ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

На правах рукописи

Антипов Сергей Геннадьевич

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ И АЛГОРИТМОВ ОБОБЩЕНИЯ ЗНАНИЙ ДЛЯ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ

Специальность 05.13.17 —
«Теоретические основы информатики»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Вагин Вадим Николаевич

Москва — 2016

Оглавление

Введение

Актуальность темы.

Интеллектуальный анализ данных является на сегодняшний день одним из активно развивающихся направлений в области искусственного интеллекта, тесно связанным с проблематикой машинного обучения и задачами выявления скрытых закономерностей. Важнейшим классом задач, решение которых требует интеллектуальной поддержки компьютерных систем, являются задачи управления сложными техническими объектами. Главной чертой подобных объектов управления следует признать то, что они являются динамическими, обладают способностью к развитию, состояния таких объектов и систем могут изменяться со временем. Появление и развитие средств управления объектами, относящимися к категории динамических, тесно связано с развитием интеллектуальных систем поддержки принятия решений (ИСППР), включая наиболее сложных их представителей – ИСППР реального времени, основным направлением развития которых является разработка динамических моделей для представления и манипулирования знаниями о событиях, фактах, действиях, процессах, отражающих динамику поведения сложного технического объекта. Поэтому актуальной является задача разработки моделей представления знаний, процедур обобщения накопленного опыта и реализации соответствующих базовых программных средств. Известен целый ряд методов и алгоритмов, способных решать задачи обобщения: индукция решающих деревьев, приближенные множества, сети Байеса и многие другие. В разработке таких методов принимали участие как выдающиеся зарубежные учёные Quinlan R., Pawlak Z., Mingers J., Utgoff P., так и российские учёные Вагин В.Н., Финн В.К., Журавлёв Ю.И. Однако характерной особенностью таких методов является то, что результаты обобщения являются статичными и не учитывают такой важный при диагностике состояний сложной технической системы фактор как время. Разработкой методов и алгоритмов, учитывающих фактор времени, занимаются такие зарубежные ученые, как Console L., Picardi C., Dvorak P., Kuipers B., Sachenbacher M., Malik A., Dupret D, Keogh E., Pazzani M., Olszewski R., Geurts P. С помощью таких моделей можно представлять не только статические, но и динамические знания о поведении сложного технического объекта. Интеллектуальные системы нового поколения ориентируются на работу с объектами, для которых характерно динамическое изменение состояний. Индуктивные модели, полученные на ос-

нове анализа таких данных, должны учитывать динамику поведения объекта, что является крайне важным, например, при диагностике текущего состояния и прогнозировании дальнейшего поведения сложной технической системы.

Объектом исследований являются интеллектуальные системы поддержки принятия решений реального времени (ИСППР РВ). **Предметом исследований** – методы и алгоритмы обобщения знаний, позволяющие учитывать фактор времени, и их применение в ИСППР РВ.

Целью данной работы является исследование и разработка методов и алгоритмов обобщения знаний, позволяющих получать обобщённые описания классов ситуаций, изменяющихся со временем.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

- исследование существующих методов и алгоритмов представления и обобщения знаний в интеллектуальных системах поддержки принятия решений;
- разработка методов и алгоритмов обобщения знаний, позволяющих получать обобщённые описания классов ситуаций (объектов), изменяющихся со временем;
- изучение возможности использования методов методов и алгоритмов обобщения знаний в интеллектуальных системах поддержки принятия решений реального времени;
- расширение понятийного аппарата: введение понятий, учитывающих динамическую природу объектов обобщения; формализация задачи обобщения для работы с динамическими данными;
- разработка методов и алгоритмов обобщения знаний для динамических объектов обобщения;
- проектирование и разработка программного комплекса, реализующего рассмотренные в работе методы и алгоритмы.

Методы исследования. Поставленные задачи решаются с использованием методов дискретной математики, теории информации, искусственного интеллекта, а также методов анализа вычислительной сложности алгоритмов.

Научная новизна. При выполнении диссертационной работы автором был получен ряд результатов, обладающих научной новизной. К их числу можно отнести следующие:

1. введено понятие динамического объекта обобщения, на основании которого решается задача обобщения понятий при наличии темпоральных данных;
2. дана постановка задачи обобщения для динамических объектов;
3. предложен метод обнаружения аномалий в наборах временных рядов, относящихся к одному или нескольким допустимым классам;
4. разработаны алгоритмы *«TS-ADEEP»*, *«TS-ADEEP-Multi»*, реализующие предложенные методы; получены оценки вычислительной сложности алгоритмов;
5. предложен метод построения темпоральных деревьев решений для динамических объектов обобщения;
6. разработан алгоритм *«Темпоральный ID3»*, реализующий предложенный метод и получена оценка его вычислительной сложности.

Практическая значимость работы заключается в создании программного комплекса, реализующего разработанные методы и алгоритмы обнаружений аномалий в наборах временных рядов и обобщения динамических объектов. Практическая значимость подтверждается использованием полученных результатов в ООО «Фактор-ТС» для анализа данных, передаваемых через сеть, и обнаружения аномалий в передаваемых данных с целью выявления несанкционированных вторжений в инфраструктуры передачи данных, а также в учебном процессе в ФГБОУ ВО «НИУ «МЭИ» при изучении дисциплин «Математическая логика» и «Дискретные математичекие модели», о чем свидетельствуют акты о внедрении.

Достоверность научных результатов подтверждена теоретическими выкладками, результатами компьютерного моделирования и сравнением полученных результатов с данными, приведенными в научной литературе.

Апробация работы. Основные результаты работы докладывались на следующих конференциях: XVI Международная научно-техническая конференция «Информационные средства и технологии» (2008), XII Московская международная телекоммуникационная конференция студентов и молодых ученых «Молодёжь и Наука» (2009), 15-ая и 16-ая международная научно-техническая конференция студентов и аспирантов «Радиоэлектроника, электротехника и энергетика» (2009, 2010, Москва), 12-ая национальная конференция по искусственному интеллекту с международным участием КИИ-2010 (Тверь), Международный конгресс по интеллектуальным системам и информационным технологиям (Ге-

ленджик, 2012), 14-ая национальная конференция по искусственному интеллекту с международным участием КИИ-2014 (Казань), а также на Общемосковском научном семинаре «Проблемы искусственного интеллекта» (2010).

Реализация результатов. Результаты диссертационной работы вошли в отчёты по НИР, выполняемым по грантам РФФИ: №08-07-00212-а «Исследование и разработка методов и инструментальных средств индуктивного формирования понятий в интеллектуальных системах поддержки принятия решений», №09-01-00076-а «Исследование и разработка методов интеллектуального анализа информации и обнаружения знаний в «зашумленных» базах данных», №11-07-00038-а «Исследование и разработка методов и инструментальных средств достоверного и правдоподобного вывода в интеллектуальных системах поддержки принятия решений», №12-01-00589-а «Исследование и разработка методов индуктивного формирования понятий в темпоральных и «зашумленных» базах данных», №14-07-00862 «Методы и инструментальные средства анализа данных в системах поддержки принятия решений», №15-01-00567 «Исследование и разработка методов и алгоритмов индуктивного формирования понятий в интеллектуальных системах поддержки принятия решений».

Публикации. Результаты диссертации представлены в 14 публикациях, в том числе в 5 научных журналах, рекомендованных ВАК РФ.

Структура и объём работы. Диссертация состоит из введения, четырех глав, заключения и приложений. Полный объем диссертации 152 страницы текста с 85 рисунками и 36 таблицами. Список литературы содержит 120 наименований.

Во введении обоснована актуальность темы диссертации, определены научная задача и цель диссертации, сформулированы её научная новизна и практическая значимость, приведено краткое содержание по разделам.

В первой главе приводится обзор методов представления знаний в интеллектуальных системах, рассматривается проблема обобщения понятий в интеллектуальных системах поддержки принятия решений реального времени, вводится понятие динамического объекта обобщения – структуры, описывающей динамическое состояние сложного технического объекта (системы), одним из параметров которой является время; приводится постановка задачи обобщения для динамических объектов.

Во второй главе подробно описаны временные ряды, способы их представления, методы работы с зашумленными данными; приводится постановка задачи обнаружения аномалий в наборах временных рядов с одним и несколькими классами, рассматриваются существующие подходы к решению таких задач; предлагаются новые методы обнаружения аномалий для наборов временных рядов с одним и несколькими классами; реализованы непараметрические алгоритмы для обнаружения аномалий в наборах временных рядов.

В третьей главе приводится подробное описание задачи обобщения динамических объектов для общего случая; описана задача технической диагностики, которую можно решить, используя обобщение динамических объектов, и рассмотрены основные подходы к ее решению; предложен подход с использованием темпоральных деревьев решений: описана модель темпоральных деревьев решений, использующаяся для решения задачи; предложен новый алгоритм формирования темпоральных деревьев решений, проведено его сравнение с уже существующим; описан способ, с помощью которого можно моделировать процесс диагностики с использованием темпоральных деревьев решений.

В четвертой главе описан реализованный программный комплекс, позволяющий производить предварительную обработку данных для задач обнаружения аномалий и технической диагностики; решать задачу обнаружения аномалий для наборов временных рядов с одним и несколькими классами; решать задачу технической диагностики для наборов временых рядов с одним и несколькими классами. Приведены результаты использования программного комплекса для решения задачи обнаружения аномалий в наборах временных рядов с одним классом (алгоритм TS-ADEEP; алгоритм дискретизации и использование деревьев решений). Приведены результаты использования программного комплекса для решения задачи обнаружения аномалий в наборах временных рядов с несколькими классами (алгоритм TS-ADEEP-Multi; алгоритм дискретизации и использование деревьев решений). Приведены результаты моделирования процесса диагностики с использованием темпоральных деревьев решений. Проведено сравнение реализованных алгоритмов с другими алгоритмами, решающими аналогичные задачи. Сделаны выводы об эффективности предложенных алгоритмов при решении задач обнаружения аномалий в наборах временных рядов и диагностики.

В заключении приведены основные результаты, полученные в диссертационной работе.

Глава 1. Модели и методы обработки и анализа данных в интеллектуальных системах

Развитие современных сложных информационных систем тесно связано с развитием наиболее совершенных их представителей, к которым относятся интеллектуальные системы. Интеллектуальная система (ИС) [15] может быть рассмотрена как компьютерная система для решения классов задач, традиционно считающихся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы и которые или не могут быть решены человеком в реальное время, или же их решение требует автоматизированной поддержки. Решение, предоставляемое интеллектуальной системой, должно давать результаты, сопоставимые с решениями, принимаемыми человеком-специалистом в некоторой области. Характеризация компьютерной системы как интеллектуальной будет неполной, если не будут уточнены как природа решаемых задач, так и средства их решения, реализуемые благодаря определенной архитектуре компьютерной системы [16].

Важнейшим классом задач, решение которых требует интеллектуальной поддержки компьютерных систем [17], являются задачи управления сложными техническими объектами. Главной чертой подобных объектов управления следует признать то, что они являются динамическими, обладают способностью к развитию, состояние таких объектов и систем может изменяться со временем.

Появление и развитие средств управления объектами, относящимися к категории динамических, тесно связано с развитием интеллектуальных систем поддержки принятия решений (ИСППР). В настоящее время интеллектуальные системы поддержки принятия решений работают со всё более сложными техническими объектами и системами. В основе интеллектуальных систем данного типа лежит интеграция моделей представления и манипулирования знаниями. Модели должны быть ориентированы на специфику предметной области и иметь развитые средства представления знаний о событиях, фактах, действиях, процессах, происходящих на сложном техническом объекте.

На рис. 1.1 представлена базовая структура ИСППР [18], включающая такие подсистемы, как база данных и база знаний, база моделей, блок поиска решений, блок анализа ситуации, средства интеллектуального интерфейса. Крайне важными компонентами ИСППР являются также блоки, предназначенные для обобщения информации: это блоки приобретения и накопления знаний, обу-

чения, адаптации, модификации. В основе функционирования этих подсистем лежат модели и методы индуктивного формирования понятий.

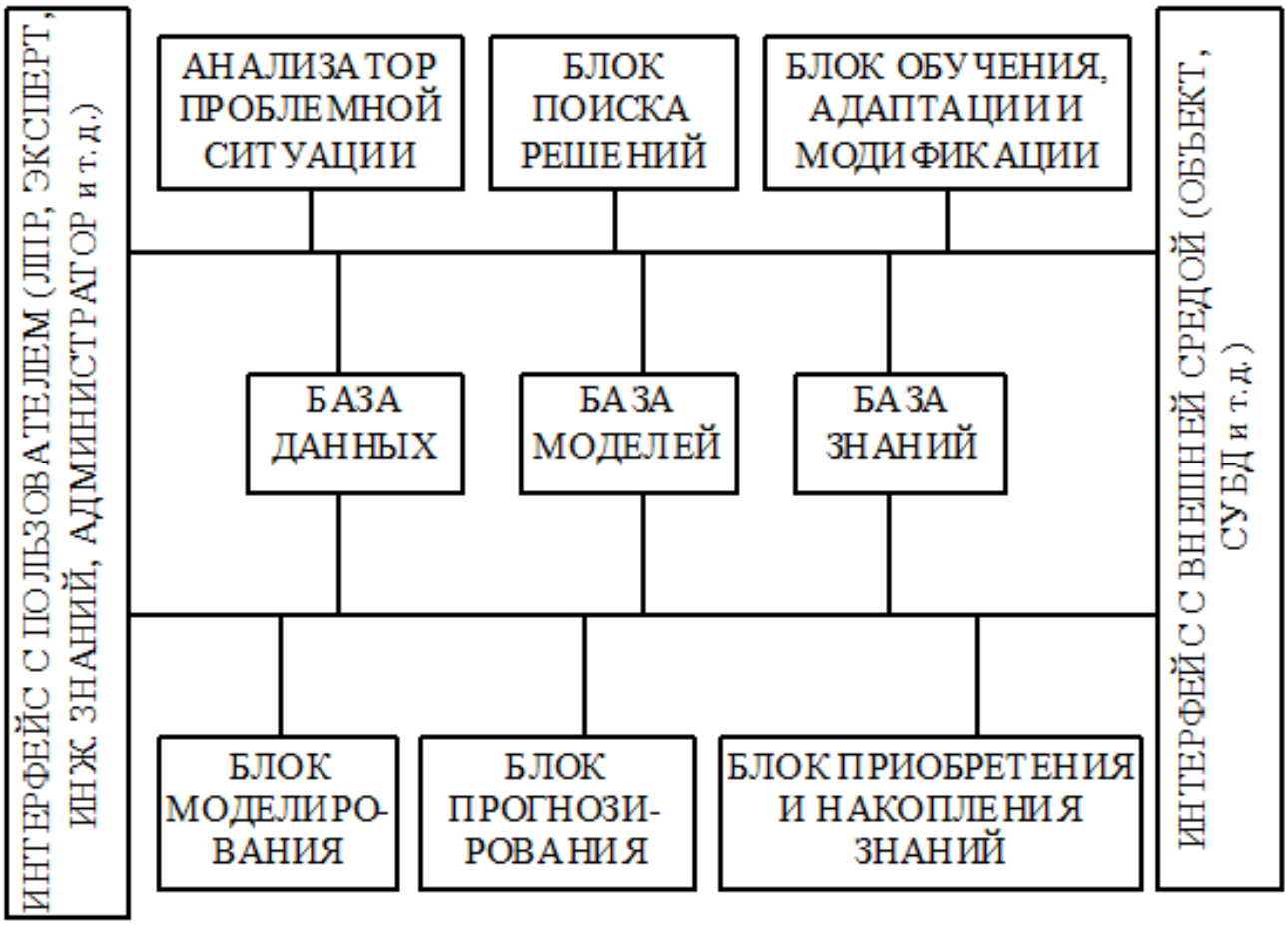


Рисунок 1.1 — Базовая структура ИСППР

Задачи индуктивного формирования понятий важны тем, что должны отображать такие аспекты естественного интеллекта, как способности обобщения – упорядочения данных и знаний с выделением существенных параметров в данных в соответствии с поставленной целью.

При решении задачи индуктивного формирования понятий (задачи обобщения информации) в рамках ИСППР необходимо [19]:

- определить методы представления знаний для решения задачи обобщения,
- выбрать метод представления полученного обобщенного описания (определяется, прежде всего, тем, для каких целей будет использоваться полученное описание),
- выбрать методы и алгоритмы обобщения, универсальные или предметно-ориентированные, т.е. предназначенные для конкретной предметной области.

Первые две задачи связаны с используемым в ИСППР методом представления знаний. Рассмотрим основные методы и модели представления знаний в интеллектуальных системах.

1.1 Методы представления знаний в интеллектуальных системах

Существует целый ряд различных методов представления знаний [20]: к ним относятся продукционные модели, семантические сети, схемы, фреймы, сценарии, искусственные нейронные сети, логические модели, деревья решений и др.

В системах, основанных на **продукционных правилах**, знания представлены в форме множества правил, которые указывают, какие заключения должны быть сделаны или не сделаны в различных ситуациях [21].

Интеллектуальная система, использующая продукционную модель представления знаний, включает в себя базу знаний, хранящую как множество правил вида «Если <условие> ТО <заключение> », так и множество фактов, истинных в данный момент. Интерпретатор управляет тем, какое правило должно быть выбрано для исполнения в зависимости от наличия истинных фактов в рабочей памяти. Система, основанная на правилах, состоит из правил IF-THEN, фактов и интерпретатора, который управляет тем, какое правило должно быть вызвано в зависимости от наличия фактов в рабочей памяти.

В экспертных системах часто используются такие правила, в которых посылкой является описание ситуации, а заключением – действия, которые необходимо предпринять в данной ситуации.

Если интеллектуальная система предназначена для решения задачи обобщения, продукционные правила в качестве посылок могут использовать условия, которым удовлетворяет описание рассматриваемого объекта, а заключением должен стать вывод о принадлежности объекта к определенному классу.

