия

**Перечень рекомендованной литературы.**


1. Э. Версански. Генетические алгоритмы на Python
2. Описание теоремы о схемах на портале machine learning.ru http://www.machinelearning.ru/wiki/index.php?title=%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D1%81%D1%85%D0%B5%D0%BC%D1%8B
3. Mohammed A.F. Al- Husainy. Image Encryption Using Genetic Algorithm. https://docsdrive.com/pdfs/ansinet/itj/2006/516-519.pdf

4. C.A. Odell. Using genetic algorithms to detect security related software parameter chains. A Thesis for the Degree of MASTER OF SCIENCE.
5. B. Ehab, H. O. Nasereddin, Hebah. (2010). Using Genetic Algorithm in Network Security. https://www.researchgate.net/publication/288345210_Using_Genetic_Algorithm_in_Network_Security
6. С. О. Иванов, Д. В. Ильин, Л. А. Ильина. Генетический алгоритм подбора оптимальной конфигурации системы защиты информации. http://itids.ugatu.su/index.php/itids/itids2018/paper/view/51
7. Официальная страница фреймворка DEAP. https://deap.readthedocs.io/en/master/

Вопросы к экзамену по модулю «Генетические алгоритмы и информационная безопасность»

| № | Вопрос |
|---|---|
| 1 | Выберите наиболее правильное определение генетического алгоритма:<br>1. Метод машинного обучения на основе обучения среды, которая взаимодействует с агентом<br>**2. Эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе**<br>3. Метод машинного обучения на основе обучения интеллектуального агента, который действует во внешней среде<br>4. Марковский процесс принятия решений с конечным множеством состояний |
| 2 | Выберите корректные методы «скрещивания»:<br>**1. Трехточечное скрещивание**<br>2. **Равномерное скрещивание**<br>3. Инвертированное скрещивание<br>4. Скрещивание перетасовкой |
| 3 | Чему равен порядок схемы:?<br>1. 1<br>2. 2<br>**3. 4**<br>4. 5 |
| 4 | Чему равна определяющая длина схемы ?<br>1. 1<br>2. 2<br>**3. 4**<br>4. 5 |

| | |
|---|---|
| 5 | Какой модуль фреймворка DEAP используется для создания класса особи?<br>1. Individual<br>2. PopulationCreator<br>3. Toolbox<br>**4. Creator** |
| 6 | Выберите наиболее важные требования для функции приспособленности:<br>1. Функция приспособленности должна быть целочисленной<br>**2. Соответствие предметной области**<br>3. Функция приспособленности должна иметь «ступенчатый» характер<br>**4. Вычислимость для всех элементов популяции** |
| 7 | Выберите корректные методы отбора?<br>**1. Правило рулетки**<br>2. Пропорциональный отбор<br>**3. Ранжированный отбор**<br>4. **Стохастическая универсальная выборка** |
| 8 | Какое количество одноточечных скрещиваний нужно для реализации k-точечного:<br>**1. k**<br>2. $k^2$<br>3. k/2<br>4. k-точечное скрещивание невозможно осуществить с применением одноточечного скрещивания |
| 9 | Какие ограничения характерны для генетических алгоритмов?<br>1. Обязательное отсутствие шума во входных данных<br>2. Задача должна иметь математическое представление<br>**3. Отсутствие гарантированного решения**<br>**4. Опасность преждевременной сходимости** |
| 10 | Выберите верные утверждения:<br>1. Операция скрещивания выполняется до отбора.<br>**2. Априорные знания могут быть заложены в начальную популяцию.**<br>**3. Если для задачи известен аналитический алгоритм решения, то вполне вероятно, что он окажется эффективнее, чем генетический алгоритм**<br>4. Для любой задачи, решаемой генетическим алгоритмом, должно быть задано только одно условие останова |

**Методы классификации из ML в ИБ**

Разница между методами машинного обучения (МО) с учителем и без, базовые алгоритмы классификации, применение алгоритмов классификации в информационной безопасности. На практическом занятии: анализ спама (классификация)

## Машинное обучение

Привести примеры определений МО (на слайде). Артур Самуэль - изобретатель первой самообучающейся компьютерной программы игры в шашки. Том Митчелл - американский учёный, основатель первой в мире кафедры машинного обучения и автор первого учебника по этому предмету. «МО – это изучение компьютерных алгоритмов, которые могут автоматически улучшаться благодаря опыту и использованию данных».
Методы МО делятся на методы без учителя и с учителем. Принципиально отличаются наличием (или отсутствием) «истинно-верных» ответов при обучении модели.
https://docs.microsoft.com/ru-ru/dotnet/machine-learning/resources/tasks

**Обучение с учителем: регрессия, классификация (общие подходы)**

Классификация – предсказание одного из конечного списка классов, регрессия – предсказание одного из бесконечного списка классов. Опирается на шаблоны данных, а не на явное программирование.
https://habr.com/ru/company/ods/blog/322534/

## Классификация: деревья принятия

Дерево принятия решений — это двоичное дерево, каждый узел которого - входная переменная и точка разделения для этой переменной (если переменная — число). Листовые узлы — это выходная переменная, которая предсказывается. Классификация выполняется с помощью прохода по дереву к листовому узлу. Смысл обучения модели – получение условий и весов для узлов.
Модель умеет только интерполировать, но не экстраполировать. То есть дерево решений делает константный прогноз для объектов, находящихся в признаковом пространстве вне параллелепипеда, охватывающего все объекты обучающей выборки.
Преимущества:

- ¾ интерпретируемость: правила классификации понятны человеку
- ¾ быстрое обучение и предсказание

Недостатки:

- ¾ чувствительны к шумам
- ¾ может только интерполировать, но не экстраполировать

Графическое представление работы – на слайде.
https://habr.com/ru/company/ods/blog/322534/

## Классификация: K-ближайших соседей

Метрический метод классификации. Объект присваивается тому классу, который является наиболее распространённым среди k соседей данного элемента, классы которых уже известны
Преимущества:

- ¾ интерпретируемость: правила классификации понятны человеку
- ¾ устойчив к выбросам (голосование)

Недостатки:

- ¾ вычислительно сложный (используются все доступные данные)
- ¾ зависит от выбранной метрики расстояния
- ¾ нет теоретических оснований выбора определенного числа соседей (на практике берут нечётные)

Применять, когда много признаков (проклятие размерности).
Графическое представление работы – на слайде.
https://habr.com/ru/company/ods/blog/322534/

**Классификация: метод опорных векторов**

Пространство разделяется многомерными плоскостями, отделяющими области разных классов

Преимущества:

- хорошо работает в условиях большого количества признаков
- может обучаться на небольших данных

Недостатки:

- долго обучается на больших данных
- чувствителен к шуму
- нет общепринятого метода подбора ядер (для плоскостей сложной формы)

Графическое представление работы – на слайде.

https://habr.com/ru/company/ods/blog/484148/

**Классификация: байесовский классификатор**

Отдельные признаки рассматриваются как независимые друг от друга. С использование теоремы Байеса апостериорные вероятности описываются через априорные.

Преимущества:

- не требует подбора гиперпараметров
- работает быстро

Недостатки:

- не обнаруживает категории, отсутствовавшие при обучении
- предположение о независимости признаков

Пример применения показать на слайде (разобран случай определения вероятности исхода)

https://labelme.medium.com/наивный-байесовский-классификатор-naive-bayes-classifier-b939578f6e

http://datareview.info/article/6-prostyih-shagov-dlya-osvoeniya-naivnogo-bayesovskogo-algoritma-s-primerom-koda-na-python/

https://linis.hse.ru/data/2016/03/23/1128107554/models-probability.pdf

**Методы регрессии: линейная регрессия**

Модель зависимости непрерывной переменной от одной или нескольких других переменных с линейной функцией зависимости

Преимущества:

- Скорость и простота обучения
- Интерпретируемость: по коэффициентам можно понять влияние факторов

Недостатки:

- Чувствительна к выбросам
- Неэффективна при нелинейной зависимости целевой переменной от входной

Графическое представление работы – на слайде.

https://neurohive.io/ru/osnovy-data-science/linejnaja-regressija/

**Методы регрессии: логистическая регрессия**

Классификация, а не регрессия (несмотря на название)

Решает задачу предсказания значения непрерывной переменной, принимающей значения от 0 до 1: вероятность класса P=1/(1+□^(□(□, □)) ), где X − признаки, b – коэффициенты

Преимущества:

- Скорость и простота обучения

- Интерпретируемость: по коэффициентам можно понять влияние факторов

Недостатки:

- Чувствительна к выбросам
- Неэффективна при нелинейной зависимости целевой переменной от входной

Графическое представление работы – на слайде.
https://habr.com/ru/company/io/blog/265007/

**Ансамблирование моделей**

Обучаются несколько базовых моделей, затем их результаты объединяются по какому-либо правилу и формулируется окончательный результат

Если использование простой модели машинного обучения не даёт достаточно качественного результата, то можно объединить несколько моделей. Ткой метод называется ансамблированием моделей. Основные варианты ансамблирования – бэггинг, бустинг, стекинг.

Бэггинг – это способ объединения нескольких моделей, при котором из исходных данных формируются выборки, каждая из которых содержит неполный набор признаков. На каждой выборке обучается одна модель, а при классификации каждая из них делает свой выбор в пользу одного из классов. Итоговое решение принимается с помощью усреднения набора решений всех моделей.

Бустинг – это способ объединения нескольких моделей, при котором каждая модель добавляет к исходным признакам своё решение, и следующая модель работает уже с дополненными исходными данными.

Стэкинг - это способ объединения нескольких моделей, при котором каждая модель принимает своё решение о классе объекта, а затем с помощью ещё одной модели, принимающей на вход уже ответы всех предыдущих моделей, выбирается итоговый класс объекта.

Есть методы AutoML, Рассказать про TPOT. http://epistasislab.github.io/tpot/

Преимущества:

- точнее, чем отдельная модель

Недостатки:

- Сложная архитектура требует больше времени на разработку
- Сложность интерпретации

https://www.soa.org/globalassets/assets/files/e-business/pd/events/2020/predictive-analytics-4-0/pd-2020-09-pas-session-014.pdf

**Применение МО (классификации) в ИБ**

В ИБ МО может применяться в случаях, когда сложно выявить детерминированные правила для работы системы – например, в случае, когда система сложна для понимания, или есть повышенные требования к скорости генерации таких пракил.

Выявление спама

Эвристические алгоритмы антивирусов

Контроль информационных потоков:

- DLP
- URL filtering
- NGFW

**Спам: определение, задача, существующие решения, их недостатки**

Спам (нежелательная корреспонденция) – телематическое электронное сообщение, предназначенное неопределенному кругу лиц, доставленное абоненту и (или) пользователю без их предварительного согласия и не позволяющее определить

отправителя этого сообщения, в том числе ввиду указания в нем несуществующего или фальсифицированного адреса отправителя.
Страдает свободное или рабочее время человека, которое легко конвертировать в финансовые потери, а во втором есть риск понести прямые убытки, случайно передав злоумышленникам конфиденциальную информацию.

**Спам: решение с помощью МО**

В задаче классификации спама исходных данных достаточно – сэмплы генерируют злоумышленники, а размечают их занимаются сами пользователи своим поведением.
Прежде чем эти данные «скармливаются» классификатору, следует решить еще одну проблему — очистить их от шумов. Ведь основная сложность фильтрации спама заключается в том, что критерии восприятия полезности сообщений у разных людей могут быть разными. Один пользователь будет воспринимать письма с предложениями по распродажам как жесткий спам, в то время как другой сочтет их потенциально полезной информацией. Письма такого рода создают шумы, затрудняющие построение качественного алгоритма машинного обучения. Говоря языком статистики, в нашем наборе данных могут быть так называемые выбросы, то есть результаты, резко выделяющиеся из общей выборки. И для решения этой проблемы мы внедрили автоматическую фильтрацию выбросов на базе адаптированного под наши нужды алгоритма Isolation Forest. Безусловно, она позволяет отсеивать только часть шумов, но это уже значительно упрощает жизнь нашим алгоритмам.
В результате мы получаем практически «чистые» данные. Наша следующая задача — преобразовать их в формат, понятный классификатору. То есть в набор признаков (или, как мы говорим, фичей). По большому счету можно выделить три основных типа таких фичей, используемых в нашем классификаторе:
Текстовые фичи. С ними все более и менее понятно. Они представляют собой отрывки текста, которые часто встречаются в спам-письмах. После предварительной обработки этого текста (препроцессинга) мы получаем достаточно устойчивые признаки.
Экспертные фичи. Это признаки, базирующиеся на экспертных знаниях, накопленных нашими базами данных за долгие годы работы. Связанные, например, с доменами, с частотностью служебных заголовков и т.д.
Все эти признаки и их различные комбинации помогут нам на последнем шаге – запуске классификатора.
Напомню, на выходе мы хотим получить систему, которая допускает минимум ложных срабатываний, быстра и при этом выполняет основное свое предназначение – ловит спам. Для этого мы строим ансамбль классификаторов, и для каждого набора признаков он свой. В результате, вердикты всех классификаторов объединяют, и система выносит финальный вердикт.
https://habr.com/ru/company/vk/blog/476714/
https://securelist.ru/machine-learning-versus-spam/29962/

**Эвристические алгоритмы антивирусов: задача**

Эвристический анализ – это метод поиска вирусов, заключающийся в выявлении последовательностей команд, предположительно присущих вредоносной программе. Несмотря на возможные ошибочные решения, эвристический анализ позволяет существенно повысить эффективность работы антивирусного сканера как по обнаружению новых видов вирусов, так и полиморфных вирусов и вирус-генераторов, которых он может выявлять по характерным алгоритмам или вызовам внешних процедур.

**Эвристические алгоритмы антивирусов: МО**

Предположим, что рассматриваемый инструмент анализирует программу с помощью некоторых элементарных единиц, которые отражают поведение программы, например,

машинные команды или системные вызовы. Машинная команда или системный вызов имеют смысловое значение. А это значит, что модель, которая состоит из статистически обработанных системных вызовов или машинных команд, тоже имеет некоторую интерпретацию. Аналогично тому, как смысл слова не равен смыслу каждой отдельной его буквы, так и понимание составных частей не дает понимания всей модели. Последовательность букв приобретает смысл лишь после того, как человек императивно назначит ему смысловую нагрузку.
https://www.okbsapr.ru/library/links/obnaruzhenie-anomalnogo-povedeniya-programm-dlya-dalneyshego-ispolzovaniya-pri-reshenii-zadachi-zashch/
https://download.geo.drweb.com/pub/drweb/windows/workstation/12.0/documentation/html/av/ru/index.html?intro_detectionmethods.html

**DLP: задача**

DLP-системе необходимо понимать, какие данные следует защищать. Как система определяет, что информация относится к конфиденциальной?
С самого начала развития DLP-системы опирались на характеристики файлов – время создания, метки и прочее. Можно по ключевым словам определить, содержит ли файл ценную информацию.
В последнее время всё активнее применяется «искусственный интеллект» в виде моделей машинного обучения, нейронных сетей. Они обучаются на известных документах разной степени ценности, затем сами предсказывают ценность обнаруживаемых документов, а при корректировке предсказаний сотрудниками отдела ИБ – дообучаются.

**DLP: применение МО**

Определение конфиденциальности новой информации – классификация на основе известных наборов данных.
Важные признаки для определения необходимости защиты файла – его свойств и контекст. Контекст – это дополнительные сведения о передаваемой информации: кто, когда, откуда, по какому каналу, как часто. Например, получение сотрудником, имеющим допуск к конфиденциальной информации, одного или двух документов в течение часа – событие рядовое, а вот запрос всей защищаемой библиотеки сразу – повод для сотрудника отдела ИБ проверить его.

**URLfiltering: задача**

Кроме контроля информации, выходящей из системы, надо контролировать ещё и входящую. В том числе – не допускать загрузку данных с URL-адресов, которые могут быть вредоносными
Существующие реализации – списки запрещённых адресов, списки частей URL-адресов, анализ сертификатов
Проблема – вредоносные URL-адреса становятся доступными раньше, чем появляются в списках запрещённых
https://meliorit.ru/palo-alto-networks/subscription/url-filtering

**URLfiltering: применнеие МО**

Известные признаки – элементы URL-адреса, HTML и JavaScript кода страницы
Вариант реализации – по списку запрещённых адресов определить зависимости между частями URL-адресов, соответствующие «зловредным» и «не-зловредным» Интернет-ресурсам
Для первого элемента исследуемого адреса предсказывать моделью, обученной на лексике «зловредных» ресурсов, следующую часть. Если предсказание подтвердилось – повысить вероятность того, что ресурс зловредный
Лексическую модель расширить с помощью известных данных из WHOIS и списков запрещённых ресурсов

В результате модель формирует итоговое решение.
https://vixra.org/pdf/1906.0402v1.pdf

**NGFW: задача**

Традиционные межсетевые экраны отстают от требований по обеспечению полной защиты

Универсальный шлюз безопасности (UTM) - устройство «всё в одном», объединившее возможности антивируса, VPN, межсетевого экрана, контент- и спам-фильтров, систем обнаружения и предотвращения вторжений

Проблема: UTM медленные и обнаруживают не все угрозы

Решение: создать FPGA-чипы и связать все сервисы в один для совместной обработки данных. Результат назвали NGFW – Next Generation FireWall

Но как связать все данные, ничего не пропустив?

