УДК 004.522

АНАЛИЗ ПОТОКА ДАННЫХ

Гылычтаганов Ш.
Заведующий кафедры «Программное обеспечение информационных технологий»
Институт Телекоммуникаций и информатики Туркменистана,
gylyctaganow@bk.ru

Ш. Ю. Тедженов
Преподаватель кафедры «Программное обеспечение информационных технологий»
Институт Телекоммуникаций и информатики Туркменистана,
tejenowshirmyrat@gmail.com

Гылычтаганов Ш.
Заведующий кафедры «Программное обеспечение информационных технологий» Институт Телекоммуникаций и информатики Туркменистан.

Ш. Ю. Тедженов
Преподаватель кафедры «Программное обеспечение информационных технологий», Институт Телекоммуникаций и информатики Туркменистана

Аннотация. В связи с распространенным использованием датчиков и сетей инструменты мониторинга, большие объемы данных или «большие данные» сегодня перемещаться по конвейерам обработки данных предприятия в потоковом режиме мода. Хотя некоторые компании предпочитают размещать свои данные инфраструктура обработки и услуги в виде частных облаков, другие полностью передают эти услуги публичным облакам. В любом случае, пытаясь сначала сохранить данные для последующих анализ приводит к дополнительным затратам ресурсов и нежелательным задержкам в получении действенной информации. В результате предприятия все чаще используют системы обработки потоков данных или событий. и в дальнейшем хотим расширить их с помощью комплексной онлайн-аналитики и возможности майнинга.

Ключевые слова: потоки данных, обработка сложных событий, ассоциация майнинг правил, потоковый майнинг, корреляция, априори, рост *FP*.

Введение. Что такое *Big Data* (большие данные)? Большие данные имеют четыре измерения, которые чрезвычайно затрудняет управление: объемом, разнообразием, Скорость и достоверность, также известные как «4*Vs of Big Data*». [13]. Многие организации сегодня, включая телекоммуникационные операторы, сайты электронной коммерции, банки, муниципалитеты, СМИ сети, а правительства генерируют терабайты данных ежедневно. Они также пытаются установить этот высокий уровень «объемные» данные в базы данных, которые уже содержат петабайты. Данные поступают из различных источников, таких как мобильные устройства, веб-журналы, датчики, камеры и т. д., что создает «разнообразие» для данных, подлежащих обработке и хранению (например, неструктурированный текстовый файл, полуструктурированный *XML*,

структурированный *CSV*, реляционный или двоичный аудио/видео данные). Новая распределенная обработка данных были изобретены такие фреймворки, как *Apache Hadoop* [2] исключительно для работы с объемом и разнообразием данных за последние десятилетие. Система *Hadoop* сегодня широко используется компаниями облачных вычислений, такими как *Google*, *Yahoo*, *Facebook* и многими другими, для крупномасштабного хранения данных (*HDFS*), параллельной обработки (*MapReduce*) и складирования (*Hive*). *Hadoop* обычно развертывается как частная облачная служба, но сегодня несколько поставщиков также предоставляют ее в качестве общедоступной облачной службы.

Хотя объем и разнообразие всегда были проблемой для управления данными, их растущая «скорость» является новой проблемой 21 века [13]. Попытка сначала сохранить эти данные, чтобы затем проанализировать их, приводит к дополнительным затратам, нежелательным задержкам в получении полезной информации и потере возможностей для бизнеса. К счастью, сейчас существуют инструменты для оперативной обработки данных по мере их перемещения от распределенных источников (например, датчиков) к выбранным пунктам назначения. Поскольку количество источников данных и частота выборки постоянно растут, обработка данных в оперативной памяти по-прежнему остается большой проблемой. Этот документ посвящен только анализу и анализу данных потокового текстового журнала и не охватывает обработку закодированных типов мультимедиа (изображений и аудио/видео).

Потоковая передача данных от датчиков также подвержена ошибкам, что снижает их «правдивость» (или точность). Кортежи могут отсутствовать, быть поврежденными, иметь неправильный порядок или иметь неверные значения. В таблице 1 показаны примеры данных из реального приложения для отслеживания автобусов, в котором показано несколько таких случаев. Система случайно удалила нули из младших цифр долготы и широты местоположений автобусов (обозначенных жирным шрифтом), пропустила несколько кортежей (на момент 7:26 существует только 1 из 3 записей) и, возможно, вставила неточные значения скорости (0, 1 км/ч).