Широкое применение продукционных моделей при разработке интеллектуальных систем обусловлено следующими причинами [20]:

– модульная организация: отдельные продукционные правила могут быть независимо добавлены в базу знаний, исключены или изменены, при этом не требуется перепрограммирование всей системы. Таким образом, продукционная модель является открытой моделью представления зна-

ний. Как следствие этого, представление больших объемов знаний не вызывает затруднений.

– наличие средств объяснения: в продукционные модели легко встраиваются средства объяснения, позволяющие отследить, запуск каких правил и в каком порядке был осуществлен, таким образом, всегда можно восстановить ход рассуждений, которые привели к определённому заключению;
– наличие аналогии с познавательным процессом человека: согласно гипотезе Ньюэлла-Саймона [22] продукционные правила, по-видимому, представляют собой естественный способ моделирования процесса решения задач человеком; кроме того, продукционные правила легки для восприятия человеком.
– с помощью продукционных правил выражаются как декларативные, так и процедурные знания.

Следующей широко известной моделью представления знаний являются **семантические сети**. Семантические сети [23–25] – классический способ представления информации, используемый в искусственном интеллекте. С точки зрения математики семантическая сеть представляет собой помеченный ориентированный граф, вершины которого обозначают некоторые сущности (объекты, события, процессы, явления, ситуации), а дуги – отношения между сущностями, которые они связывают. Отношения имеют для семантических сетей исключительно важное значение, поскольку представляют базовую структуру для организации знаний: если заданы отношения, то знания представляют собой связную структуру, исследование которой позволяет выводить логическим путем другие знания. Главным преимуществом семантических сетей является то, что вся информация, связанная с некоторым объектом, легко может быть получена по связям этого объекта.

При использовании семантических сетей в системах обобщения и извлечения знаний возникает проблема поиска общих закономерностей в описаниях объектов либо ситуаций в случае, когда каждый отдельный пример объекта (ситуации) представляется отдельной семантической сетью. Операции над такими примерами сводятся к операциям над графами, например, к поиску наибольшего подграфа, общего для всех примеров заданного класса. Недостатком данной

модели является сложность работы с неоднородными графовыми структурами и связанный с этим большой перебор.

Одной из разновидностей сетевых моделей, которая широко используется во многих системах искусственного интеллекта, является модель на основе **фреймов** [26]. Каждый отдельный фрейм является сложно организованной структурой и представляет собой сценарий, который описывает типовую ситуацию, связанную, например, с каким-либо видом деятельности. Фреймы обычно образуют сетевые структуры и иерархии, связанные взаимными ссылками. Фреймы широко используются для решения таких задач, как понимание зрительных образов, анализ текстов на естественных языках и в ряде других областей.

Для фрейма характерно представление взаимосвязанных знаний по конкретной теме или ситуации в виде набора слотов, характеризующих отдельные черты, свойства, особенности ситуации, при этом значения слотов в большей части задаются по умолчанию. Типовое, или «скелетное» описание ситуации может пополняться и изменяться за счет значений, поступающих из других фреймов, при этом типовые значения слотов уточняются и конкретизируются.

С точки зрения обобщения информации легко заметить, что фреймы предоставляют удобную структуру для описания объектов, типичных для какой-то конкретной ситуации, в частности, стереотипов объектов. Недостатком такой модели является прежде всего ее направленность на решение задачи конкретизации описания случая, объекта, ситуации, а не на получение новых обобщенных понятий путём генерации новых фреймов.

Когнитивные карты [27] (термин впервые появился в [28]) относятся к тому же классу систем представления знаний, что и фреймы. Когнитивные карты могут быть полезным инструментом для формирования и уточнения гипотезы о функционировании исследуемого объекта, рассматриваемого как сложная система [29]. Для того чтобы понять и проанализировать поведение сложной системы, целесообразно построить структурную схему причинно-следственных связей. Когнитивную карту можно понимать как схематичное, упрощенное описание картины мира индивида, точнее ее фрагмента, относящегося к данной проблемной ситуации. Психологи последнее время используют этот термин в узком смысле, только для описания пространственных отношений. Представляется, что термин

«когнитивная карта» значительно теснее связан с общепринятым пониманием картины мира, чем введенные лингвистами понятия «фрейм» и «скрипт».

Искусственная **нейронная сеть** – математическая модель, построенная по принципу организации и функционирования биологических нейронных сетей, т. е. сетей нервных клеток живого организма. Основой нейронных сетей является искусственный нейрон, который имеет следующую структуру [30]:

- входные сигналы x_i: данные, поступающие из окружающей среды или от других активных нейронов. Диапазон входных значений для различных моделей может отличаться. Обычно входные значения бывают дискретными (бинарными) и определяются множествами {0, 1} или {-1, 1}, либо принимают любые вещественные значения.
- набор вещественных весовых коэффициентов w_i: весовые коэффициенты определяют силу связи между нейронами.
- уровень активации нейрона $\sum w_i * x_i$, который определяется взвешенной суммой его входных сигналов x_i.
- пороговая функция f, предназначенная для вычисления выходного значения нейрона путем сравнения уровня активации с некоторым порогом. Пороговая функция определяет активное или неактивное состояние нейрона.

Первым примером нейросетевой модели является нейрон Мак-Каллока-Питтса [31]. В настоящее время нейронные сети применяются во множестве задач, среди которых наиболее важными являются следующие:

- классификация;
- распознавание образов;
- реализация памяти;
- прогнозирование;
- оптимизация;
- фильтрация.

Особенностью нейронных сетей является то, что они обучаются. Возможность обучения – одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения

сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искаженных данных.

Важной частью любой интеллектуальной системы является подсистема логического вывода. Традиционно основой процесса формирования рассуждений является дедуктивный логический вывод, основанный на получении заключения из посылок. Развитие **логических моделей** и их реализация на ЭВМ привели к созданию логического программирования и к разработке таких языков, основанных на логике, как PROLOG [32]. Классические логические модели играют важную роль в экспертных системах, поскольку в таких системах необходимы средства логического вывода, позволяющие проводить рассуждения от фактов к заключениям.

Однако задачи, решаемые в интеллектуальных системах, часто являются некорректными в том смысле, что они требуют применения эвристик и не предполагают полноты знаний [33;34], являющихся исходными посылками при классическом логическом выводе. Это означает, что в рамках ИС нужно отображать такие способности к рассуждению [21], как синтез различных познавательных процедур, способности к выдвижению гипотез, способности к обучению на основе позитивных и негативных примеров, и, наконец, способности к адаптации в соответствии с изменением множества фактов и знаний. Такие задачи требуют развития прежде всего неклассических логик и создания новых алгоритмических и программных систем для реализации нетрадиционного вывода.

Реализация индуктивных рассуждений или рассуждений по аналогии позволяет получить правдоподобные выводы. Одной из наиболее успешных моделей представления знаний для индуктивного вывода является модель **деревьев решений**. Представление знаний с помощью деревьев решений с успехом было использовано в ряде систем обучения с учителем, например, в алгоритме *ID3* Куинлана [35].

Теория *деревьев решений* базируется на информационных оценках; деревья решений используются при решении классификационных задач и представляют собой процедуру определения класса для предъявленного примера. Каждый узел дерева определяет или имя класса, или специфическую проверку, разделяющую пространство примеров, приписанных узлу, в соответствии с возможными результатами проверки. Каждое подмножество примеров, возникшее в результате

такого разделения, соответствует классификации подпроблемы для пространства примеров, которое получено на поддереве. Дерево решений можно представить как стратегию «дробления и продвижения вперед» для объекта, подлежащего классификации. Формально можно определить дерево решений как граф, не содержащий циклов, в котором каждая вершина – это или конечный узел, взвешенный именем класса, или промежуточный узел, содержащий проверку значений атрибута с дальнейшим расщеплением на поддеревья для каждого допустимого значения атрибута.

Например, алгоритм ID3 (см. [35]) строит дерево решений на основе множества примеров, для которых известен результат классификации, начиная с корневого узла (вершина дерева) вниз к конечным узлам (листьям). На каждом этапе построения информационная связь между классификационным и исследуемым атрибутами используется для выбора атрибута, на основании которого происходит ветвление в данной точке.

Информационная связь между классификационным атрибутом и исследуемым атрибутом называется также приростом информативности (information gain [35]), и определяется на основе частоты появления значений признаков атрибута в тестовом множестве примеров.

Дерево решений можно рассматривать как особую форму теста, который предписывает определенные проверки на каждом шаге анализа.

В общем, деревья решений представляют собой дизъюнкцию конъюнкций ограничений для значений атрибутов примеров. Каждый путь от корня дерева к конечной вершине (листу) соответствует конъюнкции проверок условий на значения атрибутов, а дерево в целом представляет дизъюнкцию таких конъюнкций. Точнее, решающие деревья классифицируют примеры путем сортировки их с помощью дерева от корневого узла к одному из конечных узлов (листьев), в которых выполняется классификация примера. Каждый узел в дереве решений определяет проверку значения некоторого атрибута примера, а каждое ветвление, выходящее из этого узла, соответствует одному из возможных значений этого атрибута.

Классификация примера начинается с корня дерева решений, где выполняется проверка атрибута, приписанного данному узлу (тест для данного атрибута), затем, выбирается путь для движения вниз по одной из ветвей дерева в соответствии со значением атрибута. Процесс повторяется в узле, которым за-

канчивается выбранная ветвь, и так далее, до тех пор, пока не будет достигнут конечный узел (лист). Конечному узлу приписан один из возможных ответов (решение)

Из разных видов обобщения для целей систем поддержки принятия решений [36] реального времени (СППР РВ) наиболее пригоден вариант обобщения на основе признакового описания как самих объектов, так и ситуаций, возникающих на сложном техническом объекте (в технической системе [37]).

Из рассмотренных выше моделей представления знаний в дальнейшем предлагается использовать такие модели, как деревья решений, и продукционные модели: их основными чертами являются универсальность, простота реализации и удобство преобразования дерева решений в продукционные правила.

1.2 Проблема обобщения понятий

В системах, моделирующих мышление, обобщение понимают как процесс получения знаний, объясняющих имеющиеся факты, и способных объяснять, классифицировать или предсказывать новые [38]. В общем виде задача обобщения была сформулирована Михальским [39] следующим образом: по совокупности наблюдений (фактов) F, совокупности требований и допущений к виду результирующей гипотезы H, и совокупности базовых знаний и предположений, включающих знания об особенностях предметной области, выбранном способе представления знаний, допустимых операторов, эвристик и др., сформировать гипотезу $H : H => F$ (H «объясняет» F).

Форма представления и общий вид гипотезы H, а также выбранные модели обобщения зависят от цели обобщения и выбранного способа представления знаний. Согласно Михальскому [39], можно выделить модели обобщения по выборкам и модели обобщения по данным. В первом случае совокупность фактов F имеет вид обучающей выборки – множества объектов, каждый из которых сопоставляется с именем некоторого класса. Целью обобщения в этом случае может быть

– формирование понятий, то есть построение по данным обучающей выборки для каждого класса максимальной совокупности его общих характеристик;

– классификация, или построение по данным обучающей выборки минимальной совокупности характеристик, которая отличала бы элементы одного класса от элементов других классов;
– определение закономерности последовательного появления событий.

К моделям обобщения по выборкам относятся лингвистические модели, методы автоматического синтеза алгоритмов и программ по примерам. В моделях обобщения по данным априорное разделение фактов по классам отсутствует. Здесь могут ставиться такие цели:

– получение гипотезы, обобщающей данные факты;
– выделение образов на множестве наблюдаемых данных, группировка данных по признакам;
– установление закономерностей, характеризующих совокупность наблюдаемых данных.

По способу представления знаний и допущений на общий вид объектов, вошедших в обучающую выборку, методы обобщения делятся на методы обобщения по признакам и структурно-логические (концептуальные) методы. В первом случае объект обучающей выборки представляются в виде совокупности значений косвенных признаков. Методы обобщения и распознавания различаются для качественных и количественных признаков. В формально-логических системах, использующих структурно-логические методы обобщения, вывод общих следствий из данных фактов называется индуктивным выводом. Правило вывода гипотезы H из фактов F называют индуктивным, если из истинности H следует истинность F, а обратное неверно.

Главной особенностью структурно-логических методов, в отличие от признаковых методов, является использование в обучающих выборках объектов, имеющих внутреннюю логическую структуру. Такими объектами могут быть последовательности событий, иерархически организованные сети, алгоритмические и программные схемы.

1.3 Задача обобщения понятий по признакам

Прежде всего, из всех возможных задач, связанных с построением индуктивных зависимостей, выделим круг задач, называемых задачами индуктивного формирования понятий; это задачи, которые моделируют возможность человека давать описания, охватывающие множество примеров некоторого понятия. В ос-

нове процесса индуктивного формирования понятий лежит умение человека выделять некоторые наиболее общие или характерные фрагменты описаний среди описаний отдельных примеров понятия, избавляясь от мелких, незначительных характеристик, присущих конкретным примерам понятия. Назовём такую задачу задачей обобщения.

Под обобщением, как правило, понимается переход от рассмотрения единичного объекта *o* или некоторого множества объектов *O* к рассмотрению обобщенного понятия *D*, которое отображает характерные для этого множества отношения между значениями признаков и является достаточным для разделения объектов, принадлежащих множеству, и объектов, не принадлежащих ему, с помощью некоторого правила распознавания [38].

Процесс обобщения тесно связан с понятием машинного обучения [40]. На основе обработки экспертной информации [41] формируется база знаний [42] СППР, в которой хранится модель функционирования системы. С помощью системы обобщения информация о характеристиках конфликтных ситуаций обрабатывается специальным образом и вводится в базу знаний. Множество признаков, характеризующих факт возникновения различных ситуаций, формируется на основе информации, циркулирующей в системе управления. Такая информация часто хранится в виде таблиц в базах данных, причем поля в таких таблицах хранят текущие значения признаков [43]. Значения признаков обычно вполне определенны и достоверны, на основании анализа этих значений необходимо автоматически выполнить обобщение с целью различать типовые и нестандартные ситуации, и выдавать сообщения о факте возникновения нестандартной или конфликтной ситуации. Описания различных ситуаций формируются на основе анализа целей и задач функционирования системы и экспертной информации. Для каждой типовой ситуации необходимо получить обобщенное описание [44] в виде моделей (гипотез) с возможностью их последующей проверки [45].

Пусть $O = \{o_1, o_2, ..., o_n\}$ – множество объектов, которые могут быть представлены в интеллектуальной системе *S*. Каждый объект характеризуется *q* признаками. Обозначим через $X_1, X_2, \ldots, X_q$ множество допустимых признаков, где $X_k = \{x_{k_1}, x_{k_2}, \ldots, x_{k_m}\} \quad (1 \leq k \leq q)$ и x_{k_i} являются значениями признаков. Каждый объект $o_i \in O, 1 \leq i \leq n$, представляется как упорядоченное множество значений признаков, то есть $o_i =< x_1, x_2, .., x_j, .., x_q >$, где $x_j \in X_j, 1 \leq j \leq q$. Такое описание объекта называется признаковым описанием. В качестве призна-

ков объектов могут использоваться количественные, качественные либо шкалированные признаки.

В основе процесса обобщения лежит сравнение описаний исходных объектов, заданных совокупностью значений признаков, и выделение наиболее характерных фрагментов этих описаний. В зависимости от того, входит или не входит объект в объем некоторого понятия, назовем его положительным или отрицательным объектом для этого понятия.

Пусть O – множество всех объектов, которые могут быть представлены в некоторой системе знаний, V – множество положительных объектов и W – множество отрицательных объектов. Будем рассматривать случай, когда положительные и отрицательные объекты образует разбиение множества O, то есть $O = V \cup W$, $V \cap W = \emptyset$, при этом множество отрицательных объектов также разбито на подмножества – в эти подмножества входят примеры, относящиеся к различным классам, отличным от класса объектов из V: $W = \cup W_i$ и $W_i \cap W_j = \emptyset, i \neq j$. Пусть K – непустое множество объектов, такое, что $K = K^+ \cup K^-$, где $K^+ \subset V$, $K^- \subset W$. Будем называть K обучающей выборкой. На основании обучающей выборки надо построить правило, разделяющее положительные и отрицательные объекты обучающей выборки.

Таким образом, понятие сформировано, если удалось построить решающее правило, которое для каждого примера из обучающей выборки указывает, принадлежит этот элемент понятию или нет. Алгоритмы формируют решение в виде набора правил «ЕСЛИ <условие> ТО <искомое понятие> ». Условие представляется в виде логической функции, в которой булевы переменные, отражающие значения признаков, соединены логическими операциями конъюнкции, дизъюнкции, отрицания. Решающее правило считается корректным, если оно в дальнейшем успешно распознает объекты, не вошедшие первоначально в обучающую выборку.

В приведённой задаче обобщения важной проблемой является описание объекта $o \in O$. Традиционно в признаковом описании объекта рассматривается как набор значений признаков $< X_1, X_2, .., X_q >$. Однако современные СППР имеют дело с объектами, требующими более сложных средств описания, чем такая модель, не имеющая возможности представить, например, динамику поведения сложной системы. В связи с этим для описания объекта o требуется

использовать более сложный аппарат, позволяющий в описании объекта учесть фактор времени и изменение значений признаков с течением времени.

Так как состояние сложных технических объектов или систем, с которыми приходится работать ИСППР (РВ), меняется со временем, необходимы использование и разработка методов, неявно или явно учитывающих фактор времени. В связи с этим возникла задача интеллектуального анализа *темпоральных* данных [46–48]. В большинстве случаев крайне затруднительно или вовсе невозможно использовать существующие методы анализа данных в таких предметных областях, где необходимо учитывать фактор времени, следовательно, возникает необходимость модификации существующих методов и разработки новых.

