БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2022. – № 1 (104). – 41–60

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.9 DOI: 10.17212/2782-2230-2022-1-41-60

АЛГОРИТМЫ И МЕТОДЫ КЛАСТЕРИЗАЦИИ ДАННЫХ В АНАЛИЗЕ ЖУРНАЛОВ СОБЫТИЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ *

Д.Н. СИДОРОВА[1], Е.Н. ПИВКИН[2]

[1] *630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, ассистент кафедры защиты информации. E-mail: d.sidorova.2013@corp.nstu.ru*
[2] *105066, РФ, г. Москва, ул. Новорязанская, 31/7, к. 2, ПАО АКБ «Связь-Банк», кандидат технических наук. E-mail: evpiv@yandex.ru*

Файлы журналов регистрации событий безопасности дают представление о состоянии инфосистем и возможности находить аномалии в поведении пользователей, а также диагностировать происшествия кибербезопасности. В работе рассмотрены существующие журналы событий (журналы событий приложений, систем, безопасности). Следует отметить, что автоматический анализ данных журналов событий осложнен, так как они содержат большое количество неструктурированных данных, которые собираются из разных источников. Поэтому в настоящей статье представлена и описана проблема анализа журналов событий информационной безопасности. Для решения проблемы анализа журналов безопасности были рассмотрены новые и не особо изученные методы и алгоритмы кластеризации данных, как Randomforest («случайный лес»), инкрементальная кластеризация, алгоритм Iterative Partitioning Log Mining (IPLoM) – итеративный анализ журналов секционирования. Алгоритм Randomforest создает деревья решений для выборок данных, после чего делается прогноз по каждой выборке и с помощью голосования выбирается наилучшее решение. Такой метод сокращает переобучение путем усреднения показателей. Также алгоритм применяется в таких типах задач, как регрессия и классификация. Инкрементальная кластеризация определяет кластеры как группы объектов, которые принадлежат одному классу или концепту. Когда кластеры определяются, то они могут перекрываться, поэтому допускается степень «размытости для выборок», которые лежат на границах разных кластеров. Алгоритм итеративного анализа журналов секционирования использует уникальные характеристики сообщений журналов для их итеративного разделения, что способствует эффективному извлечению типов сообщений.

Ключевые слова: алгоритмы, методы, кластеризация данных, информационная безопасность, «случайный лес», инкрементальная кластеризация, итеративный анализ журнала секционирования, журналы событий

* Статья получена 07 февраля 2022 г.

ВВЕДЕНИЕ

В научной литературе на данном этапе отсутствует устоявшееся определение понятия «кластеризация данных (кластерный анализ)». В настоящей работе будем использовать определение из словаря [1]: кластерный анализ – совокупность математических методов, предназначенных для формирования относительно «отдаленных» друг от друга групп «близких» между собой объектов по информации о расстояниях или связях (мерах близости) между ними. Кластеризация используется в различных областях деятельности человека, и в каждой индивидуальной задаче ее использование имеет свои особенности. На сегодняшний день существует множество различных методов и алгоритмов кластеризации данных, которые наиболее распространены и используемы [2–14]. Поэтому нашей задачей в настоящей статье будем считать рассмотрение наименее известных алгоритмов и методов кластеризации данных.

1. ЖУРНАЛЫ СОБЫТИЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Событием называется любое существенное изменение состояния системы либо программы, о котором следует уведомить пользователей, также событием называется запись в журнале событий. Служба журнала событий записывает события приложений, системные события и события безопасности в средство просмотра событий. При помощи журналов – инструмента просмотра событий – возможно принимать информацию об аппаратном и программном обеспечении компьютера, о системных компонентах, а также проверять события безопасности на локальных или удаленных компьютерах. Журналы событий допускается применять при распознавании источников текущих системных проблем и для недопущения вероятных проблем [15].

Принято подразделять журналы событий на три вида: журналы событий и приложений, журналы систем, журналы безопасности.

Журналы событий приложений включают события, которые сформированы приложениями, а не системой. Сервер базы данных записывает ошибки, которые происходят при его работе в журнале приложений. Есть возможность для разработчиков самостоятельно принимать решения, какие события оставлять в протоколах журналов событий приложений. К примеру, Майкрософт SQL Server протоколирует детализированные данные о принципиальных аварийных ситуациях, которые возникают при работе SQL-сервера, таких как «недостаточно памяти», «сбой при запасном копировании базы данных» и т. д. При всем этом события, сгенерированные различными приложениями, попадают в единый журнал приложений. Приложения распознаются как различные «источники» в первоначальном свойстве событий. Поэтому нетрудно

выделить события определенного приложения. Коды событий (ID) распознаются приложением, которое сформировало эти коды. События могут повторяться для разных источников.

Журналы событий систем включают события, которые сгенерированы системными элементами. Например, отказы драйверов либо остальных системных компонентов при запуске систем фиксируются в системных журналах событий. Типы и коды событий системных компонентов заранее назначены разработчиками операционных систем (например, Windows). Подобно журналу приложений, системный журнал включает события из различных источников. Стоит обратить внимание, что определенные события идентифицируются и кодом, и источником. Журналы событий систем являются важным информационным источником при поиске обстоятельств отказов и вопросов системных администраторов и технических экспертов [16, 17].

Журналы событий безопасности содержит события, которые оказывают влияние на безопасность систем. Это попытки (удачные и не удачные) входа в профили систем, внедрение ресурсов (файлов, списка, устройств), управление учетными записями, изменения прав и преимуществ учетных записей, пуск и остановка процессов (программ) и т. д. Администратор может изменять категории событий, необходимых для регистрации. По умолчанию система всегда сконфигурирована так, чтобы регистрировать события управления учетными записями, события входа в систему. Как правило, аудит доступа к объектам не включается в записи журналов. В данном случае важно быть осторожным при настройке аудита доступа к файлам: некомпетентная настройка может привести к возникновению значительного числа событий, что, в свою очередь, плохо отразится на общей продуктивности системы и может привести к скорому переполнению журнала безопасности.