Большинство этих проблем анализа больших данных можно решить за счет использования систем управления потоками данных (*Data Stream Management Systems- DSMS*) [1,4] в конвейерах данных организаций. Поэтому предприятия все чаще используют эти системы и расширяют свои базовые средства фильтрации за счет комплексных возможностей онлайн-аналитики и интеллектуального анализа данных. Полученные в результате программные средства иногда называют в литературе механизмами обработки сложных событий (*Complex Event Processing- CEP*) [6]. Преимущества использования систем *DSMS* и *CEP* заключаются как минимум в трех аспектах:

1 Они могут удалять нежелательные данные на ранних этапах конвейера, экономя дополнительные затраты на процессор, память, хранилище и электроэнергию.

2 Они могут быстро превращать необработанные данные в полезную информацию, тем самым помогая предприятиям использовать выгодные возможности или избегать потерь из-за мошенничества или операционной неэффективности.

3 Они могут выявить временные или возникающие закономерности, которые никогда не проявляются при автономном анализе данных.

Кратко суть нашей статьи такова: мы реализуем и демонстрируем, что как статистический анализ (например, корреляция потоков), так и интеллектуальный анализ данных (например, интеллектуальный анализ правил) могут быть реализованы в одной и той же системе и использоваться для различных приложений реального времени. Мы также показываем, что как для инструментов аналитики, так и для инструментов интеллектуального анализа семантика временных окон (тип и размер) может иметь

большое влияние как на производительность, так и на удобство использования в различных приложениях.

Таблица 1: Пример данных из реальной системы слежения автобусов

BUSID	LONGITUTE	LATTITUDE	SPEED	DATE & TIME
00-123	28,863,169	4,105,348	42	12/8/2009 7:23
00-123	2,886,469	41,052,845	3	12/8/2009 7:23
00-123	28,866,064	41,052,856	26	12/8/2009 7:23
00-123	28,867,975	410,522	37	12/8/2009 7:24
00-123	2,886,879	4,105,189	1	12/8/2009 7:24
00-123	28,869,068	41,051,792	6	12/8/2009 7:24
00-123	28,869,884	41,051,376	16	12/8/2009 7:25
00-123	28,870,121	41,051,258	0	12/8/2009 7:25
00-123	2,887,055	41,051,044	16	12/8/2009 7:25
00-123	28,870,613	4,105,191	15	12/8/2009 7:26
00-123	28,868,597	4,105,249	46	12/8/2009 7:27
00-123	28,866,816	4,105,319	19	12/8/2009 7:27
00-123	288,657	41,053,898	20	12/8/2009 7:27

Остальная часть статьи состоит в следующем. В разделе 2 мы даем обзор систем *DSMS* и *CEP* и описываем архитектуру нашей системы анализа и интеллектуального анализа потоков данных. В разделе 3 мы покажем приложение для использования статистических корреляций, в частности *Pearson Product Moment Correlation* (*PPMC*), в потоках *GPS* и предоставим некоторые результаты по производительности и удобству использования. В разделе 4 мы обсуждаем интеграцию двух алгоритмов *ARM* (*Apriori* и *FP-Growth*) в *Esper*, даем сравнение их производительности и описываем, как их можно использовать для поддержки механизма рекомендаций музыки в реальном времени. Мы обсуждаем соответствующую работу в разделе 5. Мы завершаем статью в разделе 6 и обсуждаем будущую работу.

Кратко суть нашей статьи такова: мы реализуем и демонстрируем, что как статистический анализ (например, корреляция потоков), так и интеллектуальный анализ данных (например, интеллектуальный анализ правил) могут быть реализованы в одной и той же системе и использоваться для различных приложений реального времени. Мы также показываем, что как для инструментов аналитики, так и для инструментов интеллектуального анализа семантика временных окон (тип и размер) может иметь большое влияние как на производительность, так и на удобство использования в различных приложениях.

Остальная часть статьи состоит в следующем. В разделе 2 мы даем обзор систем *DSMS* и *CEP* и описываем архитектуру нашей системы анализа и интеллектуального анализа потоков данных. В разделе 3 мы покажем приложение для использования статистических корреляций, в частности *Pearson Product Moment Correlation* (*PPMC*), в потоках *GPS* и предоставим некоторые результаты по производительности и удобству использования. В разделе 4 мы обсуждаем интеграцию двух алгоритмов *ARM* (*Apriori* и *FP-Growth*) в *Esper*, даем сравнение их производительности и описываем, как их можно использовать для поддержки механизма рекомендаций музыки в реальном времени. Мы

обсуждаем соответствующую работу в разделе 5. Мы завершаем статью в разделе 6 и обсуждаем будущую работу.