Выделяют 4 категории данных, явным или неявным образом содержащих время [49]:

- статические данные — в таких данных нет и не может быть темпорального контекста; тем не менее фактор времени можно учесть за счет использования журналов рагистрации событий, логов и т. п.
- последовательности данных— упорядоченные списки событий. Эта категория включает упорядоченные совокупности событий, не помеченные временными метками. Массивы данных, включающие в себя потребительские или рыночные корзины чаще всего рассматриваются как последовательности. В то время как большинство совокупностей обычно ограничены отношениями «до» и «после», эта категория позволяет ввести большее количество отношений, описанных в логике Аллена [50] и других;
- данные с временными метками: помеченные временными метками последовательности статических данных, зафиксированные через более или менее регулярные промежутки времени. Примеры включают в себя ценовые и метеорологические данные, а в некоторых случаях – биржевые транзакции или сетевую активность;
- непосредственно темпоральные данные: каждый кортеж в изменяющемся во времени отношении в базе данных может иметь одну или несколько временных размерностей: время транзакции и/или время действия.

1.4 Динамический объект обобщения

Рассмотрим теперь проблему обобщения при наличии темпоральных данных. Важной задачей в таких системах является обработка данных, зависящих от времени. Обычно для контроля за состоянием сложного объекта используется набор датчиков, отображающих и, возможно, контролирующих, значения основных параметров системы. Пусть в системе имеется q датчиков, показания которых снимаются в некоторые дискретные моменты времени: $t = 0, 1, 2, 3, \ldots$. Тогда показания имеющихся датчиков в некоторый момент времени i можно представить в виде вектора (1.1).

$$s_i =< x_1(t = i), x_2(t = i), \ldots, x_q(t = i), t = i > \tag{1.1}$$

Очевидно, такое признаковое описание объектов позволяет лишь взглянуть на мгновенный «слепок» состояния системы. Для того, чтобы проследить динамику развития системы, изменение ее состояния, тенденции, очевидно, необходимо рассмотреть упорядоченное множество таких векторов, полученных на конечном временном интервале $(t_i, t_{i+r-1}), r > 1$. Пусть рассматриваются q параметров на временном интервале длины r. Представим такие данные в следующем виде (табл 1.1):

Таблица 1.1 — Динамический объект обобщения

	Параметр$_1$	Параметр$_2$	...	Параметр$_q$	Время (t)
(s_i)	$x_1(t = i)$	$x_2(t = i)$	...	$x_q(t = i)$	i
(s_{i+1})	$x_1(t = i + 1)$	$x_2(t = i + 1)$	...	$x_q(t = i + 1)$	i+1
(s_{i+2})	$x_1(t = i + 2)$	$x_2(t = i + 2)$	...	$x_q(t = i + 2)$	i+2
...	...	...	...	...	...
(s_{i+r-1})	$x_1(t = i + r - 1)$	$x_2(t = i + r - 1)$	...	$x_q(t = i + r - 1)$	i+r-1

Тогда каждая из строк указанной матрицы, обозначенная $(s_i), (s_{i+1}), .., (s_{i+r-1})$, представляет собой слепок состояния рассматриваемой системы на моменты времени соответственно $i, i + 1, .., i + r - 1$. Каждая ячейка матрицы представляет собой значение соответствующего параметра в определенный момент времени (будем далее называть эти величины *наблюдениями*). Каждый столбец матрицы, обозначенный Параметр$_1$, Параметр$_2$, .., Параметр$_q$, представляет собой значения соответствующего параметра, изме-

няющегося за интервал времени $t^* = r$. Назовем структуру, представленную в табл. 1.1, *динамическим объектом обобщения*.

Сам динамический объект обобщения можно рассматривать, с одной стороны, как совокупность статических слепков состояния системы (строки матрицы), которые, тем не менее, тесно связаны между собой, так как отражают изменение (динамику) состояния системы за определённый интервал времени; с другой стороны, так как параметры обычно являются вещественными, динамический объект обобщения можно рассматривать как набор временных рядов (столбцы матрицы), которые соответствуют изменению значений каждого из рассматриваемых параметров за интервал времени $t^* = r$.

Также динамический объект обобщения может рассматриваться как описание одной конкретной динамической ситуации на сложном техническом объекте, развивающейся за промежуток времени в $N = r$ тактов.

Эквивалентное представление для динамического объекта, которое потребуется нам в дальнейшем, представлено в табл. 1.2 (транспонированная матрица из представления в табл. 1.1):

Таблица 1.2 — Динамический объект обобщения (эквивалентное представление)

	(s_i)	(s_{i+1})	(s_{i+2})	...	(s_{i+r-1})
Время (t)	i	$i+1$	$i+2$	...	$i+r-1$
Параметр$_1$	$x_1(t=i)$	$x_1(t=i+1)$	$x_1(t=i+2)$	...	$x_1(t=i+r-1)$
Параметр$_2$	$x_2(t=i)$	$x_2(t=i+1)$	$x_2(t=i+2)$	...	$x_2(t=i+r-1)$
...	...	...	...	...	...
Параметр$_q$	$x_q(t=i)$	$x_q(t=i+1)$	$x_q(t=i+2)$	...	$x_q(t=i+r-1)$

Приведем теперь постановку задачи обобщения для темпорального случая. Пусть динамические объекты обобщения рассматриваются на временно**м** интервале длиной r, причем каждый такой объект представлен в виде (табл. 1.1) или (табл. 1.2).

Пусть $\hat{O} = \{DynO_1, DynO_2, ..., DynO_n\}$ – множество динамических объектов обобщения, которые могут быть представлены в интеллектуальной системе S. Пусть $\hat{V}$ – множество положительных объектов и $\hat{W}$ – множество отрицательных объектов. Будем рассматривать случай, когда положительные и отрицательные объекты образует разбиение множества $\hat{O}$, то есть $\hat{O} = \hat{V} \cup \hat{W}$, $\hat{V} \cap \hat{W} = \emptyset$, при этом множество отрицательных объектов также разбито на подмножества –

в эти подмножества входят примеры, относящиеся к различным классам, отличным от класса объектов из $\hat{V}$: $\hat{W} = \cup\hat{W}_i$ и $\hat{W}_i \cap \hat{W}_j = \emptyset$. Пусть $\hat{K}$ – непустое множество объектов, такое, что $\hat{K} = \hat{K^+} \cup \hat{K^-}$, где $\hat{K^+} \subset \hat{V}$, $\hat{K^-} \subset \hat{W}$. Будем называть $\hat{K}$ обучающей выборкой. На основании обучающей выборки надо построить правило, разделяющее положительные и отрицательные объекты обучающей выборки.

Задача обобщения в такой постановке является гораздо более сложной и требует разработки как новых способов представления решающих правил, так и новых алгоритмов получения таких правил.

В связи с этим сначала будет рассмотрен простейший случай: единственный параметр, рассматриваемый на некотором временном интервале. В этом случае динамический объект обобщения вырождается во временной ряд. В главе 2 конкретизируется постановка задачи обобщения для такого случая, который фактически может быть сведен к задаче обнаружения аномалий в наборах временных рядов; приводится обзор методов решения подобных задач, предлагаются новые методы решения.

После этого будет рассмотрен и общий случай: в главе 3 конкретизирована постановка задачи для общего случая, описан аппарат темпоральных деревьев решений, который совместно с некоторыми соображениями, полученными в главе 2, позволяет решить задачу диагностики для динамических объектов обобщения в интеллектуальных системах поддержки принятия решений реального времени.

1.5 Выводы к первой главе

В первой главе было сделано следующее:

1. Дан обзор методов представления знаний в современных интеллектуальных системах. Для разработки подсистемы индуктивного формирования понятий в ИСППР обоснован выбор таких моделей, как деревья решений и продукционные модели.
2. Приведена постановка задачи обобщения понятий в интеллектуальных системах для случая признакового описания объектов.
3. Рассмотрена проблема работы с данными, явно зависящими от времени – темпоральными данными. Выделены основные категории таких данных, которые могут использоваться в ИСППР реального времени.

4. Введено понятие динамического объекта обобщения – структуры, описывающей динамическое состояние сложного технического объекта (системы), одним из параметров которой является время.
5. Дана постановка задачи обобщения для случая, когда исходными данными для обобщения являются динамические объекты. Показано, что в случае, когда для описания объектов используется единственный атрибут, явно зависящий от времени, задачу можно свести к задаче анализа временных рядов. Для общего случая, при использовании нескольких зависящих от времени атрибутов, динамический объект принимает вид динамической ситуации; при этом задача обобщения связана с необходимостью анализировать такие динамические ситуации и строить классы сходных ситуаций.

Глава 2. Задача обобщения для динамических объектов. Частный случай.

В данной главе рассмотрен наиболее простой случай задачи обобщения для динамических объектов. Пусть рассматривается ситуация на конечном временном интервале длины $t^* = r, r > 1$, динамический объект описан единственным атрибутом и представлен в виде табл. 1.2. Тогда динамический объект фактически является временным рядом. Рассмотрим временные ряды более подробно.

2.1 Временные ряды

Описание объекта в виде набора значений его свойств используется, в основном, для представления тех объектов, которые со временем не изменяются. Для описания же состояния сложной технической системы требуется способ, позволяющий каким-то образом учитывать фактор времени. Обычно для контроля за состоянием сложной системы используется набор датчиков, отображающих и контролирующих значения основных параметров системы. Изменение значений датчиков со временем позволяет отслеживать изменение состояния системы в целом. Последовательность значений каждого из датчиков представляет собой временной ряд, который, будучи правильным образом проанализирован, может многое сказать о состоянии и изменении состояния сложного объекта. Именно по этим причинам в последнее время уделяется большое внимание интеллектуальному анализу временных рядов [51], которые используются не только в технике, но и в экономике, медицине, банковском деле и т. п.

В данной главе рассмотрена задача обобщения для динамических объектов, поведение которых характеризуется изменением единственного параметра (или атрибута). В этом случае динамический объект представляет собой временной ряд.

Временным рядом [52] называют последовательность наблюдений, обычно упорядоченную по времени, хотя возможно упорядочение и по какому-то другому параметру. Основной чертой, выделяющей анализ временных рядов среди других видов анализа является существенность порядка, в котором производятся наблюдения. Если во многих задачах наблюдения статистически независимы, то во временных рядах они, как правило, зависимы и характер этой зависимости может определяться положением наблюдений в последовательности. Природа ряда и структура порождающего ряд процесса могут предопределять порядок образования последовательности.

В общем случае временной ряд *TS* – это конечная упорядоченная последовательность значений $TS =< ts_1, ts_2, .., ts_r >$, описывающая протекание какого-либо длительного процесса, где $ts_i, 1 \leq i \leq r$ – некоторое вещественное число, индекс *i* соответствует метке времени. Время, как было введено в главе 1, будем считать дискретным, принимающим целочисленные значения 0, 1, 2... . Значениями ts_i могут быть показания датчиков, цены на какой-либо продукт, курс валюты и т. п. Пример временного ряда приведен в табл. 2.1 и на рис. 2.1. По горизонтальной оси отложены метки времени, по вертикальной – соответствующие значения. Пример представления временного ряда в табл. 2.1 в точности соответствует представлению динамического объекта обобщения, введенному в первой главе (табл. 1.2).

Таблица 2.1 — Пример временного ряда

Время	0	1	2	3	4	5	6	7	8	9
Значение	-1.07	0.13	0.85	0.96	0.81	0.84	-0.07	-1.01	-0.90	-1.14

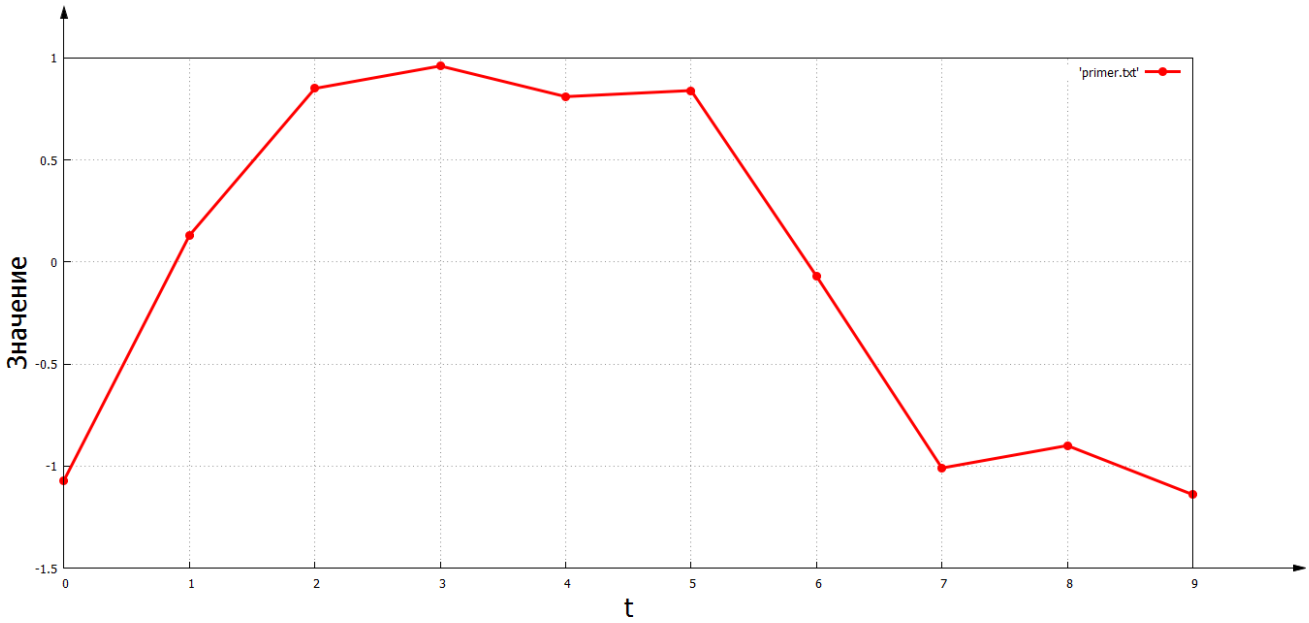


Рисунок 2.1 — Пример временного ряда.

Большинство алгоритмов обобщения понятий работает с дискретными данными, в то время как данные во временных рядах представляют собой вещественные числа. Поэтому представляет интерес способ дискретизации времен-

ных рядов таким образом, чтобы полученные данные могли использоваться в алгоритмах обобщения понятий.

2.1.1 Способы представления временных рядов

Для создания алгоритмов обобщения информации, представленной временными рядами, требуется, безусловно, разработка методов предварительного преобразования самих рядов [53]. Временные ряды, которые представляют данные из разных областей, в различных единицах измерения, требуется привести к некоторым типовым, удобным для дальнейшего анализа формам. Существует более 200 способов представления временных рядов для различных задач в различных предметных областях [54], можно назвать такие виды представлений как кусочно-линейная и кусочно-констатная аппроксимации [55], представление с помощью вейвлетов [56], спектральное [57], символьное и другие [58].

Известные статистические методы анализа временных рядов позволяют получать нормализованные представления числовых последовательностей. Однако для целей обобщения крайне важно перейти от числового представления временного ряда к более общему, абстрактному представлению – представлению временного ряда в виде набора символов.

В основу преобразования, позволяющего получить символьное представление для временных рядов, был положен алгоритм **SAX** [58] (Symbolic Aggregate approXimation). Этот алгоритм преобразует нормализованное представление числового ряда в ряд, состоящий из символов. Данное представление временных рядов успешно используется [59] в различных предметных областях для задач классификации и кластеризации временных рядов, индексирования объемных баз данных временных рядов и поиска в них [60], обнаружения «необычных» [61] или наоборот - наиболее часто встречающихся шаблонов во временных рядах, визуализации длительных временных рядов [62] и многих других. Данное представление является достаточно универсальным и позволяет «сравнивать» временные ряды, имеющие сильно различающиеся параметры - среднее, среднеквадратичное отклонение, размерность.

Рассмотрим временной ряд, представленный на рис. 2.2 (табл. 2.3, вторая колонка). Для получения символьного представления временного ряда сначала необходимо временной ряд «нормализовать»: нормализацией называется приведение временного ряда к такому виду, что среднее для него было бы равно

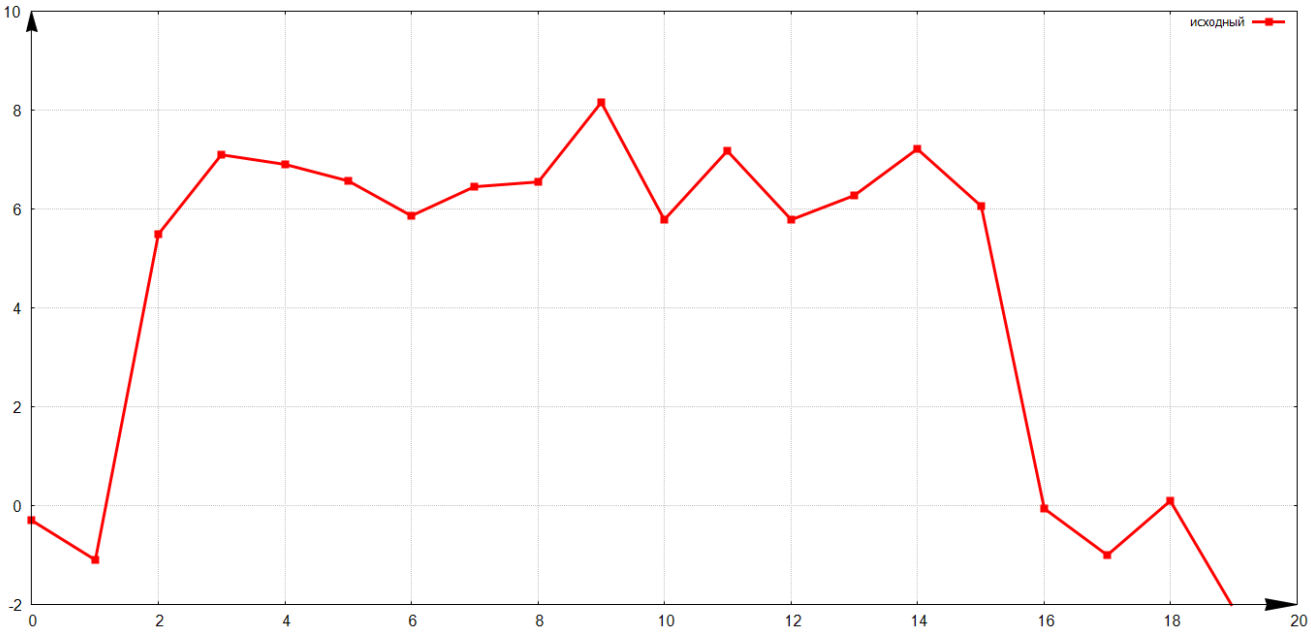


Рисунок 2.2 — Исходный временной ряд

нулю, а среднеквадратичное отклонение – единице; такое преобразование является необходимым процессом при предварительной обработке данных [53; 58]. Именно за счет данной предварительной обработки временного ряда появляется возможность «сравнивать» ряды с различными параметрами. Нормализованный временной ряд, соответствующий рассматриваемому исходному ряду, представлен на рис. 2.3 (табл. 2.3, третья колонка).