Для решения этой задачи уместно использование ИИ

https://www.smart-soft.ru/blog/chto_takoe_utm_ngfw_i_chem_oni_otlichajutsja/

**NGFW: применение МО**

Новое устройство появилось в сети. Что это? Классификация по слепку сетевого трафика даст ответ

Много данных, непонятно, как составить правила для связи разных систем в составе NGFW? Подход «предобработка – выбор модели – оценка метрик» может помочь создать рекомендации

Непонятно, как защищаться от новых атак? Рассмотренные выше для антивирусов эвристические алгоритмы здесь также эффективны

https://blog.tiger-optics.ru/2021/01/iot-security/

https://www.paloaltonetworks.com/cyberpedia/what-is-an-ml-powered-ngfw

Вопросы к экзамену по модулю «Методы классификации из ML в ИБ»

| № | Вопрос |
|---|---|
| 1 | Выберите классы методов машинного обучения:<br>**1. «Без учителя» и «с учителем»**<br>2. «С явным программированием» и «без явного программирования»<br>3. «Графические» и «графовые»<br>4. «Обучения» и «предсказания» |
| 2 | Выберите наиболее правильное определение «классификация»:<br>**1. Предсказание для объекта одного из конечного списка классов**<br>2. Предсказание для объекта одного из бесконечного списка классов<br>3. Предсказание для класса его объектов<br>4. Предсказание для объекта вероятности попадания в интервал |
| 3 | Выберите наиболее правильное определение «регрессия»:<br>1. Предсказание для объекта одного из конечного списка классов<br>**2. Предсказание для объекта одного из бесконечного списка классов**<br>3. Предсказание для класса его объектов<br>4. Предсказание для объекта вероятности попадания в интервал |

| | |
|---|---|
| 4 | Выберите качества, присущие деревьям решений:<br>1. Нечувствительность к шумам<br>**2. Возможность интерполировать знания**<br>3. Возможность экстраполировать знания<br>**4. Интерпретируемость** |
| 5 | Выберите качества, присущие методу K-ближайших соседей<br>**1. Устойчивость к выбросам**<br>2. Теоретическая обоснованность выбора всех параметров<br>**3. Зависимость от способа измерения расстояния между объектами**<br>**4. Интерпретируемость** |
| 6 | Выберите качества, присущие методу опорных векторов:<br>**1. Хорошие результаты работы в условиях большого количества признаков**<br>2. Нечувствительность к шуму<br>**3. Возможность замены ядра**<br>4. Наличие общепринятого метода подбора ядра |
| 7 | Выберите качества, присущие наивному байесовскому классификатору:<br>1. Малое количество гиперпараметров<br>**2. Отсутствие гиперпараметров**<br>3. Умение обнаруживать категории, отсутствовавшие при обучении<br>4. Работа в предположении о том, что признаки зависимы |
| 8 | Выберите качества, присущие линейной регрессии:<br>1. Возможность работы с конечным количеством классов<br>2. Эффективная работа при любой функции зависимости целевой переменной от входной<br>**3. Метод регрессии, а не классификации**<br>4. Метод классификации, а не регрессии |
| 9 | Выберите качества, присущие логистической регрессии:<br>**1. Возможность работы с конечным количеством классов**<br>2. Эффективная работа при любой функции зависимости целевой переменной от входной<br>3. Метод регрессии, а не классификации<br>**4. Метод классификации, а не регрессии** |

| | |
|---|---|
| 10 | Выберите характерные для метода ансамблирования «Бэггинг» особенности:<br>1. Каждая слабая модель принимает решение о классе объекта на основе ограниченного набора признаков, затем добавляет недостающие, принимает решение ещё раз и передаёт неизменённый исходный набор данных следующей модели. Решение последней модели является общим решением.<br>**2. Слабые модели обучаются на ограниченном наборе признаков, общее решение принимается по усреднению набора решений.**<br>3. Каждая слабая модель добавляет к исходным данным своё решение и передаёт следующей. Общее решение – это решение последней модели.<br>4. Каждая слабая модель принимает решение о классе объекта, а ещё одна модель, принимая в качестве входных данных ответы всех предыдущих моделей, принимает общее решение. |
| 11 | Выберите характерные для метода ансамблирования «Бустинг» особенности:<br>1. Каждая слабая модель принимает решение о классе объекта на основе ограниченного набора признаков, затем добавляет недостающие, принимает решение ещё раз и передаёт неизменённый исходный набор данных следующей модели. Решение последней модели является общим решением.<br>2. Слабые модели обучаются на ограниченном наборе признаков, общее решение принимается по усреднению набора решений.<br>**3. Каждая слабая модель добавляет к исходным данным своё решение и передаёт следующей. Общее решение – это решение последней модели.**<br>4. Каждая слабая модель принимает решение о классе объекта, а ещё одна модель, принимая в качестве входных данных ответы всех предыдущих моделей, принимает общее решение |
| 12 | Выберите характерные для метода ансамблирования «Стэкинг» особенности:<br>1. Каждая слабая модель принимает решение о классе объекта на основе ограниченного набора признаков, затем добавляет недостающие, принимает решение ещё раз и передаёт неизменённый исходный набор данных следующей модели. Решение последней модели является общим решением.<br>2. Слабые модели обучаются на ограниченном наборе признаков, общее решение принимается по усреднению набора решений.<br>3. Каждая слабая модель добавляет к исходным данным своё решение и передаёт следующей. Общее решение – это решение последней модели.<br>4. **Каждая слабая модель принимает решение о классе объекта, а ещё одна модель, принимая в качестве входных данных ответы всех предыдущих моделей, принимает общее решение**. |

| | |
|---|---|
| 13 | Выберите типовые случаи применения машинного обучения в информационной безопасности:<br>**1. Сложно выявить детерминированные правила для работы системы**<br>2. Требуется абсолютная точность работы системы<br>**3. Повышенные требования к скорости создания правил для работы системы**<br>4. Любой другой случай – методы машинного обучения имеет смысл использовать в любой задаче |
| 14 | Что нужно сделать в задаче выявления спама при создании модели после получения размеченных данных от пользователей?<br>1. Передать полученные данные в качестве входных данных для обучения модели<br>**2. Очистить полученные данные**<br>3. Передать полученные данные в качестве входных данных для предсказания модели<br>4. Определить верный ответ для каждого сэмпла в данных |
| 15 | Как имеет смысл обработать полученные от пользователей данные в задаче выявления спама?<br>1. Оставить как есть – модель сама разберётся<br>2. Выделить текстовые признаки<br>3. Выделить экспертные признаки<br>**4. Выделить текстовые и экспертные признаки** |
| 16 | Выберите наиболее правильное завершение определения. «Эвристический анализ – это метод поиска вирусов, заключающийся в …»:<br>1. генерации вирусов и обучении модели поиску сгенерированных сэмплов<br>2. выявлении последовательностей команд, встречавшихся ранее в обнаруженных вирусах<br>**3. выявлении последовательностей команд, предположительно присущих вредоносной программе**<br>4. выявлении сигнатур, соответствующих исходным кодам вирусов |
| 17 | Выберите, на что опирались в работе первые DLP-системы:<br>**1. Характеристики файла**<br>2. Контекст<br>3. Представление текстового содержимого файла, приведённое в двумерное пространство<br>4. Особенности операционной системы, в которой установлена DLP-система |

| | |
|---|---|
| 18 | Выберите наиболее правильное утверждение:<br>1. Решение модели машинного обучения важнее, чем решение сотрудника отдела ИБ<br>2. В случае расхождения решений модели машинного обучения и сотрудника отдела ИБ не обходимо обратиться к руководителю отдела ИБ<br>**3. Решение модели машинного обучения менее важно, чем решение сотрудника отдела ИБ**<br>4. Решение модели машинного обучения не может отличаться от решения сотрудника отдела ИБ |
| 19 | Выберите одно или несколько правильных утверждений:<br>1. Межсетевой экран, UTM, NGFW – это синонимы<br>2. UTM – это развитие NGFW<br>**3. NGFW – это развитие UTM**<br>**4. UTM и NGFW – это развитие межсетевых экранов** |
| 20 | Что реализация NGFW может предложить в случае появления неопознанного узла в защищаемой сети?<br>1. Классификация по первому пакету<br>**2. Классификация по слепку сетевого трафика**<br>3. Классификация с применением активного сканирования<br>4. Классификация на основе экспертного знания об адресном пространстве |

**Поведенческий анализ**

Определение Gartner (Если член Garnter: https://www.gartner.com/en/documents/2831117).

Смена парадигмы. Ориентир на людей и систему а целом, а не на защиту данных.

UBA или UEBA? Или же SIEM и SOAR?

- https://blog.varonis.ru/user-entity-behavior-analytics-ueba/
- https://www.exabeam.com/siem/uba-ueba-siem-security-management-terms-defined-exabeam/
- https://www.cshub.com/executive-decisions/articles/unlock-hidden-threats-with-uba-and-ueba
- https://www.s3-uk.com/user-entity-behaviour-analytics-siem/
- https://www.securitylab.ru/blog/personal/80na20/342749.php
- https://xakep.ru/2020/09/10/trend-micro-xdr/#toc01.
- https://can-topay.medium.com/yet-another-soar-design-3278194a71f5
- https://www.paloaltonetworks.com/cyberpedia/what-is-ueba

UBA или UEBA? Или же SIEM и SOAR?

Три столпа UEBA.

- https://habr.com/ru/company/varonis/blog/468989/
- https://www.researchgate.net/publication/336259455_The_role_of_User_Entity_Behavior_Analytics_to_detect_network_attacks_in_real_time

Что нужно знать при интеграции UEBA?

https://www.logpoint.com/en/blog/ueba-user-and-entity-behavior-analytics/

Источники данных UEBA

https://www.splunk.com/en_us/data-insider/user-behavior-analytics-ueba.html
Playbooks. Разработка сценариев атак на систему. Слабые места систем. Причинно-следственные связи. Определение целей обнаружения.
Поиск метрик и очистка данных. Влияние избыточности данных на ложное принятие решений.
Создание профилей пользователей и других сущностей.
Моделирование шаблонов поведения. Принцип формирования, группировки по необходимым направлениям.
Моделирование шаблонов поведения. Кластеризация пользователей на группы по поведению.
Определение нормального состояния. Обучение модели.
Корреляция событий и поведенческого шаблона.
Корреляция событий и поведенческого шаблона. Примеры отклонения от нормального поведения.
Регистрация отклонений. Привести примеры возможных отклонений. Регистрация может идти в SIEM, на базе которой располагается UEBA.
Обнаружение отклонений. Способы
Классификация вектора атаки. Таблица Mitre ATT&CK.
https://xakep.ru/2021/03/17/mitre-att-ck/
Классификация вектора атаки. Таблица Mitre ATT&CK русский вариант от Positive Technologies.
Оценка риска. Управление рисками компании на основе определения метрики у каждого субъекта. Аномальное поведение повышает уровень риска.
Оценка риска. Отслеживание изменения показателей. Превышение допустимого порога влечет оповещение сотрудников, которые специализируются на конкретной группе отклонений (администратор информационной безопасности, DLP-систем или же отдел кадров). Приоритезация событий дает быструю реакцию на инциденты, что позволяет предотвратить причинение ущерба.
Сценарии. Манипуляции с данными.
Как можно избежать? События, которые бы отслеживались UEBA.
Сценарии. Фишинговая атака.
Как можно избежать? События, которые бы отслеживались UEBA.
Community open-source UBA. Решении под лицензией GNU, не зависит от SIEM.
https://openuba.org/
https://github.com/GACWR/ouba-paper/blob/master/openuba.pdf
Рекомендуется к ознакомлению:
Анализ поведенческих данных на R и Python / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2022. – 368 с.

Вопросы к экзамену по модулю «Поведенческий анализ»

| № | Вопрос |
|---|---|

| | |
|---|---|
| 1 | Аналитика поведения пользователей ("UBA"), по определению Gartner, — это процесс кибербезопасности, направленный на:<br>1. обнаружение внешних угроз, целевых атак и финансового мошенничества<br>**2. обнаружение внутренних угроз, целевых атак и финансового мошенничества**<br>3. обнаружение внутренних угроз, массовых атак и финансового мошенничества<br>4. обнаружение внутренних угроз, целевых атак и имущественного мошенничества |
| 2 | Решения UBA изучают модели поведения:<br>**1. людей**<br>2. машин<br>3. IoT-устройств<br>4. узлов в сети |
| 3 | Какой подход применяется в системах U(E)BA?<br>1. Датацентричный<br>**2. Человекоцентричный**<br>**3. Системоцентричный**<br>4. Событийноцентричный |
| 4 | Что означает E в аббревиатуре UEBA?<br>1. Essence (существо)<br>2. Existence (наличие)<br>**3. Entity (сущность)**<br>4. Effect (влияние) |
| 5 | UEBA может входить в систему:<br>**1. SIEM**<br>**2. SOAR**<br>3. Treat Intelligence<br>4. Все перечисленное |
| 6 | Преимущества UEBA от классических SIEM:<br>**1. Большая вероятность найти внутреннюю угрозу**<br>2. Простота интеграции<br>**3. Самообучаема**<br>**4.** Все перечисленное |
| 7 | Три столпа UEBA:<br>1. Конфиденциальность, целостность, доступность<br>**2. Решаемые задачи, данные, аналитика**<br>3. Очистка данных, моделирование, представление<br>4. Наследование, инкапсуляция, полиморфизм |

| | |
|---|---|
| 8 | Источники данных UEBA:<br>**1. Письма электронной почты**<br>**2. Журналирование систем**<br>**3. Публикации в социальных сетях**<br>**4. Телефонные разговоры с рабочего телефона** |
| 9 | С помощью UEBA возможно:<br>**1. Оценить угрозу действий каждого сотрудника**<br>2. Обеспечивать мониторинг производительности сущностей<br>3. Отправлять оповещения подозрительным сотрудникам<br>4. Ограничить доступ злоумышленнику |
| 10 | Решения UBA изучают модели поведения людей, а затем применяют алгоритмы и _____ для обнаружения значимых аномалий в этих моделях – аномалий, указывающих на потенциальные угрозы<br>1. иммунные системы<br>2. кластерный анализ<br>**3. статистический анализ**<br>4. методы машинного обучения |

**RL в ИБ**

Лекция Семёна Козлова из курса DLcourse https://www.youtube.com/watch?v=_x0ASf9jV9U

Оригинальная статья Andrej Karpathy http://karpathy.github.io/2016/05/31/rl/

Перевод статьи Andrej Karpathy https://habr.com/ru/post/439674/

Основы RL и про его применение в ИБ). На практическом занятии: анализ спама (классификация)

**Обучение с подкреплением**

Что такое обучение с подкреплением в целом. Пояснить, кто такой «агент» и какие у него есть возможности.

http://ai.lector.ru/?go=lection07

**Среда**

Что такое марковский процесс и как среда видит агента (с выбором действий).

https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

**Состояние**

Состояние с точки зрения среды и с точки зрения агента. Рисунок.

**Действие**

Определение действия в RL, классификация пространств действий. Действие с точки зрения агента и среды. Примеры действий в понятных играх (задачах). Вероятность перехода в состояние после действия.

**Награда**

Определение награды в RL. Методы оценки действий – немедленный и отсроченный. Показать на рисунке.

**Траектории (эпизоды)**

Описание траектории в RL. Объяснение способа их формирования. Табличное представление траекторий и шагов в них.

**Политики**

Описание политики в RL. Задача политики. Формулы. Коэффициент дисконтирования (жадные и e-жадные политики). Будущие награды.

**Функции значения**

Определение. Пояснить формулы для политики в целом и для оптимальной политики. Объяснить, как выбирается оптимальное действие.

**Policy Gradients: общая идея**

Объяснение формулы функции политики. Объяснение формулы ожидаемой награды. Демонстрация изменения политика при обучении. Демонстрация анимации с пояснением того, что такое градиентный спуск. Объяснение формирования вероятности траектории.

**Policy Gradients: немного математики**

Напоминание вероятность прохождения стратегии. Показать трюк с градиентом и логарифмом. Показать, как перейти к логарифму вероятности прохождения траектории. https://spinningup.openai.com/en/latest/spinningup/rl_intro3.html

**Policy Gradients: в поисках градиента**

Напоминание про то, что ищем (градиент политики) и про то, что доказано (логарифму вероятности прохождения траектории). Подстановка второго в первое, оценка по выборочному среднему. Возможность оценить по сыгранным траекториям. Демонстрация того, как считается изменение параметров политики.

**Policy Gradients: Reward-to-Go**

Рассказ о том, как улучшить базовую политику и работать с наградой за каждый шаг в траектории. Демонстрация на рисунке, к каким положительным изменениям это ведёт (уменьшение шума и пр.)

**Policy Gradients: baseline**

Лемма EGLP. Использование baseline для дополнительной мотивации нейронной сети (101>100, но 1>>0).

**Policy Gradients: хватит математики, как это работает?**

Описание алгоритма обучения модели RL с применением нейронной сети для оптимизации параметров.

**Policy Gradients: actor-critic**

Расширенный (адаптивный) вариант baseline. Дополнительный выходной нейрон.

**Q-learning**

Решение проблемы неообходимости переигрывания эпизодов, характерной для классического Policy Gradients. Q-функция – принципиально важное нововведение: выбирается не оптимальное действие, а награда, которую можно получить за каждое действие.

**Уравнение Беллмана**

Демонстрация и объяснение формулы ожидаемой награды, расписанной на «следующий и остальные» шаги. Демонстрация неизменности при оптимальной политике.

**Q-learning: идея**

Описание возможности «заглядывать в будущее» с помощью «старой версии себя». Уточнённая награда. Демонстрация на схеме.

**Q-learning: алгоритм**

Описание алгоритма обучения Q-learning алгоритма – схема и таблица на слайде.

**Q-learning vs Policy Gradients**

Сравнение двух алгоритмов с описанием достоинств и недостатков каждого.

**Классификация алгоритмов RL**

Таксономия алгортмов RL, разработанная сообществом OpenAI. Верхнеуровнево описать и
https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

**Примеры применения RL в ИБ**

Описать основные примеры применения RL в ИБ.

Li, C., & Qiu, M. (2019). Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies (1st ed.). Chapman and Hall/CRC, главы 7-9

**Противодействие радиопомехам при передаче информации: существующие методы**

Принципиальная задача. Существующие решения – отстройка по частоте.

**Противодействие радиопомехам при передаче информации: применение RL**

Описание метода, представленного в статье ниже. Смысл игры – угадать по расположению занятых каналов на текущем шаге, какие каналы будут заняты на следующем шаге.

B. Wang, Y. Wu, K. J. R. Liu, and T. C. Clancy. An anti-jamming stochastic game for cognitive radio networks. IEEE Journal on Selected Areas in Communications, 29(4):877–889, 2011.

**Мобильные периферийные вычисления: существующие методы ИБ**

Точки доступа, в связи с внедрением 5G, планируют хранить информацию, чтобы быстрее обслуживать пользователей, и экономить передаваемый трафик.

Потенциальные риски – потратить ресурсы на хранение невостребованных данных и дать доступ к конфиденциальной информации тем людям, у которых его не должно быть.

**Мобильные периферийные вычисления: применение RL для защиты**

Описание среды как множества состояний и состояний как сэмплов информации. Всё в соответствии со статьёй:

L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani. Security in mobile edge caching with reinforcement learning. arXiv preprint arXiv:1801.05915, 2018.

**Распределение задач между работниками: существующие методы**

Задача массового обслуживания. Распределение аналитиков по событиям (или событий по аналитикам – в зависимости от масштаба).

**Распределение задач между работниками: применение RL для защиты**

Идея – определить модель игры и возможности каждого аналитика, и обучить модель играть так, чтобы получать выигрыш в достаточно долгосрочной перспективе.

R. Ganesan, S. Jajodia, A. Shah, and H. Cam. Dynamic scheduling of cybersecurity analysts for minimizing risk using reinforce- ment learning. ACM Transactions on Intelligent Systems and Technology (TIST), 8(1):4, 2016

**Поиск безопасного пути передачи информации: задача**

Задача – найти безопасный путь соединения через доверенные узлы. Для простых случаев есть готовые решения, но для сложных сетей они вычислительно сложны. Пример алгоритма Дейкстры – в анимации на слайде.