Системная архитектура. Механизмы *DSMS* обеспечивают эффективную организацию очередей, планирование, поддержку окон времени и счета, а также быструю обработку в памяти высокоскоростных, непрерывных, неограниченных потоков данных [1]. Они анализируют, оптимизируют и выполняют запросы, написанные на декларативных языках, таких как язык обработки событий (*Event Processing Language-EPL*) в *Esper*. Синтаксис и семантика *EPL* очень похожи на синтаксис и семантику языка структурированных запросов (*SQL*) в базах данных, но есть дополнительные предложения, такие как *WINDOW*, для поддержки скользящего или переворачивающегося анализа на основе окон по потокам данных. На рисунке 1 показаны эти два типа окон. Скользящие временные окна используются для буферизации кортежей событий, время возникновения которых попадает в определенный период времени (например, последняя 1 минута), а также для замены событий, которые старше временного окна. Окно будет перемещаться или «скользить» во времени с периодом, который обычно меньше размера окна, и поэтому две эпохи событий перекрываются. Аналогично, окна скользящего счетчика буферизуют последние *X* событий. С другой стороны, переворачивающиеся окна переходят к следующей эпохе, перемещаясь на размер окна, и данные между двумя эпохами событий не перекрываются. Другие типы окон включают «Ориентир» и «Затухание» [9], где первый учитывает события от прошлого ориентира до настоящего времени, а второй придает больший вес недавним событиям. В этой статье мы используем только раздвижные окна, а остальное оставляем на будущее.

Рисунок 1. Семантика «переворачивающегося» и «скользящего» окна

Запросы *EPL* можно использовать для непрерывной фильтрации (например, *SELECT x,y FROM Stream<x,y,z> WHERE*…), а также для агрегирования: алгебраического (*COUNT*, *SUM*, *AVERAGE*) или целостного (*MIN*, *MAX*). Также можно найти или реализовать сложные функции агрегирования, такие как *TOP-K*, *DISTINCT*, *QUANTILES* и *SKYLINE*.

Потоковая аналитика и архитектура майнинга. На рисунке 2 показаны компоненты и высокоуровневая архитектура нашей системы анализа потоков данных. Высокоскоростные потоки необработанных данных сначала обрабатываются операторами *Select* и *Project* внутри *DSMS*, чтобы избавиться от нежелательных кортежей в дальнейшем в конвейере данных. Объем данных можно уменьшить по строкам (Выбрать) и/или по столбцам (Проект). По сути, на этом этапе выполняется этап предварительной обработки потокового майнинга. Остальные пункты сложной аналитики и майнинга реализованы внутри системы *CEP* [6], которая развертывается либо как частная, либо как общедоступная облачная система (см. часть *B*). Данные из потоков можно коррелировать между собой или с данными, хранящимися в постоянных репозиториях, таких как *DBMS* и системы *NoSQL*. *NoSQL* («*Not-only SQL*») – это общий термин для широкомасштабируемых распределенных механизмов хранения и обработки данных в сочетании с декларативным или процедурным языком для аналитики. Примеры включают *Apache HBase*, *Cassandra*, *MongoDB* и многие другие. Сегодня во многих частных облаках с высокоскоростной обработкой потоков событий системы *NoSQL* используются для поддержки очередей событий для расширения окон корреляции.

Новые операторы *CEP*, разработанные в этой статье, обозначены знаком бабочки на рисунке 2: агрегат (корреляция) и потоковый анализ (*Apriori* и *FP-Growth*). Мы описываем интеграцию алгоритмов машинного обучения с *Esper* в разделе *IV.C*. Статистические результаты и оповещения отправляются в системы управления бизнес-процессами и средства визуализации для дальнейших действий. Цикл обратной связи, показанный на выходе системы *CEP*, означает регистрацию правил, полученных в результате прогнозного анализа, обратно в механизм *CEP* для более быстрой описательной обработки. Например, правило ассоциации, такое как ($A\&B \Rightarrow C$), можно зарегистрировать как последовательность <<*A*,*B*>,*C*>. Однако мы откладываем эту автоматическую регистрацию правил в системе частного облака *CEP* до будущих работ.