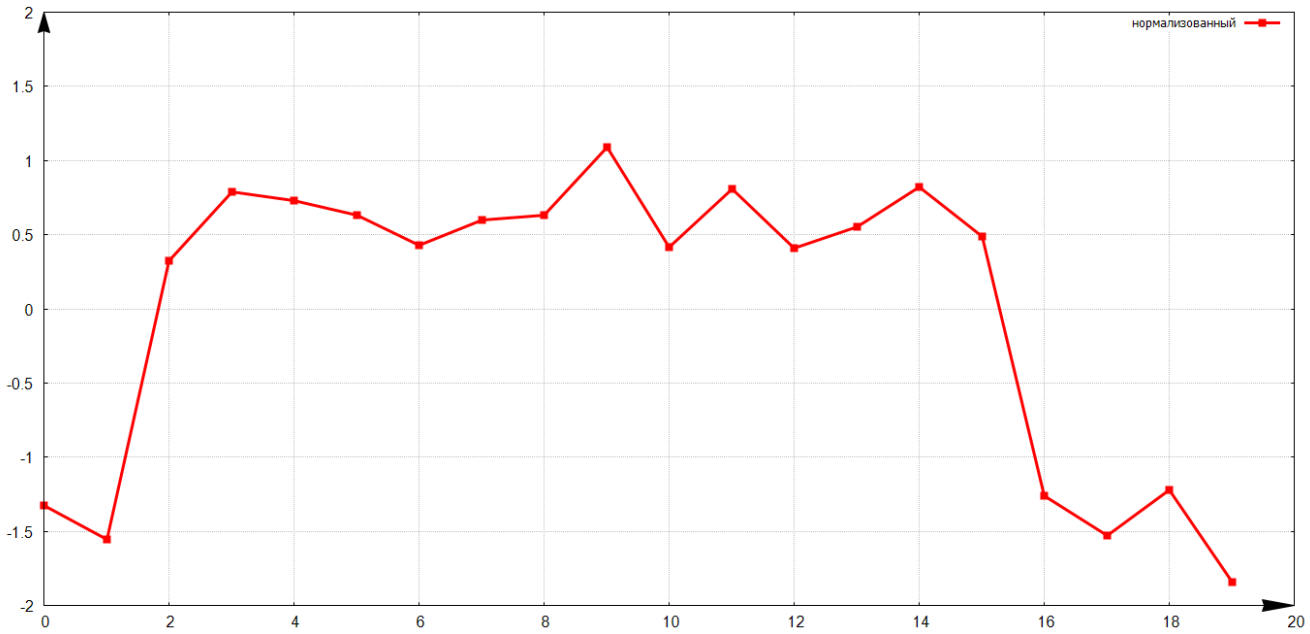


Рисунок 2.3 — Нормализованный временной ряд

Рассмотрим способ преобразования числового ряда в символьное представление SAX [58]. В основу такой процедуры положена дискретизация норма-

лизованного временного ряда. Пусть имеется некоторый алфавит A – конечный набор символов: $A = a_1, a_2, \ldots, a_{|A|}$. При создании данного алгоритма авторами было сделано допущение о том, что было бы желательно иметь равные вероятности появления символов алфавита A [58]. Для реализации этой идеи предлагается для нормализованного временного ряда найти упорядоченное множество таких точек $B = \beta_0, \beta_1, \beta_2, \ldots, \beta_{|A|-1}, \beta_{|A|}$ $(\beta_0 = -\infty, \beta_{|A|} = +\infty)$, которые делили бы область под графиком стандартной нормальной (гауссовой) кривой $N(0, 1)$ на равные площади, равные $1/|A|$. Символьное представление для временного ряда TS получается далее по следующему правилу: если очередной элемент ts_i меньше β_1, то он отображается в первый символ алфавита A, если элемент ts_i больше $\beta_{|A|-1}$, то он отображается в последний символ алфавита A. Если же элемент ts_i попадает в интервал (β_k, β_{k+1}), т. е. $\beta_k \leq ts_i \leq \beta k + 1, k = 1 \ldots |A| - 2$, то он отображается в символ алфавита, соответствующий данному интервалу. Указанные интервалы и соответствующие им символы алфавита (для примера выбран алфавит из 10 символов) представлены на рис. 2.4. Соответствующие значения коэффициентов $\beta_i, i = 0, ..,10$ приведены в табл. 2.2. Символьное представление, соответствующее исходному временному ряду, приведено в табл. 2.3 (четвертая колонка).

Таблица 2.2 — Алфавит из 10 символов: значения $\beta_i, i = 0, ..,10$

β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
$-\infty$	-1.28	-0.84	-0.52	-0.25	0.0	0.25	0.52	0.84	1.28	∞

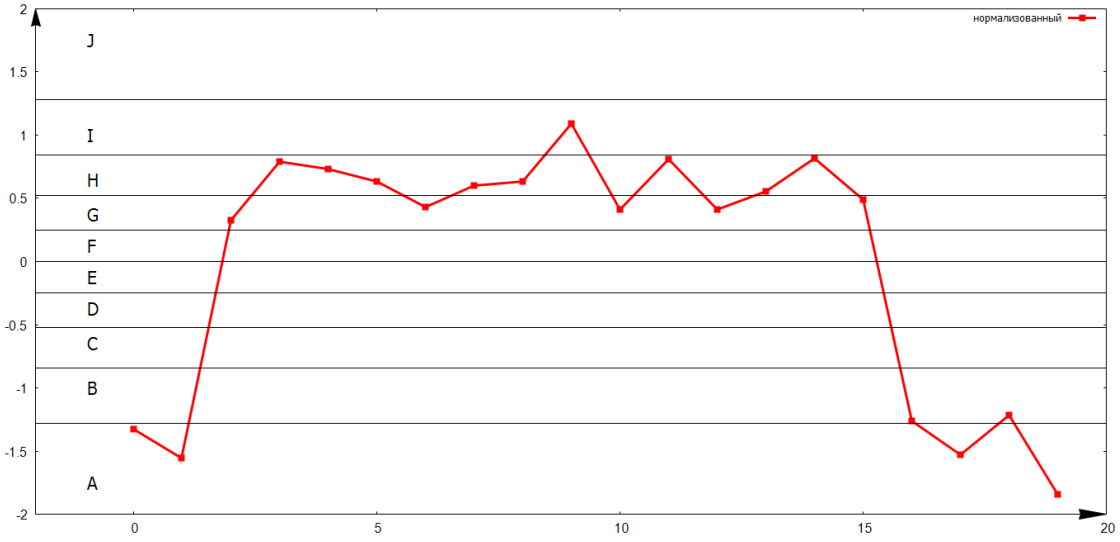


Рисунок 2.4 — Соответствие символов

Таблица 2.3 — Различные представления временного ряда

(Время)	Исходное	Нормализованное	Символьное (SAX)
0	-0.2955	-1.32813	A
1	-1.0959	-1.55709	A
2	5.4844	0.325309	G
3	7.1003	0.787564	H
4	6.9060	0.731981	H
5	6.5670	0.635005	H
6	5.8532	0.43081	G
7	6.4491	0.601278	H
8	6.5588	0.632659	H
9	8.1564	1.08968	I
10	5.7925	0.413446	G
11	7.1705	0.807646	H
12	5.7883	0.412245	G
13	6.2838	0.553991	H
14	7.2119	0.819489	H
15	6.0642	0.491171	G
16	-0.0503	-1.25798	B
17	-0.9934	-1.52777	A
18	0.0919	-1.2173	B
19	-2.0988	-1.84399	A

В процессе преобразования временного ряда из нормализованного представления в символьное (SAX) мы получаем таблицу расстояний *dist* между символами. Данные расстояния потребуются далее при выполнении операций над временными рядами в символьном представлении. Пример для алфавита из 10 символов приведен в табл. 2.4. Значения в таблице вычисляются по следующим формулам:

$$dist[i,j] = \begin{cases} 0, |i-j| \leq 1 \\ \beta_{max(i,j)-1} - \beta_{min(i,j)}, |i-j| > 1 \end{cases}$$

Соответствующие значения коэффициентов $\beta_i, i = 0,..,10$ приведены в табл. 2.2.

Одним из важных свойств алгоритма SAX является возможность уменьшения размерности [58] для временного ряда. Пусть имеется исходный временной ряд длины N. Его можно рассматривать как вектор в N-мерном пространстве. При нормализации данного временного ряда его можно преобразовать в ряд дли-

Таблица 2.4 — Таблица расстояний между символами для алфавита из 10 символов

–	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
A	0	0	0.44	0.76	1.03	1.28	1.53	1.8	2.12	2.56
B	0	0	0	0.32	0.59	0.84	1.09	1.36	1.68	2.12
C	0.44	0	0	0	0.27	0.52	0.77	1.04	1.36	1.8
D	0.76	0.32	0	0	0	0.25	0.5	0.77	1.09	1.53
E	1.03	0.59	0.27	0	0	0	0.25	0.52	0.84	1.28
F	1.28	0.84	0.52	0.25	0	0	0	0.27	0.59	1.03
G	1.53	1.09	0.77	0.5	0.25	0	0	0	0.32	0.76
H	1.8	1.36	1.04	0.77	0.52	0.27	0	0	0	0.44
I	2.12	1.68	1.36	1.09	0.84	0.59	0.32	0	0	0
J	2.56	2.12	1.8	1.53	1.28	1.03	0.76	0.44	0	0

ны $M, M < N$, и рассматривать его как вектор уже в M-мерном пространстве, то есть в пространстве меньшей размерности.

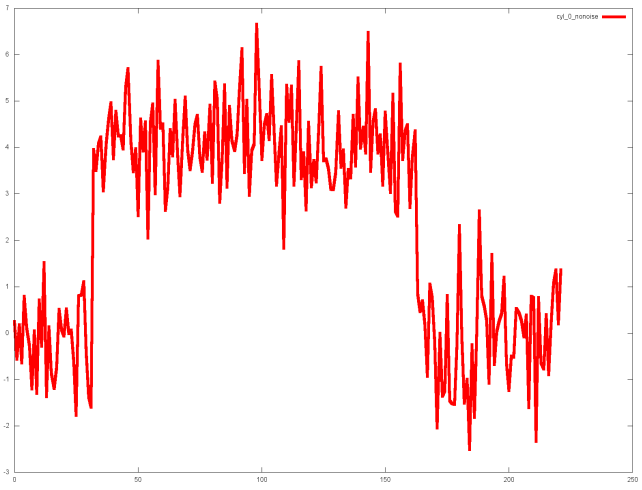


Рисунок 2.5 — Исходный временной ряд (222 точки)

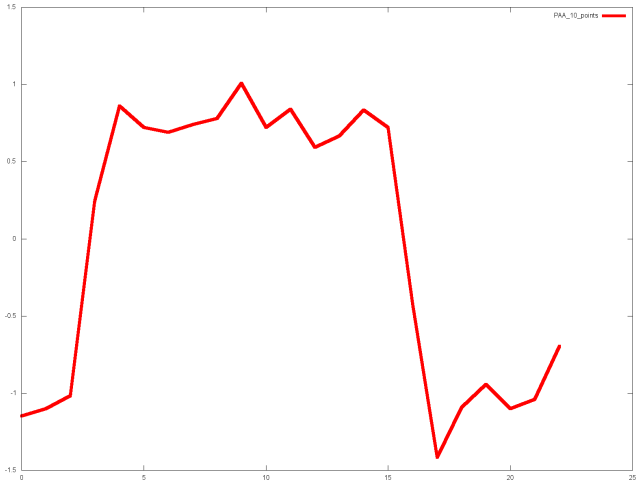


Рисунок 2.6 — Преобразованный временной ряд (меньшая размерность, 23 точки)

Пример исходного временного ряда приведен на рис. 2.5 (222 точки), пример преобразованного в ряд меньшей размерности временного ряда приведен на рис. 2.6 (для представления ряда меньшей размерности использованы 23 точки), примерное соответствие между этими двумя рядами приведено на рис. 2.7.

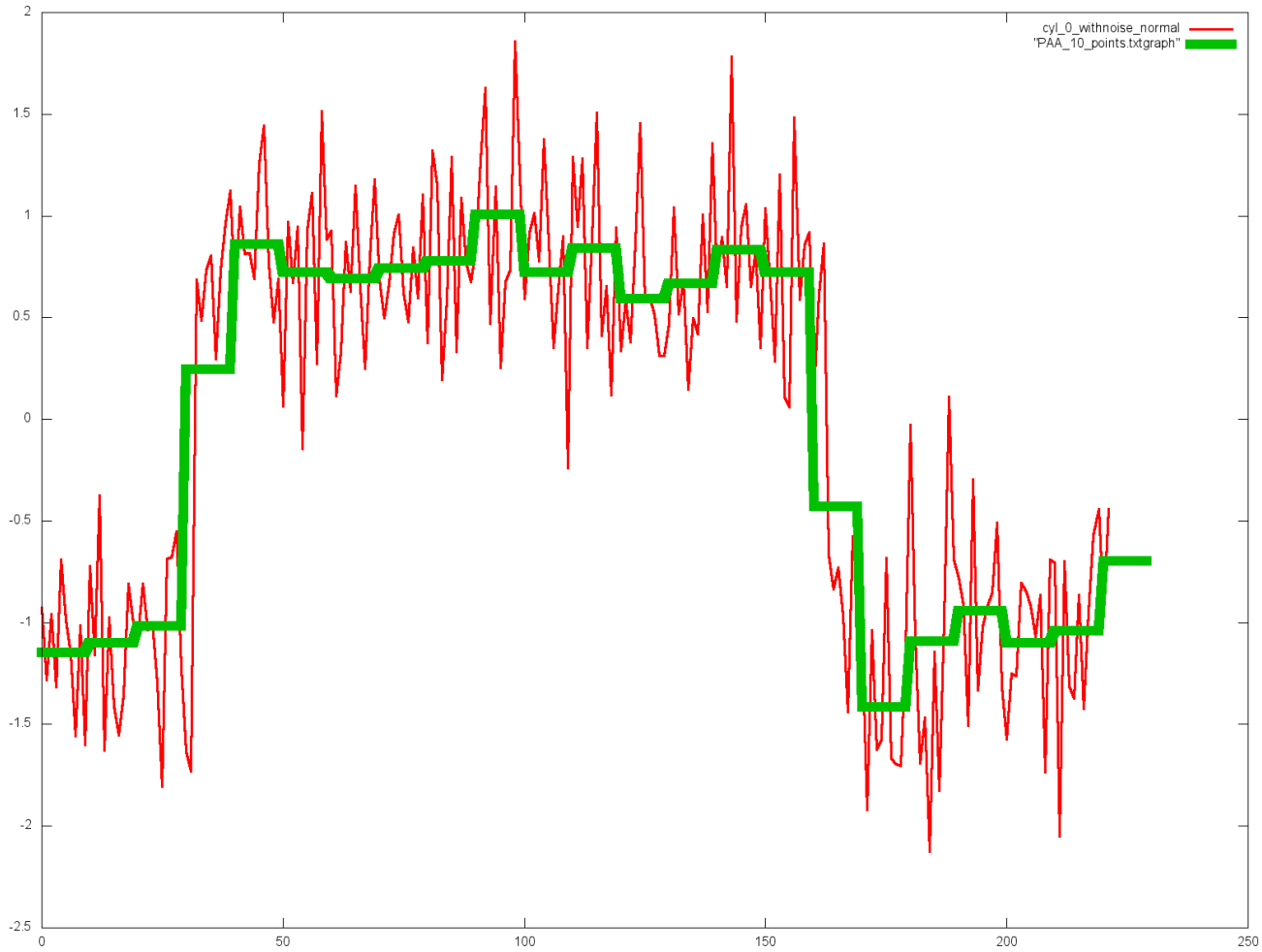


Рисунок 2.7 — Примерное соответствие между исходным и преобразованным рядами

2.1.2 Ограничения задачи

В работе не ставилась задача изучения природы исходных данных, способов передачи и характеристик сигналов, а также прочих понятий, затрагивающих область цифровой обработки сигналов. В нашем распоряжении имелись уже сформированные наборы данных, состоящие из динамических объектов обобщения (временные ряды или наборы временных рядов), которые априори были отнесены к определенным классам. При этом сами временные ряды, составляющие динамические объекты обобщения, имеют конечную длину – разную для различных наборов данных и, возможно, разную и в пределах каждого отдельного набора данных.

Наборы данных, большинство из которых уже разделенно на обучающую и экзаменационные выборки, используются для проверки того, насколько успешно методы обобщения знаний, обзор которых приведен в работе, а также предло-

женные в самой работе, позволяют на основании обучающей выборки построить модель, которая смогла бы относить объекты экзаменационной выборки условно к «нормальным» или «аномальным» (задача обнаружения аномалий) или к различным классам, определенным в каждом наборе данных (задача классификации, задача диагностики). Наличие же открытых и общедоступных репозиториев *UCR Time Series Data Mining Archive* [63; 64], *UC Irvine Repository* [65] позволяет сравнить эффективность предложенных алгоритмов с существующими.