**Поиск безопасного пути передачи информации: применение RL**

Схожесть с задачей объезда пробок. Задача в терминологии RL. Потребность в адаптивной модели и возможности RL в этой области.

1. R. Nannapaneni «Optimal path routing using reinforcement learning», Dell EMC
2. Li, C., & Qiu, M. (2019). Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies (1st ed.). Chapman and Hall/CRC

**Автоматизированное тестирование на проникновение: существующие методы**

Экспертов пока заменить не получается, но есть много ПО для облегчения им жизни. Но ПО работает по детерминированным алгоритмам, что не даёт возможности обнаруживать новые уязвимости.

**Автоматизированное тестирование на проникновение: применение RL**

RL в этой области может помочь обнаруживать новые уязвимости с помощью комбинирования известных подходов и способов. Существующий инструмент – DeepExplot, основанный на Metasploit.
https://www.vulnhub.com/
https://github.com/rapid7/metasploitable3
Мясников А.В. «Применение машинного обучения с подкреплением в задаче тестирования на проникновение» (2020).
УК РФ ст. 272. Неправомерный доступ к компьютерной информации, УК РФ ст. 273. Создание, <>.. программ <..> предназначенных для <..> нейтрализации СЗИ <..> до 7 лет лишения свободы

Вопросы к экзамену по модулю «Методы классификации из ML в ИБ»

| № | Вопрос |
|---|---|
| 1 | Выберите наиболее правильное определение «обучение с подкреплением»:<br>1.Метод машинного обучения на основе обучения среды, которая взаимодействует с агентом<br>2.Эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе<br>**3.Метод машинного обучения на основе обучения интеллектуального агента, который действует во внешней среде**<br>4.Марковский процесс принятия решений с конечным множеством состояний |
| 2 | Выберите наиболее правильное определение «награда»:<br>1. Обратная связь для измерения скорости обучения агента<br>2. **Обратная связь для измерения успеха или неудачи действий агента**<br>3. Мера, определяющая расположение агента по отношению к набору состояний<br>4. Определяемая агентом метрика успеха его действий |
| 3 | Выберите наиболее правильное определение «действие»:<br>1. Переход из одного состояния в другое<br>2. Решение агента о переходе в конкретное состояние<br>3. Факт перехода из одного перехода в другое и получения за это награды<br>**4. Решение, которое агент принимает и передаёт среде** |

| | |
|---|---|
| 4 | Выберите наиболее правильное определение «траектория»:<br>**1. Последовательность состояний и действий, которые влияли на переходы между этими состояниями**<br>2. Последовательность состояний, которые прошёл агент от начала до конца эпизода<br>3. Последовательность действий, которые предпринял агент и наград, которые он получил за эти действия<br>4. Последовательность наград |
| 5 | Выберите наиболее правильное определение «политика»:<br>1. Способ выбора действия для получения максимальной награды на следующем шаге<br>2. Метод оценки возможного выигрыша на бесконечном числе шагов<br>**3. Правило для выбора следующего действия на основе текущего состояния**<br>4. Способ оценки награды на следующем шаге для каждого возможного действия |
| 6 | Выберите наиболее правильное определение «функция значения»:<br>1. Ожидаемое значение награды на следующем шаге<br>2. Оптимальное действие из заданного состояния<br>3. Функция для определения состояния на следующем шаге по состоянию и действию на текущем шаге<br>**4. Ожидаемая сумма будущих наград из текущего состояния в соответствии с заданной политикой** |
| 7 | Выберите качества, присущие Policy Gradients:<br>**1. Может работать с непрерывными действиями**<br>2. Может обучаться на исторических данных<br>**3. Принимает решение об оптимальном действии на каждом шаге**<br>4. Оценивает размер возможной награды для каждого действия |
| 8 | Выберите качества, присущие Q-learning:<br>1. Может работать с непрерывными действиями<br>**2. Может обучаться на исторических данных**<br>3. Принимает решение об оптимальном действии на каждом шаге<br>**4. Оценивает размер возможной награды для каждого действия** |
| 9 | Выберите варианты разделения на одном уровне алгоритмов обучения с подкреплением (по классификации, предложенной OpenAI):<br>**1. «Model-Free» и «Model-Based»**<br>2. «Model-Free» и «Learn the Model»<br>**3. «Policy Optimization» и «Q-Learning»**<br>4. «Policy Optimization» и «Model-Based» |

| | |
|---|---|
| 10 | Выберите улучшения, которые могут произойти в существующих инструментах тестирования на проникновение после интеграции в них алгоритмов обучения с подкреплением:<br>1. Инструменты смогут заменить экспертов.<br>**2. Инструменты получат возможность обнаруживать новые способы эксплуатации уязвимостей в программном обеспечении.**<br>**3. Инструменты станут менее заметными для систем обнаружения вторжений.**<br>4. Инструменты получат возможность получать первый прообраз значения хеш-функции без полного перебора. |

## Проблема конфиденциальности данных при использовании ML

### Постановка проблемы

Как вы уже знаете, глубокое обучение, являясь подразделом машинного обучения, основано на изучении данных. Но часто изучаемые данные являются глубоко личными. Многие модели анализируют частную информацию, рассказывающую о жизни людей то, что иным способом трудно было бы узнать.

Основным ресурсом в глубоком обучении являются обучающие данные (естественные или синтетические). Без этих данных глубокое обучение невозможно; а поскольку самые ценные модели часто используют наборы личных данных, глубокое обучение нередко становится причиной, почему компании стремятся собрать такие данные. Они нужны им для использования в конкретных сферах.

Кроме отсутствия желания пользователей делиться своими данными, существует еще ряд законодательных ограничений. Их мы рассмотрим на следующем слайде.

Все эти пункты можно разобрать на примере формирования модели, позволяющей диагностировать раковые опухоли на ранних этапах.

### Классификация информации по видам доступа

Классификация информации по видам доступа приведена на схеме ниже (в соответствии с российским законодательством). Кратко привести примеры по каждому из видов информации.

Отсюда важный вывод, что конфиденциальные данные и персональные данные это не одно и тоже. Персональные данные — это подмножество конфиденциальных данных. Детальное определение будет дальше. В лекции мы сосредоточимся на персональных данных, так как часто именно они представляют интерес для обучения моделей. Однако сделанные выводы могут быть перенесены и на другие виду конфиденциальной информации.

На самом деле зачастую одни и те же сведения могут относится к различным категориям тайн. Например, сведения, составляющие врачебную тайну, наверняка являются чьими-то персональными данными. Поэтому это деление весьма условно.

Подробнее и с примерами: https://www.securitylab.ru/blog/personal/aguryanov/29908.php

### Нормативные правовые акты, регламентирующие вопросы обработки персональных данных

Видим довольно большой перечень документов разного уровня, направленных на защиту персональных данных, поэтому вопрос является актуальным. Детально на этом вопросе останавливаться не будем, а лишь поверхностно посмотрим на положения 152-ФЗ.

**ФЗ-152** Персональные данные - любая информация, относящаяся к определенному физическому лицу. Разобрать примеры персональных данных. Кратко разобрать все

определения. Разобрать категории персональных данных и их состав. Привести примеры.

Основные требования:

· обработка персональных данных должна ограничиваться достижением конкретных, заранее определенных и законных целей;

· содержание и объем обрабатываемых персональных данных должны соответствовать заявленным целям обработки;

· обрабатываемые персональные данные не должны быть избыточными по отношению к заявленным целям их обработки;

· должно быть получено согласие субъекта на размещение его персональных данных в форме, установленной ч. 4 ст. 9 Федерального закона № 152-ФЗ.

· при обработке персональных данных должны быть обеспечены точность персональных данных, их достаточность, а в необходимых случаях и актуальность по отношению к целям обработки персональных данных. Оператор должен принимать необходимые меры либо обеспечивать их принятие по удалению или уточнению неполных или неточных данных;

· хранение персональных данных должно осуществляться в форме, позволяющей определить субъекта персональных данных, не дольше, чем этого требуют цели обработки персональных данных;

· обрабатываемые персональные данные подлежат уничтожению либо обезличиванию по достижении целей обработки или в случае утраты необходимости в достижении этих целей.

**Ответственность за нарушение законодательства в области ПД**

За нарушение законодательства в области персональных данных для физических и юридических лиц предусмотрена как административная, так и уголовная ответственность. С некоторыми наиболее «популярными» статьями можно ознакомить на слайде.

**Потенциальное решение – обезличивание ПД**

Ученые предложили множество методов обеспечения защиты конфиденциальности при анализе данных. Популярным методом является удаление личных данных или замена случайными значениями перед анализом данных. Как правило, такие детали, как номера телефонов и почтовые индексы, анонимны. Однако анонимных данных не всегда достаточно для удовлетворения требований. Когда злоумышленник получает вспомогательную информацию о лицах, представленных в наборе данных, конфиденциальность, обеспечиваемая этой анонимной операцией, будет значительно снижена. Таким образом, сложно определить и защитить конфиденциальность, а также оценить объем информации, которую может получить злоумышленник.

Разобрать определение обезличивания ПД со слайда. В РФ процедура по обезличиванию персональных данных регламентирована Приказом Роскомнадзора от 5 сентября 2013 г. № 996 "Об утверждении требований и методов по обезличиванию персональных данных". Так, в соответствии с Приказом, к наиболее перспективным и удобным для практического применения относятся следующие методы обезличивания:

· введение идентификаторов – замена части сведений идентификаторами с созданием таблицы соответствия идентификаторов исходным данным;

· изменение состава или семантики – изменение состава или семантики персональных данных путем замены результатами статистической обработки, обобщения или удаления части сведений;

· декомпозиция – разбиение множества персональных данных на несколько частей с последующим раздельным хранением подмножеств;

· перемешивание – перестановка отдельных записей, а так же групп записей в массиве персональных данных.

Важно обратить внимание, что есть различие между обезличенными и анонимизированными данными. Полностью анонимизированные данные представляют собой статистику, которая доступна в свободном доступе и относится к открытым данным (например, статистика Росстата и соцопросы). Такие данные не несут той же ценности, как обезличенные, на основании которых можно определить некоторые особенности разных видов социальных групп. Анализ поведенческих особенностей малых социальных групп, прогнозирование возрастных трендов, измерение настроения людей и определение их отношения к тем или иным явлениям – все эти функции на основании анонимизированных данных невозможны. Другими словами, полностью анонимизированные данные не представляют ценности для бизнеса, а для некоторых областей искусственного интеллекта даже обезличенные данные не представляют ценности – для его обучения требуется опыт, а если такой опыт с пробелами, его обучение будет соответственным.

Подробнее - https://www.garant.ru/news/1464529/

**Потенциальное решение – Федеративное обучение**

В 2017 году компания Google опубликовала очень интересную статью и пост в блоге, которые внесли значительный вклад в обсуждение этой темы. В Google предположили, что для обучения моделей не нужен централизованный набор данных. Компанией было предложено рассмотреть вопрос: что если вместо сбора данных в одном месте, попытаться перенести модель в данные? Этот новый и увлекательный раздел машинного обучения получил название федеративное обучение. Рассмотреть определение.

Эта простая перестановка имеет чрезвычайно большое значение. Во-первых, это означает, что для участия в цепочке глубокого обучения людям не нужно отправлять свои данные кому бы то ни было. Ценные модели в здравоохранении, управлении личными активами и других чувствительных областях можно обучать без необходимости раскрывать личную информацию.

Основная идея приведена на изображении из блога Google. Кратко: на девайс загружается модель, дообучается (обновляет коэффициенты модели) на данных конкретного пользователя. Эти изменения модели собираются в некий апдейт небольшого размера, которая отправляется в облако с использование зашифрованного соединения. Затем эти изменения вливаются в основную модель. При этом все данные, на которых обучается модель не покидают пределы пользовательского устройства.

Подробнее - https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

**Пример Федеративного обучения – Gboard**

Первое крупномасштабное применение федеративного обучения нашлось в Gboard. При старом подходе машинного обучения разработка более совершенных предсказаний клавиатуры была бы чрезвычайно инвазивной - все, что мы печатали, все наши личные сообщения и странные поиски в Google должны были быть отправлены на центральный сервер для анализа.

Федеративный подход работает по-другому. При выводе предлагаемых слов для ввода смартфон локально хранит информацию о текущем контексте и выбранных предложениях. С помощью федеративного обучения эти накопленные данные обрабатываются локально на устройстве с целью улучшения работы модели на данном устройстве.

На устройстве для дообучения используется миниатюрная версия TensorFlow. Дообучения осуществляется только когда телефон не используется, подключен к зарядке. Для передачи используются только не тарифицируемые сети. Это сделано для сведения к минимуму неудобств, доставляемых пользователю.

Подробнее - https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

**Федеративное обучение для сферы здравоохранения**

В 2018 году Intel установила партнерские отношения с Центром биомедицинских вычислений и аналитики изображений в Пенсильванском университете, чтобы продемонстрировать, как федеративное обучение может быть применено к медицинским данным в качестве доказательства концепции.

Сотрудничество показало, что в рамках федеративного подхода к обучению их конкретная модель глубокого обучения может быть подготовлена с точностью 99% по сравнению с той же моделью, обученной традиционными методами.

**Пример практического использования федеративного обучения. NVIDIA**

Исследователи из NVIDIA и Королевского колледжа Лондона использовали федеративную архитектуру клиент-сервер с центральным сервером для поддержки глобальной глубокой нейронной сети. При таком подходе участвующим больницам будет предоставлена копия их нейронной сети для обучения по их собственному набору данных.

Подробнее - https://servernews.ru/995564

**Безопасное агрегирование. Google Secure Aggregation**

Подход федеративного обучения предполагает агрегирование данных моделей от большого количества пользователей. В общем случае параметры модели от конкретного пользователя могут отображать специфичную информацию о пользователе. Поэтому встает вопрос безопасного агрегирования получаемых данных. Основная идея в том, чтобы усреднять веса до того, как кто-то сможет их увидеть (в том числе и сервер).

Есть два основных пути, используемых на практике. В действительности их больше, но рассмотрим основные.

Протокол безопасной агрегации использует многоступенчатые вычисления для определения среднего значения группы сводок пользовательских данных, не раскрывая сводки данных какого-либо отдельного лица на сервере или любой другой стороне.

В этой системе каждая из пользовательских сводок шифруется перед тем, как покинуть пользовательское устройство, и они не могут быть расшифрованы сервером до тех пор, пока они не будут добавлены вместе и усреднены с заданным числом других пользовательских сводок. Это позволяет серверу обучать свою модель в среднем по пользователю, не раскрывая отдельных сводок, которые могут быть использованы для раскрытия личных данных отдельных лиц.

Secure Aggregation не только предотвращает доступ сервера к пользовательским сводкам, но также человек посередине атаки гораздо сложнее.

При разработке были учтены особенности мобильных устройств: узкий канал передачи данных и возможные частые сбои (пользователь выключил телефон и т.д.).

Теоретическая основа - схема разделения секрета Шамира. Схема Шамира позволяет реализовать (k,n) — пороговое разделение секретного сообщения (секрета) между n сторонами так, чтобы только любые k и более сторон $k \leq n$) могли восстановить секрет. При этом любые k-1 и менее сторон не смогут восстановить секрет.

Кратко рассмотреть алгоритм, изображенный на картинке. Подробнее про Google Secure Aggregation - https://eprint.iacr.org/2017/281.pdf

**Дифференциальная приватность. Определение**

Другой способ – использование модели дифференциальной приватности.

Дифференциальная приватность — это математическое определение понятия «наличия приватности». Это не какой-то конкретный процесс, а, скорее, свойство, которым может обладать процесс. Например, можно рассчитать (доказать), что данный конкретный процесс удовлетворяет принципам дифференциальной приватности.

Этот термин был введён Синтией Дворк в 2006 году. Рассмотреть иллюстрацию на слайде, вывод по ней приведен в рамке. Кто бы ни посмотрел на результаты, он не сможет сказать, в каком случае использовались данные Ивана, а в каком не использовались.
Подробнее - https://habr.com/ru/company/domclick/blog/526724/

**Пример потребности в дифференциальной приватности**

Предположим, что у нас есть база данных медицинских записей $D_1$, где каждая запись представляет собой пару (Имя, X), где X является нулём или единицей, обозначающим, имеет ли человек гастрит или нет. Теперь предположим, что злоумышленник хочет найти, имеет ли Михаил гастрит или нет. Также предположим, что он знает, в какой строке находится информация о Михаиле в базе данных. Теперь предположим, что злоумышленнику разрешено использовать только конкретную форму запроса $Q_i$, который возвращает частичную сумму первых i строк столбца X в базе данных. Чтобы узнать, есть ли гастрит у Михаила, злоумышленник выполняет запросы: $Q_4(D_1)$ и $Q_3(D_1)$, затем вычисляет их разницу. В данном примере, $Q_4(D_1)=3$, а $Q_3(D_1)=2$, поэтому их разность равна 1. Это значит, что поле «Наличие гастрита» в строке Михаила должно быть равно 1. Этот пример показывает, как индивидуальная информация может быть скомпрометирована даже без явного запроса данных конкретного человека.
Если бы злоумышленник получал значения $Q_i$ через ε-дифференциально приватный алгоритм, для достаточно малого ε, то он не смог бы отличить два набора данных по определению дифференциальной конфиденциальности, так как $Q_4(D_1)$ и $Q_3(D_1)$ были бы неотличимы.
Посмотрим, как на практике может быть реализована дифференциальная конфиденциальность.

**Формальное определение дифференциальной приватности**

Мы контролируем требуемый уровень приватности через изменение параметра приватности ε, который также называют потерей приватности (privacy loss) или бюджетом приватности (privacy budget). Чем меньше значение ε, тем менее различимы результаты и тем больше защищены данные отдельных людей. Рассмотреть определение.
В соответствии с этим определением дифференциальная приватность является условием механизма публикации данных (то есть определяется доверенной стороной, выпускающей информацию о наборе данных), а не самим набором. Интуитивно это означает, что для любых двух схожих наборов данных, дифференциально-приватный алгоритм будет вести себя примерно одинаково на обоих наборах. Определение также даёт сильную гарантию того, что присутствие или отсутствие индивидуума не повлияет на окончательный вывод алгоритма.
Случай, когда $\varepsilon = 0$, является идеальным для сохранения конфиденциальности, поскольку наличие или отсутствие любой информации о любом человеке в базе данных никак не влияет на результат алгоритма, однако такой алгоритм является бессмысленным с точки зрения полезной информации, так как даже при нулевом количестве людей он будет давать такой же или подобный результат.
Подробнее - https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%84%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%B8%D0%B2%D0%B0%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C

**Реализация дифференциальной приватности – метод Лапласа**

В связи с тем, что дифференциальная приватность является вероятностной концепцией, любой её метод обязательно имеет случайную составляющую. Некоторые из них, как и

метод Лапласа, используют добавление контролируемого шума к функции, которую нужно вычислить.
Разобрать вывод формулы на слайде.
Если мы попытаемся использовать эту концепцию в вышеприведённом примере про наличие гастрита, то Δ□=1. То есть □ явно определяет параметр приватности.
Кроме шума Лапласа также можно использовать другие виды шума (например, гауссовский), но они могут потребовать небольшого ослабления определения дифференциальной приватности.