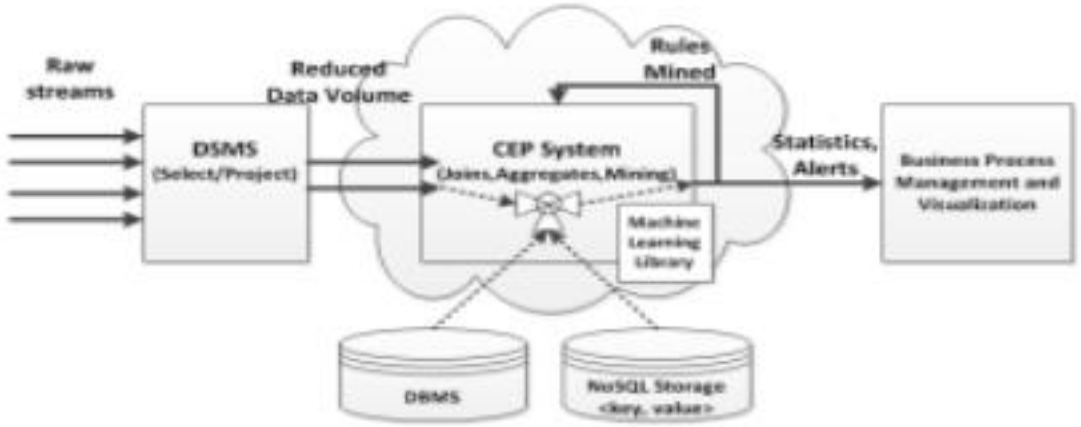


Рисунок 2. Аналитика потоков данных и архитектура системы интеллектуального анализа данных.

Развертывания в публичном и частном облаке для потоков. Большинство организаций сохраняют потоковые данные в своих частных сетях, поскольку эти данные обычно имеют критически важное значение. Поэтому они также предпочитают модель развертывания аналитики потоков данных на основе частного облака (совместного использования). Однако также возможно перенаправлять потоки данных в общедоступную или общественную облачную службу аналитики и майнинга для предварительной или последующей обработки, если эта служба ближе к данным, чем собственные серверы компаний, и надежна. Определения частных и публичных облаков также довольно расплывчаты: некоторые организации могут считать облака внутри (соответственно снаружи) сетевого домена компании или страны частными (соответственно публичными). Что касается проблем с производительностью, потоковая передача данных на ближайшее вычисление или передача вычислений на данные, когда это возможно, являются более дешевой альтернативой, чем попытка переместить большие объемы данных на удаленные вычисления.

ПОТОКОВАЯ АНАЛИТИКА – КОРРЕЛЯЦИЯ. В этом разделе мы описываем нашу реализацию оператора корреляции момента продукта *Pearson* (*PPMC*) для потоков и показываем его применение для сопоставления маршрутов в потоках данных *GPS*. Короче говоря, корреляция – это ковариация двух переменных, деленная на их стандартные отклонения. Значение корреляции может изменяться в диапазоне [-1,+1], где +1 обозначает высокоположительную корреляцию, 0 обозначает отсутствие корреляции и -1 обозначает сильноотрицательную корреляцию. В этой статье, чтобы сопоставить автобус с его ранее записанным маршрутом или обеспечить совместное движение двух

транспортных средств, мы реализовали и использовали следующие непрерывные запросы по широте и долготе транспортных средств (на рис. 3 показана долгота):

```
SELECT CorrelationLong

FROM VehiclePairStream.

WIN:length(50).stat:correl(a.long, b.long)

SELECT CorrelationLat

FROM VehiclePairStream.

WIN:length(50).stat:correl(a.lat, b.lat),
```

где *VehiclePairStream* создается путем объединения двух потоков (*a*, *b*). На рисунке 3 показаны временные ряды данных о долготе, собранные для двух разных автобусов на одном и том же маршруте. Две переменные потока также могут принадлежать текущему исследуемому автобусу и его заранее записанному маршруту.