2.2 Описание ситуаций с помощью временных рядов

Рассмотрим следующий пример. Пусть на сложном техническом объекте используется некоторый датчик, отображающий состояние какого-то произвольного параметра – $q = 1$ в данной выше постановке задачи обобщения понятий. Зададим некоторое $r = t^*$ – максимальный интервал времени, на котором будем рассматривать ситуацию – такой промежуток времени соответствует максимальной длине временного ряда. Тогда временной ряд, приведенный в табл. 2.1 является одновременно и динамическим объектом, или ситуацией на объекте. Рассмотрим случай, когда контроль за состоянием некоторой системы осуществляется на основании показаний единственного датчика. Пример набора динамических объектов (ситуаций), которые могут быть использованы как исходные данные для задачи обобщения, приведен в табл. 2.5: здесь временной интервал $r = 10$, каждая строка таблицы – это набор числовых значений контролируемого параметра, поступающих с единственного датчика в течение указанного времени. Тогда каждая строка представляет собой одну ситуацию (Сит1-Сит9), при этом каждая ситуация относится к некоторому классу («КС» – класс ситуации) $NORM$ – обозначает ситуации, которые соответствуют нормальному состоянию объекта.

Будем использовать описания динамичеких объектов вида (табл. 1.2), пример которых приведён в табл. 2.5, для построения некоторой модели, с помощью которой можно будет далее относить возникающие на объекте ситуации к «нормальным» или «анмомальным» (неисправностям). При этом обычно существует два варианта: в таблице наблюдений могут быть приведены или только «нормальные» ситуации, или наоборот – только «аномальные». Обычно в случае, если таблица содержит «нормальные» ситуации, решаемая задача относится

Таблица 2.5 — Набор динамических объектов (ситуаций) для случая 1 параметра

t	0	1	2	3	4	5	6	7	8	9	КС
Сит1	-1.07	-0.13	0.85	0.96	0.81	0.84	-0.08	-1.01	-0.90	-1.13	NORM
Сит2	-0.72	-0.70	1.25	1.23	1.27	0.03	-0.76	-0.71	-0.71	-0.74	NORM
Сит3	-0.94	-0.84	1.06	0.97	1.01	1.04	-0.35	-0.92	-0.83	-0.80	NORM
Сит4	-0.56	-0.62	-0.19	0.64	1.45	1.39	-0.69	-0.61	-0.66	-0.62	NORM
Сит5	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99	NORM
Сит6	-0.54	-0.44	-0.28	0.75	1.61	0.40	-0.45	-0.53	-0.38	-0.61	NORM
Сит7	-0.45	1.05	1.25	0.61	-0.35	-0.50	-0.39	-0.27	-0.89	-0.28	NORM
Сит8	-0.68	-0.67	1.63	1.07	0.69	0.01	-0.59	-0.70	-0.64	-0.53	NORM
Сит9	-1.01	0.50	1.35	0.89	0.33	0.18	-0.34	-0.75	-0.98	-0.65	NORM

также и к классу задач, называемых обнаружением аномалий. В другом случае - если в таблице содержатся «аномальные» ситуации (соответствующие неисправностям), задачу часто относят к классу задач диагностики. Рассмотрим более подробно задачу обнаружения аномалий в наборах временных рядов.

2.3 Задача обнаружения аномалий

Для задачи обнаружения аномалий обычно имеется описание нормальной работы системы – например, набор состояний системы, при которых неполадки остутствуют. Описание же ситуаций, соответствующих неполадкам на объекте, часто отсутствует либо является неполным. При обучении на таких данных требуется построить модель нормальной работы системы, которая в дальнейшем могла бы предказывать, является ли текущая ситуация на объекте «нормальной» или «аномальной», то есть присутствуют ли в данный момент какие-либо неисправности или нет.

Задача определения, или обнаружения, аномалий [66] ставится как задача поиска в наборах данных образцов, не удовлетворяющих некоторому предполагаемому типовому поведению. Возможность найти аномалии в некотором наборе данных важна в различных предметных областях — при анализе работы сложных технических систем (например, телеметрии спутников), анализе сетевого трафика, в медицине (анализ снимков магнитно-резонансной томографии, электрокардиограмм), в банковском деле (анализ транзакций, производимых с помощью кредитных карт) и др.

Аномалия, или «выброс», определяется как элемент, который явно выделяется из набора данных, к которому он принадлежит, и существенно отличается от других элементов выборки. Неформально задача определения аномалий в наборах временных рядов ставится следующим образом. Пусть имеется коллекция временных рядов, описывающих некоторые процессы. Эта коллекция используется для описания нормального протекания процессов. Требуется на основании имеющихся данных построить модель, которая является обобщенным описанием нормальных процессов и позволяет различать нормальные и аномальные процессы.

Задача усложняется, тем, что набор исходных данных ограничен и не содержит примеров аномальных процессов; также не задан критерий, по которому можно было бы различать «нормальные» и «аномальные» временные ряды. В связи с этим трудно точно оценить качество работы алгоритма (процент правильно определенных аномалий, число ложных срабатываний и число пропущенных аномалий). К тому же многие алгоритмы, хорошо показавшие себя на одних наборах данных, совершенно не подходят для других предметных областей. Также может отличаться и критерий, на основании которого определяется «нормальность» рядов.

Важным при решении такой задачи является то, что обнаружение аномалий позволяет получить информацию, требующую принятия мер: аномальный шаблон поведения в компьютерной сети, выявленый при анализе трафика, может говорить о том, что к компьютеру был получен несанкционированный доступ и он может рассылать данные посторонним лицам; аномалия на снимке, полученном при проведении магнитно-резонансной томографии, может свидетельствовать о наличии злокачественной опухоли; аномалии, выявленные при анализе действий с кредитными картами, могут быть показателем того, что карта или персональные данные пользователя были скомпрометированы; аномалии в показаниях датчика могут свидетельствовать о неисправности устройства.

Первые работы, связанные с обнаружением аномалий статистическими методами, были опубликованы еще в XIX веке. Со временем появились новые методы обнаружения аномалий, разработанные разными научными сообществами: некоторые методы разрабатывались для конкретных предметных областей, некоторые были достаточно общими.

Аномалии в наборах данных могут быть вызваны различными причинами (например, действиями злоумышленников) но их объединяет то, что они представляют *интерес* для эксперта-аналитика.

Очевидный подход к решению задачи обнаружения аномалий следующий: необходимо определить область, соответствующую нормальному поведению. Тогда любое наблюдение, лежащее в этой области, будет считаться нормальным, а вне области - аномальным. Тем не менее, даже при таком простом подходе возникает много трудностей:

- определить область, которая охватывает все возможные варианты нормального поведения непросто; кроме того, граница между нормальными и аномальными наблюдениями очень часто размыта: аномальное наблюдение, лежащее близко к границе области, на самом деле может быть нормальным, и наоборот;
- в случае, если аномалии являются результатом злонамеренных действий, злоумышленники обычно стремятся замаскировать свои действия таким образом, чтобы аномалии выглядели нормальным поведением, таким образом, делая задачу определения области нормального поведения еще более сложной;
- во многих предметных областях область нормального поведения изменяется со временем – следовательно, текущее описание нормального поведения может стать недостаточно представительным в будущем;
- точное определение термина «аномалия» различается в зависимости от выбранной предметной области. Например, в медицине небольшие отклонения от нормального состояния (например, температура тела) могут считаться аномалиями, в то время как небольшие колебания цен на бирже могут считаться нормальными. Таким образом, достаточно часто нельзя напрямую применять методы, разработанные для конкретной предетной области, в других предметных областях;
- большой проблемой является доступность данных, которые используются для обучения и проверки моделей, помеченных как нормальные и аномальные;
- часто данные содержат шум, за счет которого некоторые наблюдения становятся похожими на аномальные, при этом не являясь таковыми.

2.3.1 Природа исходных данных

Ключевой вопрос любого метода обнаружения аномалий - *природа* исходных данных. Входными данными обычно является коллекция экземпляров данных (которые могут называться объектами, записями, точками, шаблонами, образцами, событиями, наблюдениями, сущностями и т. п.). Каждый объект может описываться набором атрибутов (переменных, характеристик, полей). Атрибуты могут быть различных типов - бинарные, категорийные, непрерывные. Каждый объект может быть описан одним атрибутом (одномерный) или несколькими (многомерный).

Применимость методов обнаружения аномалий определяется характером атрибутов. Например, для статистических методов при работе с непрерывными и категорийными данными должны использоваться различные статистические модели. Аналогично, для методов, основанных на методе ближайших соседей, характер атрибутов будет определять метрику.

За счет описанных выше проблем задача обнаружения аномалий в общей постановке является достаточно трудной. На самом деле все методы обнаружения аномалий решают частные задачи: на постановку задачи влияет природа исходных данных, доступность/наличие данных, помеченных как нормальные и аномальные, тип аномалий, которые необходимо обнаружить, и т. п. Часто эти факторы определяются предметной областью: используются понятия из статистики, машинного обучения, интеллектуального анализа данных, теории информации, теории обработки сигналов – и применяются к определенным постановкам задачи.

В общем случае экземпляры данных, используемых в задачах обнаружения аномалий, могут быть связаны между собой: например, последовательности, пространственные данные, графовые данные. В последовательностях экземпляры данных линейно упорядочены - это временные ряды, геномы, протеиновые последовательности. В пространственных данных каждый экземпляр данных связан с соседними - например, данные о городском трафике, экологии. Если в пространственных данных есть временной компонент, то говорят о пространственно-временных данных (напр., климат). В графовых данных экземпляры данных представляют собой вершины в графе, которые связаны с другими вершинами дугами.

Важным аспектом в методах обнаружения аномалий является природа рассмативаемых аномалий. Аномалии могут быть разделены на 3 категории:

- точечные аномалии: если отдельный объект может считаться аномалией по отношению к остальному набору данных, то он считается точечной аномалией. Это самый простой тип аномалий и предмет большинства исследований;
- контекстные аномалии: если объект является аномалией в каком-то контексте (но не иначе), то он считается контекстной, или условной, аномалией. Контекст определяется структурой данных и формулируется при постановке задачи. Примером такой аномалии может быть температура: зимой температура -10 градусов вполне обычное явление, в то время как летом такая температура аномальна;
- групповые аномалии: если некоторый набор объектов является аномалией по отношению ко всему набору данных, то он считается аномалией; отдельные элементы в этом наборе сами по себе могут и не быть аномалиями, но вместе они образуют то, что называется коллективной аномалией. Например события «переполнение буфера» и «копирование файлов по протоколу ftp» могут быть вполне обычными событиями в некоторой компьютерной системе, однако их совместное последовательное появление может свидетельствовать об удаленной атаке на компьютерную систему, в ходе которой данные копируются на удаленную машину по протоколу ftp.

2.3.2 Обучающие выборки для задачи обнаружения аномалий

Обычно для данных указано, относятся ли они к нормальным или аномальным, но при этом получить репрезентативную выборку, которая будет достаточно точной и при этом описывающей все возможные варианты поведения, чрезвычайно трудно. Часто объекты, представленные в выборке, относит к нормальным или аномальным эксперт, при этом следует отметить, что получить выборку для нормального поведения объекта или системы проще, чем для аномального, так как в некоторых случаях аномальное поведение встречается крайне редко и приводит к катастрофическим последствиям (например, безопасность на транспорте). Более того, аномальное поведение динамично по своей природе, а зна-

чит, могут появляться новые типы аномалий, которые не были представлены в исходной выборке.

В зависимости от наличия или отсутствия меток данных выделяют три категории методов обнаружения аномалий:

– обнаружение аномалий «с учителем» (методы управлямого обнаружения аномалий): для методов, относящихся к данной категории, требуется наличие в обучающей выборке объектов, относящихся как к нормальным, так и к аномальным. На основании таких данных строится модель, которая сможет определять класс объектов, поступающих к ней на вход. Для успешного функционирования подобных моделей необходима репрезентативная выборка и точное отнесение исходных объектов к правильным классам. Кроме того, данных, относящихся к «нормальным», обычно гораздо больше чем «аномальных», и за счет этого создается некий дисбаланс, который может повлиять на способность модели точно относить объекты к правильным классам.
– обнаружение аномалий «без учителя»: для данной категории методов предполагается, что данные для обучения не требуются. Но при этом делается предположение о том, что «нормальные» объекты встречаются горазо более часто, чем «аномальные». В противном случае данные методы страдают от большого числа ложных срабатываний.
– обнаружение аномалий при частичном обучении с «учителем» – нечто среднее между первыми двумя: предполагается, что в обучающей выборке есть только примеры «нормальных» объектов. Соответствующие методы строят модель, описывающую нормальное поведение системы, и используют её для обнаружения аномалий в тестовых данных. Также существует небольшое количество методов, для которых исходными данными является набор «аномалий». Данные методы не нашли широкого применения, так как очень трудно получить исчерпывающий набор данных, описывающий аномальное поведение.

2.3.3 Представление результатов для методов обнаружения аномалий

При использовании методов обнаружения аномалий результаты обычно представляются в следующем виде:

– коэффициенты - метод относит объект к нормальным или аномальным с некоторой степенью уверенности; в конечном итоге будет получен список аномалий, ранжированный по степени уверенности, после чего эксперт может задать некоторый порог, чтобы отсечь из списка объекты, скорее всего, не являющиеся аномалиями;
– метки - метод относит объект к одному из нормальных или аномальных классов.

2.3.4 Области применения методов обнаружения аномалий

Методы обнаружения аномалий используются в следующих областях:
– системы обнаружения вторжений [67]. Под *вторжением* понимаются факты неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими (в основном, через Интернет). Системы обнаружения вторжений – это программные и/или аппаратные средства, которые предназначены для выявления подобных фактов и используются для обнаружения некоторых типов вредоносной активности, которая может нарушить безопасность компьютерной системы. К такой активности относятся сетевые атаки против уязвимых сервисов, атаки, направленные на повышение привилегий, неавторизованный доступ к важным файлам, а также действия вредоносного программного обеспечения (компьютерных вирусов, троянов и червей). Данная предметная область характеризуется огромными объемами данных, соответственно, методы обнраужения аномалий должны иметь низкую вычислительную сложность и в большинстве случаев обрабатывать поступающие данные «на лету»;
– фрод (от англ. **fraud** – мошенничество, афера, подделка): вид мошенничества в области информационных технологий, в частности, несанкционированные действия и неправомочное пользование ресурсами и услугами в сетях связи, мошеннические действия с банковскими дебетовыми и кредитными картами;
– медицина и здоровье: методы обнаружения аномалий в данной области работают с данными о пациентах, которые обычно включают в себя рост, возраст, вес, группу крови и другие данные, при этом имеется большое число данных о нормальном состоянии пациентов, а следовательно, ча-

сто применяются методы частичного обучения с учителем. Кроме того, данные могут иметь привязку к географии (пространственные) или ко времени (темпоральные) – сюда же относится и анализ электрокардиограмм, эхоэнцефалограмм;

- обнаружение неисправностей в сложных технических объектах, системах: промышленные объекты, из-за их постоянной работы, имеют свойство изнашиваться и приходить в негодность, в связи с этим необходимо вовремя обнаруживать возникающие неисправности. Данная категория, в свою очередь, делится на следующие: обнаружение неисправностей в механических компонентах и обнаружение дефектов в физических структурах;
- обработка изображений: данная предметная область включает в себя обработку снимков со спутников, распознавание образов, спектроскопию, анализ маммологических снимков, видеонаблюдение. Аномалии могут быть вызваны движением, наличием посторонних объектов, ошибками приборов, а объекты могут содержать пространственные и временн**ы**е данные;
- обнаружение аномалий в текстовых данных: в данной предметной области основные задачи - обнаружение новых тем, событий, историй в больших массивах документов или статей. Аномалии могут быть вызваны появлением нового интересного события или темы;
- сенсорные сети: в последнее время всё большее внимание уделяется изучению беспроводных сенсорных сетей, это связано с тем, что данные, полученные с датчиков, входящих в сеть, обладают рядом уникальных характеристик. Обнаружение аномалий в таких данных может говорить о том, что один из датчиков неисправен или о том, что произошло некоторое событие, представляющее интерес для анализа;
- другие области включают в себя более специфические задачи, такие как распознавание речи, анализ поведения робота, анализ дорожного трафика, анализ астрономических данных, определение ошибок в веб-приложениях и т. п.

2.3.5 Обзор и классификация методов обнаружения аномалий

Методы обнаружения аномалий разделяются на следующие широкие категории. Способы обнаружения аномалий, **основанные на методе ближайшего соседа**, используют следующее предположение: нормальные экземпляры объектов расположены в тесном соседстве, в то время как аномалии находятся на значительном расстоянии от своих ближайших соседей. Для использования данной категории методов необходимо, чтобы была задана метрика или функция, определяющая расстояние между объектами. Расстояние или мера сходства могут вычисляться различными способами – например, для непрерывных атрибутов обычно используется евклидово расстояние [68]; для дискретных атрибутов используется коэффициент сходства или другие, более сложные меры расстояния [69]; для данных со множеством атрибутов вычисляется расстояние между каждыми из них, а полученные результаты каким-то образом объединяют.

Выделяют две группы методов:

– использующие расстояние до k-ого ближайшего соседа;

– использующие относительную плотность для каждого объекта.

К достоинствам таких методов относят то, что данные не должны быть изначально отнесены экспертом к каким-либо классам – это методы, управляемые данными, и никаких априорных предположений о природе и свойствах данных не делается. Тем не менее, даже небольшое участие эксперта в обучении позволяет повысить качество определения аномалий. Также методы, относящиеся к данному классу, легко адаптировать для использования с другими типами данных (в других предметных областях), так как для это часто требуется всего лишь определить новую метрику.