Запись в журнал безопасности делается лишь системными элементами, коды событий совершенно точно идентифицируют события. Журналы событий безопасности являются значительным информационным источником при расследовании происшествий безопасности, и их изучение существенно для администраторов безопасности, специалистов по информационной безопасности и специалистов по цифровой криминалистической экспертизе.

Журнал регистрации событий безопасности – это электронный журнал, который включает записи о ситуации кибербезопасности, о действиях пользователей и эксплуатирующего персонала автоматической системы [18].

Журналы событий можно разделить на следующие типы.

1. Журналы, отслеживающие вход / выход пользователя:
   а) журнал регистрации неудачных попыток начала сеанса;
   б) журнал регистрации начала / завершения сеанса.

2. Журналы, отслеживающие события:
   а) журнал регистрации событий;
   б) архив журнала регистрации событий.
3. Журнал регистрации сообщений об ошибках.

2. АЛГОРИТМЫ И МЕТОДЫ КЛАСТЕРИЗАЦИИ

Существует множество различных методов и алгоритмов кластеризации данных. В основном методы кластеризации классифицируют на иерархические и неиерархические (основанные на плоском разделении). Иерархические методы разделяют на агломеративные и дивизимные, неиерархические методы – на четкие и нечеткие. К наиболее популярным алгоритмам и методам кластеризации можно отнести следующие: наивный байесовский подход, методы опорных векторов, ближайших соседей, деревьев решений, искусственных нейронных сетей, нечеткой логики, генетические алгоритмы [19–28].

Поэтому мы рассмотрим наименее известные методы и алгоритмы кластерного анализа, такие как Randomforest («случайный лес»), инкрементальную кластеризацию, алгоритм IPLoM (Iterative Partitioning Log Mining), так как они применимы для больших объемов данных и характеризуются высокой точностью.

Randomforest

«Случайный лес» – это популярная процедура машинного обучения, которую можно использовать для разработки моделей прогнозирования. Впервые представленные Брейманом в 2001 году «случайные леса» представляют собой простые модели из наборов деревьев классификации и регрессии, использующие двоичное разбиение переменных-предикторов для определения прогнозов результатов [29–31].

Деревья решений просты в использовании на практике, так как представляют собой интуитивно понятный метод прогнозирования результата, который разделяет «высокие» и «низкие» значения предиктора, связанного с результатом. Несмотря на то что данный метод предлагает множество преимуществ, методология дерева решений часто обеспечивает низкую точность для сложных наборов данных (например, для больших наборов данных и наборов данных со сложными взаимодействиями переменных).

В настройке «случайного леса» многие деревья классификации и регрессии строятся с использованием случайно выбранных наборов, обучающих данных и случайных подмножеств переменных-предикторов для результатов моделирования. Результаты каждого дерева агрегированы, чтобы дать прогноз для каждого наблюдения [32, 33]. Следовательно, «случайный лес» часто обеспечивает

более высокую точность по сравнению с моделью с одним деревом решений, сохраняя при этом некоторые полезные качества моделей деревьев. «Случайные леса» неизменно предлагают одну из самых высоких точность прогнозов по сравнению с другими моделями при настройке классификации.

Основным преимуществом использования «случайного леса» для моделирования прогнозирования является возможность обрабатывать наборы данных с большим количеством переменных-предикторов, однако часто на практике количество предикторов, необходимых для получения прогнозов результатов, следует минимизировать для повышения эффективности. Например, вместо использования всех переменных, имеющихся в электронной медицинской карте, можно предпочесть использовать только подмножество наиболее важных переменных при разработке модели медицинского прогнозирования [34]. В прогнозном моделировании часто возникает интерес определить наиболее важные предикторы, которые следует включить в сокращенную и экономную модель. Это может быть достигнуто путем выбора переменных, при которых оптимальные предикторы определяются на основе статистических характеристик, таких как важность или точность. Разработка моделей прогнозирования с использованием выбора переменных может снизить нагрузку на сбор данных и повысить эффективность прогнозирования на практике [35]. Поскольку многие современные наборы данных имеют сотни или тысячи возможных предикторов, выбор переменных часто является необходимой частью разработки модели прогнозирования.

Выбор переменных в структуре «случайного леса» является важным аспектом для многих приложений в экспертных системах и приложениях. Таким образом, общая цель многих экспертных систем – помочь в принятии решений по сложной проблеме.

«Случайный лес» использует такие алгоритмы, как бэггинг и случайности признака.

Случайность признака. В обычном дереве решений, когда нужно разделить узел, мы рассматриваем каждый возможный признак и выбираем тот, который «сильнее» делит значения в узлах. В «случайном лесу» каждое дерево может делать выбор исключительно из случайного подмножества объектов. Из этого следует еще большая вариация между деревьями в модели. В конечном итоге более слабая корреляция между деревьями соответствует большему разнообразию.

Бэггинг. Деревья решений очень чувствительны к данным, на которых обучаются: небольшие изменения в наборе могут привести к значительно отличающимся древовидным структурам. «Случайный лес» использует это преимущество, позволяя каждому отдельному дереву произвольно выбирать данные с заменой, что приводит к различным деревьям.

Существует несколько методов выбора переменных при случайной классификации лесов. Многие пакеты **Randomforest** предоставляют процедуры случайного выбора переменных леса.

Инкрементальная кластеризация. Инкрементальная кластеризация – это задача разделения набора данных на k кластеров, в которых точки в одном кластере похожи, а точки в разных кластерах не похожи [36]. Контекст инкрементальной кластеризации выглядит следующим образом: для некоторых текущих кластеров инкрементальная кластеризация является однопроходной кластеризацией, цель которой – идентифицировать метку кластера для точек инкрементальных данных.

Инкрементальная кластеризация очень выгодна для динамических данных или потока данных (хранилища данных). Как правило, инкрементальная кластеризация комбинируется с несколькими процессами удаления и вставки. Учитывая набор кластеров, этап вставки направлен на идентификацию меток, новой точки данных на основе текущих кластеров. В некоторых случаях будут созданы новые кластеры или новые точки данных будут интегрированы с текущими кластерами.