**Реализация дифференциальной приватности – пример**

Дифференциальная приватность основана на введении случайности в данные.
Простой пример заключается в том, чтобы попросить человека ответить на вопрос «Есть ли у вас атрибут A?» в соответствии со следующей процедурой:

- Человек подбрасывает монету
- Если выпал орел, отвечает честно на вопрос.
- Иначе подбрасывает ещё раз, если выпадет орел, ответь «Да», если решка — «Нет»

Конфиденциальность возникает, так как невозможно по ответу точно узнать, обладает ли человек данным атрибутом. Но тем не менее эти данные значительны, так как положительные ответы дают четверть от тех людей, у которых нет этого атрибута, и три четверти от тех, кто на самом деле им обладают. Таким образом, если p — истинная доля людей с A, то мы ожидаем получить (1/4) (1- p) + (3/4) p = (1/4) + p / 2 положительных ответов. Следовательно, можно оценить p.
То есть общая закономерность сохраняется, ценность выборки не теряется. Однако достоверно определить наличие атрибута A у конкретного человека не представляется возможным.

**PATE – алгоритм, реализующий дифференциальную приватность**

Локальные модели тренируются независимо друг от друга (без какого-либо взаимодействия). Каждая модель обучается на своем наборе данных.
Приватный набор данных разделяется на подмножество данных (разделов). На каждом из этих данных учится своя модель. Метод обучения модели учителя неограничен, что также является одним из основных преимуществ PATE.
Подробнее - https://arxiv.org/pdf/2004.06567.pdf

**PATE. Предсказание**

Как использовать дальше эти отдельно обученные модели? Каждая модель выполняет предсказание, для того чтобы внести конфиденциальность используется подход дифференциальной приватности, а именно добавляется случайный шум Лапласа или Гаусса (для нарушения статистики).
Однако в таком подходе есть явные проблемы. Во-первых, каждый прогноз, сделанный механизмом агрегирования, увеличивает набор статистики, в далекой перспективе снижая конфиденциальность. Во-вторых, набор моделей учителей не может быть опубликован с открытым исходным кодом, в противном случае злоумышленник может проверить опубликованные параметры модели, чтобы узнать об обучении.

**PATE. Модель студента**

Для решения описанных выше проблем создается дополнительная модель студента, которая обучается на за счет наличия моделей учителей. Этот процесс отображен на слайде, рассмотрим его подробнее. Модель студента обучается с соблюдением конфиденциальности путем интеграции знаний, полученных учителями. Модель студента выбирает входные данные из набора немаркированных общедоступных данных и отправляет эти входные данные в модели учителей для получения меток, а затем модель ученика использует помеченные данные для обучения. Защита

конфиденциальности и правильность тегов, предсказываемых механизмом агрегирования, являются результатом консенсуса, достигнутого среди учителей.

## Проблемы и ограничения федеративного обучения

Федеративное обучение страдает двумя большими проблемами, особенно трудноразрешимыми, когда у каждого человека имеется лишь маленькая горстка обучающих примеров, — скорость и конфиденциальность.

Как оказывается, если у кого-то имеется лишь несколько обучающих примеров (или модель, присланная вам, была обучена лишь на нескольких примерах: обучающем пакете), вы все еще можете довольно много узнать об исходных данных. Если представить, что у вас есть 10 000 человек (и у каждого имеется очень небольшой объем данных), большую часть времени вы потратите на пересылку модели туда и обратно и не так много — на обучение (особенно если модель очень большая).

Перехват при передаче данных уточненной модели – простое решение: шифрование.

«Отравление» модели: злоумышленник может испортить модель через свое собственное устройство или путем захвата устройств других сторон, участвующих в обучении алгоритмической модели. Это актуальная проблема, особенно для протокола безопасной агрегации Google, так как там мы не сможем вычислить злоумышленника, чтобы перестать принимать от него данные.

Компании должны защищать свою интеллектуальную собственность, и похоже, что отправка модели напрямую на устройства пользователей может легко привести к тому, что эти модели будут выставлены любому желающему. Однако есть решения, которые компании могут использовать для защиты своих алгоритмических моделей.

Одним из них является использовать секретный обмен многопартийных вычислений. Это позволяет организациям скрывать взвешивание модели, распределяя ее фрагменты по устройствам. В рамках этой системы ни одна из секретных сторон не может знать всю модель.

Это позволяет организациям передавать свои алгоритмические модели обучения на устройства, не беспокоясь о краже их интеллектуальной собственности.

Обсудим ограничения, которые накладывает федеративное обучение. Во-первых, конечные устройства должны обладать достаточной вычислительной мощностью и памятью. Именно поэтому 10-15 лет назад идея федеративного обучения была просто нереализуемой. Современные носимые устройства уже обладают достаточным потенциалом.

Другое техническое ограничение связано с пропускной способностью. Федеративное обучение проводится через Wi-Fi или 4G, в то время как традиционное машинное обучение происходит в центрах обработки данных. Пропускная способность Wi-Fi или 4G на порядок ниже, чем между рабочими узлами и серверами в этих центрах. Это узкое место, которое увеличивает задержку и замедляет процесс обучения по сравнению с традиционным подходом.

Несмотря на попытки оптимизировать время запуска обучения это может вызвать замедление работы устройства (например, когда пользователь неожиданно для системы снимает телефон с зарядки и начинает им активно пользоваться).

## Достоинства федеративного обучения

Федеративное обучение может также помочь организациям улучшить свои алгоритмические модели, не раскрывая данные клиента и не нарушая законодательные нормы. Законы, такие как Европейское общее положение о защите данных (GDPR) и Закон о переносимости медицинского страхования США от 1996 года, ФЗ-152 содержат строгие правила в отношении данных отдельных лиц и способов их использования.

Другая ключевая причина принятия федеративного подхода к обучению заключается в том, что он потенциально может снизить задержку. В вероятном будущем сценарии,

когда на наших дорогах будет большое количество автомобилей с автоматическим управлением, они должны быть в состоянии быстро реагировать друг на друга во время происшествий, связанных с безопасностью.
Традиционное облачное обучение включает в себя передачу больших объемов данных и более медленный темп обучения, поэтому существует вероятность того, что федеративное обучение может позволить автономным транспортным средствам действовать быстрее и точнее, уменьшая количество аварий и повышая безопасность.
В завершении лекции еще раз напомнить основные моменты и прокомментировать картинку, демонстрирующую отличия традиционного обучения, федеративного обучения и федеративного обучения с безопасной агрегацией.

**Перечень рекомендованной литературы.**


1. Dominik, Jens Sjlund, Tobias J. Oechtering. Decentralized Differentially Private Segmentation with PATE. https://arxiv.org/pdf/2004.06567.pdf
2. Э. Траск. Грокаем машинное обучение (глава 16).
3. Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical Secure Aggregation for Privacy-Preserving Machine Learning. https://eprint.iacr.org/2017/281.pdf
4. Блог SecurityLab. Виды информации ограниченного доступа. https://www.securitylab.ru/blog/personal/aguryanov/29908.php
5. Информационно-правовой портал «Гарант.ру». Обезличивание данных: сохранение баланса между правами граждан и развитием инноваций. https://www.garant.ru/news/1464529/
6. Google AI blog. Federated Learning: Collaborative Machine Learning without Centralized Training Data. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
7. Новостной портал servernews.NVIDIA использовала федеративное машинное обучение при создании ИИ для здравоохранения https://servernews.ru/995564
8. Портал «Хабр». Дифференциальная приватность — анализ данных с сохранением конфиденциальности. https://habr.com/ru/company/domclick/blog/526724/
9. Википедия. Дифференциальная приватность. https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%84%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%B8%D0%B2%D0%B0%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C


Вопросы к экзамену по модулю «Проблема конфиденциальности данных при использовании ML»

| № | Вопрос |
|---|---|
| 1 | Выберите наиболее правильное определение федеративного обучения:<br>1. Метод машинного обучения на основе обучения среды, которая взаимодействует с агентом<br>**2. Метод машинного обучения, который функционирует на нескольких децентрализованных периферийных устройствах, хранящих локальные данные, без обмена ими**<br>3. Метод машинного обучения на основе обучения интеллектуального агента, который действует во внешней среде<br>4. Метод машинного обучения с частичным привлечением учителя |

| | |
|---|---|
| 2 | Что означает термин «дифференциальная приватность»?<br>**1. Математическое определение понятия «наличия приватности»**<br>2. Подход, при котором конфиденциальные данные разделяются на части и хранятся в разных хранилищах<br>**3. Совокупность методов, направленных на минимизацию возможности идентификации отдельных записей в базе данных**<br>4. Свойство процесса, алгоритма, модели позволяющее надежно шифровать данные |
| 3 | С какой целью используют модели учителей в алгоритме PATE?<br>1. Для увеличения скорости обучения и сокращения объема передаваемых по сети данных<br>2. Для повышения общей точности предсказания алгоритма<br>**3. Для последующей разметки общедоступных данных для модели студента**<br>4. Для усреднения этих моделей в модель студента |
| 4 | Назовите проблемы, характерные для федеративного обучения?<br>**1. Скорость обучения**<br>2. Передача тренировочных данных на сервер<br>**3. «Отравление» модели**<br>4. Высокие требования к пропускной способности канала связи |
| 5 | На какие две группы разделяется информация согласно № 149-ФЗ?<br>1. Конфиденциальная информация<br>**2. Ограниченного доступа**<br>3. Государственная тайна<br>**4. Общедоступная** |
| 6 | Выберете то, что относится к персональным данным:<br>**1. Фамилия, имя, отчество**<br>**2. Размер одежды**<br>**3. Образование**<br>**4. Модель телефона** |
| 7 | К какой категории персональных данных относится информация о состоянии здоровья человека?<br>**1. Специальные категории персональных данных**<br>2. Биометрические персональные данные<br>3. Иные персональные данные<br>4. Общедоступные персональные данные |

| | |
|---|---|
| 8 | Что означает термин «распространение персональных данных»?<br>**1. Действия, направленные на раскрытие персональных данных неопределенному кругу лиц**<br>2. Действия, направленные на раскрытие персональных данных определенному кругу лиц<br>3. Действия, направленные на получение доступа к персональным данным<br>4. Действия, связанные с обработкой запроса на предоставление персональных данных |
| 9 | Какой шум чаще всего применяют в алгоритмах дифференциальной приватности?<br>1. Равномерный шум<br>2. Белый шум<br>**3. Шум Лапласа**<br>4. Гауссовский шум |
| 10 | Выберите верные отверждения:<br>1. За нарушение законодательства в области персональных данных предусмотрена только административная ответственность<br>**2. Клавиатура Gboard является одним из первых масштабных применений федеративного обучения**<br>**3. В основе протокола безопасной агрегации Google лежит схема разделения секрета Шамира**<br>4. Обезличенные персональные данные можно по-другому назвать анонимизированными |

**Системы организации знаний в информационной безопасности**

В модуле рассматриваются системы организации знаний в информационной безопасности: онтологии, таксономии, словари и другие виды баз знаний. В настоящее время уже разработаны сообществом базы знаний, например MITRE, CVE, CWE, STIX и другие. Подход с использованием баз знаний хорошо зарекомендовал себя в поиске (пример Knowledge Graph). Для описания онтологий используются графы и специальные средства из описания и работы с ними.
https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/

**Актуальность использования экспертных знаний**

В различных видах обеспечения безопасности (информационная безопасность, кибербезопасность, киберфизическая безопасность) существует множество актуальных на сегодняшний день задач: сетевой мониторинг, повышение осведомленности о киберобстановке, выявление аномалий, оценка уязвимости, противодействие атакам. Эффективное решение этих задач становится невозможным без специальных средств автоматизации, поскольку увеличивается количество и тип атак и ручного труда даже самых высококвалифицированных специалистов недостаточно для противодействия атакам. Один из способов повысить эффективность автоматизированных систем является использование накопленных экспертных знаний в машиночитаемом виде. Различные системы организации знаний являются хорошим инструментом для решения

этой задачи. Для использования накопленных экспертных знаний в автоматизации необходимо формализованное представление знаний Формализованное представление знаний это область искусственного интеллекта, которая создает структурированную информацию на основе выявленной семантики (смысла) концепций, свойств, связей, отдельных объектов из какой-то предметной области (например, медицина, финансы, информационная безопасность, промышленное производство).
https://controleng.ru/wp-content/uploads/8332.pdf

## Системы организации знаний

Для хранения знаний используются следующие структуры:

· Контролируемые словари: обеспечивают способ организации знаний для последующего поиска, используются в схемах предметной индексации, предметных рубриках, тезаурусах, таксономиях и других системах организации знаний

· Тезаурусы: объединяют термины в группы по определенному признаку, например, с учетом похожести (синонимы)

· Таксономии: категорированные слова, упорядоченные по иерархическому признаку

· Онтологии: формальное описание знаний из какого-то домена (предметной области) с учетом имеющихся сложных правил и связей между элементами, позволяющим сделать автоматическое извлечение знаний

· Датасеты: наборы машиночитаемых данных

Также существует множество других видов представления формальных знаний, более подробно см., например, здесь:
https://www.researchgate.net/publication/324993820_Improving_semantic_interoperability_in_the_obstetric_and_neonatal_domain_through_an_approach_based_on_ontological_realism

## Понятие онтологии

Онтология является самым популярным представлением знаний о предметной области (домена), позволяющем описать объекты этого домена, их свойства и связи между объектами. Онтология опирается на «гипотезу открытого мира», это означает, что если искомый объект не найден в онтологии, то он не обязательно не существует. В противоположность этому базы данных работают в соответствии с гипотезой «закрытого мира»: если запрос к базе данных не нашел объект, то система считает, что такого объекта нет. Например

**:Alice :knows :Bob .**

не означает, что только Алиса знает Боба. Базы данных хорошо подходят для хранения и поиска конкретной информации, а онтологии большое подходят для выявления новых знаний, например для создания экспертных систем.

Мощь онтологии заключается в том, что она основана на дескриптивной логике предикатов первого порядка, что позволяет выявлять различные логические заключения. В подобных системах всегда ищется баланс между наглядностью и сложностью извлечения фактов (ризонинг). Для эффективной работы ризонинга в онтологии необходимо максимально подробно определить свойства сущностей и связей между ними.

Онтологии бывают разного вида, например онтология верхнего уровня (общая онтология, применимая ко многим различным областям) или доменная (онтология для конкретной области знаний). Для области информационной безопасности доменные онтологии могут описывать семантику работы сети и устройств, уязвимости, информационные потоки и многое другое.

https://www.ontology-of-designing.ru/article/2020_4(38)/Ontology_Of_Designing_4_2020_final_10_Ontology_Summit_2020_Knowledge_Graphs.pdf

## Примеры использования онтологий

Системы организации знаний на основе онтологий уже очень распространены и используются во многих отраслях. Самый яркий пример это knowledge graph для поиска информации в Интернет, благодаря этой технологии качество поиска стало очень высоким. Другие примеры использования онтологий на практике:

· банки используют графы знаний для анализа транзакций (fraud detection)

· в консалтинге используются графы на основе юридических документов

· в здравоохранении используются накопленные сведения на основе данных о здоровье пациентов, Health Electronic Record (HER).

· в промышленной индустрии графы знаний используются для анализа цепочек поставщиков (supply-chain management), в целом для Индустрии 4.0 характерно взаимодействие киберфизических систем между собой, что приводит к автоматизации и необходимости управлять знаниями

· во многих отраслях базы знаний используются для организации работы чат-ботов, в том числе и для обработки сложных запросов на естественном языке (например, сервис asknow).

· онтологии могут применяться для широкого круга задач обработки естественного языка, например: аннотирование текстов с помощью онтологий, извлечение знаний, NER, Named Entity Linking, Relation Linking, автоматический вывод новых знаний, ризонинг.

В целом направление SemTech переживает в настоящее время бурное развитие, в том числе в России, но информации на русском языке по данной тематике по-прежнему очень мало.

Важно осознавать, что далеко не всегда онтологии оказываются наиболее эффективным инструментом для решения задачи, необходимо подходить к выбору способа решения с учетом особенностей предметной области.

## Дескриптивная логика

Дескриптивная логика это язык представления знаний, позволяющий описывать понятия предметной области в однозначном формализованном виде. Обеспечивает наглядность описываемой информации (см. пример на слайде), при этом, благодаря близости к математической логике, сохраняется возможность вычислений для извлечения сведений с использованием логических конструкций. Благодаря этим свойствам дескриптивная логика активно применяется на практике, обеспечивая компромисс между выразительностью и разрешимостью.

Дескриптивные логики были выбраны в качестве логической основы для языка веб-онтологий OWL. Имеющиеся в дескриптивных логиках понятия «концепт», «роль», «индивид» и «база знаний» в OWL соответствуют понятиям «класс», «свойство», «объект» и «онтология» соответственно.

## Средства описания онтологий

Основные инструменты для хранения онтологий (графов знаний)

- RDF (Resource Description Framework)
- OWL (Ontology Web Language)

RDF часто хранятся в xml в виде троек (субъект, предикат, объект)

- Форматы: RDF/XML, Turtle, N-Triples, JSON-LD, RDFa, HTML5 Microdata
- Примеры RDF хранилищ – Virtuoso, 4store (4store.org), stardog.

RDF описывает различные информационные модели, но не учитывает семантику того, что описывает. По сути, RDF это граф. Для дополнительного выражения семантики требуются:

· Словари – набор терминов, имеющих одинаковый смысл во всей описываемой предметной области
· Таксономия: словарь иерархически организованных терминов
· Онтология: описание отношений между терминами

Для описания RDF с учетом семантики используется RDFS (RDF Schema) и OWL (Web Ontology Language).

RDFS (RDF Schema): для описания словарей, таксономий, тезаурусов, простых онтологий

OWL позволяет дополнительно описывать логические правила над данными (ограничения), используется для сложных онтологий.