К недостаткам подобных методов относят неудовлетворительную их работу в случае, когда у нормальных объектов слишком мало соседей или наоборот – у аномальных экземпляров соседей слишком много, что приводит к большому числу ошибок первого и второго рода (соответственно ложных срабатываний и пропусков событий). Также способы, основанные на методе ближайшего соседа, требуют значительного количества вычислений, так как при отнесении объекта к нормальным или аномальным требуется вычислить расстояние до всех объектов из обучающей выборки. Кроме того, выбор метрики может значительно повлиять на качество распознавания, а выбор метрики для сложных объектов может вызвать затруднение.

Методы поиска аномалий, основанные на кластеризации. Кластеризация [70] используется для разбиения заданной выборки объектов на подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов. При использовании данного класса методов полагаются на одно из следующих предположений:

- нормальные объекты принадлежат кластеру, аномальные - нет;
- нормальные объекты лежат близко к центру кластера, аномальные - вдали от центра;
- нормальные объекты принадлежат большим, плотным кластерам, в то время как аномалии принадлежат небольшим и разреженным.

Методы, основанные на кластеризации, оценивают расстояние до объектов на основании информации о кластере, к которому принадлежат объекты, в то время как методика использования ближайших соседей пользуется локальным окружением каждого объекта.

Достоинствами данного класса методов являются возможность обучения без учителя, адаптация методов к различным типам данных и небольшое число вычислений при отнесении объектов к нормальным или аномальным (так как число кластеров обычно незначительно). К недостаткам следует отнести сильную зависимость от выбранного алгоритма кластеризации и специфики его работы (метод кластеризации не оптимизирован для задачи обнаружения аномалий, аномалии – лишь побочный результат от работы алгоритма кластеризации и т. п.).

Основной принцип **статистических методов обнаружения аномалий** следующий: элементы выборки распределены по некоторому закону, а аномалия – это наблюдение, которое явно выделяется из набора данных, к которому оно принадлежит, существенно отличается от других элементов выборки так как, скорее всего, было получено по некоторому другому закону. Соответственно, и предположение, которым оперируют статистические методы обнаружения аномалий, состоит в том, что нормальные наблюдения попадают в районы стохастической модели с высокой вероятностью, а аномалии – в районы с низкой вероятностью. Статистические методы применяют статистическую модель к исходным данным (обычно – определяющим нормальное поведение) и пользуются статистическим выводом для того чтобы определить, является наблюдение аномалией или нет. Те наблюдения, которые имеют низкую вероятность быть полученными в рас-

сматриваемой модели, считаются аномалиями. Данное предположение и определяет как основные достоинства, так и основные недостатки статистических методов: если наблюдения дейтвительно распределены по некоторому закону и закон определен верно, то статистичекие методы дают хороший результат.

Теоретико-информационные методы обнаружения аномалий анализируют количество информации в данных, используя различные теоретико-информационные величины, такие как колмогоровская сложность [71], энтропия [72], относительная энтропия и т. п. При этом предполагается, что аномалии в данных приводят к неравномерности информационного содержания набора данных.

Спектральные методы обнаружения аномалий пытаются найти приближение данных с использованием набора атрибутов таким образом, чтобы отразить все разнообразие данных. Спектральные методы используют в предположении, что данные можно представить в пространстве меньшей размерности, причем в новом представлении различие между нормальными и аномальными данными будет значительным. Следовательно, основная задача - определить такое пространство, в котором можно было бы легко обнаружить аномалии [73].

Методы поиска аномалий, основанные на классификации. Классификация используется для обучения модели на данных, отнесенных к различным классам (этап обучения), и отнесения экземпляров данных к одному из имеющихся классов с использованием полученной модели (этап экзамена). Методы обнаружения аномалий, основанные на классификации, предполагают, что если классификатор, может быть обучен в имеющемся пространстве признаков, то он сможет разделить нормальные и аномальные объекты.

Среди методов обнаружения аномалий, основанных на классификации, выделяют методы, использующие:

– нейронные сети [74; 75];
– байесовские сети доверия [76; 77];
– метод опорных векторов [78];
– продукционные правила [79; 80].

В данной задаче принято выделять два случая [66]: первый случай – обучающее множество содержит примеры единственного класса; второй случай – обучающее множество содержит примеры нескольких классов. В первом случае важен сам факт принадлежности рассматриваемых объектов к классу из обучающего множества, здесь требуется каким-то образом определить «границу», в

соответствии с которой временной ряд принадлежит классу из обучающего множества (не является аномалией) или не принадлежит ему (является аномалией). Во втором случае дополнительно нужно определить принадлежность объекта к конкретному классу.

К преимуществам методов обнаружения аномалий, основанных на классификации, относится возможность использовать огромное количество методов и алгоритмов, разработанных в области машинного обучения - в особенности для случая, когда обучающее множество содержит примеры нескольких классов. Кроме того этап «экзамена» проходит быстро по сравнению с другими классами методов, так как используется предварительно построенная модель (классификатор).

2.4 Использованные в работе наборы данных

Моделирование процесса обнаружения аномалий было проведено на данных из репозиториев *UCR Time Series Data Mining Archive* [63; 64], *UC Irvine Repository* [65]. Также использовались данные, собранные с помощью специальных систем анализа трафика при передаче файлов по различным протоколам (набор данных «трафик»).

Набор данных *«трафик»*. «Трафик» – данные, полученные на основе анализа трафика при передаче файлов по протоколу ftp в различных условиях (в том числе при одновременной передаче нескольких файлов по нескольким протоколам).

Для получения данных был собран специальный стенд, на котором осуществлялась передача данных по сети между двумя компьютерами по различным протоколам в различных условиях. Фиксировалась только длина передаваемого пакета данных. Обратная передача, даже если и была, не фиксировалась.

Рассмотрим пример зафиксированного пакета данных

```
No. Time Source Destination Protocol Info
4339 23.071158 10.10.10.50 10.10.10.100 FTP-DATA FTP Data: 1448
   bytes
Frame 4339 (1514 bytes on wire, 1514 bytes captured)
Ethernet II, Src: 00:27:0e:2d:06:df (00:27:0e:2d:06:df), Dst:
   00:27:0e:2d:06:17 (00:27:0e:2d:06:17)
Internet Protocol, Src: 10.10.10.50 (10.10.10.50), Dst:
   10.10.10.100 (10.10.10.100)
```

```
Transmission Control Protocol, Src Port: 59022 (59022), Dst Port:
   9680 (9680), Seq: 288153, Ack: 1, Len: 1448
FTP Data
```

Здесь передача осуществляется от компьютера с IP-адресом 10.10.10.50 (порт 59022) к компьютеру с IP-адресом 10.10.10.100 (порт 9680). Номер пакета 4339, время принятия 23.071158 с от начала передачи, в пакете передаются данные (не служебная информация), длина пакета 1448 байт. Передача осуществлялась по протоколу FTP.

Исследовались следующие варианты передачи данных:

– передача по протоколу FTP (эталон);
– одновременная передача по протоколам FTP и ping (анализировался FTP-трафик);
– одновременная передача по протоколам FTP и UDP (анализировался FTP-трафик).

Имея информацию подобного вида о передаче данных по сети, необходимо определить, не является ли передача данных «подозрительной», что может свидетельствовать о возможной компрометации сетевой инфрастуктуры, наличии программных и/или аппаратных закладок.

В качестве тестовых данных, помимо прочих, использовались специальным образом сгенерированные временные ряды, имитирующие передачу данных.

2.4.1 Наборы данных из *UCR Time Series Data Mining Archive*

Набор данных *«цилиндр-колокол-воронка»* (*«cylinder-bell-funnell»*, *«CBF»*), как следует из названия, содержит три различных класса временных рядов, условно названных «цилиндр», «колокол», «воронка». Это известный набор данных, широко применяемый для проверки алгоритмов, работающих с временными рядами [53; 81; 82].

Временные ряды, относящиеся к классу «цилиндр», характеризуются наличием на графике плато, перед которым наблюдается резкий рост значения параметра, после – резкий спад. Классу «колокол» соответствует постепенный рост значения от момента времени, после чего наблюдается резкое падение значения. Для класса «воронка» характерен резкий скачок значения, после которо-

го наблюдается постепенный спад. Временные ряды – типичные представители данных классов приведены соответственно на рис 2.8-2.10.

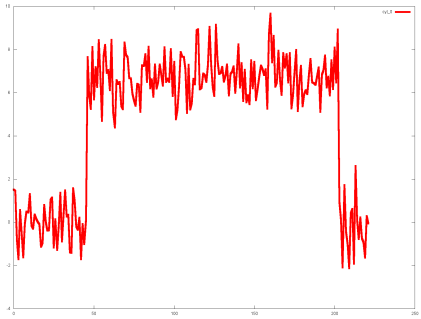


Рисунок 2.8 — «Цилиндр»

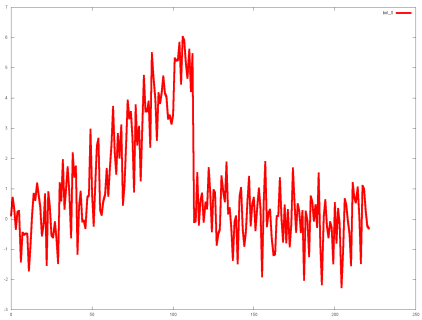


Рисунок 2.9 — «Колокол»

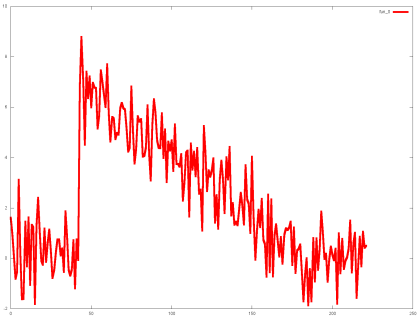


Рисунок 2.10 — «Воронка»

Набор данных *«контрольные карты»* [83–85] (*«control chart», «CC», «synthetic control»*) – искусственный набор данных, который содержит шесть различных классов, описывающих тренды, которые могут присутствовать в процессах: цикличность, уменьшение значения, резкое падение, увеличение значения, постоянная величина, резкое возрастание. Примеры рядов из данного набора приведены на рисунках 2.11-2.16.

Для приведённого набора данных, как и в случае набора «цилиндр-колокол-воронка», параметр, определяющий ход процесса, является абстрактной величиной. Это даёт возможность использоваться такие модели для очень широкого круга реальных задач, где наблюдаются подобные тренды.

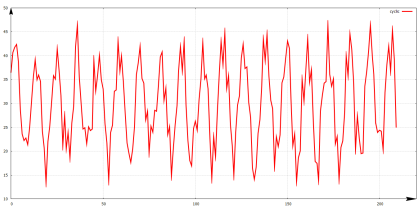


Рисунок 2.11 — «Цикличность»

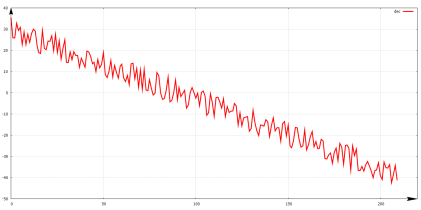


Рисунок 2.12 — «Уменьшение значения»

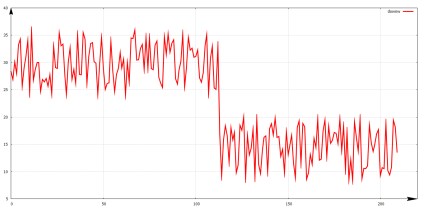


Рисунок 2.13 — «Резкий спад»

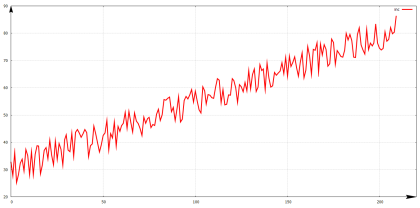


Рисунок 2.14 — «Увеличение значения»

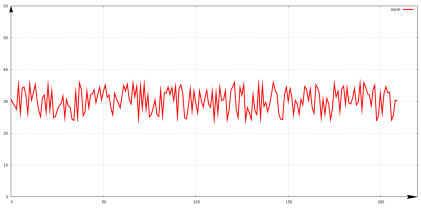


Рисунок 2.15 — «Нормальное значение»

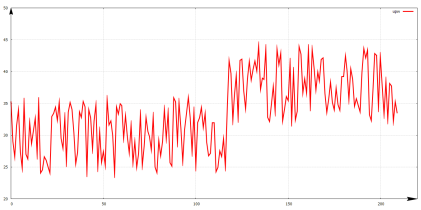


Рисунок 2.16 — «Резкое возрастание»

В **наборе данных *«wafer»*** [86] содержатся временные ряды, соответствующие показаниям датчиков при производстве полупроводниковых пластин.

Полупроводниковая пластина (англ. *wafer*) – полуфабрикат в технологическом процессе производства полупроводниковых приборов и микросхем. Представляет собой тонкую (250—1000 мкм) пластину из полупроводникового материала диаметром до 450 мм, на поверхности которой с помощью операций планарной технологии формируется массив дискретных полупроводниковых приборов или интегральных схем. После создания необходимой полупроводниковой структуры пластину разрезают на отдельные кристаллы (чипы).

Производство таких пластин (травление) – сложный технологический процесс, включающий в себя более 250 этапов обработки, на каждом из которых может произойти ухудшение характеристик или надежности, уменьшение выхода продукта или даже отбраковка, если параметры вышли за требуемые пределы. Наиболее критичными являются 6 параметров, среди которых экспертами выделены 2, которые по результатам экспериментов показали наиболее точные результаты по определению качественных и бракованных изделий: это *405 nanometer (nm) emission, 520 nanometer (nm) emission* – интенсивность излучения плазмы с длиной волны 405 нм и 520 нм во время изготовления полупроводниковых пластин. На рис. 2.17 и 2.18 представлены шесть временных рядов, анализ которых позволяет различать классы качественных и бракованных пластин.

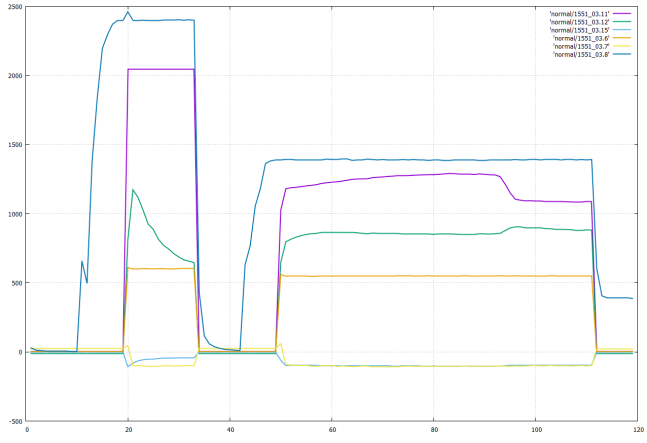


Рисунок 2.17 — «wafer» – нормальное протекание процесса

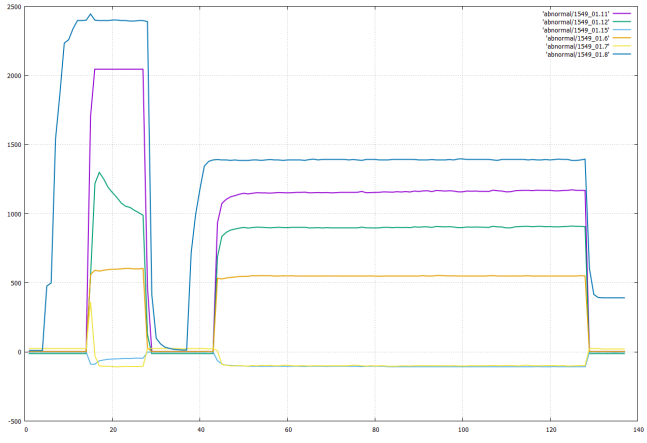


Рисунок 2.18 — «wafer» – ненормальное протекание процесса

В **наборе данных *«ECG»***. [86] (ЭКГ – электрокардиография) содержатся показания электрических сигналов кардиологической активности, записанных с электродов, прикрепленных в различных местах. При записи электрокардио-

граммы использовались два электрода, при этом каждый временной ряд соответствует записи сигнала с одного электрода в течение одного сердечного сокращения.

Наборы данных *«Beef»*, *«Coffee»*, *«Olive oil»* – спектрограммы продуктов [87].

Спектрографы для продуктов используются в хемометрике [1] для классификации типов продуктов – задачи, имеющей практическое применение при контроле качества и безопасности продуктов.