В процессе удаления, если мы хотим удалить одну или несколько точек данных, нам необходимо реформировать кластеры, потому что эти операции могут повлиять на некоторые уже существующие кластеры. Для любого типа кластеризации в литературе предложены некоторые алгоритмы инкрементальной кластеризации, такие как Incremental k-means, Incremental DBSCAN (Density Based Spatial Clustering of Applications with Noise) или инкрементальная кластеризация графов [37]. Ключевая идея этих алгоритмов заключается в том, что нам необходимо идентифицировать ситуацию для каждого типа алгоритма для шага вставки и шага удаления.

Инкрементальная кластеризация решает проблему идентификации метки для нового объекта данных или обновления кластеров, когда мы удаляем точки в текущих кластерах. Эта проблема очень важна, когда мы занимаемся большими данными, набор данных которых слишком велик, чтобы поместиться в доступной памяти. Для каждого вида кластеризации в литературе предлагается несколько вариантов инкрементальной кластеризации.

Существует кластеризация на основе инкрементальной плотности (Incremental DBSCAN). Опираясь на концепцию кластеризации на основе плотности, Incremental DBSCAN может результативно прибавлять и удалять точки для текущих кластеров. Процесс добавления новой точки имеет несколько случаев (например, новая точка может быть шумом, новая точка будет добавлена в кластер, новая точка может объединить несколько кластеров). Для процесса удаления точка может быть точкой шума, может разбивать на некоторые кластеры или не влиять на текущие кластеры.

Также есть однопроходная инкрементальная кластеризация для большого набора данных на основе *k*-средних, которую обозначили как GenIC (Generalized Incremental Algorithm for Clustering). GenIC обновляет каждый центр с каждой новой точкой данных и объединяет кластеры только в конце генерации (то есть окна данных). С помощью обобщенного инкрементального алгоритма алгоритм GenIC может перемещать центр в списке центров, используя взвешенную сумму существующего центра и представленной новой точки.

Сущность алгоритма GenIC состоит в том, чтобы разбить поток данных на блоки или окна, как это заведено в алгоритмах потоковой передачи. Мы рассматриваем каждый блок из *n* точек данных как поколение и думаем, что «приспособленность» центра измеряется количеством назначенных ему точек. В целом наиболее приспособленные центры доживают до следующего поколения, но иногда выбираются новые центры, а старые центры уничтожаются. Алгоритм GenIC сравнивается с *k*-средними, показывает эффективность по времени выполнения и меньше зависит от выбора начальных центров, чем *k*-средние.

Предлагается вариант инкрементальной кластеризации *k*-средних. В алгоритме кластеры строятся постепенно, добавляя по одному центру кластера за один раз. Представлены новый двухфазный статический однопроходный алгоритм, а также динамический двухфазный однопроходный алгоритм, основанный на методе нечеткой кластеризации *C*-средних, которые демонстрируют высокую полезность [38]. Идея, лежащая в основе многоступенчатых методов, заключается в том, что оценка матрицы разделения и расположения центров кластеров может быть получена путем кластеризации выборки данных. Ожидается, что небольшая выборка дает быструю, но менее надежную оценку центров кластеров.

Это приводит к многоступенчатому подходу, который включает несколько этапов выборки (с заменой) данных и оценки матрицы членства для следующего этапа. Проведенные эксперименты показывают эффективность предложенного метода, схему инкрементальной кластеризации локальной плотности для поиска плотных подграфов в потоковых данных, то есть, когда данные поступают инкрементально (ILDC – incremental local density of clusters) [39].

Схема инкрементальной кластеризации захватывает избыточность в источнике потоковых данных, находя плотные подграфы, которые соответствуют заметным объектам и сценам. Процесс ILDC выполняет операции, такие как расширение кластера, добавление кластера и слияние кластера, на принципе подобия между определенными кластерами. ILDC показывает эффективность при использовании в приложениях для поиска изображений.

Далее рассмотрим алгоритм инкрементальной полууправляемой ансамблевой кластеризации, названный ISSCE (incremental semi-supervised clustering ensemble). Алгоритм ISSCE использует ограничения для обновления инкрементальных элементов. Разрабатывается процесс возрастающего выбора членов ансамбля на основе глобальной целевой функции и локальной целевой функции для удаления избыточных членов ансамбля.

Представлено улучшение ISSCE по сравнению с традиционными подходами к ансамблям полууправляемой кластеризации или с традиционными методами ансамбля кластеров на шести реальных наборах данных из репозитория машинного обучения UCI и на 12 реальных наборах данных профилей экспрессии [40]. В контексте классификации находят метку для нового объекта данных с помощью классификатора, обученного на предлагаемых данных. Проблема идентификации метки для нового объекта в инкрементальной кластеризации может рассматриваться аналогично контексту классификации.

Алгоритм IPLoM (Iterative Partitioning Log Mining). Алгоритм IPLoM разработан для кластеризации данных журнала. Он работает путем итеративного разбиения наборов журналов событий, используемых в качестве учебных образцов. На каждом этапе процесса разбиения результирующие разделы становятся ближе к содержанию журналов событий. В конце процесса разбиения алгоритм пытается обнаружить форматы строк, которые создали строки в каждом разделе. Эти обнаруженные разделы и форматы строк являются выходными данными алгоритма [41].

Алгоритм IPLoM проходит четыре этапа:

1) разделение по количеству токенов;

2) разделение по позиции токена;

3) разбиение поиском на биекцию;

4) извлечение шаблона журнала.

Этапы описаны более подробно ниже. Алгоритм предназначен для обнаружения всех возможных форматов строк в начальном наборе журналов событий. Алгоритм находит только форматы строк, поддержка которых превышает определенный порог, функция сокращения файла включена в алгоритм [42]. Функция сокращения файла работает, избавляясь от всех разделов, которые опускаются ниже порога поддержки файлов. Новое значение фиксируется в конце каждого шага разделения. Таким образом, мы способны производить только линейные форматы, которые соответствуют желаемому порогу поддержки файлов в конце алгоритма. Работа IPLoM без порога поддержки файлов является его состоянием по умолчанию.