Язык веб-онтологий OWL разрабатывается как язык, на котором можно формулировать и публиковать в веб так называемые сетевые онтологии — формально записанные утверждения о понятиях и объектах некоторой предметной области. Одним из требований к таким онтологиям заключается в том, чтобы содержащиеся в них знания были «доступны» для машинной обработки, в частности, для автоматизированного логического вывода новых знаний из уже имеющихся. Для этого требуется, чтобы язык, на котором формулируются онтологии, имел точную семантику, а соответствующие логические проблемы были разрешимы (и имели практически допустимую вычислительную сложность). Кроме того, желательно, чтобы такой язык имел довольно большую выразительную силу, пригодную для формулировки на нём практически значимых фактов.

## RDF

Resource Description Framework (RDF) разработан консорциумом W3C (World Wide Web Consortium), www.w3c.org. RDF предназначен для формализованного представления данных, пригодном для машинной обработки. RDF является важной технологической частью сети Интернет, однако нашел применения и в других областях, например для описания графов знаний.

Ресурсом в RDF называется любая сущность, например: компьютер, сотрудник, документ, изображение, атака. Триплет RDF состоит из объекта и субъекта (вершины графа), связанных предикатом. Для обозначения субъектов, объектов и отношений в RDF используется URI (Uniform Resource Identifier).

## Примеры баз знаний

DBPedia (https://www.dbpedia.org/), с 2007 года, более 6 миллиардов фактов, получена парсингом информации в текстовом виде в Wikipedia (разные языки). https://www.wikipedia.org/ один из первых успешных применений семантических технологий.

Yago (https://yago-knowledge.org/), обработка WikiPedia и семантического тезауруса (словаря) WordNet (wordnet.princeton.edu), более 120 миллионов фактов

NELL, Never Ending Language Learner (http://rtw.ml.cmu.edu/rtw/), читает web страницы и выделяет в граф знаний, более 14,4 миллиардов фактов

WikiData (https://www.wikidata.org/wiki/Wikidata:Main_Page), поставщик данных для WikiPedia, 7 миллиардов знаний о 90 миллионов сущностей, сюда загружают данные крупные компании, например Facebook и Google.

Google Knowledge Graph (https://developers.google.com/knowledge-graph), развитие графа Freebase, который изначально был большой онтологической базой, собранной Интернет сообществом.
Кроме того: открытые графы знаний, специализированные графы знаний, например, для медицины: BioPortal – репозиторий биомед-графов (более 140 миллионов фактов), PubMed – аннотации медицинский статей
В настоящее время созданно и поддерживается сообществами или коммерческими компаниями огромное множество баз знаний, они связанны между собой, актуальную картину можно увидеть на https://lod-cloud.net/

**Хранение информации в графах**

Существуют различные виды графовых баз знаний.

- OpenLink Virtuoso
- Stardog
- Blazegraph (ранее Bigdata)
- GraphDB (ранее OWLIM)
- RDF4J (ранее Sesame)
- Apache Jena
- Allegrograph
- 4Store

Одним из популярных инструментов создания запросов к базам знаний является SPARQL.

**Ризонинг**

Сила онтологии проявляется в том случае, если подробно и качественно описаны взаимосвязи между ее элементами, с использованием математического аппарата дескриптивной логики. Например, для отношений можно задать их свойства (функциональное, транзитивное, рефликсивное). И тогда можно автоматически из онтологии извлекать факты, этот процесс называется ризонинг (reasoning), есть типовые алгоритмы ризонинга, основаны на графах.

Транзитивность
Симметричность
Ассиметричность
Рефлексивность
Иррефлексивность
Функциональность
Инверсная функциональность
Несовместность

Возможные применения: уточнение характеристик объекта и выделение из набора похожих объектов уникального, поиск похожих объектов, «понимание текста» и отнесение текста к определенному классу, помощь в NLP задачах (NER, Relation Extraction), анализ корневых причин, выявление паттернов в данных.
Наиболее популярный редактор онтологий, поддерживающий ризонинг, это Protégé. Еще фреймворки: IBM Watson, Wolfram Alpha, news360.com.

**Редактор Protege**

Изначально создавался как редактор медицинских данных.
Программа Protege для создания, редактирования и использования онтологий разработана в Стэнфордском университете, распространяется бесплатно, ее можно

скачать себе на компьютер по ссылке https://protege.stanford.edu/products.php, либо воспользоваться web-версией https://webprotege.stanford.edu/
Сейчас это популярный open-source продукт для создания онтологий в любых областях
Позволяет описывать Классы, их свойства, объекты классов
Содержит механизм ризонинга, позволяющий извлекать из онтологии скрытые факты, находить противоречия

**Онтологии кибербезопасности**

Описательные онтологии для кибербезопасности структурируют информацию из предметной области и объединяют сведения из различных связанных предметных областей с точки зрения кибербезопасности.
К таким глобальным широко используемым стандартам относятся

- ISO/IEC, https://www.iso.org
- OASIS, https://www.oasis-open.org
- NIST, https://www.nist.gov
- ITU-T, https://www.itu.int
- MITRE, https://www.mitre.org
- the Open Grid Forum, https://www.ogf.org
- IEEE, https://www.ieee.org

Использование информации из существующих баз знаний

- об угрозах (Common Vulnerabilities Enumeration, CVE, https://cve.mitre.org/), National Vulnerability Database, https://nvd.nist.gov
- об уязвимостях (Common Weakness Enumeration, CWE, https://cwe.mitre.org/)
- о паттернах атак (Common Attack Pattern Enumeration and Classification, CAPEC, https://capec.mitre.org)

позволяет получать более полную информацию и происходящем инциденте и предпринимать наиболее правильные и эффективные действия.

**Онтологии кибербезопасности**

Описательные онтологии для кибербезопасности структурируют информацию из предметной области и объединяют сведения из различных связанных предметных областей с точки зрения кибербезопасности.
К таким глобальным широко используемым стандартам относятся

- ISO/IEC, https://www.iso.org
- OASIS, https://www.oasis-open.org
- NIST, https://www.nist.gov
- ITU-T, https://www.itu.int
- MITRE, https://www.mitre.org
- the Open Grid Forum, https://www.ogf.org
- IEEE, https://www.ieee.org

Использование информации из существующих баз знаний позволяет получать более полную информацию и происходящем инциденте и предпринимать наиболее правильные и эффективные действия.

- угрозы (Common Vulnerabilities Enumeration, CVE, https://cve.mitre.org/),
- уязвимости (Common Weakness Enumeration, CWE, https://cwe.mitre.org/)
- паттерны атак (Common Attack Pattern Enumeration and Classification, CAPEC, https://capec.mitre.org)

**Матрица MITRE**

Матрица MITRE ATT&CK описывает тактики и техники, которыми злоумышленники пользуются в своих атаках на корпоративную инфраструктуру.

Матрица MITRE D3Fend описывает методы защиты от атак.
https://cryptoworld.su/mitre-attck-i-informacionnaya-bezopasnost/
Компания Positive Technologies выяснила, 37% специалистов по ИБ (https://www.ptsecurity.com/ru-ru/research/analytics/kak-rossijskie-kompanii-zashchishchayutsya-ot-celevyh-atak/) планируют начать использовать матрицу ATT&CK для мониторинга и расследования атак. Эксперты компании перевели её на русский язык и опубликовали
https://mitre.ptsecurity.com/ru-RU/techniques?utm_source=telegram&utm_medium=seclab
в интерактивном формате. В адаптированной версии видно, какие угрозы из международной базы знаний можно выявить с помощью системы анализа трафика PT Network Attack Discovery и как она это делает.

**CVE**

https://cve.mitre.org/
База знаний об угрозах.

**CWE**

https://cwe.mitre.org/
База знаний об уязвимостях.

**CAPEC**

https://capec.mitre.org/
База знаний о паттернах атак.

**Использование графов знаний в информационной безопасности**

В информационной безопасности графы знаний могут использоваться для следующих задач:

- хранение сложноструктурированной информации: графы являются полуструктурированными объектами, можно гибко модифицировать
- Поиск новых фактов на основе имеющихся: восстановление ребер по паттернам
- Выявление противоречий в фактах: проверка логической непротиворечивости и единственности в графовых конструкциях
- Выявление закономерностей в фактах: эмбеддинги на графах и кластеризация

Область машинного обучения на графах активно развивается и в ближайшем будущем появится множество новых результатов и алгоритмов в этом направлении.

Тестовые вопросы по модулю «Системы организации знаний в ИБ»

| № | Вопрос |
|---|---|
| 1 | Укажите несколько причин актуальности использования систем поддержки принятия решений на основе экспертных знаний:<br>**1. Существенное увеличение объема данных для анализа**<br>2. Стоимость решений<br>3. Государственная поддержка систем на базе искусственного интеллекта<br>**4. Высокая производительности аппаратных платформ** |

| | |
|---|---|
| 2 | Выберите системы организации знаний:<br>**1. Онтологии**<br>**2. Таксономии**<br>3. Журналы<br>4. Временные ряды |
| 3 | Что такое онтология:<br>**1. Описание объектов какой-то предметной области и взаимосвязей между ними**<br>2. Раздел психологии<br>3. Наука о семантике<br>4. Набор алгоритмов для работы с графами |
| 4 | Выберите элементы, которые входят в онтологию<br>**1. Объект**<br>**2. Субъект**<br>3. Примат<br>**4. Предикат** |
| 5 | Выберите средства, с помощью которых описывается онтология<br>**1. OWL**<br>**2. RDF**<br>3. HTML<br>4. SQL |
| 6 | Для чего может быть использован SPARQL<br>**1. Для запросов в базу знаний**<br>**2. Для автоматизированного извлечения факто**<br>3. Для ускорения SQL запросов<br>4. В качестве альтернативы HTML |
| 7 | Что такое ризонинг<br>1. Средство тестирования модели машинного обучения на графе<br>**2. Автоматическое извлечение фактов из базы знаний**<br>3. Выявление вершины с наибольшим количеством связей<br>4. Средство визуализации графа знаний |
| 8 | Укажите программное обеспечение для работы с графами знаний<br>**1. Protege**<br>2. TensorFlow<br>**3. IBM Watson**<br>4. Google Chrome |

| | |
|---|---|
| 9 | Выберите онтологии, относящиеся к информационной безопасности<br>**1. MITRE**<br>**2. CAPEC**<br>3. WikiPedia<br>4. ФЗ-187 |
| 10 | Какие задачи можно решать с использованием графов знаний в информационной безопасности<br>**1. Обогащать дополнительными сведениями процесс расследования инцидента в SoC**<br>2. Выполнять требования ФЗ-187<br>3. Уменьшать стоимость решения для кибербезопасности.<br>**4. Выявлять неизвестные факты об уязвимостях защищаемого объекта** |

**Состязательные модели в задачах информационной безопасности**

В модуле рассказывается про различные технологии применения генеративных (то есть что-то создающих) моделей машинного обучения, в том числе созданных с использованием состязательных подходов, к решению задач информационной безопасности. Отличие от типовых дискриминационных (то есть что-то описывающих) задач: классификация, регрессия.

**Обучение без учителя**

Типовое применение.

· Обработка данных
- o Понижение размерности
- o Сжатие информации
- o Очистка изображений

· Анализ данных
- o Поиск аномалий
- o Кластеризация
- o Обнаружение паттернов

· Генерация данных
- o Создание объектов

**Понижение размерности**

Одной из проблем в машинном обучении является избыточность признаков. Данные с большим количеством признаков сложно визуализировать, для их обработки требуется больше вычислительных ресурсов.

Практические задачи, которые решаются с использованием методов понижения размерности:

· сжатие информации,
· визуализация (например, для кластеризации)
· обнаружение паттернов.

Наиболее популярные методы для решения этой задачи методы: анализ главных компонент (Principal Component Analize, PCA) и TSNE. В результате получаем скрытый (латентный) вектор для каждого экземпляра в датасете, который содержит в себе сжатую информацию об этом экземпляре.

**Автокодировщики**

Автокодировщик это нейронная сеть, имеющая специальную архитектуру, состоит из двух частей: кодировщик и декодировщик.
Сначала модель обучают кодировать и декодировать сигналы из имеющейся выборки, при этом закодированный сигнал имеет гораздо меньшую размерность, чем исходный, поэтому модели приходится учиться «сжимать» информацию.
Автокодировщик, внутренний слой которого больше, чем входной, используется для удаления шумов из сигнала (denoising autoencoder)

## Машина Больцмана

Изобретатели: Джеффри Хинтон, Терри Сейновски, 1985 год. Представляет собой стохастическую рекуррентную сеть. Для обучения сети используется алгоритм имитации отжига.
Машина Больцмана с ограничениями (Restricted Boltzman Machine, RBM), используется в сетях глубокого доверия
«Интуитивное основание»: скрытая марковская модель

## Скрытая марковская модель

- наблюдаемый (известный) и скрытый (неизвестный) слои
- a и b – вероятности
- Алгоритм Витерби позволяет установить наиболее вероятную цепочку
- Обратный алгоритм позволяет уточнить значения матриц A (вероятности перехода состояний в скрытом слое) и B (вероятности того, что мы будем наблюдать на внешнем слое)

## Генеративные состязательные сети

Генеративно-состязательная сеть (Generative adversarial network, GAN) это модель машинного обучения, умеющая создавать искусственные объекты (тексты, изображения, видео) таким образом, что они максимально похожи на их естественные аналоги. Подход GAN был изобретен Яном Гудфеллоу и описан в статье: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
Основные элементы GAN: генератор и дискриминатор. Процесс обучения:

- Генератор создает фейковые изображения
- Обучаем генератор распознавать фейк (подаем на вход размеченные реальные и фейковые изображения)
- Замораживаем веса дискриминатора
- Учим генератор обманывать дискриминатор (размечаем фейк как реальные изображения)
- Повторяем (долго J)
- Закончиться все должно незначительной победой генератора (чем позднее, тем лучше)

## CycleGAN

Перенос стилей (CycleGAN) – преобразование одного изображения в соответствии со стилем других изображений (например, картин известного художника);

https://www.tensorflow.org/tutorials/generative/cyclegan

## StyleGAN

Генерация человеческих лиц (StyleGAN), реалистичные примеры доступны на сайте “This Person Does Not Exist”, https://www.thispersondoesnotexist.com/

## Лучшие практики в обработке последовательностей

Для обработки последовательностей (текстов, изображений) наилучшие результаты (State-Of-the-Art, SOTA) показали модели трансформеров с использованием механизма внимания (Attention). Отдельные части этой архитектуры получили самостоятельное существование. OpenAI разработал сеть под названием Generative Pre-trained Transformer (GPT), которая использовала модифицированый декодер трансформера. Затем Google создал Bidirectional Encoder Representations from Transformers (BERT), используя энкодер трансформера. Помимо трансформера, новые модели объединяет стратегия обучения на большом корпусе неразмеченных текстов.

GPT предсказывает очередное слово текста, а BERT - "закрытые" слова внутри предложения. В результате такого обучения формируется языковая модель, включающая в себя грамматику, семантику и даже определённые знания. После предварительного обучения производится тонкая настройка параметров модели под конкретную задачу уже на размеченных данных.

https://medium.com/techvariable/transformers-and-gpt-3-f7bbc6df2c29
https://habr.com/ru/post/341240/

## Применение GAN

- Создание наборов данных
- Аватары
- Видеоклипы
- Создание произведений искусства
- Стилизация (нарисовать фотографию в стиле Ван Гога)
- DeepFake
- …
- До конца возможности не изучены (новая технология) J

## DeepFake

Deepfake (дипфейк): «глубинное обучение» (англ. deep learning) и «подделка» (англ. fake), методика синтеза изображения, основанная на искусственном интеллекте. Она используется для соединения и наложения существующих изображений и видео на исходные изображения или видеоролики. Для DeepFake часто используют генеративно-состязательные нейросети (GAN).

Deepfake может быть использован для замены определённых элементов изображения на желаемые образы, в частности для создания фальшивых видео со знаменитостями. Deepfake может быть использован для создания поддельных новостей и вредоносных обманов.

Deepfake-ролики можно легко найти на популярных сайтах потокового видео, таких как YouTube или Vimeo.

Методы выявления движения и превращения в целевое видео, которое похоже на целевой образ, были представлены в 2016 году и позволяют создавать поддельные мимические изображения в существующем 2D-видео в режиме реального времени. 1 сентября 2020 года компания Microsoft анонсировала свою новую разработку – Microsoft Video Authenticator, программное обеспечение, которое позволяет определить вмешательство в видео.

## Применение генеративно-состязательных сетей в кибербезопасности

Как и любой инструмент, модели машинного обучения, и GAN в частности, могут применяться как со стороны атаки, так и со стороны обороны.

**Генерация фейковых сообщений**

- Текстовые сообщения, максимально приближенные по стилю к имитируемому автору
- Использование для фишинга
- Цель: спровоцировать объект атаки (собеседника) на нарушение правил информационной безопасности (клик на ссылку, вредоносный файл, неправомерные или ошибочные действия)
- Сочетание с методами социальной инженерии

**DeepFake**

- Deepfake (дипфейк): «deep learning» + fake
- Синтез изображения, используется ИИ
- Часто используют генеративно-состязательные нейросети (GAN).
- Поддельные видео, новости и вредоносные обманы.
- Ролики на YouTube или Vimeo.
- Есть ПО для распознавания факта вмешательства в видео, например, Microsoft Video Authenticator

**Атаки на модели машинного обучения**

- Извлечение: кража и обратный инжиниринг модели
- Уклонение: незначительное изменение значений признаков для обмана модели
- Отравление данных: манипуляция входными данными для изменения логики работы модели

**Генерация фишинга**

- Используется персонифицированная информация об объекте атаки, позволяющая сделать фишинговое письмо эффективным
- Контент и характеристики фишингового письма модифицируются для сокрытия природы фишинга
- Внешний вид фейкового сайта максимально похож на оригинал внешне (для человеческого глаза), но имеет иную структуру, чем оригинал (невозможно обнаружить похожесть оригинала и фейка автоматизированными методами)

Тестовые вопросы по модулю «Состязательные модели в ИБ»

| № | Вопрос |
|---|---|
| 1 | В каких задачах может быть использовано «обучение без учителя»:<br>**1. Понижение размерности**<br>2. Классификация<br>3. Регрессия<br>**4. Кластеризация** |
| 2 | Какие методы могут быть использованы для понижения размерности<br>**1. Анализ главных компонент**<br>**2. TSNE**<br>3. Анализ второстепенных компонент<br>4. TSMNG |

| | |
|---|---|
| 3 | Какие два важным элемента входят в архитектуру автокодировщика:<br>**1. Кодировщик**<br>2. Шифровальщик<br>**3. Декодер**<br>4. Транскриптор |
| 4 | Какие важные элементы входят в архитектуру генеративных состязательных сетей<br>**1. Генератор**<br>2. Дешифратор<br>3. Анализатор<br>**4. Дескриминатор** |
| 5 | Какое утверждение лучше всего описывает процесс обучения GAN<br>**1. Борьба дискриминатора и генератора**<br>2. Заморозка весов дискриминатора<br>3. Оптимизация весов генератора<br>4. Быстрое повышение точности генератора до 1. |
| 6 | Для каких задач могут быть использованы Transformers+Attention<br>**1. Машинный перевод текстов**<br>**2. Аннотация картинок**<br>3. Предсказание погоды<br>4. Классификация посетителей магазина |
| 7 | Для решения каких задач могут применяться генеративные состязательные сети<br>**1. Генерация аватаров**<br>**2. Создание произведений искусства**<br>**3. DeepFake**<br>4. Классификация рукописных цифр |
| 8 | С использованием каких инструментов создается DeepFake<br>**1. GAN**<br>**2. CycleGAN**<br>3. Логистическая регрессия<br>4. SVM |
| 9 | Какие модели можно отнести к генеративным?<br>**1. GAN**<br>**2. GPT-3**<br>3. DBSCAN<br>4. Линейная регрессия |

| | |
|---|---|
| 10 | Какие бывают виды атак на машинное обучение<br>**1. Кража и обратный инжиниринг модели**<br>**2. Уклонение (незначительное изменение признаков)**<br>**3. Отравление данных**<br>4. Переименование модели |

**Управление ролями пользователей**

Знакомство с процессом извлечения ролей (role mining), рассмотрение типовых алгоритмов role mining, оптимизация результата.