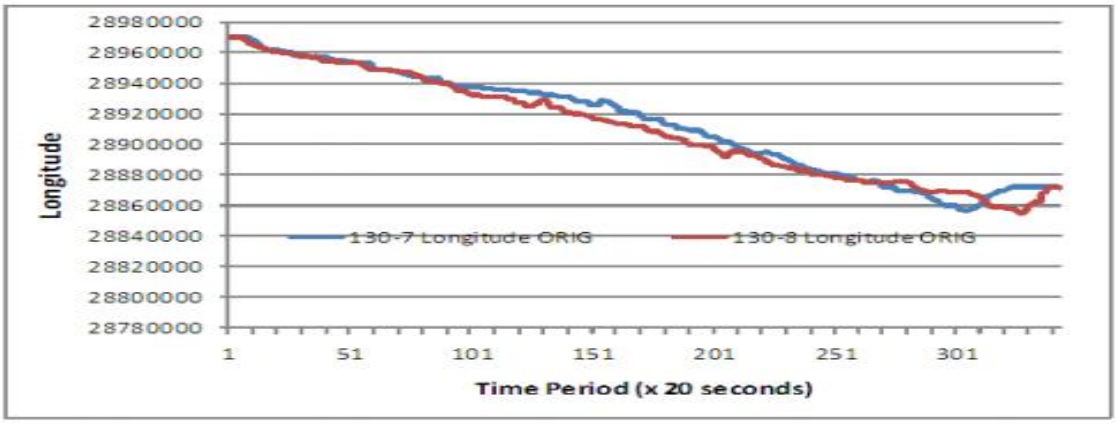


Рисунок 3. Долготная информация в реальном времени по сравнению с записанной или сгруппированные долготы транспортных средств. Эти данные используются для корреляции.

Получен путем усреднения долгот автобусов, ежедневно курсирующих по одному и тому же маршруту. Цель состоит в том, чтобы постоянно отслеживать автобусы и обнаруживать аномалии в режиме реального времени, такие как разделение групп, выезд за пределы маршрута или значительные задержки движения.

На рисунке 4 показаны результаты корреляции для разных размеров скользящего и переворачивающегося окна (25-50-100). Если корреляция C ниже определенного порога (например, $C < 0{,}8$), может быть сгенерирован сигнал тревоги. Автобус либо сбился с маршрута, либо не движется вовремя, и то, и другое указывает на аномалии. Обратите внимание, что окна меньшего размера содержат меньше данных для сравнения, поэтому, когда автобусы движутся, даже немного по-другому. друг с другом, значение корреляции за этот период резко падает. Таким образом, маленькие окна приводят к высокому уровню ложных срабатываний. Для больших размеров окон, например, 50–100, этот эффект компенсируется, поскольку одна шина обычно догоняет другую (или одна и та же шина компенсирует свою переходную задержку на одном и том же маршруте в разное время). Чтобы уменьшить объем обработки и вывода, мы могли бы использовать переключающиеся окна, которые публикуют результаты только в конце периода времени или периода подсчета. Мы обнаружили, что «переворачивающееся окно» по сути является

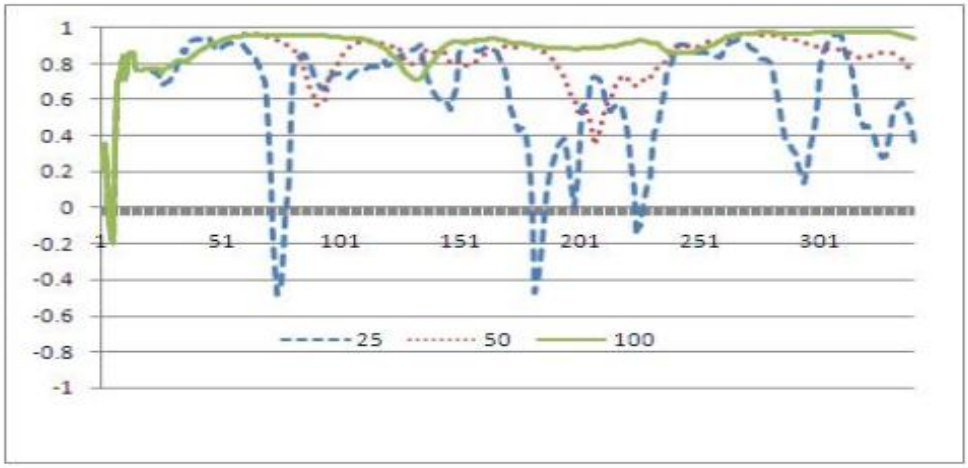
дискретной версией непрерывного скользящего окна, и оба публикуют аналогичные результаты корреляции. Поэтому для краткости мы опускаем результаты в переворачивающихся окнах и отсылаем пользователей к нашей предыдущей работе [3]. Как показано на рисунке 4*b*, изменение размера скользящего окна не влияет на задержку обработки, поскольку части формулы корреляции вычисляются постепенно. Для переключающихся окон задержка увеличивается с размером окна, поскольку все данные, собранные до конца временного интервала, обрабатываются одновременно. Этот вывод согласуется с мотивами быстрых инкрементных обновлений (*FUP*), используемых при поиске часто встречающихся наборов элементов в потоках [7,8,9].