Шаг 1. Разделение по величине событий. Журналы разбиты на различные кластеры по длине. В настоящих журналах может быть такое, что журналы,

принадлежащие одному шаблону, могут иметь переменную длину. В этом случае результат IPLoM следует обработать вручную.

Шаг 2. Разделение по позиции токена. На этом шаге каждый кластер содержит журналы схожей длины. Рассчитывая, что в кластере есть m журналов, длина которых равна n, этот кластер можно рассматривать как матрицу размером $m \times n$. Данный шаг сделан на предположении, что столбец с минимальным числом неповторимых слов (позиция разбитого слова) содержит константы. В итоге позиция разбитого слова применяется для разделения каждого кластера, то есть каждый сгенерированный кластер имеет одно и то же слово в позиции разбитого слова.

Шаг 3. Разбиение с помощью поиска по отображению. На данном шаге два столбца журналов выбираются для предстоящего разбиения на базе соотношения отображения между ними. Чтобы найти два столбца, определяется количество уникальных слов в каждом столбце и выбирается два столбца с чаще всего встречающимся числом слов. Существует четыре отношения отображения: 1-1, 1-M, M-1, M-M. В случае отношений 1-1 журналы содержат одинаковые отношения 1-1 в двух выбранных столбцах, разделенных на один и тот же кластер. Для отношений 1-M и M-1 мы должны сначала решить, содержит ли столбец стороны M константы или переменные. Если сторона M содержит константы, то столбец стороны M используется для разделения журналов в отношениях 1-M / M-1. В противном случае используется первая боковая колонка. Наконец, журналы в отношениях M-M делятся на один кластер.

Шаг 4. Извлечение шаблона журнала. Алгоритм IPLoM обрабатывает все кластеры, которые были созданы на прошлых шагах, и генерирует по одному шаблону журнала для каждого из них. Для каждого столбца в кластере подсчитывается число неповторимых слов. Если в столбце только одно неповторимое слово, оно считается неизменным. Иначе слова в столбце являются переменными и в выходных данных будут сменены знаком подстановки.

3. ПРОБЛЕМА АНАЛИЗА ЖУРНАЛОВ БЕЗОПАСНОСТИ

Файлы журнала безопасности содержат данные практически обо всех событиях, которые происходят в системе, независимо от уровня журнала. Для этого развернутая инфраструктура ведения журналов автоматически собирает, объединяет и хранит журналы, которые постоянно создаются большинством компонентов и устройств (например, веб-серверами, базами данных или межсетевыми экранами). Текстовые сообщения журнала обычно читабельны и привязываются к отметке времени, сообщая момент времени,

когда была создана запись журнала. Доступ к длительным данным журнала имеет огромное количество преимуществ, в особенности для больших компаний: журналы дают возможность проводить изучение прошедших событий, системные администраторы получают возможность отследить корни наблюдаемых проблем. Кроме того, журналы могут содействовать возврату системы к исправному состоянию, сбросить некорректные операции, вернуть данные, предупредить утрату информации и воспроизвести сценарии, которые приводят к неверным состояниям во время тестирования [43].

Главная трудность изучения журналов состоит в том, что инциденты обнаруживаются лишь постфактум. Также изучение журналов – это трудозатратная и ресурсоемкая задача, которая требует познания предметной области о системе. В настоящее время обнаружение дефектов в системах становится вероятным благодаря неизменному мониторингу системных журналов в реальном времени, т. е. online. Это позволяет вовремя проявлять реакцию на происшествия кибербезопасности [44] и снижает вызванные ими потери. Также индикаторы грядущего неверного поведения системы нередко можно отследить заблаговременно. Довольно раннее обнаружение таковых индикаторов и принятие соответствующих мер может предупредить определенные неисправности.

Но эта задача навряд ли вероятна для человека, так как данные журнала генерируются в больших объемах [45] и с большой скоростью. При рассмотрении больших корпоративных систем часто число каждый день создаваемых строк журнала исчисляется миллионами. К примеру, общедоступные журналы распределенной файловой системы Hadoop Distributed File System (HDFS) содержат свыше 4 млн строк журнала в день, а маленькие организации имеют дело с наивысшими показателями 22 тыс. событий за секунду. Разумеется, что это делает неосуществимым ручной анализ, и потому уместно применять методы машинного обучения [46], которые автоматом обрабатывают линии и распознают достойные внимания шаблоны, представляя их в сжатой форме.

Одним из способов изучения огромных объемов данных журнала является кластеризация. Данные журнала владеют определенными чертами, которые нужно учесть при разработке метода кластеризации. Во-первых, файл журнала обычно состоит из набора однострочных либо многострочных строк, перечисленных в определенном хронологическом порядке, который обычно подкрепляется отметкой времени сообщений журнала [47]. Сообщения могут быть очень структурированными (к примеру, перечень значений, разбитых запятыми), отчасти структурированными (к примеру, пары атрибут-значений), неструктурированными (к примеру, вольный текст случайной длины) либо смешанными, а также сообщения журнала время от времени включают иден-

тификаторы действий, относящихся к задаче, которая их сгенерировала. В данном случае достаточно просто извлечь трассировки журнала, другими словами, последовательности связанных строк журнала, и выполнить интеллектуальный анализ процессов. Остальные артефакты, время от времени включаемые в сообщения журнала, – это номера строк, индикатор уровня либо серьезности сообщения и статический идентификатор, указывающий на оператора, который создает сообщение.

Эти характеристики разрешают группировать системные журналы двумя различными методами. Во-первых, кластеризация некоторых строк журнала по схожести их сообщений дает обзор всех событий, которые происходят в системе. Во-вторых, кластеризация последовательностей сообщений журнала дает представление о базисной логике программы и открывает при другом варианте скрытые зависимости событий и элементов.

ЗАКЛЮЧЕНИЕ

В настоящей работе были рассмотрены наименее известные, но не менее эффективные и обладающие высокой точностью методы и алгоритмы кластеризации. Рассмотрены типы журналов событий. Особое внимание уделено журналам безопасности. Следует отметить, что обсуждаемые в статье методы и алгоритмы кластеризации могут быть применены для анализа журналов безопасности.