Спектрограммы для трех видов продуктов приведены на рисунке

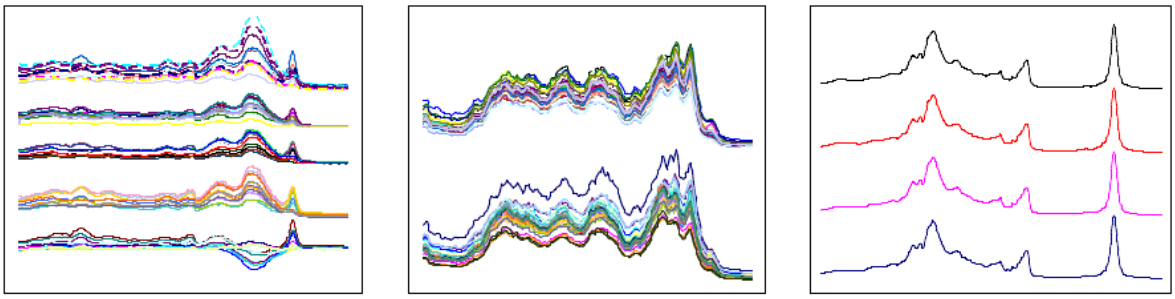
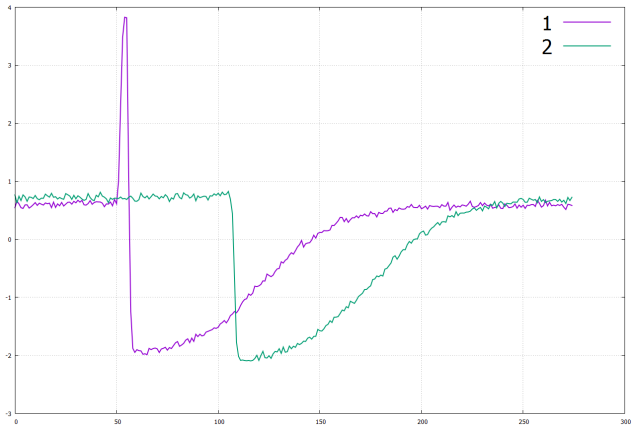


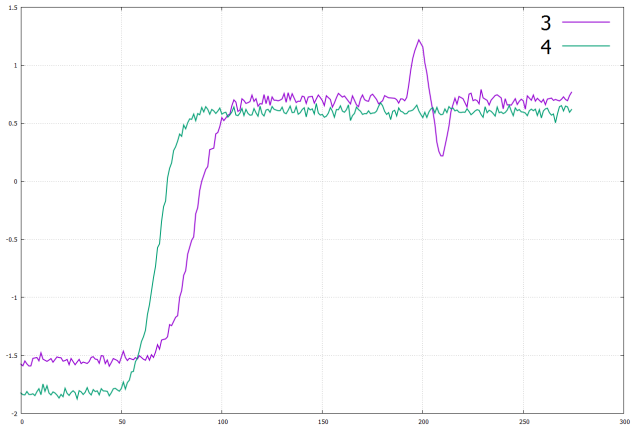
Рисунок 2.19 — Спектрограммы: мясо – кофе – оливковое масло

- **Beef** (мясо): набор данных содержит спектрограммы, соответствующие различной степени содержания побочных продуктов в мясе.
- **Coffee** (кофе): набор данных содержит спектрограммы, соответствующие двум классам (двум видам) кофе: арабика и робуста.
- **Olive oil** (оливковое масло): набор данных содержит спектрограммы оливкового масла экстракласса фильтрованного (extra virgin olive oil) из различных географических регионов.

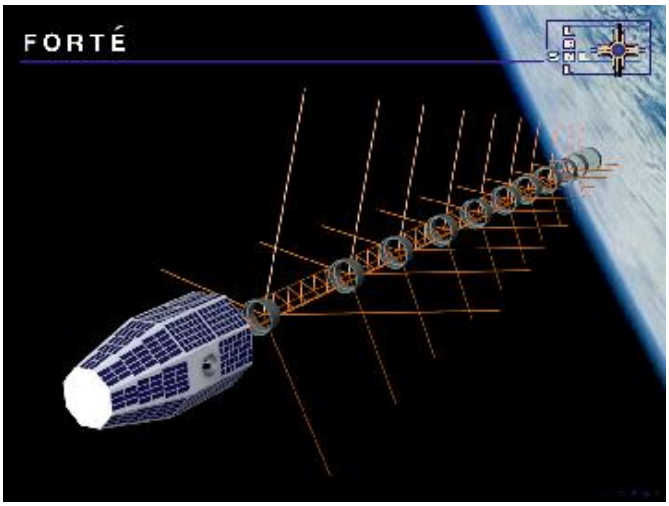
Набор данных *«Trace»* содержит временные ряды, соответствующие показаниям некоторых датчиков при определенных переходных процессах на атомной электростанции [88; 89].

[1] Хемометрика - раздел аналитической химии, ставящий целью получение химических данных с помощью математических методов обработки и добычи данных; химическая дисциплина, применяющая математические, статистические и другие методы, основанные на формальной логике, для построения или отбора оптимальных методов измерения и планов эксперимента, а также для извлечения наиболее важной информации при анализе экспериментальных данных





Набор данных *«Lightning 7»* [90].

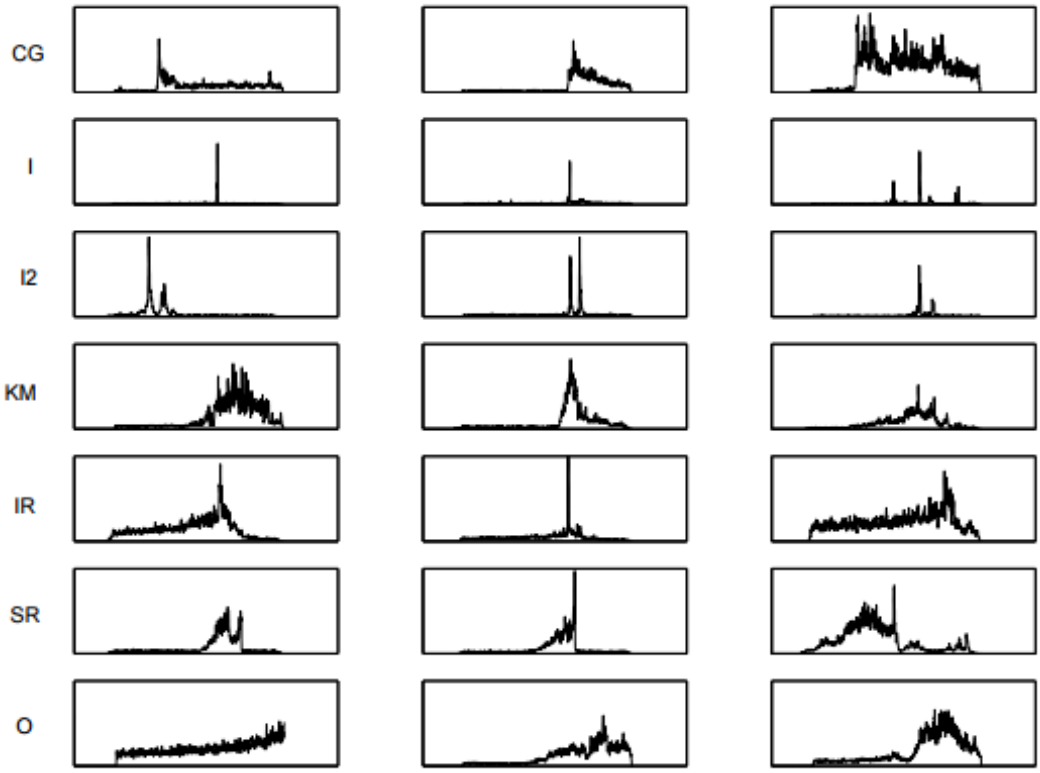


Fast On-orbit Rapid Recording of Transient Events (FORTE) – легкий спутник, запущенный в 1997 году на круговую низкую околоземную орбиту (800 км от Земли). Совместная разработка Сандийских национальных лабораторий (США) и Лос-Аламосской национальной лаборатории (США). Изначально планировалось использовать спутник для обнаружения ядерных взрывов, позднее – для изучения молний из космоса. Содержит оптические и радиочастотные датчики и «классификатор событий» для наблюдения в том числе и за высокочастотными излучениями молний в ионосфере на высоте от 80 до 966 км.

Примеры графиков – power density profile (плотность мощности энергии), соответствующая каждому классу (по три примера на класс).

Таблица 2.6 — Классы

Метка	Название
CG	Positive Initial Return Stroke
IR	Negative Initial Return Stroke
SR	Subsequent Negative Return Stroke
I	Impulsive Event
I2	Impulsive Event Pair
KM	Gradual Intra-Cloud Stroke
O	Off-record



Набор данных *«Lightning 2»* [90] аналогичен *«Lightning 7»*, только все молнии разделены на два класса: наземные – включают в себя классы *CG, IR, SR* – и внутриоблачные *I, I2, KM*.

2.4.2 Наборы данных из *UC Irvine Repository*

Набор данных *«Activities of Daily Living Recognition with Wrist-worn Accelerometer Data Set»*.

Акселерометр – прибор, измеряющий проекцию кажущегося ускорения (разности между истинным ускорением объекта и гравитационным ускорением).

Как правило, акселерометр представляет собой чувствительную массу, закреплённую в упругом подвесе. Отклонение массы от её первоначального положения при наличии кажущегося ускорения несёт информацию о величине этого ускорения.

Акселерометры реагируют на ускорение или силу, действующую на сенсорный элемент датчика. Ускорение, статическое или динамическое, возникает под действием силы, ускоряющей датчик, например, вследствие действия гравитации. Следовательно, акселерометры могут применяться для измерения силы, ускорения, вибрации, движения или перемещения, а также положения и угла наклона (инклинометры).

Акселерометры можно использовать в любом устройстве, работа которого связана с перемещением, наклоном, вибрацией.

По конструктивному исполнению акселерометры подразделяются на однокомпонентные, двухкомпонентные, трёхкомпонентные. Соответственно, они позволяют измерять ускорение вдоль одной, двух и трёх осей. Некоторые акселерометры также имеют встроенные системы сбора и обработки данных. Это позволяет создавать завершённые системы для измерения ускорения и вибрации со всеми необходимыми элементами.

В наборе данных «Activities of Daily Living Recognition with Wrist-worn Accelerometer Data Set» (ADL, набор данных «повседневная активность, записанная с помощью акселерометра») [65] представлены записи с помощью акселерометров выполнения некоторых простых действий, которые обозначены как «примитивы движения человека» (Human Motion Primitives, HMP), и перечислены ниже:

1. чистить зубы;
2. подниматься по ступенькам;
3. причесываться;
4. спускаться по ступенькам;
5. пить воду из стакана;
6. есть мясо (с вилкой и ножом);
7. есть суп (ложкой);
8. вставать с кровати;
9. ложиться в кровать;
10. наливать воду;

11. садиться на стул;
12. вставать со стула;
13. звонить по телефону;
14. ходить.

Спецификация акселерометра

– Тип: трёхосный акселерометр.
– Пределы измерений: [- 1.5g; + 1.5g].
– Чувствительность: 6 бит на ось.
– Частота обновления сигнала: 32 Гц.

Расположение - прикреплен к запястью, при этом:

– ось x : направлена вдоль руки (pointing toward the hand)
– ось y: направлена влево (pointing toward the left)
– ось z: перпендикулярна плоскости руки (perpendicular to the plane of the hand)

Ускорение кодируется по следующим правилам: [0; +63] = [-1.5g; +1.5g].

Правило преобразования оцифрованного сигнала в реальное значение ускорения следующее: $real\_val = -1.5g + (coded\_val/63) * 3g$.

Показания акселерометра, соответствующие примерам некоторых действий, приведены на рис. 2.20-2.23.

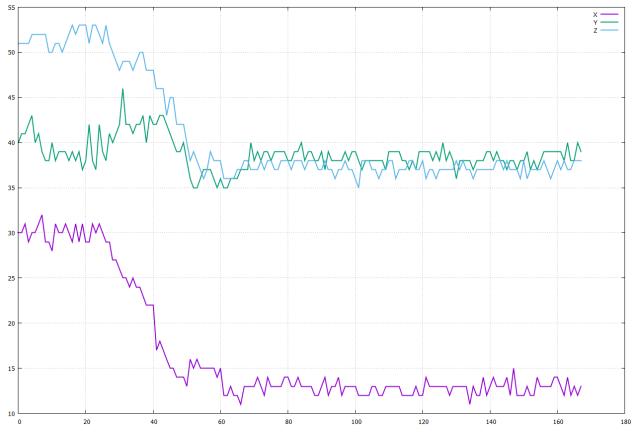


Рисунок 2.20 — Пример показаний акселерометра для действия «вставать со стула»

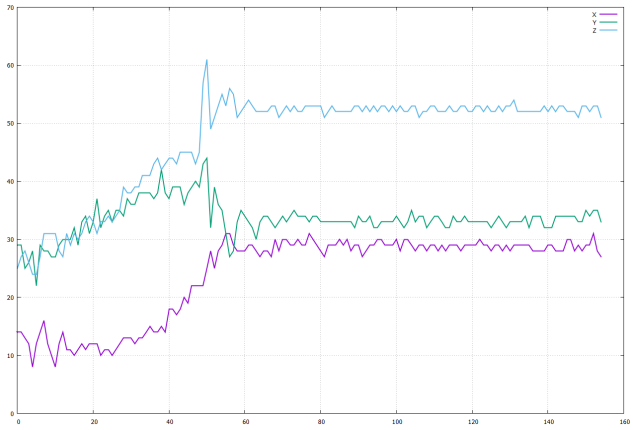


Рисунок 2.21 — Пример показаний акселерометра для действия «садиться на стул»

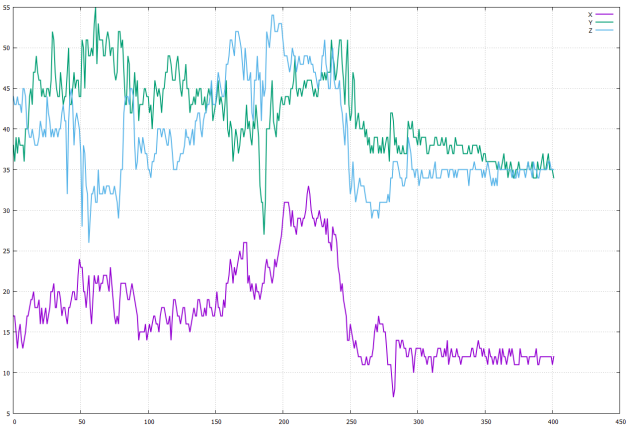


Рисунок 2.22 — Пример показаний акселерометра для действия «вставать с кровати»

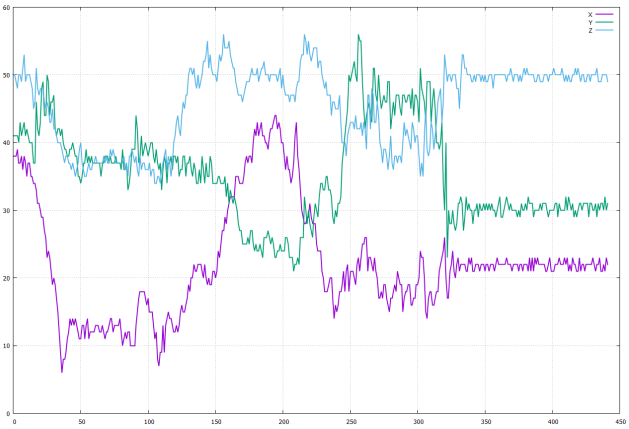


Рисунок 2.23 — Пример показаний акселерометра для действия «ложиться в кровать»

2.5 Модель шума в данных

При изучении вопросов поиска классифицирующих правил [38] неявно предполагалось, что такие правила существуют. В частности, предполагалось, что существуют детерминированные классифицирующие правила. Хотя такое предположение может быть верным для искусственно созданных обучающих множеств, используемых в машинном обучении, оно наверняка не выполняется применительно к реальным базам данных. Использование базы данных в качестве обучающего множества вызывает следующие трудности. Во-первых, информация в базе данных ограничена, так что не вся информация, необходимая для определения класса объекта, доступна. Во-вторых, доступная информация может быть повреждена или частично отсутствовать. Наконец, большой размер баз данных и их изменение со временем рождает дополнительные проблемы. Далее если база данных содержит всю информацию, необходимую для корректной классификации объектов, некоторые данные могут не соответствовать действительности. Например, значения каких-либо атрибутов могут содержать ошибки в результате измерений или субъективных суждений. Ошибка в значениях предсказываемых атрибутов приводит к тому, что некоторые объекты в обучающем множестве классифицированы неправильно. Несистематические ошибки такого рода обычно называются шумом.

Шум в обучающих примерах может вызываться различными причинами: во-первых, ошибками при описании объектов – это могут быть ошибки при описании предметной области: ошибки в измерениях, погрешности измеряю-

щих приборов, неправильное распределение примеров по классам экспертом и т. п. Второй причиной является то, что сам язык описания предметной области недостаточен для того, чтобы полностью и корректно описать все возможные ситуации. Такая ситуация обычно имеет место в медицине – например, когда по определенному набору симптомов можно поставить несколько диагнозов и для точного диагностирования требуются дальнейшие наблюдения и большее количество информации. Третья причина – изменение информации со временем: данные могут теряться и искажаться при хранении и пересылке, меняться со временем.

В работах [5; 6; 13] проводилось исследование влияния шума на работу алгоритмов обобщения понятий при наличии шума во входных данных. При этом одним из основных параметров исследования являлся *уровень шума* – величина $p_0, 0 < p_0 < 0.5$, которая показывает, что с вероятностью p_0 значение признака в обучающем или экзаменационном множестве искажено. Также эта величина показывает, что среди всех N значений признаков в среднем $N * p_0$ значений признаков будет искажено.

Для случая признакового описания объектов данная величина была достаточно информативной для оценки степени влияния шума на имеющиеся в распоряжении данные. Однако когда объекты представлены временными рядами или *наборами* временных рядов, такая оценка неприменима.

2.5.1 Набор данных *«цилиндр-колокол-воронка»*

Помимо наборов данных из самого *UC Irvine Repository* [65], временные ряды для данного набора можно получить искусственно по следующим формулам [91]:

1. «цилиндр»: $c(t) = (6 + \zeta) * \chi_{[a,b]}(t) + \epsilon(t), 1 \leq t \leq M$,
2. «колокол»: $b(t) = (6 + \zeta) * \chi_{[a,b]}(t) * \frac{(t-a)}{(b-a)} + \epsilon(t), 1 \leq t \leq M$,
3. «воронка»: $f(t) = (6 + \zeta) * \chi_{[a,b]}(t) * \frac{(b-t)}{(b-a)} + \epsilon(t), 1 \leq t \leq M$,

где

1. M – длина временного ряда;
2. $\chi_{[a,b]} = \begin{cases} 0, t < a \\ 1, a \leq t \leq b \\ 0, t > b \end{cases}$,

3. ζ – случайная величина, подчиняющаяся стандартному нормальному распределению $N(0,1)$;
4. $\epsilon(t)$ – случайные величины, подчиняющиеся стандартному нормальному распределению $N(0,1)$;
5. a – случайная величина, подчиняющаяся равномерному распределению на отрезке $[16, 32]$;
6. b – случайная величина, подчиняющаяся равномерному распределению на отрезке $[32, 96]$.