ПРАВИЛА ДОБЫЧИ НАД ПОТОКАМИ. В этом разделе описываются реализации алгоритмов *Apriori* и *FPGrowth* для потоков данных и их применение в качестве механизма рекомендаций музыки в реальном времени.

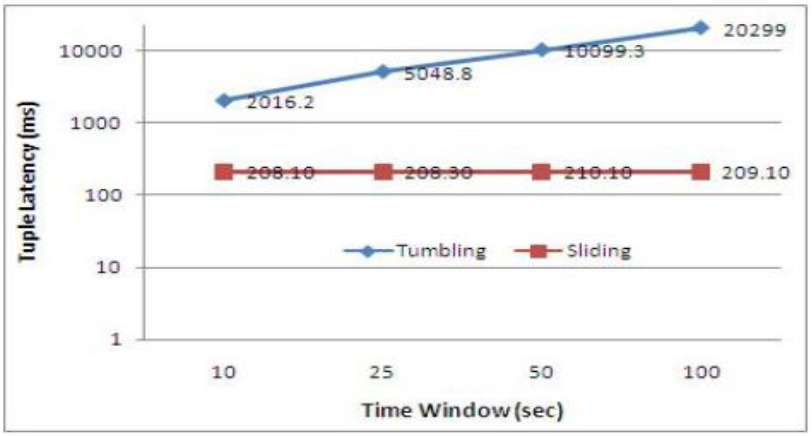
Алгоритмы интеллектуального анализа правил ассоциации (*Association Rule Mining-ARM*)

В правиле ассоциации, обозначаемом $X \Rightarrow Y$ (S,C), X и Y относятся к часто встречающимся наборам элементов, S – к поддержке, а C – к уверенности для правила. Поддержка набора элементов – это

(а)



Изменение значения корреляции с различными скользящими окнами подсчета.



(б) Сравнение задержек обработки скользящих и переворачивающихся окон.

Рисунок 4. (а) Изменение значений широтных корреляций во времени для разных размеры раздвижных окон (б) Сравнение производительности раздвижных и переворачивающихся окон.

процент записей в базе данных, содержащих этот набор элементов (*X*, *Y* или оба). Доверие к вышеуказанному правилу рассчитывается как процент записей, содержащих *X*, которые также содержат *Y*. Формула уверенности также может быть представлена как $Conf(X \Rightarrow Y) = Support(X \cup Y)/Support(X)$. Априорный алгоритм подсчитывает частые наборы элементов, генерирует наборы-кандидаты, используя минимальное значение поддержки (например, 0,1), сокращает редкие, вычисляет достоверность всех перестановок частых наборов элементов и выбирает те, которые превышают заданный порог достоверности. Алгоритм *FP-Growth* выполняет первый проход по транзакциям, создавая базу данных элементов, отсортированную по частоте, опускает редкие элементы и, наконец, создает *FP*-дерево [5,8].

Почему интеллектуальный анализ правил в *Streams* имеет решающее значение?

Мы живем в эпоху, когда тенденции не держатся долго. Поэтому временные аспекты рекомендаций чрезвычайно важны. К сожалению, когда объемы потоковых данных и правила вывода увеличиваются, аналитики данных реагируют на увеличение значений поддержки и достоверности, чтобы получить меньше правил с более сильным подъемом. Тем не менее, правила, которые демонстрируют высокие значения подъема в течение длительного периода времени (например, месяца), возможно, уже устареют к концу этого периода. Например, продажа мороженого, прохладительных напитков и пластиковых стаканчиков будет чрезвычайно популярна в самый жаркий месяц в году, так как мешки с песком и лопаты востребованы во время урагана. После того, как тренд исчезнет, возможностей для продаж не будет. Эти временные закономерности возникают еще быстрее в случае онлайн-продаж или на фондовых рынках, где каждую секунду происходят миллионы транзакций. Потоковый анализ правил может выявить такую тенденцию, как «когда акции *HPQ* и *MSFT* падают более чем на 1%, *DELL* следует за ними» в течение одного часа или дня. Предположим, что недавняя тенденция (т. е. правило с минимальной поддержкой и доверием) возникает только через определенный период времени. Его доверительное значение может не соответствовать самый высокий в мире, но его ценность для местного бизнеса может быть довольно высокой. Предлагаемые нами системы разработаны с учетом этих правил. Другие приложения потокового анализа включают кластеризацию и классификацию потоков [7].