БЛАГОДАРНОСТИ

Авторы выражают глубокую благодарность д-ру техн. наук, профессору Белову Виктору Матвеевичу за ценные советы и замечания, высказанные при работе над статьей.

СПИСОК ЛИТЕРАТУРЫ

1. *Королев М.А.* Статистический словарь. – М.: Финансы и статистика, 1989. – 623 с.

2. *Воронцов К.В.* Алгоритмы кластеризации и многомерного шкалирования: курс лекций. – М.: МГУ, 2007.

3. *Jain A., Murty M.*, *Flynn P.* Data clustering: a review // ACM Computing Surveys. – 1999. – Vol. 31, iss 3. – P. 264–323.

4. *Котов А.*, *Красильников Н.* Кластеризация данных. – СПб.: СПбГУ ИТМО, 2006.

5. *Мандель И.Д.* Кластерный анализ. – М.: Финансы и статистика, 1988. – 176 с.

6. Прикладная статистика: классификация и снижение размерности / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин. – М.: Финансы и статистика, 1989. – 607 с.

7. MachineLearning.Ru. Информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. – URL: www.machinelearning.ru (дата обращения: 04.03.2022).

8. *Чубукова И.А.* Курс лекций «DataMining» / Интернет-университет информационных технологий. – URL: www.intuit.ru/department/database/datamining (дата обращения: 04.03.2022).

9. *Farid D.M., Rahman M.Z., Rahman C.M.* Adaptive intrusion detection based on boosting and naïve Bayesian classifier // International Journal of Computer Applications. – 2011. – Vol. 24 (3). – P. 12–19.

10. *Лепский А.Е., Броневич А.Г.* Математические методы распознавания образов: курс лекций. – Таганрог, 2009. – URL: https://lepskiy.ucoz.ru/Posobie/MMPR_.pdf (дата обращения: 04.03.2022).

11. Интуит. Национальный открытый университет. Лекция 9: Методы классификации и прогнозирования. Деревья решений. – URL: http://www.intuit.ru/studies/courses/6/6/lecture/174 (дата обращения: 14.03.2022).

12. *Круглов В.В., Голунов Р.Ю.* Нечеткая логика и искусственные нейронные сети. – М.: Физматлит, 2001. – 224 с.

13. *Воронцов К.В.* Лекции по искусственным нейронным сетям. – 2007, 21 декабря. – URL: http://www.ccas.ru/voron/download/NeuralNets.pdf (дата обращения: 04.03.2022).

14. *Барский А.Б.* Нейронные сети: распознавание, управление, принятие решений. – М.: Финансы и статистика, 2004. – 176 с.

15. *Панченко Т.В.* Генетические алгоритмы / под ред. Ю.Ю. Тарасевича. – Астрахань: Астраханский университет, 2007. – 87 с.

16. CompoWiki. Журнал событий. – URL: https://wiki.compowiki.info/ЖурналСобытий (дата обращения: 04.03.2022).

17. Журналы событий Windows. – URL: https://eventlogxp.com/rus/essentials/windowseventlog.html (дата обращения: 04.03.2022).

18. Журнал регистрации событий информационной безопасности. – URL: https://safe-surf.ru/glossary/ru/849/ (дата обращения: 04.03.2022).

19. *Makanju A., Zincir-Heywood A.N., Milios E.E.* Clustering event logs using iterative partitioning // KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. – ACM, 2009. – P. 1255–1264. – DOI: 10.1145/1557019.1557154.

20. On vulnerability and security log analysis: a systematic literature review on recent trends / J. Svacina, J. Raffety, C. Woodahl, B. Stone, T. Cerny, M. Bures, D. Shin, K. Frajtak, P. Tisnovsky // RACS '20: Proceedings of the International Conference on Research in Adaptive and Convergent Systems. – ACM, 2020. – P. 175–180. – DOI: 10.1145/3400286.3418261.

21. Process mining and hierarchical clustering to help intrusion alert visualization / S.C. de Alvarenga, S. Barbon, R.S. Miani, M. Cukier, B.B. Zarpelão // Computers and Security. – 2018. – Vol. 73. – P. 474–491. – DOI: 10.1016/j.cose.2017.11.021.

22. *Alaba A., Maitanmi S., Ajayi O.* An ensemble of classification techniques for Intrusion detection systems // International Journal of Computer Science and Information Security. – 2019. – Vol. 17, N 11. – P. 24–33.

23. *Chauhan A., Mishra G., Kumar G.* Survey on data mining techniques in intrusion detection // International Journal of Scientific and Engineering Research. – 2011. – Vol. 2, iss. 7. – P. 1–4.

24. A multi-level intrusion detection method for abnormal network behavior / S.-Y. Ji, S. Choi, B.-K. Jeong, D.H. Jeong // Journal of Network and Computer Applications. – 2016. – Vol. 62. – P. 9–17.

25. *Onan A., Korukoğlu S., Bulut H.* A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification // Expert Systems with Applications. – 2016. – Vol. 62. – P. 1–16. – DOI: 10.1016/j.eswa.2016.06.005.

26. Implementation of naïve Bayes classification method for predicting purchase / F. Harahap, A.Y.N. Harahap, E. Ekadiansyah, R.N. Sari, R. Adawiyah, C.B. Harahap // 2018 6th International Conference on Cyber and IT Service Management (CITSM). – Parapat, Indonesia, 2018. – P. 1–5. – DOI: 10.1109/CITSM.2018.8674324.

27. Deep Learning techniques for traffic speed forecasting with side information / P. Farajiparvar, N. Hoseinzadeh, L.D. Han, A. Hedayatipour // 2020 IEEE Green Energy and Smart Systems Conference (IGESSC). – Long Beach, CA, 2020. – P. 1–5. – DOI: 10.1109/IGESSC50231.2020.9285132.

28. *Aklani S.A.* Metode fuzzy logic untuk evaluasi kinerja pelayanan perawat (Studi Kasus: RSIA Siti Hawa Padang) // Edik Informatika. – 2014. – Vol. 1, N 1. – P. 35–43.