**Реализация разграничения доступа в компьютерной системе**

Ввести понятие монитора безопасности, как основного метода разграничения доступа в КС.

Ссылки:

Н.А. Гайдамакин, «Учебно-методический комплекс. Теоретические основы компьютерной безопасности», раздел 1.3.

**Политика разграничения доступа**

Ввести множество всех возможных доступов (P), которое делится на 2 класса (легитимные и нелегитимные доступы). Ввести понятие политики разграничения доступа как способа задания класса легитимных доступов.

Рассказать о матрице доступа на примере дискреционного доступа.

Рассказать о мандатной модели на примере упрощенной модели Белла Лападулы (правила No read up и No write down).

Н.А. Гайдамакин, «Учебно-методический комплекс. Теоретические основы компьютерной безопасности», разделы 2.1, 2.2.

**Процесс управления доступом**

Ввести общее понятия процесса управления доступом и дать обзор его процедур.

Пояснить, что он относится к процедурному уровню обеспечения ИБ.

Ссылки: https://www.invensislearning.com/blog/itil-access-management/

**Процесс управления доступом**

Пояснить какие варианты запроса доступа могут быть (различные виды инициации процесса управления доступом).

**Процесс управления доступом**

Уточнить, что проверка разбивается на две подпроцедуры: кто инициировал запрос (точно ли тот, за кого он себя выдает) и собственно, легитимность самого запроса – можно ли предоставить доступ, фигурирующий в запросу.

**Процесс управления доступом**

Пояснить, что процедура корректировки доступа в КС одна из самых быстрых и не требующих больших трудозатрат, но при этом чаще всего пытаются автоматизировать именно эту процедуру.

**Процесс управления доступом**

Отметить, что предоставление доступа – это далеко не конец процесса.

Привести примеры основных событий, требующих пересмотра прав доступа: отпуск, смена должности, перевод в другое подразделение и пр.

**Процесс управления доступом**

Отметить, что доступы также рекомендуется журналировать, но акцентировать внимание на затратах, связанных с журналированием (нагрузка на оборудование, необходимость хранения журналов).

Также отметить, что, помимо выявления подозрительных действий, журналирование – хороший способ получения информации для актуализации прав доступа (например, удаления неиспользуемых прав у пользователя).

**Процесс управления доступом**

Отметить, что задача прекращения доступа может быть даже сложнее, чем задача предоставления доступа. Так как при блокировании доступа надо учесть все виды предоставленного доступа, а также различные изменения, которые вносились по ходу мониторинга (например, передача прав заместителю и пр.).

**Проблемы практической реализации управления доступом**

Процесс предоставления доступа порождает противоречие при попытках его оптимизации:

С одной стороны, сложность процесса предполагает необходимость укрупнения прав – например, давать доступ к целой компьютерной системе разом.

С другой стороны, есть базовые принципы наименьших привилегий и подход «запрещено все, что не разрешено явно» - это требует делать гранулированные права доступа.

Но без укрупнения прав не обойтись – если согласовывать индивидуальные доступы к сотням тысяч объектов защиты, это займет годы, что противоречит здравому смыслу.

Таким образом, надо решать задачу укрупнения прав, но с соблюдением принципа минимальных привилегий и пр.

**Кто согласует доступ?**

Пояснить, что для организации, в первую очередь, существуют бизнес-процессы. Процессы делятся на отдельные операции. И что у процессов, что у операций есть хозяева – лица, ответственные за все, в том числе, за предоставление полномочий персоналу.

Внутри бизнес-операции силами хозяина можно определить задействованные информационные активы и компьютерные системы, а также субъектов доступа – персонал операции. Именно хозяин(владелец) должен определять правила корректного использования активов и выстраивать работу персонала в рамках операции.

Таким образом, именно хозяева операций – основные согласующие доступ лица для активов, которые задействованы в их операции. Также хозяева операций могут назвать отличия доступа разных ролей персонала внутри своей операции.

Ссылки:

ГОСТ Р ИСО/МЭК 27002-2012 «Информационная технология. Методы и средства обеспечения безопасности. Свод норм и правил менеджмента информационной безопасности», раздел 7.

**Что нужно чтобы найти владельцев активов?**

Отметить, что в реальной жизни процесс поиска владельца активов непрост: он предполагает, что в организации высокий уровень зрелости процессов, а это далеко не всегда так. Нередко в организации не выполнен даже первый шаг: идентификация процессов и назначение хозяев.

**Ролевая модель управления доступом**

Идея ролевой модели управления доступом – объединить в рамках одной сущности сразу много объектов и операций (видов доступа). При этом пару «операция», «объект» будем называть транзакцией. При описании ролевой модели, как правило, используют именно транзакции, а не отдельно объекты и отдельно операции.

**Как формируется роль?**

Отметить, что роль достаточно естественно формируется из описания процесса (операции). При этом не нарушается принцип минимальности привилегий, так как в роль включаются именно те транзакции, что нужны для выполнения операции. Уменьшение

множества транзакций хотя бы на один элемент приведет к невозможности выполнения операции.
Опять отметить, что все это верно только при наличии хорошо задокументированных процессов в организации.

**Стандарт Role-based access control (RBAC)**

Познакомить со стандартом RBAC – наиболее знаменитым стандартом, описывающим принципы ролевого управления доступом.
Упомянуть про уровни RBAC и образуемую ими «решетку».
Ссылки: https://csrc.nist.gov/Projects/Role-Based-Access-Control

**RBAC. Иерархия ролей**

Отметить, что нередко роли могут вкладываться друг в друга – при этом нижележащая роль включает в себя все транзакции роли-предка.

**RBAC. Ограничения**

Также отметить, что RBAC отличает роли авторизованные (которые потенциально назначены пользователю) и активные (те, которые пользователь использует в настоящий момент времени). Такой подход позволяет проводить дополнительные проверки правил разделения полномочий и блокировать активацию роли, если возникает такое нарушение.

**Процесс управления доступом**

Заметить, что при переходе к ролевой модели сам процесс управления доступом особо не меняется. Отличие только в том, что теперь запрашивается и предоставляется не отдельный доступ к объекту, а сразу роль.

**Связь с процессом управления ролями**

Также отметить, что для ролевой модели управления доступом нужен каталог ролей, который нужно поддерживать в актуальном состоянии. Это приводит к необходимости реализации отдельного процесса – управления ролями.

**Процесс управления ролями**

Процесс начинается с первичного заполнения каталога ролей. Это наиболее трудоемкий шаг, на всех последующих итерациях он будет требовать на порядки меньших трудозатрат.
При этом есть два способа наполнения: сверху вниз – от бизнес процессов и снизу вверх – от фактически назначенных транзакций.
В организации не бывает абсолютно одинаковых работников. А значит и обязанности хоть немного, но будут отличаться. Это приводит к тому, что роль может иметь вполне типовое наполнение транзакциями, но у отдельных сотрудников будут возникать свои отклонения от основной роли – исключения. Все исключения должны идентифицироваться и уточняться. Возможно, исключение может быть отброшено и наблюдаемое отклонение просто следствие некорректного предоставления доступа работнику.
Предыдущие шаги являются зоной ответственности выделенного аналитика ролевой модели организации, а хозяева процессов и операций выступают лишь в роли консультанта. Но шаг «сертификация ролей» выводит хозяев на первый план – именно они должны рассмотреть результаты работы аналитиков ролевой модели и принять или не принять сформированные роли.
После завершения формирования каталога ролей в него неизбежно будут вноситься изменения: меняются процессы, инструменты, организационно-штатная структура. Внесение изменений в ролевую модель запускает весь цикл процесса по новой.

**Построение ролевой модели сверху вниз**

Построение ролевой модели сверху вниз – очень трудоемкий процесс, но он гарантирует наилучший результат. Пояснить, что здесь мы идем от более крупных сущностей к

меньшим: анализируя процесс, определяем состав операций в нем, далее определяем бизнес-роли – т.е. различные наборы обязанностей персонала, задействованного в операции. На основе бизнес ролей и сведений об автоматизации операции выясняем набор системных ролей (т.е. непосредственно сущностей отдельных КС), из которых состоит бизнес роль. После этого можно сформировать конкретные наборы транзакций, входящие в системную роль.

Самое важное, что при таком подходе проще всего преодолеть границу «бизнеса» и «техники», т.е. тот масштаб формирования ролевой модели, где зона ответственности от представителей бизнес-подразделений переходит к представителям ИТ-подразделения.

**Построение ролевой модели сверху вниз – алгоритм**

Пройти по шагам алгоритма. Еще раз отметить, что он трудоемок, так как выполняется вручную и требует участия руководителей различного уровня.

**Построение ролевой модели снизу вверх**

Описать вариант снизу вверх, когда мы движемся от транзакций в сторону процесса. Этот подход хорош тем, что первые шаги достаточно легко автоматизируются (сбор текущих назначений прав, формирование системных ролей на их основе).

Но отметить, что в этом подходе очень сложно преодолеть «границу бизнеса и техники». Особенно, если в организации низкий уровень зрелости управления процессами.

**Общие подходы к извлечению ролей**

Остановиться на общих принципах role mining: мы не смотрим внутрь системы, для нас она – просто набор доступных транзакций.

Системные роли в большинстве случаев будут плоским списком, а вся иерархия будет реализовываться на уровне бизнес ролей.

И отметить, что границей role mining является именно построение системных ролей.

**Извлечение ролей: входные и выходные данные**

Проговорить входные данные и RBAC-состояние. Пояснить суть каждого элемента.

Ссылки:

SACMAT '09: Proceedings of the 14th ACM symposium on Access control models and technologies, June 2009 Pages 95–104, https://doi.org/10.1145/1542207.1542224

**Извлечение ролей: оценка качества результата**

Ввести взвешенную структурную сложность RBAC-состояния, напомнить понятия транзитивного замыкания и $L_1$ нормы. Заметить, что для матриц RBAC-состояния $L_1$ норма – это просто количество единиц.

Продемонстрировать разные оптимизационные задачи, задаваемые разными значениями вектора W

Ссылки:

SACMAT '09: Proceedings of the 14th ACM symposium on Access control models and technologies, June 2009 Pages 95–104, https://doi.org/10.1145/1542207.1542224

**Последующие задачи**

Отметить, что есть также много других задач, связанных с управлением ролями, где могут применяться технологии искусственного интеллекта и машинного обучения.

В частности, сопоставление системных и бизнес ролей на основе различных атрибутов пользователя (должность, место работы, положения должностной инструкции и т.д.).

Отметить, что многие из этих задач значительно сложнее первичного извлечения ролей и в настоящее время там нет универсальных решений.

Вопросы к экзамену по модулю «Ролевая модель в ИБ»

| № | Вопрос |
|---|---|

| | |
|---|---|
| 1 | Что такое политика разграничения доступа?<br>1. общее руководство для действий и принятия решений, которое облегчает достижение целей разграничения доступа<br>2. Конкретная конфигурация ПО, содержащая сведения о допустимых действиях субъектов над объектами<br>**3. Способ задания множества санкционированных операций субъектов над объектами**<br>4. Способ описания множества всех допустимых операций (доступов) в компьютерной системе |
| 2 | Что такое управление доступом?<br>1. Установка паролей<br>2. Внесение пользователей в базу данных<br>3. **Любые действия, связанные с изменением политики разграничения доступом**<br>4. Разрешение администрирования информационных систем |
| 3 | Что относится к актуальным проблемам управления доступом?<br>**1. Большое количество сотрудников (субъектов)**<br>2. Отсутствие механизма разграничения доступа в ПО<br>3. Необходимость получать сертификат администратора<br>**4. Большое количество информационных активов (объектов)** |
| 4 | Кто является владельцем объекта доступа (информационного актива)?<br>1. Тот, кто создал этот объект в компьютерной системе<br>2. Администратор компьютерной системы<br>**3. Хозяин бизнес-операции**<br>4. Руководитель организации |
| 5 | Что такое *транзакция* в ролевой модели?<br>1. минимальная логически осмысленная операция, которая имеет смысл и может быть совершена только полностью<br>**2. совокупность операции и объекта доступа**<br>3. совокупность субъекта доступа и объекта доступа<br>4. совокупность всех допустимых операций субъектов над объектами |
| 6 | Что из перечисленного **входит** в стандарт RBAC?<br>**1. Базовые принципы управления доступом на основе ролей**<br>**2. Описание иерархии ролей**<br>**3. Ограничения применения ролей**<br>4. Порядок формирования базового списка ролей |

| | |
|---|---|
| 7 | Расположите процессы управления ролями в правильном порядке:<br>1. а) Формирование каталога ролей, б) актуализация каталога ролей, в) сертификация каталога ролей, г) уточнение состава ролей, ограничений и исключений.<br>**2. а) Формирование каталога ролей, б) уточнение состава ролей, ограничений и исключений, в) сертификация каталога ролей, г) актуализация каталога ролей.**<br>3. а) Формирование каталога ролей, б) сертификация каталога ролей, в) уточнение состава ролей, ограничений и исключений, г) актуализация каталога ролей,<br>4. а) Сертификация каталога ролей, б) актуализация каталога ролей, в) уточнение состава ролей, ограничений и исключений, г) Формирование каталога ролей |
| 8 | Что из перечисленного относится к области знаний хозяина бизнес-процесса или бизнес-операции?<br>**1. Бизнес-функция**<br>2. Системная роль<br>3. Права доступа<br>**4. Бизнес-процесс** |
| 9 | Какая проблема является общей для подхода к формированию списка ролей сверху вниз и снизу вверх?<br>1. Необходимость интервьюирования руководящего состава организации<br>2. Ручной сбор исходных данных<br>3. Необходимость предобработки данных о фактически назначенном доступе пользователям<br>4. **Сложность сопоставления бизнес ролей и системных ролей** |
| 10 | Как оценивается качество алгоритма извлечения ролей?<br>1. По количеству сформированных ролей (чем меньше, тем лучше)<br>2. По количеству прямых назначений прав пользователей (чем меньше, тем лучше)<br>3. По суммарному количеству назначений ролей пользователям и прав ролям (чем меньше, тем лучше)<br>**4. Все вышеперечисленное, в зависимости от весовой функции** |

## Предиктивная аналитика в ИБ
## Аналитика больших данных

Аналитика данных – это поиск, интерпретация и информирование о закономерностях в массивах данных, позволяющих оптимизировать анализируемый процесс: предотвращение инцидентов, повышение эффективности процессов, сокращение трудозатрат на сопровождение.

Аналитику данных можно разделить условно на несколько видов:

**Описательная (дескриптивная) аналитика** отвечает на вопрос «Что случилось?» и выполняется при возникновении инцидента. В качестве примера можно привести логи операционной системы, полученные во время очередного инцидента.

**Диагностическая аналитика** отвечает на вопрос «Почему случилось?». На этом уровне аналитики выясняется причины произошедшего инцидента, или делается оценка вероятности.
**Предиктивная аналитика** отвечает на вопрос «Что может случиться?». Данный вид аналитики отвечает за прогнозирование вероятностей будущих событий: раннее предотвращение неисправностей, инцидентов.
И, наконец, **предписывающая аналитика** позволяет не только прогнозировать поведение системы, но и давать рекомендации о возможных способах предотвращения негативных сценариев развития. Здесь идет речь об автоматической реакции системы на события, которые с высокой долей вероятности приведут к негативным последствиям.
С увеличением уровня аналитики как правило возрастает сложность применяемых моделей и методов, но снижается степень участия человека в управлении и контроле за системами.
https://www.bigdataschool.ru/blog/types-of-data-analytics.html

## Жизненный цикл аналитики данных

Данные могут помочь ответить на многие вопросы, возникающие у бизнеса. Чтобы извлечь пользу из данных, необходимо для начала сформулировать проблему, вопрос или гипотезу, которую вы хотите проверить. Без формулировки целей невозможно создать качественную аналитику, вы можете в итоге ответить совсем не на тот вопрос, который волнует постановщика, или потратить большое количество времени на анализ элементов, не относящихся к поставленной проблеме. Постановка задачи влияет на все этапы жизненного цикла аналитики: выбор данных, их обработку, выбор типа аналитики и, конечно, на то, в каком виде вы эти результаты представите.
Для того, чтобы извлечь полезную информацию из данных обычно необходимо вначале эти данные собрать, потом обработать и привести к единому формату, только после этого можно использовать данные для аналитики. Результаты аналитики, если они достигают изначально поставленной цели, должны быть сохранены в базе данных, переданы в другую информационную систему, на их основании может быть подготовлен отчет, произведено оповещение или другие действия, включая реакцию системы в случае предписывающей аналитики.
Таким образом, для того, чтобы правильно подобрать тип аналитики, собрать нужные данные и корректно их обработать, необходимо представлять какую задачу вы хотите решить и в каком виде от вас требуется результат.
https://www.jigsawacademy.com/blogs/hr-analytics/data-analytics-lifecycle/

## Источники данных

Источниками данных могут быть различные базы данных. В текущий момент существует множество различных баз, отличающихся быстродействием, ориентацией на определенный тип данных, на определенные запросы и т.п.
https://jino.ru/journal/articles/7-baz-dannyh/
Кроме того, данные можно забирать из открытых источников, например веб-сервисов или открытых хранилищ данных.
https://habr.com/ru/post/331036/
https://te-st.ru/2014/02/18/open-data-sources-russia/
https://yandex.ru/promo/oda/useful
Кроме того, данные могут собираться в реальном времени от сетевого оборудования, различных датчиков, устройств, систем мониторинга и так далее.