Подумайте о переворачивающемся окне, которое настолько велико, что может охватить все данные, используемые для автономного анализа. В этом случае оффлайн-анализ и онлайн-анализ с одним большим окном дадут одинаковые результаты набора правил. В системах *CEP* и меняющихся окнах достаточно одного параметра (т.е. размера окна), чтобы переключиться с автономного анализа на анализ в режиме, близком к реальному. Если от анализа скользящего временного окна ожидаются дополнительные выгоды, переключение типа окна опять же будет незначительным усилием.

Реализация запроса *ARM*. Мы получили *Java*-реализации алгоритмов *Apriori* и *FP-Growth* из библиотеки правил ассоциации известного инструмента машинного обучения *Weka* [12] и интегрировали эти алгоритмы в движок *Esper*, который также основан на *Java*. Для добавления новых операторов необходимо реализовать специальную функцию агрегирования в *Esper* (класс *AggreationSupport*). Мы реализовали этот интерфейс, чтобы добавить алгоритмы непосредственно в *Esper* для анализа правил потока:

ЗАПРОС 1:

```
SELECT Apriori(parameters, table.feature1, table.feature2)
FROM event.win:length(5) AS table
```

Параметры, которые мы использовали для инициализации алгоритмов *Weka Apriori* внутри *Esper*, были следующими:

'-*N* 10 -*T* 0 -*C* 0,9 -*D* 0,05 -*U* 1,0 -*M* 0,1 -*S* -1,0 -*c* -1'

где *N* – количество выводимых правил, *T* – тип метрики, по которой ранжируются правила (0 = уверенность | 1 = рост | 2 = рычаг | 3 = убежденность), *C* – минимальная оценка метрики (например, минимальная достоверность = 0,9). правила, *U*/*M* – верхняя/нижняя границы минимальной поддержки (по умолчанию = 1,0 и = 0,1), *D* – дельта, на которую уменьшается минимальная поддержка на каждой итерации (по умолчанию = 0,05), *S* – уровень значимости. , а *c* – индекс класса (по умолчанию = последний).

Можно присвоить равные значения границам *U*/*M* (например, 0,3), чтобы избежать итераций, что было бы правильным выбором для сред потоковой обработки. Однако в этом случае пользователь должен хорошо знать домен и правильно установить значения, чтобы найти нужное количество правил для каждого временного окна. В средах динамической потоковой передачи фиксированные ручные настройки могут привести к тому, что будет извлечено слишком много или слишком мало правил. Поэтому мы предпочли, чтобы эту динамическую настройку выполняла система *Weka*. Выбранные функции описаны в части *D*.

ЗАПРОС2:

```
SELECT FPgrowth(parameter, table.feature1, table.feat2)
FROM event.win:length(5) AS table
```

Параметры, которые мы использовали для инициализации алгоритмов *Weka FP-Growth* внутри *Esper*, были следующими:

'-П 2 -И -5 -Н 10 -Т 0 -С 0,9 -Д 0,05 -У 1,0 -М 0,7'

где *P* – индекс атрибута для двоичных атрибутов в обычных плотных экземплярах (используется индекс по умолчанию 2 для разреженных экземпляров), *I* – максимальное количество элементов, включаемых в большие наборы элементов (и правила) (по умолчанию = -1, т. е. без ограничений). .), *N* – необходимое количество выходных правил, *T* – тип метрики, по которому ранжируются правила (0 = достоверность), *C* – минимальная оценка метрики (например, минимальная достоверность = 0,9) правила, *D* – дельта, по которой минимальная поддержка уменьшается на каждой итерации (по умолчанию = 0,05), *U*/*M* – верхняя/нижняя границы минимальной поддержки (по умолчанию = 1,0 и = 0,1).