29. Recognition of driving postures by contourlet transform and random forests / C.H. Zhao, B.L. Zhang, J. He, J. Lian // IET Intelligent Transport Systems. – 2012. – Vol. 6 (2). – P. 161–168.

30. *Probst P., Wright M.N., Boulesteix A.-L.* Hyperparameters and tuning strategies for random forest // WIREs Data Mining and Knowledge Discovery. – 2019. – Vol. 9. – P. e1301.

31. Applying a random forest method approach to model travel mode choice behavior / L. Cheng, X. Chen, J. De Vos, X. Lai, F. Witlox // Travel Behaviour and Society. – 2019. – Vol. 14. – P. 1–10.

32. GIS-based landslide susceptibility evaluation using a novel hybrid integration approach of bivariate statistical based random forest method / W. Chen, X. Xie, J. Peng, H. Shahabi, H. Hong, D.T. Bui, Z. Duan, S. Li, A-X. Zhu // Catena. – 2018. – Vol. 164. – P. 135–149.

33. Identifying core driving factors of urban land use change from global land cover products and POI data using the random forest method / H. Wu, A. Lin, X. Xing, D. Song, Y. Li // International Journal of Applied Earth Observation and Geoinformation. – 2021. – Vol. 103. – P. 102475.

34. *Cai Y., Lin H., Zhang M.* Mapping paddy rice by the object-based random forest method using time series Sentinel-1/Sentinel-2 data // Advances in Space Research. – 2019. – Vol. 64 (11). – P. 2233–2244.

35. Prediction of consumer behaviour using random forest algorithm / H. Valecha, A. Varma, I. Khare, A. Sachdeva, M. Goyal // 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON). – Gorakhpur, India, 2018. – P. 1–6. – DOI: 10.1109/UPCON.2018.8597070.

36. *Кутуков Д.С.* Применение методов кластеризации для обработки новостного потока // Технические науки: проблемы и перспективы: материалы I Международной научной конференции. – СПб.: Реноме, 2011. – С. 77–83. – URL: https://moluch.ru/conf/tech/archive/2/207/ (дата обращения: 09.03.2022).

37. *Kailing K., Kriegel H.-P., Kröger P.* Density-connected subspace clustering for high-dimensional data // Proceedings of the 4th SIAM International Conference on Data Mining (SDM). – Philadelphia, PA, 2004. – P. 246–257.

38. *Braun R.K.*, *Kaneshiro R.* Exploiting topic pragmatics for new event detection in TDT-2004 // DARPA Topic Detection and Tracking Workshop. – Gaithersburg, 2004.

39. *Peters M.*, *Zaki M.J.* Click: clustering categorical data using K-partite maximal cliques / Computer Science Department Rensselaer Polytechnic Institute. – Troy, NY, 2004. – 31 p.

40. Clustering uncertain data based on probability distribution similarity / B. Jiang, J. Pei, Y. Tao., X. Lin // IEEE Transactions on Knowledge and Data Engineering. – 2013. – Vol. 25 (4). – P. 751–763. – DOI: 10.1109/TKDE.2011.221.

41. *Makanju A., Zincir-Heywood A.N., Milios E.E.* A lightweight algorithm for message type extraction in system application logs // IEEE Transactions on Knowledge and Data Engineering. – 2012. – Vol. 24 (11). – P. 1921–1936. – DOI: 10.1109/TKDE.2011.138.

42. *Makanju A., Zincir-Heywood A.N., Milios E.E.* Clustering event logs using iterative partitioning // ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 09). – ACM, 2009. – P. 1255–1263.

43. *Oliner A., Ganapathi A., Xu W.* Advances and challenges in log analysis: logs contain a wealth of information for help in managing systems // ACM Queue. – 2011. – Vol. 9 (12). – DOI: 10.1145/2076796.2082137.

44. Best practices for incident response. – 2020, September 3. – URL: https://www.securitymagazine.com/articles/93235-best-practices-for-incident-response (accessed: 09.03.2022).

45. Operational-log analysis for big data systems: challenges and solutions / A. Miranskyy, A. Hamou-Lhadj, E. Cialini, A. Larsson // IEEE Software. – 2016. – Vol. 33 (2). – P. 52–59. – DOI: 10.1109/MS.2016.33.

46. HDFS Architecture / The Apache Software Foundation. – URL: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html (accessed: 09.03.2022).

47. *Brownlee J.* A tour of machine learning algorithms. – 2019, August 12. – URL: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ (accessed: 09.03.2022).

Сидорова Диана Николаевна, ассистент кафедры защиты информации Новосибирского государственного технического университета. В настоящее время специализируется в области информационной безопасности. E-mail: d.sidorova.2013@corp.nstu.ru

Пивкин Евгений Николаевич, кандидат технических наук, руководитель направления отдела защиты информации департамента безопасности ПАО АКБ «Связь-Банк». Основное направление научных исследований – применение математических методов в различных областях науки, техники, общества. Автор более 80 публикаций. E-mail:evpiv@yandex.ru

DOI: 10.17212/2782-2230-2022-1-41-60

Algorithms and methods of data clustering in the analysis of information security event logs*

D.N. Sidorova[1], E.N. Pivkin[2]

[1] *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, assistant of the Department of Information Security. E-mail: d.sidorova.2013@corp.nstu.ru*
[2] *PJSC JSCB "Svyaz-Bank", 31/7 Novoryazanskaya Street, Moscow, 105066, Russian Federation, candidate of technical sciences, E-mail: evpiv@yandex.ru*

Security event log files give an idea of the state of the information system and allow you to find anomalies in user behavior and cybersecurity incidents. The existing event logs (application, system, security event logs) and their division into certain types are considered. But automated analysis of security event log data is difficult because it contains a large amount of unstructured data that has been collected from various sources. Therefore, this article presents and describes the problem of analyzing information security event logs.
And to solve this problem, new and not particularly studied methods and algorithms for data clustering were considered, such as Random forest (random forest), incremental clustering, IPLoM algorithm (Iterative Partitioning Log Mining - iterative analysis of the partitioning log). The Random forest algorithm creates decision trees for data samples, after which it is provided with a forecast for each sample, and the best solution is selected by voting. This method reduces overfitting by averaging the scores. The algorithm is also used in such types of problems as regression and classification. Incremental clustering defines clusters as groups of objects that belong to the same class or concept, which is a specific set of pairs. When clusters are defined, they can overlap, allowing for a degree of "fuzziness for samples" that lie at the boundaries of different clusters. The IPLoM algorithm uses the unique characteristics of log messages to iteratively partition the log, which helps to extract message types efficiently.