## Визуализация данных

Визуализация данных – важный этап анализа, так как зачастую некачественная визуализация может испортить самый хороший анализ. Важно донести до конечного потребителя выводы в понятной форме, поэтому на самом первом этапе визуализации

данных необходимо четко понимать, кому вы будете ее показывать и на какой вопрос вы будете отвечать.
Часто бывает, что после сбора данных, подбора типов и структуры визуализации становится понятно, что визуализация не достигает изначально поставленной цели, тогда нужно вернуться на несколько шагов назад и повторять эти шаги до тех пор, пока визуализация не будет отвечать на изначально сформулированный вопрос.
https://www.owox.ru/blog/articles/data-visualization/
https://vc.ru/services/204235-top-15-luchshih-instrumentov-vizualizacii-dannyh-v-2020-2021-godah-s-primerami

**Предиктивная аналитика**

Предиктивная аналитика, как мы говорили ранее, отвечает на вопрос «Что может случиться?». На первый взгляд очевидно, что речь идет о прогнозировании, но это не совсем так. Если углубиться в вопрос, то станет понятно, что недостаточно только прогнозировать ситуации и параметры, но и следить за текущей ситуацией, отслеживая возникающие аномалии. Да, мы не всегда сможем сказать, к чему эти аномалии приведут, но мы как минимум уведомим оператора системы о том, что система работает некорректно. Кроме выявления аномалий важно оценить уже произошедшие инциденты и разобраться в их причинах. Этот тип анализа называется анализ корневых причин (RCA, Root Cause Analysis) и он также является инструментом предоставления новой информации для этапа выявления аномалий.
Таким образом, задача предиктивной аналитики состоит не только в прогнозировании, но и в оценке существующей ситуации, а также в диагностике произошедших инцидентов.
Предиктивная аналитика может выполняться вручную, но это достаточно неэффективно, поэтому для ускорения процесса возможно и необходимо использовать методы машинного обучения.

**Типы анализируемых данных**

В процессе анализа данных вы будете сталкиваться с различными типами данных: трафик, логи, видео, тексты и так далее. Для каждого типа данных и каждой задачи существует множество способов анализа. В нашей лекции мы остановимся на временных рядах. Следует учитывать, что в виде временного ряда можно представить, например, логи. Для этого можно брать количество сообщений в момент времени, или частоту возникновения определенных сообщений.

**Временные ряды**

Временной ряд – собранный в разные моменты времени статистический материал о значении каких-либо параметров (в простейшем случае одного) исследуемого процесса. Каждая единица статистического материала называется измерением или отсчётом, также допустимо называть его уровнем на указанный с ним момент времени. Во временном ряде для каждого отсчёта должно быть указано время измерения или номер измерения по порядку. Временной ряд существенно отличается от простой выборки данных, так как при анализе учитывается взаимосвязь измерений со временем, а не только статистическое разнообразие и статистические характеристики выборки.
https://ru.wikipedia.org/wiki/%D0%92%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B9_%D1%80%D1%8F%D0%B4
https://blog.skillfactory.ru/glossary/vremennoj-ryad-2/
https://www.ibm.com/docs/ru/spss-statistics/SaaS?topic=forecasting-introduction-time-series

**Примеры временных рядов**

Временной ряд может отображать только один или несколько параметров во времени. В первом случае ряд называется одномерным, во втором – многомерным. Из временного ряда можно извлечь сезонность и тренд, это можно сделать, например, при помощи

библиотеки statsmodels (Python). Но некоторые ряды могут не включать в себя тренд и/или сезонность, тогда их прогнозирование становится сложнее.

**Прогнозирование для временных рядов**

Для прогнозирования временных рядов могут использоваться статистические модели, особенно это относится к рядам с ярко выраженным трендом и сезонностью. Кроме того, неплохо с задачей прогнозирования справляются рекуррентные нейронные сети, в модель которых заложена долговременная зависимость. Наконец, существуют специализированные библиотеки и фреймворки для прогнозирования.

https://habr.com/ru/post/553658/

https://habr.com/ru/post/559796/

https://vc.ru/dev/286478-prognozirovanie-vremennyh-ryadov-s-pomoshchyu-prophet

**Прогнозирование. SARIMA**

Модель SARIMA моделирует следующий шаг в последовательности как линейную функцию разностных наблюдений, ошибок, а также их сезонных составляющих. Модель SARIMA расширяет основную модель ARIMA, включая в моделирование сезонную компоненту.

Модель ARIMA включает в себя модель авторегрессии, дифференцирования и скользящего среднего.

где – измерение в момент времени $t$

$c$ – начальный коэффициент (значение модели при нулевых влияющих факторах)

$p$ – порядок авторегрессии (количество предыдущих шагов, которые учитывает модель)

– коэффициент авторегрессии (описывает влияние предыдущих шагов на модель)

$q$ – порядок модели скользящего среднего

– коэффициент модели скользящего среднего

– оператор разности временного ряда порядка d

– случайная компонента (погрешность модели)

Для качественного анализа временные ряды должны быть стационарными, проверить это можно при помощи критерия Дики-Фуллера. В случае нестационарности ряда его приводят к стационарности при помощи логарифмирования, метода Бокса-Кокса, разностного дифференцирования. При применении преобразований важно не забыть применить обратные преобразования при построении прогноза.

https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/

В.В. Домбровский. Эконометрика. http://sun.tsu.ru/mminfo/2016/Dombrovski/start.htm

**Прогнозирование. LSTM**

Рекуррентные нейронные сети (РНС, англ. Recurrent neural network, RNN) — вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки. В отличие от многослойных перцептронов, рекуррентные сети могут использовать свою внутреннюю память для обработки последовательностей произвольной длины. В последнее время наибольшее распространение получили сеть с долговременной и кратковременной памятью (LSTM) и управляемый рекуррентный блок (GRU).

https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C

https://habr.com/ru/post/487808/

LSTM (long short-term memory, дословно (долгая краткосрочная память) — тип рекуррентной нейронной сети, способный обучаться долгосрочным зависимостям. LSTM специально разработаны для устранения проблемы долгосрочной зависимости. Их специализация — запоминание информации в течение длительных периодов времени.
https://neurohive.io/ru/osnovy-data-science/lstm-nejronnaja-set/
http://colah.github.io/posts/2015-08-Understanding-LSTMs/
LSTM можно использовать для прогнозирования значений временных рядов, причем существует множество разновидностей архитектур LSTM, в том числе для прогнозирования многомерных временных рядов.
https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

**Выявление аномалий**

Используемые для поиска аномалий во временных рядах методы принято разделять на группы:

· proximity-based: выявление аномалии на основе информации о близости параметров или последовательности параметров фиксированной длины, подходит для выявления точечных аномалий и выбросов, но не позволит выявить изменения в форме сигнала

· prediction-based: построение прогнозной модели и сравнение прогноза и фактической величины, лучше всего применимо ко временным рядам с выраженными периодами, циклами или сезонностью

· reconstruction-based: методы, основанные на реконструкции фрагментов данных, используют восстановление (реконструкцию) фрагмента данных, поэтому может выявлять как точечные аномалии, так и групповые аномалии, в том числе изменения в форме сигнала.

Proximity-based методы ориентированы на поиск значений, существенно отклоняющихся от поведения всех остальных точек. Самый простой и наглядный пример реализации такого метода – контроль превышения заданного порога значений.
В prediction-based методах основная задача – построить качественную модель процесса, чтобы смоделировать сигнал и сравнить полученные смоделированные значения с исходными (истинными). Если предсказанный и истинный сигнал близки, то поведение считается «нормальным», а если значения в модели сильно отличаются от истинных, то поведение системы на этом участке объявляется аномальным.
Оригинальный подход используется в reconstruction-based моделях – сначала модель обучают кодировать и декодировать сигналы из имеющейся выборки, при этом закодированный сигнал имеет гораздо меньшую размерность, чем исходный, поэтому модели приходится учиться «сжимать» информацию.
Методы прогнозирования мы рассмотрели в предыдущем модуле, примеры Proximity-based и Reconstruction-based методов мы рассмотрим чуть ниже.
Перед выбором метода выявления аномалий следует определиться, что является для нас аномалией и выбрать метрику, по которой мы будем оценивать качество работы модели.
https://habr.com/ru/post/588320/

**Выявление аномалий. LOF**

Локальный уровень выброса – один из алгоритмов поиска аномалий, основанный на близости (Proximity-based). Это алгоритм кластеризации, имеющий общие концепции с DBSCAN и OPTICS, базирующийся на идее локальной плотности, которое в свою очередь использует определение «расстояния достижимости» (формулы локальной плотности и расстояния достижимости представлены на слайде). В отличие от других методов LOF позволяет выявлять выбросы при разных плотностях кластеров. С другой стороны не определены правила определения порогового значения LOF, оно

подбирается эмпирически, поэтому зачастую результаты анализа сложно интерпретировать.
https://towardsdatascience.com/local-outlier-factor-lof-algorithm-for-outlier-identification-8efb887d9843
https://medium.com/datasciencearth/local-outlier-factor-7821b5651bc5
https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C_%D0%B2%D1%8B%D0%B1%D1%80%D0%BE%D1%81%D0%B0#:~:text=%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9%20%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C%20%D0%B2%D1%8B%D0%B1%D1%80%D0%BE%D1%81%D0%B0%20%D1%8F%D0%B2%D0%BB%D1%8F%D0%B5%D1%82%D1%81%D1%8F%20%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%BC,%D1%82%D0%BE%D1%87%D0%BA%D0%B8%20%D1%81%20%D1%83%D1%87%D1%91%D1%82%D0%BE%D0%BC%20%D0%B5%D1%91%20%D1%81%D0%BE%D1%81%D0%B5%D0%B4%D0%B5%D0%B9.

**Выявление аномалий. AE**

Autoencoder (автокодер, автоэнкодер, AE) — специальная архитектура искусственных нейронных сетей, позволяющая применять обучение без учителя при использовании метода обратного распространения ошибки. Простейшая архитектура автокодировщика — сеть прямого распространения, без обратных связей, наиболее схожая с перцептроном и содержащая входной слой, промежуточный слой и выходной слой. В отличие от перцептрона, выходной слой автокодировщика должен содержать столько же нейронов, сколько и входной слой.
Автоэнкодер состоит из двух частей:
· Энкодер: отвечает за сжатие входа в латентное пространство. Представлен функцией кодирования h = f (x);
· Декодер: предназначен для восстановления ввода из латентного пространства. Представлен функцией декодирования h = f (x).
Основными практическими приложениями автокодировщиков остаются уменьшение шума в данных, а также уменьшение размерности многомерных данных для визуализации.
Для выявления аномалий во временных рядах автоэнкодеры чаще всего строятся на LSTM ячейках (для учета долговременных связей). Вначале автоэнкодер обучается на данных без аномалий, и определяется пороговое значение ошибки восстановления на основании разницы входного и восстановленного вектора. Превышение этого значения будет в последствии означать потенциальную аномалию.
После обучения автоэнкодер анализирует данные в реальном времени, позволяя выявлять практически все типы аномалий (точечные, групповые, изменение частоты, появление тренда).
https://neurohive.io/ru/osnovy-data-science/avtojenkoder-tipy-arhitektur-i-primenenie/
https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BA%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D1%89%D0%B8%D0%BA

**Анализ корневых причин (RCA)**

Анализ корневых причин заключается в поиске первопричины произошедшего инцидента.
Анализ корневых причин применяется практически во всех сферах жизни:
- Производство
- Медицина
- ИТ инфраструктура

- Менеджмент качества

Чаще всего это описательные методы и ручная аналитика:

- Байесовский вывод
- Анализ видов и последствий отказов
- Анализ дерева отказов
- Диаграмма Исикавы
- Правило Парето и т.д.

С другой стороны, существует множество методов машинного обучения для анализа корневых причин.
https://www.researchgate.net/publication/313097743_Survey_on_Models_and_Techniques_for_Root-Cause_Analysis
https://medium.datadriveninvestor.com/root-cause-analysis-in-the-age-of-industry-4-0-9516af5fb1d0
Несмотря на то, что анализ корневых причин – это по сути диагностическая аналитика, информация, параметры и их зависимости, определенные на этапе RCA могут стать дополнительной информацией при выявлении аномалий, поэтому процесс RCA должен быть связан с процессом выявления аномалий и прогнозирования.

**RCA с учителем**

Чаще всего применение RCA с учителем заключается в изначальном обучении и классификации возможных аномалий и их причин. Допустим, можно разбить все инциденты с ПК на проблемы с оперативной памятью, процессором, винчестером и видеокартой. Конечно, чаще всего классификация гораздо сложнее, тем не менее основная проблема подхода с учителем – в случае возникновения неизвестной аварии, алгоритм вряд ли поможет найти ее причину.
https://aaltodoc.aalto.fi/bitstream/handle/123456789/36347/master_Kahles_Bastida_Julen_2019.pdf?sequence=1&isAllowed=y
https://www.zebrium.com/resources/wp-ml-based-autonomous-incident-detection
https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0261-9
https://www.researchgate.net/publication/347268378_Assembly_Line_Anomaly_Detection_and_Root_Cause_Analysis_Using_Machine_Learning
Методы классификации и выявления аномалий могут быть любыми, в зависимости от поставленной задачи.

**RCA без учителя**

Здесь рассматривается класс методов, который не требует предварительной классификации корневых причин. При этом возможно первоначальное обучение модели на данных без аномалий для последующего определения аномалий. Формально такой вид обучения можно назвать semi-supervised.
Из-за разнообразия сфер применения и данных, нет единого подхода или SOTA для этих типов задач. Для решения чаще всего используются Байесовские и Марковские сети, самоорганизующиеся карты, автокодировщики и так далее.
В качестве примера на слайде приведен кейс, рассмотренный в статье https://uu.diva-portal.org/smash/get/diva2:1178780/FULLTEXT01.pdf, где исследователи применять самоорганизующиеся карты (SOM) для выявления аномалий и поиска корневых причин в работе облачного видеосервиса.
На слайде приведена схема работы их модуля:

- Определение аномалии (SOM)
- При достижении порогового количества аварий запуск механизма RCA (SOM)
- Вычисление векторов различий (dissimilarity вектор) и определение трех наиболее отличающихся векторов, которые и будут предположительно являться причинами инцидента.

**RCA. Self-organizing map (SOM)**

Самоорганизу́ющаяся ка́рта Ко́хонена (англ. Self-organizing map — SOM) — нейронная сеть с обучением без учителя, выполняющая задачу визуализации и кластеризации. Идея сети предложена финским учёным Т. Кохоненом. Является методом проецирования многомерного пространства в пространство с более низкой размерностью (чаще всего, двумерное), применяется также для решения задач моделирования, прогнозирования, выявление наборов независимых признаков, поиска закономерностей в больших массивах данных, разработке компьютерных игр, квантизации цветов к их ограниченному числу индексов в цветовой палитре: при печати на принтере и ранее на ПК или же на приставках с дисплеем с пониженным числом цветов, для архиваторов [общего назначения] или видео-кодеков, и прч. Является одной из версий нейронных сетей Кохонена.

https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BC%D0%BE%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D1%83%D1%8E%D1%89%D0%B0%D1%8F%D1%81%D1%8F_%D0%BA%D0%B0%D1%80%D1%82%D0%B0_%D0%9A%D0%BE%D1%85%D0%BE%D0%BD%D0%B5%D0%BD%D0%B0

Метод работает по следующему циклу:

1. Инициализация весов нейронов
2. Выбор произвольного вектора из обучающей выборки
3. Выбор ближайшего к вектору узла сети SOM (в оригинале используется Евклидово расстояние, но также применяется Манхэттенское и DTW).
4. Ближайший узел сети называется победителем (Best Matching Unit, BMU) и веса всех узлов пересчитываются.
5. Шаги 2-4 повторяются для всех векторов обучающей выборки.

В конечном итоге узлы сети "растягиваются" в соответствии с положением векторов.