Набор данных *LastFM* и предварительная обработка. Данные *LastFM* содержат информацию примерно о 1000 людях (набор данных *Lastfm*-1*K*) [10], которые слушают песни в базах данных *LastFM*. В этом наборе данных размером около 3 ГБ около 75 000 уникальных исполнителей, несколько сотен тысяч уникальных песен и миллионы транзакций. Кратко, поля включают в себя <идентификатор пользователя, временную метку, *mbid* исполнителя, имя исполнителя, *mbid* песни, название песни>. На этапе предварительной обработки мы сначала очистили записи с отсутствующей информацией об исполнителе и удалили поля временных песен, которые не способствовали извлечению правил. Этот процесс выполнялся в автономном режиме, и наша будущая работа включает в себя предварительную онлайн-обработку. Мы использовали скользящие окна на основе счетчика. Наконец, у нас были две функции набора данных (<*user-id*, *Artist-mbid*>). Поскольку алгоритм *Apriori* использует большой объем памяти, мы дополнительно обрезали данные, включив в них пользователей, которые прослушивали более 100 песен,

и песни, которые в целом были прослушаны более 3000 раз. В результате 967 уникальных пользователей прослушали 1105 уникальных исполнителей.

Результаты деятельности. Результаты офлайн-анализа, приведенные в таблице 2, показывают, что рост *FP* генерирует результаты «Топ-10 правил» в 75–613 раз быстрее, чем алгоритм *Apriori* в потоке данных. Это соответствует большинству предыдущих работ [9], поскольку *FP*-рост позволяет избежать итеративных поколений кандидатов, рассчитанных с помощью *Apriori*.

Онлайн-анализ проводился в небольших окнах с скользящим подсчетом пользователей и исполнителей размером 10*x*10. На рисунке 5 показаны динамические изменения поддержки *Weka* (*minSupport*, *minConfidence*) и соответствующее количество правил, генерируемых в каждом интервале для набора данных *LastFM*. Ось *X* для графиков скользящего окна подсчета увеличивается на 1 при каждом 1 отсчете события, тогда как переворачивающиеся окна увеличиваются на 1 при каждом слайде, который перемещает окно на 10 событий. Следовательно, для получения одного и того же временного региона между двумя графиками требуется сопоставление 10 к 1 (например, 88 8 или 9). Мы видим, что операция обновления динамической поддержки *Weka* (*U*/*M*) работает правильно и генерирует наборы правил *TOP*10 для каждого периода.

Таблица 2: Результаты оффлайн-анализа (*i*: экземпляры, *a*: атрибуты)

	[i:967 a:1105]	[i: 1105 a: 967]
Apriori	61.403s	226.502s
FPGrowth	0.811s	0.369s

Мы также обнаружили, что переворачивающиеся окна практически генерируют кумулятивную функцию распределения (*CDF*) или «агрегированный» набор правил для правил, найденных с помощью скользящих окон. Тем не менее, наиболее важным выводом из этих результатов является то, что в анализе скользящего окна существуют временные наборы правил, которые упускаются из виду онлайн-анализом скользящего окна из-за агрегирования. Наборы правил были рассчитаны примерно за 300–500 мс для каждого интервала, как показано на рисунке 7 для обоих типов окон. Для краткости мы пропускаем результаты с окнами большего размера.

Заключение. В этой статье мы представили детали реализации для корреляционного анализа и анализа правил для потоков. Мы проанализировали различные типы и размеры скользящих окон с наборами данных движущихся объектов и веб-журналов. В будущем мы планируем (1) тестировать модели с демпфированными окнами (или с затуханием времени), (2) автоматически регистрировать извлеченные правила в *CEP*, (3) предварительно обрабатывать потоковые данные в режиме онлайн и (4) использовать наборы *Java* или несколько наборов. классы для объединения наборов правил (*Union*, *Intersection* и *Difference*), найденных *Apriori* и *FP-Growth*.

Список литературы

[1] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. Aurora: A new model and architecture for data stream management. VLDB Journal, 12(2):120–139, August 2003.

[2]C. Giannella, J. Han, J. Pei, X. Yan, P. S. Yu; Mining Frequent Patterns in Data Streams at Multiple Time Granularities; Data Mining: Next Generation Challenges and Future Directions, AAAI/MIT; 2003.

Авторский вклад
Авторы внесли равноценный вклад в написании статьи

DATA STREAM ANALYSIS

Gylychtaganov Sh.
Head of the Department of Software of information technologies,
The Institute of Telecommunications and Informatics of Turkmenistan

Sh. Y. Tedzhenov
Lecturer of the department "Software information technologies" department
Institute of Telecommunications and Informatics of Turkmenistan

Annotation. Due to prevalent use of sensors and network monitoring tools, big volumes of data or “big data” today traverse the enterprise data processing pipelines in a streaming fashion. While some companies prefer to deploy their data processing infrastructures and services as private clouds, others completely outsource these services to public clouds.

Keywords: Data streams, Complex Event Processing, Association Rule Mining, Stream mining, Correlation, Apriori, FP-growth.