Keywords: algorithms, methods, data clustering, information security, random forest, incremental clustering, iterative partitioning log analysis, event logs

REFERENCES

1. Korolev M.A. *Statisticheskii slovar'* [Statistical dictionary]. Moscow, Finansy i statistika Publ., 1989. 623 p.

2. Vorontsov K.V. *Algoritmy klasterizatsii i mnogomernogo shkalirovaniya*: kurs lektsii [Clustering and multidimensional scaling algorithms. Lecture course]. Moscow State University, 2007.

3. Jain A., Murty M., Flynn P. Data clustering: a review. *ACM Computing Surveys*, 1999, vol. 31, iss. 3, pp. 264–323.

* *Received 07 February 2022.*

4. Kotov A., Krasil'nikov N. *Klasterizatsiya dannykh* [Data clustering]. St. Petersburg, ITMO University, 2006.

5. Mandel' I.D. *Klasternyi analiz* [Cluster analysis]. Moscow, Finansy i statistika Publ., 1988. 176 p.

6. Aivazyan S.A., Bukhshtaber V.M., Enyukov I.S., Meshalkin L.D. *Prikladnaya statistika: klassifikatsiya i snizhenie razmernosti* [Applied statistics: classification and dimensionality reduction]. Moscow, Finansy i statistika Publ., 1989. 607 p.

7. *MachineLearning.Ru*. Information and analytical resource dedicated to machine learning, pattern recognition and data mining. (In Russian). Available at: www.machinelearning.ru (accessed 04.03.2022).

8. Chubukova I.A. *Kurs lektsii "DataMining"* [Lecture course "Data Mining"]. Internet University of Information Technologies. Available at: www.intuit.ru/department/database/datamining (accessed 04.03.2022).

9. Farid D.M., Rahman M.Z., Rahman C.M. Adaptive intrusion detection based on boosting and naïve Bayesian classifier. *International Journal of Computer Applications*, 2011, vol. 24 (3), pp. 12–19.

10. Lepskiy A.E., Bronevich A.G. *Matematicheskie metody raspoznavaniya obrazov*: kurs lektsii [Mathematical methods for pattern recognition]. Taganrog, 2009. Available at: https://lepskiy.ucoz.ru/Posobie/MMPR_.pdf (accessed 04.03.2022).

11. Intuit. National Open University. Lektsiya 9: *Metody klassifikatsii i prognozirovaniya. Derev'ya reshenii* [Lecture 9: Classification and forecasting methods. decision trees]. Available at: http://www.intuit.ru/studies/courses/6/6/lecture/174 (accessed 14.03.2022).

12. Kruglov V.V., Golunov R.Yu. *Nechetkaya logika i iskusstvennye neironnye seti* [Fuzzy logic and artificial neural networks]. Moscow, Fizmatlit Publ., 2001. 224 p.

13. Vorontsov K.V. *Lektsii po iskusstvennym neironnym setyam* [Lectures on artificial neural networks], 2007, December 21. Available at: http://www.ccas.ru/voron/download/NeuralNets.pdf (accessed 04.03.2022).

14. Barskii A.B. *Neironnye seti: raspoznavanie, upravlenie, prinyatie reshenii* [Neural networks: recognition, control, decision making]. Moscow, Finansy i statistika Publ., 2004. 176 p.

15. Panchenko T.V. *Geneticheskie algoritmy* [Genetic algorithms]. Astrakhan, Astrakhanskii universitet Publ., 2007. 87 p.

16. CompoWiki. *Zhurnal sobytii* [CompoWiki. Event log]. Available at: https://wiki.compowiki.info/EventLog (accessed 04.03.2022).

17. *Zhurnaly sobytii Windows* [Windows event log]. Available at: https://eventlogxp.com/rus/essentials/windowseventlog.html (accessed 04.03.2022).

18. *Zhurnal registratsii sobytii informatsionnoi bezopasnosti* [Information security event log]. Available at: https://safe-surf.ru/glossary/ru/849/ (accessed 04.03.2022).

19. Makanju A., Zincir-Heywood A.N., Milios E.E. Clustering event logs using iterative partitioning. *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009, pp. 1255–1264. DOI: 10.1145/1557019.1557154.

20. Svacina J., Raffety J., Woodahl C., Stone B., Cerny T., Bures M., Shin D., Frajtak K., Tisnovsky P. On vulnerability and security log analysis: a systematic literature review on recent trends. *RACS '20: Proceedings of the International Conference on Research in Adaptive and Convergent Systems*. ACM, 2020, pp. 175–180. DOI: 10.1145/3400286.3418261.

21. Alvarenga S.C. de, Barbon S., Zarpelão B.B., Miani R.S., Cukier M. Process mining and hierarchical clustering to help intrusion alert visualization. *Computers and Security*, 2018, vol. 73, pp. 474–491. DOI: 10.1016/j.cose.2017.11.021.

22. Alaba A., Maitanmi S., Ajayi O. An ensemble of classification techniques for Intrusion detection systems. *International Journal of Computer Science and Information Security*, 2019, vol. 17, no. 11, pp. 24–33.

23. Chauhan A., Mishra G., Kumar G. Survey on data mining techniques in intrusion detection. *International Journal of Scientific and Engineering Research*, 2011, vol. 2, iss. 7, pp. 1–4.

24. Ji S.-Y., Choi S., Jeong B.-K., Jeong D.H. A multi-level intrusion detection method for abnormal network behavior. *Journal of Network and Computer Applications*, 2016, vol. 62, pp. 9–17.