Особенности:

· Зависимость от изначально выбранной топологии и инициализации весов;

· Возможность кластеризации и поиска аномалии в многомерных рядах (но при этом теряется временная зависимость);

· Если поверх нейронов обученной SOM наложить еще один алгоритм кластеризации, можно улучшить результат работы сети.

https://towardsdatascience.com/kohonen-self-organizing-maps-a29040d688da
https://loginom.ru/blog/som
https://habr.com/ru/post/334810/
https://habr.com/ru/post/334220/
https://habr.com/ru/post/338868/

Вопросы к экзамену по теме «Предиктивная аналитика в ИБ»

| № | Вопрос |
|---|---|

| | |
|---|---|
| 1 | Выберите тип аналитики, который наиболее точно решает следующую задачу «В компании, занимающейся аналитикой и предоставляющей облачный сервис аналитики данных случилась неприятность: хранилище данных оказалось переполнено и по этой причине сервис был недоступен около 4 часов, что привело к оттоку пользователей. Руководство компании поставило задачу, что о скором переполнении хранилища оно должно узнавать минимум за неделю»:<br>1. Описательная аналитика<br>2. Диагностическая аналитика<br>**3. Предиктивная аналитика**<br>4. Предписывающая аналитика |
| 2 | Насколько важным является определение задачи, проблемы или бизнес-цели в жизненном цикле аналитики данных:<br>**1. Очень важно практически на любом этапе жизненного цикла**<br>2. Важно только на этапе выбора метода аналитики<br>3. Важно, но не принципиально, сам процесс аналитики даст понимание проблемы<br>4. Не важно, так как аналитика является самодостаточным инструментом |
| 3 | Что **НЕ** является источником данных для аналитики:<br>1. База данных транзакций пользователей<br>2. Показания загрузки ЦПУ сервера<br>3. Веб-сервис, предоставляющий данные о температуре уличного воздуха<br>4. Сетевой коммутатор |
| 4 | Вам поставлена задача отобразить динамику изменения количества пользователей вашей системы за последний год, чтобы понять в какой момент произошел основной рост, визуализация будет продемонстрирована топ-менеджеру, у вас есть данные о количестве пользователей за каждый день в течение года. Какой тип визуализации вы выберете:<br>1.Таблицу с отображением количества пользователей за каждый день года, так будет понятен объем обработанных данных, возможно на основании объема руководитель поймет мою ценность<br>2. Круговую диаграмму, где секторы – это количество пользователей в месяц, так будет понятен общий объем пользователей<br>**3. Линейный график количества пользователей за год, для наглядности добавлю линию основного тренда и линию тренда для периода бурного роста**<br>4.Гистограмму количества пользователей по месяцам, для наглядности добавлю показания по дисперсии и математическому ожиданию количества пользователей за год |
| 5 | Выберите наиболее правильную расшифровку аббревиатуры «RCA»:<br>1. Причинно-следственный метод оценки зависимостей<br>**2. Анализ корневых причин**<br>3. Анализ корреляционного взаимодействия<br>4. Причинный анализ |

| | |
|---|---|
| 6 | Могут ли являться логи журнала событий Windows временным рядом?<br>1. Да, как и любой тип данных<br>**2. Да, но понадобится преобразование логов с учетом временных меток событий**<br>3. Нет, логи – это текстовые данные, и они не могут быть преобразованы во временной ряд<br>4. Нет, отсчеты во временном ряде обязаны идти с равным интервалом, а логи в журнал записываются без какой-либо периодичности |
| 7 | Выберите основные характеристики временного ряда (несколько):<br>1. Амплитуда<br>**2. Тренд**<br>**3. Цикличность**<br>4. Количество отсчетов |
| 8 | Какой метод (фреймворк, библиотека) **НЕ** применяется для прогнозирования временных рядов:<br>1. Prophet<br>2. ARIMA<br>3. Рекуррентные нейронные сети<br>**4. Байесовские сети** |
| 9 | Что означает коэффициент *p* в модели ARIMA:<br>1. Коэффициент авторегрессии<br>**2. Порядок авторегрессии**<br>3. Порядок разности временного ряда<br>4. Количество отсчетов в анализируемом временном ряду |
| 10 | Чем отличается архитектура сети LSTM от других нейронных сетей и почему их используют для прогнозирования:<br>1. Каждая ячейка сети LSTM обладает уникальной структурой, которая «подстраивается» под временной ряд.<br>2. **Каждая ячейка сети LSTM имеет встроенные механизмы, которые реализуют долгосрочную зависимость и исключают проблему градиентного взрыва и обнуления градиента**.<br>3. Сеть LSTM в отличие от других нейронных сетей может работать с многомерными временными рядами.<br>4. Для прогнозирования можно использовать любые разновидности рекуррентных сетей, поскольку они сохраняют всю информацию, когда-либо поступившую в сеть. |
| 11 | Какого метода выявления аномалий **НЕ** существует:<br>1. Proximity-based<br>2. Reconstruction-based<br>3. Prediction-based<br>**4. Stochastic-based** |

| | |
|---|---|
| 12 | Алгоритм LOF является разновидностью методов:<br>1. Классификации<br>**2. Кластеризации**<br>3. Регрессии<br>4. Локализации |
| 13 | Отличительной особенностью автоэнкодеров является:<br>**1. Одинаковое количество нейронов на входном и выходном слое**<br>2. Наличие скрытого слоя<br>3. Использование метода обратного распространения ошибки<br>4. Высокая точность прогнозирования |
| 14 | Возможно ли использовать метод LOF для решения задачи RCA с учителем:<br>1. Нет, поскольку LOF решает задачу кластеризации, являющейся задачей обучения без учителя<br>2. Да, поскольку LOF решает задачу классификации, являющейся задачей обучения с учителем<br>3. Да, LOF может применяться непосредственно на этапе поиска корневых причин<br>**4. Да, LOF может применяться на этапе выявления аномалий** |
| 15 | Почему SOM – это «самоорганизующаяся» карта:<br>1. Поскольку инициализация весов происходит произвольно и не влияет на качество обучения<br>**2. Поскольку нейроны SOM организуют кластеры вокруг векторов входного пространства**<br>3. Поскольку веса нейронов SOM вычисляются в строго организованном порядке, который выбирается на этапе инициализации<br>4. Поскольку это алгоритм кластеризации, т.е. обучение без учителя, т.е. самообучение |

**Анализ вредоносного программного обеспечения**

Анализ вредоносного программного обеспечения (malware analysis) — это область исследования функциональных возможностей. целей, происхождения и потенциального воздействия вредоносного ПО.

**Вредоносное программное обеспечение**

Любое программное обеспечение, предназначенное для несанкционированного доступа к вычислительным ресурсам ЭВМ или к информации, хранящейся на ЭВМ, с целью несанкционированного использования ресурсов ЭВМ, причинение вреда владельцу ЭВМ сети ЭВМ или информации путём копирования, искажения, удаления или подмены информации.

Существует множество различных типов вредоносного ПО и также, как различные типы обычного ПО, вредоносное ПО может быть написано на любом языке программирования. с учётом различий целевых сред выполнения и требований к условиям выполнения.

В упакованном и развёрнутом состоянии большинство вредоносных программ существуют в виде бинарных файлов

**Обнаружение вредоносного ПО**

При наличии доступа к коду высокого уровня, относительно просто можно определить, что делает программа, однако большинство вредоносных программ распространяется в бинарном виде, перехватываются и накапливаются “в полевых условиях”:

- отлавливаются в ловушках-песочницах,
- продаются на нелегальных форумах,
- обнаруживаются на компьютерах жертв.

Чаще всего человек не может прочитать бинарные файлы, так как они предназначены для непосредственного выполнения компьютером, следовательно, профилирование характеристик и поведения вредоносной программы становится процессом реверс-инжиниринга., цель которого — понять. что делает эта программа, как если бы имелся исходный код на языке высокого уровня. Без знания контекста интерпретации, стандарта кодирования и алгоритма декодирования сами по себе бинарные данные фактически бессмысленны. Однако система машинного обучения хороша настолько, насколько качественными являются её входные данные. Исходные бинарные данные требуют составления плана сбора данных, их очистки и валидации перед отправкой в модель машинного обучения. Предварительная обработка таких исходных данных важна для выбора их оптимального формата для передачи в модели машинного обучения.

## Пути распространения вредоносного ПО

Вредоносное ПО может быть встроено в бинарные файлы разнообразных форматов, работа которых совершенно отличается друг от друга. Например, PE-файлы в ОС Windows, ELF--файлы в Unix--системах (Executable and Linkable Format) и APK--файлы в ОС Android (формат Android Package Kit с расширением .apk и др.) имеют совершенно различную внутреннюю структуру файлов и требуют разного контекста выполнения.

Широко распространены вредоносные компоненты, внедряющиеся в файлы документов, например с расширениями .doc, .pdf и .rtf, и использующие макросы и динамически выполняемые элементы в структуре документа, чтобы выполнить вредительские действия.

Вредоносное ПО также может быть представлено в форме расширений и динамически подключаемых компонентов (plug-ins) для распространенных программных платформ, таких как веб-браузеры и комплексные рабочие веб-среды.

## Классификация вредоносного программного обеспечения

Группы классификации вредоносного ПО объединяют отдельные экземпляры на основе общих свойств. Можно классифицировать вредоносное ПО разными способами в зависимости от конкретной задачи и поставленной цели.

Например, классификация вредоносного ПО по степени его опасности и функциональности будет полезна группе обеспечения безопасности для определения степени угрозы.

Для обобщённого анализа вредоносного ПО распространённой промышленной методикой является группировка экземпляров по семействам (family).

## Семейства вредоносного ПО (Malware families)

Семейства позволяют прослеживать авторство, коррелировать информацию и идентифицировать новые варианты обнаруженных вредоносных программ. Экземпляры вредоносного ПО из одного семейства могут иметь одинаковый код, возможности, авторство, функциональные характеристики, цели и/или исходные предпосылки.

Различия между экземплярами вредоносного ПО из одного семейства могут определяться по разным компиляторам или секциям исходного кода, добавляемым и/или удаляемым для изменения функциональности вредоносной программы. Экземпляры вредоносных программ, которые со временем эволюционируют в ответ на

изменения стратегий их выявления и нейтрализации, зачастую также демонстрируют сходство между старой и новой версиями,

**Пример семейства вредоносного ПО — Conficker**

Червь, предназначенный для внедрения в ОС Microsoft Widows. Существует множество вариаций червя Conficker с различным кодом, авторами и поведением, однако определённые характеристики этих червей позволяют причислить их к одному семейству, обозначая их общее происхождение: все черви Conficker

- используют уязвимости Windows,
- предпринимают перебор по словарю для подбора пароля к учётной записи администратора.
- Устанавливают скрытое ПО для использования хоста в деятельности ботнета.

**Место безопасных файлов в классификации вредоносного ПО**

В классификации вредоносного ПО также может быть класс бинарных файлов, не являющихся вредоносными, так как основная цель антивирусного ПО — определение вероятности того, что файлу можно доверять и выполнять его в защищаемой среде. Обычно решение такой задачи основано на методе сравнения сигнатур.

**Метод сигнатур**

Метод статически сигнатур позволяет, имея достаточный набор свойств и образцов поведения ранее обнаруженного и исследованного вредоносного ПО, сравнивать новые появляющиеся в системе бинарные файлы с этим набором данных, чтобы определить, нет ли совпадений с каким--либо признаками вредоносного ПО, выявленными ранее.

**Метод сигнатур. Полиморфные вредоносное ПО**

Метаморфные или полиморфные вирусы и черви используют статические и динамические методики маскировки для изменения характеристик своего кода, поведения и свойств, применяемые для алгоритмов генерации сигнатур в механизмах идентификации вредоносного ПО. Такой подход в настоящее время становится все более распространенным из-за успешного создания помех и препятствий для механизмов синтаксических сигнатур вредоносного ПО.

Механизмы статических сигнатур продолжают погоню за постоянно сужающимся набором статических сигналов, которым авторы вредоносного ПО пренебрегают или не могут изменить коренным образом.

**Методы машинного обучения в классификации вредоносного ПО**

Методы машинного обучения могут помочь устранить проблемы, возникающие при использовании метода сигнатур, благодаря следующим свойствам:

- Нечёткое сравнение — алгоритмы машинного обучения могут определять сходство между двумя и более объектами с помощью вещественной метрики расстояния. Экземпляры данных, отображаемые в векторное пространство признаков, могут быть сгруппированы на основании относительны расстояний меду ними. Такая возможность определения приблизительных совпадений между объектами полезна при классификации вредоносного ПО. различия в свойствах которого приводят в замешательство методы статического сравнения сигнатур.
- Автоматизированный выбор свойств — автоматическое определение веса (значимости) признака и выбор самых значимых признаков являются главными аспектами машинного обучения, которые помогают осуществить классификацию вредоносного ПО. На основе статистических свойств тренировочного набора данных можно ранжировать признаки по их относительной важности при определении отличия экземпляра, принадлежащего классу A, от другого экземпляра, принадлежащего классу B. Кроме того, появляется возможность объединения в группу двух экземпляров класса A. Некоторые алгоритмы снижения размерности и выбора признаков способны обнаруживать скрытые свойства экземпляров, которые чрезвычайно трудно выявить

даже эксперту в этой области. Методы машинного обучения освобождают аналитиков вредоносного ПО от некоторых трудоемких операций по определению значимости каждого признака. Позволяя самим данным выявлять и определять набор признаков для использования в схеме классификации.

- Адаптируемость: вечное противостояние между создателями вредоносного ПО и защитниками систем приводит к непрерывному изменению типов и шаблонов предпринимаемых атак. Как и при разработке обычного ПО, вредоносное ПО развивается и улучшается со временем, поскольку его авторы добавляют функциональные возможности и исправляют ошибки.

Кроме того, как было отмечено выше, авторы вредоносного ПО всегда стремятся к непрерывным усовершенствованиям, изменяя поведение своих программ, чтобы избежать обнаружения. С помощью методики нечеткого сравнения и управляемого данными процесса выбора признаков системы классификации вредоносного ПО, реализованные на основе машинного обучения, способны адаптироваться к изменениям исходных условий (входных данных) и отслеживать развитие вредоносного ПО во времени.

Машинное обучение может помочь существенно снизить объем ручной работы и уровень экспертных знаний, требуемый для классификации вредоносного ПО.

## Современные процессы выполнения кода

Существуют 2 основных типа выполнения кода: выполнение компилируемого кода и выполнение интерпретируемого кода.

## Типовой процесс атаки вредоносного программного обеспечения

При исследовании и классификации вредоносного ПО важно понимать, что именно делает вредоносная программа и как возникла уязвимость. Разные типы вредоносного ПО используют различные методы распространения, преследуют разнообразные цели и создают разные уровни угрозы. Однако существует типовой процесс атаки вредоносного ПО.

## Типы поведения вредоносного ПО

Вредоносное ПО часто демонстрирует определённые типы поведения:

- Маскировка своего присутствия — вредоносное ПО использует методики сжатия и шифрования для придания своему коду более компактной и замаскированной формы.
- Стремление к выполнению своей функции. Вредоносная программа должна обеспечить достаточную степень живучести, чтобы не быть уничтоженной изменениями в системе или при обнаружении системным администратором. Возможно принудительное завершение антивирусных программ.
- Сбор данных и оповещение — после выполнение своей функции вредоносная программа отправляет собранные данные на внешний пункт сбора, либо отправляет оповещение на удалённый сервер для получения дальнейших инструкций.

## Генерация признаков. Сбор данных

Если просто позволить приложению принимать бесконечный поток информации из интернета, то вряд ли удастся достичь приемлемого качества данных для машинного обучения. В итоге будет собрана масса ненужных данных вместе с теми данными, которые действительно необходимы, но и они могут быть искажены или неточны.

Специалисты по исследованию данных используют следующие положения для улучшения процесса сбора данных:

- Важность знаний предметной области — глубокое экспертное исследование предметной области может помочь быстро оценить важные признаки, которые нужно собрать.

- Масштабируемые процессы сбора данных —для получения действительно полезных результатов часто необходимо предоставить алгоритму машинного обучения огромные объёмы данных.
- Валидация данных
- Итеративные эксперименты

**Генерация признаков**

Для генерации признаков применяются следующие методики:

- Статические методы
  - анализ структуры
  - статический анализ
- Динамические методы
  - анализ поведения
  - отладка
  - динамические контрольные измерения

Более подробно методики генерации признаков будут рассмотрены в лабораторной работе.

**Выбор признаков (Feature selection)**

В большинстве случаев бездумная загрузка огромного количества признаков в алгоритм машинного обучения создают ненужный шум и пагубно влияют на точность и эффективность модели. Поэтому важно выбирать только самые важные и значимые признаки.

Один из широко распространённых способов выбора признаков — использование человеческого опыта. Люди-эксперты могут обеспечить процесс руководства моделями машинного обучения, который проявляется главным образом в форме добытых вручную признаков, считающихся наиболее важными элементами информации, используемыми в процессе обучения человека.

Статистически управляемые алгоритмы выбора признаков — методы снижения размерности наборов данных:

- Одномерный анализ (univariate analysis) — модели поочерёдно подаётся на вход по одному признаку. С помощью итеративно выполняемых одномерных статистических тестов по каждому отдельном признаку можно вывести относительную оценку того, насколько хорошо каждый признак соответствует распределению меток в тренировочном наборе.
- Рекурсивное исключение признаков (recursive feature elimination): действуя с противоположного направления, такие методы начинают с обработки полного набора признаков и рекурсивно рассматривают постоянно уменьшающиеся подмножества признаков с анализом того, как исключение признаков влияет на точность тренировки модели оценки, предложенной исследователем;
- неявное представление признаков (latent feature representations): такие методы, как сингулярное разложение (Singular Value Decomposition – SVD) и метод главных компонентов (Principal Component Analysis – PCA), выполняют преобразование данных с высокой размерностью в пространства данных с более низкой размерностью. Эти алгоритмы предназначены для минимизации потерь информации при сокращении количества признаков, необходимых для эффективной работы моделей машинного обучения.
- классификация признаков в зависимости от конкретной модели (model- specific feature ranking) — когда важность признака пропорциональна весам обученной модели, соответствующим этому признаку.

**Обучение без учителя и глубокое обучение**

Существует класс алгоритмов глубоких нейронных сетей, способных обучаться без учителя, например автокодировщики

Рекомендуемая литература:

1. Чио К., Фримэн Д. Машинное обучение и безопасность / trans. Снастина О.В. М.: ДМК Пресс, 2020. 388 с. Глава 7

Вопросы к экзамену по модулю «Методы классификации из ML в ИБ»

| № | Вопрос |
|---|---|
| 1 | Выберите наиболее правильное определение «вредоносное программное обеспечение»:<br>1. Любое программное обеспечение, предназначенное для санкционированного доступа к вычислительным ресурсам ЭВМ или к информации, хранящейся на ЭВМ.<br>2. Системное программное обеспечение, предназначенное для несанкционированного доступа к вычислительным ресурсам ЭВМ или к информации, хранящейся на ЭВМ,.<br>**3. Любое программное обеспечение, предназначенное для несанкционированного доступа к вычислительным ресурсам ЭВМ или к информации, хранящейся на ЭВМ.**<br>4. Прикладное программное обеспечение, предназначенное для несанкционированного доступа к вычислительным ресурсам ЭВМ или к информации, хранящейся на ЭВМ, |
| 2 | Большинство вредоносных программ:<br>1. Предоставляют доступ к своему исходному коду<br>2. **Отлавливаются в ловушках-песочницах**<br>3. Блокируются Интернет-провайдерами<br>4. Возможно обнаружит и обезвредить рядовому пользователю |
| 3 | Выберите операционную систему, под которую невозможно создать вредоносное ПО<br>1. Windows<br>2. Mac<br>3. Linux<br>**4. Все варианты неверны** |
| 4 | Conficker —это<br>**1. Семейство червей, нацеленных на ОС Windows**<br>2. Семейство троянов, нацеленных на ОС Android<br>3. Антивирус<br>4. Дистрибутив Linux для поиска уязвимостей |
| 5 | Преимущество метода сигнатур перед методами машинного обучения:<br>1. Нечёткое сравнение<br>2. Адаптируемость<br>**3. Меньшее количество ложных срабатываний**<br>4. Устойчивость к изменению кода вредоносного ПО |

| | |
|---|---|
| 6 | Что не является общим типом поведения вредоносного ПО<br>1. Маскировка присутствия<br>2. Стремление к выполнению своей функции<br>3. Сбор данных<br>**4. Адаптируемость** |
| 7 | Выберите динамические методы генерации признаков<br>**1. Анализ поведение**<br>2. Анализ структуры<br>**3. Отладка**<br>4. Метод сигнатур |
| 8 | Выберите качества, присущие Q-learning:<br>1. Может работать с непрерывными действиями<br>**2. Может обучаться на исторических данных**<br>3. Принимает решение об оптимальном действии на каждом шаге<br>**4. Оценивает размер возможной награды для каждого действия** |
| 9 | Выберите методы неявного представления признаков:<br>**1. SVD**<br>2. ANOVA<br>**3. PCA**<br>4. RFE |
| 10 | Выберите методы глубокого обучения без учителя:<br>1. CNN<br>**2. Autoencoder**<br>**3. Policy gradient**<br>4. SGD |