25. Onan A., Korukoğlu S., Bulut H. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications*, 2016, vol. 62, pp. 1–16. DOI: 10.1016/j.eswa.2016.06.005.

26. Harahap F., Harahap A.Y.N., Ekadiansyah E., Sari R.N., Adawiyah R., Harahap C.B. Implementation of naïve Bayes classification method for predicting purchase. *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, Parapat, Indonesia, 2018, pp. 1–5. DOI: 10.1109/CITSM.2018.8674324.

27. Farajiparvar P., Hoseinzadeh N., Han L.D., Hedayatipour A. Deep Learning techniques for traffic speed forecasting with side information. *2020 IEEE Green Energy and Smart Systems Conference (IGESSC)*, Long Beach, CA, 2020, pp. 1–5. DOI: 10.1109/IGESSC50231.2020.9285132.

28. Aklani S.A. Metode fuzzy logic untuk evaluasi kinerja pelayanan perawat (Studi Kasus: RSIA Siti Hawa Padang). *Edik Informatika*, 2014, vol. 1, no. 1, pp. 35–43.

29. Zhao C.H., Zhang B.L., He J., Lian J. Recognition of driving postures by contourlet transform and random forests. *IET Intelligent Transport Systems*, 2012, vol. 6 (2), pp. 161–168.

30. Probst P., Wright M.N., Boulesteix A.-L. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 2019, vol. 9, p. e1301.

31. Cheng L., Chen X., Cheng L., De Vos J., Witlox F., Lai X., Witlox F., Witlox F. Applying a random forest method approach to model travel mode choice behavior. *Travel Behaviour and Society*, 2019, vol. 14, pp. 1-10.

32. Chen W., Xie X., Peng J., Shahabi H., Hong H., Bui D.T., Duan Z., Li S., Zhu A-X. GIS-based landslide susceptibility evaluation using a novel hybrid integration approach of bivariate statistical based random forest method. *Catena*, 2018, vol. 164, pp. 135–149.

33. Wu H., Lin A., Xing X., Song D., Li Y. Identifying core driving factors of urban land use change from global land cover products and POI data using the random forest method. *International Journal of Applied Earth Observation and Geoinformation*, 2021, vol. 103, p. 102475.

34. Cai Y., Lin H., Zhang M. Mapping paddy rice by the object-based random forest method using time series Sentinel-1/Sentinel-2 data. *Advances in Space Research*, 2019, vol. 64 (11), pp. 2233–2244.

35. Valecha H., Varma A., Khare I., Sachdeva A., Goyal M. Prediction of consumer behaviour using random forest algorithm. *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, Gorakhpur, India, 2018, pp. 1–6. DOI: 10.1109/UPCON.2018.8597070.

36. Kutukov D.S. [Application of clustering methods for news flow processing]. *Tekhnicheskie nauki: problemy i perspektivy*: materialy I Mezhdunarodnoi nauchnoi konferentsii [Technical sciences: problems and prospects: materials of the I International scientific conference], St. Petersburg, Renome Publ., 2011, pp. 77–83. (In Russian). Available at: https://moluch.ru/conf/tech/archive/2/207/ (accessed 09.03.2022).

37. Kailing K., Kriegel H.-P., Kröger P. Density-connected subspace clustering for high-dimensional data. *Proceedings of the 4th SIAM International Conference on Data Mining (SDM)*, Philadelphia, PA, 2004, pp. 246–257.

38. Braun R.K., Kaneshiro R. Exploiting topic pragmatics for new event detection in TDT-2004. *DARPA Topic Detection and Tracking Workshop*, Gaithersburg, 2004.

39. Peters M., Zaki M.J. *Click: clustering categorical data using K-partite maximal cliques*. Computer Science Department Rensselaer Polytechnic Institute, Troy, NY, 2004. 31 p.

40. Jiang B., Pei J., Tao Y, Lin X. Clustering uncertain data based on probability distribution similarity. *IEEE Transactions on Knowledge and Data Engineering*, 2013, vol. 25 (4), pp. 751–763. DOI: 10.1109/TKDE.2011.221.

41. Makanju A., Zincir-Heywood A.N., Milios E.E. A lightweight algorithm for message type extraction in system application logs. *IEEE Transactions on Knowledge and Data Engineering*, 2012, vol. 24 (11), pp. 1921–1936. DOI: 10.1109/TKDE.2011.138.

42. Makanju A., Zincir-Heywood A.N., Milios E.E. Clustering event logs using iterative partitioning. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 09)*, ACM, 2009, pp. 1255–1263.

43. Oliner A., Ganapathi A., Xu W. Advances and challenges in log analysis: logs contain a wealth of information for help in managing systems. *ACM Queue*, 2011, vol. 9 (12). DOI: 10.1145/2076796.2082137.

44. *Best practices for incident response*. 2020, September 3. Available at: https://www.securitymagazine.com/articles/93235-best-practices-for-incident-response (accessed 09.03.2022).

45. Miranskyy A., Hamou-Lhadj A., Cialini E., Larsson A. Operational-log analysis for big data systems: challenges and solutions. *IEEE Software*, 2016, vol. 33 (2), pp. 52–59. DOI: 10.1109/MS.2016.33.

46. *HDFS Architecture*. The Apache Software Foundation. Available at: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html (accessed 09.03.2022).

47. Brownlee J. *A tour of machine learning algorithms*. 2019, August 12. Available at: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ (accessed 09.03.2022).

Для цитирования:

Сидорова Д.Н., Пивкин Е.Н. Алгоритмы и методы кластеризации данных в анализе журналов событий информационной безопасности // Безопасность цифровых технологий. – 2022. – № 1 (104). – С. 41–60. – DOI: 10.17212/2782-2230-2022-1-41-60.

For citation:

Sidorova D.N., Pivkin E.N. Algoritmy i metody klasterizatsii dannykh v analize zhurnalov sobytii informatsionnoi bezopasnosti [Algorithms and methods of data clustering in the analysis of information security event logs]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2022, no. 1 (104), pp. 41–60. DOI: 10.17212/2782-2230-2022-1-41-60.