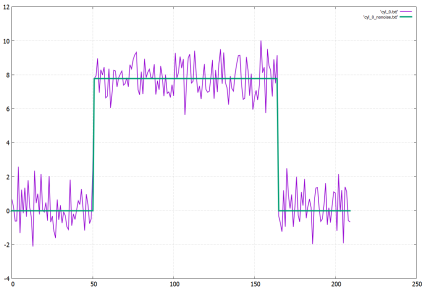


Рисунок 2.24 — Класс «цилиндр» (1)

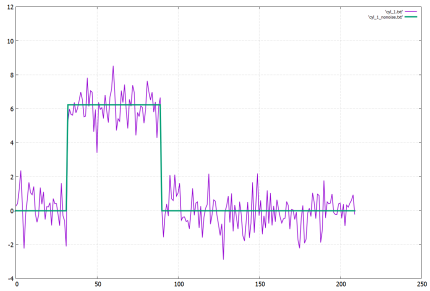


Рисунок 2.25 — Класс «цилиндр» (2)

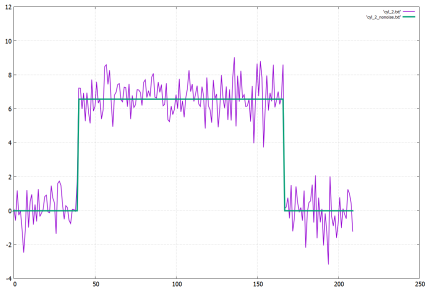


Рисунок 2.26 — Класс «цилиндр» (3)

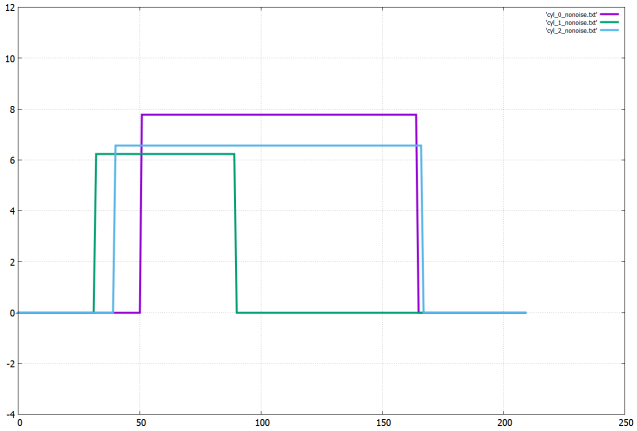


Рисунок 2.27 — Примеры временных рядов класса «цилиндр» без шума

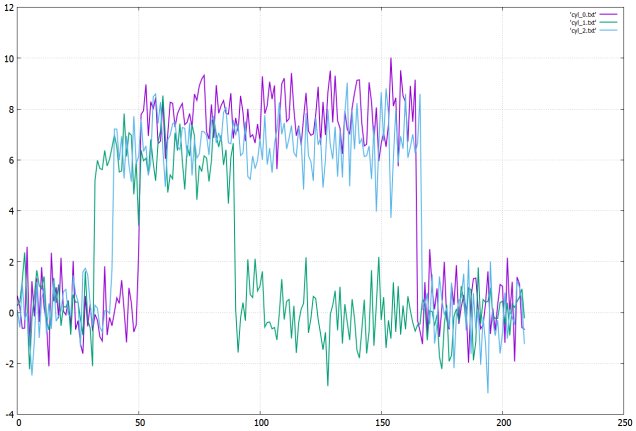


Рисунок 2.28 — Примеры временных рядов класса «цилиндр» с шумом

На рисунках 2.24-2.26 приведены примеры искусственно построенных на основании вышеприведённых формул временных рядов класса «цилиндр», по 2 на каждом графике. При этом каждый из графиков содержит исходный, незашумленный, временной ряд и соответствующий ему временной ряд с внесенным в него шумом. Графики выполнены в одном масштабе. На рисунках 2.27 и 2.28 для сравнения на одном графике приведены отдельно незашумленные и зашумленные временные ряды.

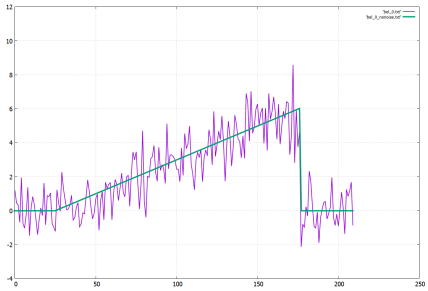


Рисунок 2.29 — Класс «колокол» (1)

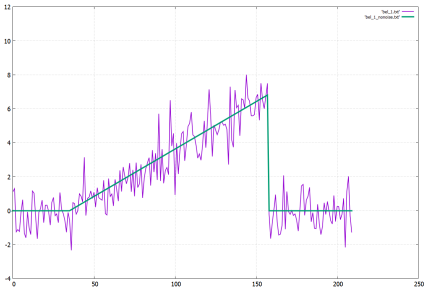


Рисунок 2.30 — Класс «колокол» (2)

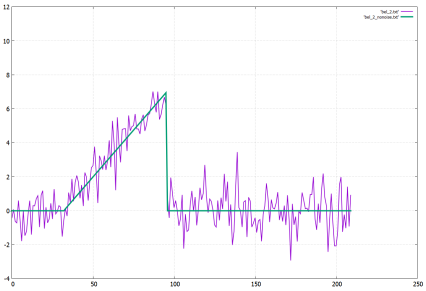


Рисунок 2.31 — Класс «колокол» (3)

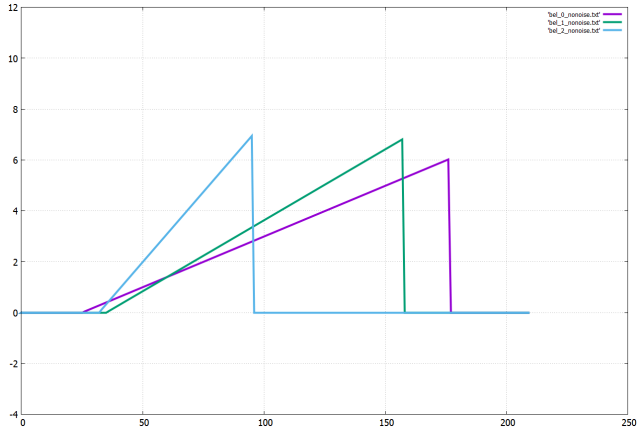


Рисунок 2.32 — Примеры временных рядов класса «колокол» без шума

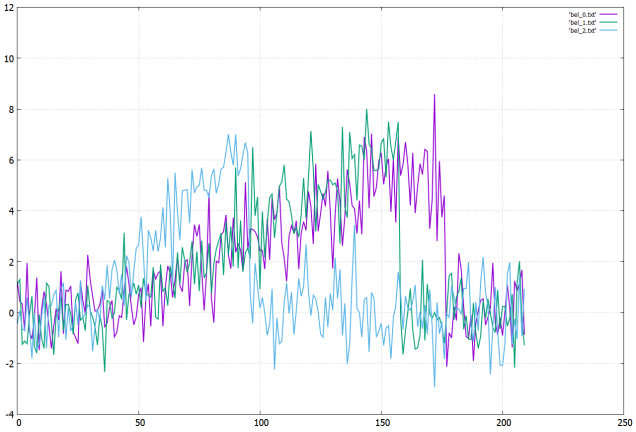


Рисунок 2.33 — Примеры временных рядов класса «колокол» с шумом

Аналогично приведены примеры для временных рядов, относящихся к классам «воронка» (рис. 2.29-2.33) и «колокол» (рис. 2.34-2.38).

Полученные таким образом наборы временных рядов использовались для изучения влияния шума на работу алгоритмов обнаружения аномалий и классификации. Для оценки уровня шума в данных следует обратить внимание на формулы, по которым можно генерировать временные ряды каждого класса. В формулах содержится слагаемое $\epsilon(t)$ – случайная величина, подчиняющаяся стандартному нормальному распределению $N(0,1)$, что является аддитивным гауссовским шумом.

2.5.2 Набор данных *«контрольные карты»*

Помимо наборов данных из самого *UC Irvine Repository* [65], временные ряды для данного набора можно получить по следующим формулам [91]:

1. «нормальное значение»: $norm(t) = m + s * \epsilon(t), 1 \leq t \leq M$,
2. «цикличность»: $cyclic(t) = m + a * sin(\pi * \frac{t}{T}) + s * \epsilon(t), 1 \leq t \leq M$,
3. «уменьшение значения»: $dec(t) = m - g * t + s * \epsilon(t), 1 \leq t \leq M$,

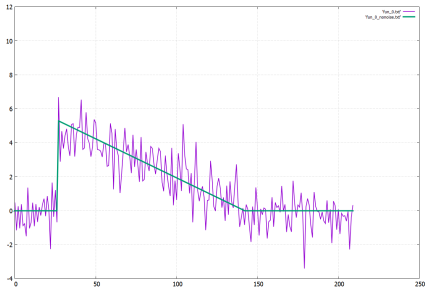


Рисунок 2.34 — Класс «воронка» (1)

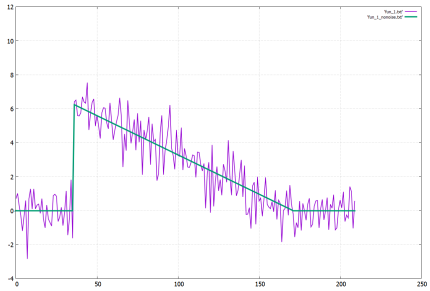


Рисунок 2.35 — Класс «воронка» (2)

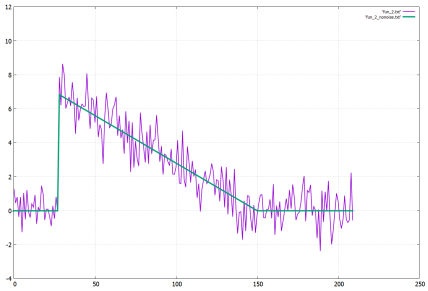


Рисунок 2.36 — Класс «воронка» (3)

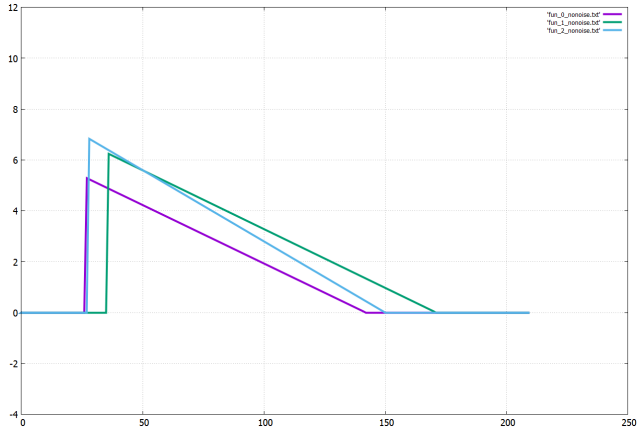


Рисунок 2.37 — Примеры временных рядов класса «воронка» без шума

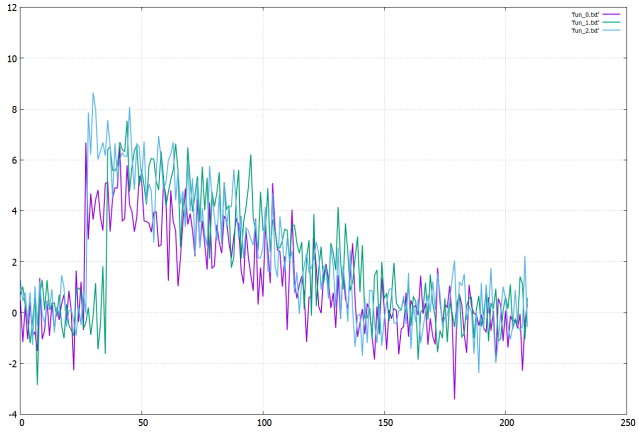


Рисунок 2.38 — Примеры временных рядов класса «воронка» с шумом

4. «увеличение значения»: $inc(t) = m + g * t + s * \epsilon(t), 1 \leq t \leq M$,
5. «резкий спад»: $downw(t) = m - k(t) * x + s * \epsilon(t), 1 \leq t \leq M$,
6. «резкое возрастание»: $upw(t) = m + k(t) * x + s * \epsilon(t), 1 \leq t \leq M$,

где

1. M – длина временного ряда;
2. $m = 30, s = 2$;
3. $\epsilon(t)$ – случайная величина, подчиняющаяся равномерному распределению на отрезке $[-3, 3]$;
4. a, T – случайные величины, подчиняющиеся равномерному распределению на отрезке $[10, 15]$;
5. g – случайная величина, подчиняющаяся равномерному распределению на отрезке $[0.2, 0.5]$;
6. x – случайная величина, подчиняющаяся равномерному распределению на отрезке $[7.5, 20]$;

7. $k(t) = \begin{cases} 0, t < t3 \\ 0, t > t3 \end{cases}$, где $t3$ – случайная величина, подчиняющаяся равномерному распределению на отрезке $[\frac{M}{3}, \frac{2*M}{3}]$.

Полученные таким образом наборы временных рядов использовались для изучения влияния шума на работу алгоритмов обнаружения аномалий и классификации. Для оценки уровня шума в данных следует обратить внимание на формулы, по которым можно генерировать временные ряды каждого класса. В формулах содержится слагаемое $\epsilon(t)$ – случайная величина, подчиняющаяся стандартному нормальному распределению $N(0,1)$, что является аддитивным гауссовским шумом.

2.6 Методы работы с зашумлёнными данными

На рис. 2.39 приведён пример временного ряда без шума, на рис. 2.40 приведен пример того же временного ряда с шумом. Выбранное представление для временных рядов позволяет успешно работать с шумом в данных: за счет сокращения размерности можно «сгладить» крайние значения для временного ряда, сохранив его форму и основные параметры.

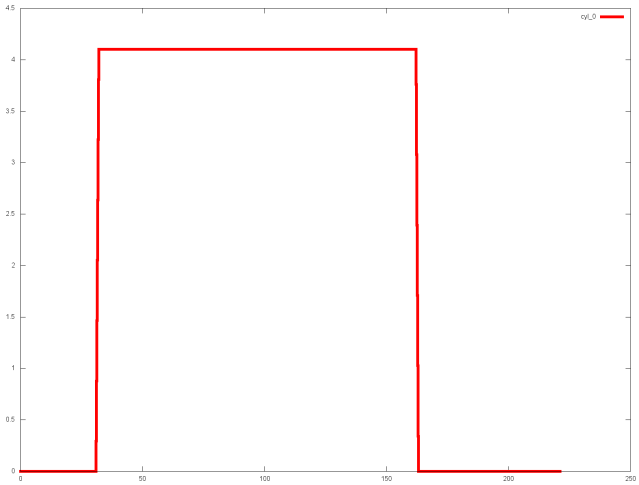


Рисунок 2.39 — Временной ряд без шума

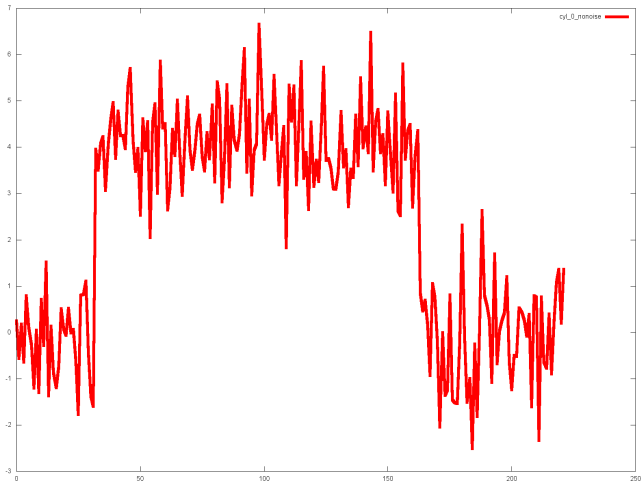


Рисунок 2.40 — Временной ряд с шумом

Рассмотрим несколько нормализованных представлений исходного временного ряда с различными параметрами. На рис. 2.41–2.44 изображены нормализованные преставления для временного ряда с шумом. На первом из рисунков (рис. 2.41) одна точка нормализованного временного ряда соответствует пяти точкам исходного временного ряда, на втором (рис. 2.42) – 10 точкам, на

третьем (рис. 2.43) - 20 точкам и на четвертом (рис. 2.44) - 30 точкам. Как видно, с увеличением числа точек исходного временного ряда, соответствующих одной точке нормализованного ряда, временной ряд «сглаживается» и нормализованный временной ряд становится все больше похожим на исходный временной ряд без шума (рис. 2.39).

Рисунок 2.41 — «Сжатие» в 5 раз

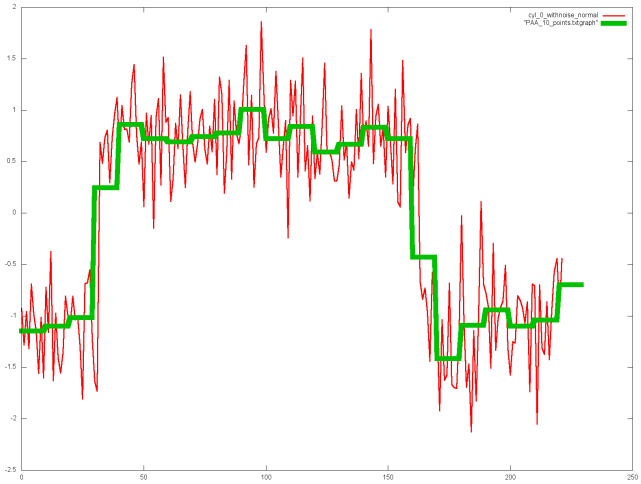


Рисунок 2.42 — «Сжатие» в 10 раз

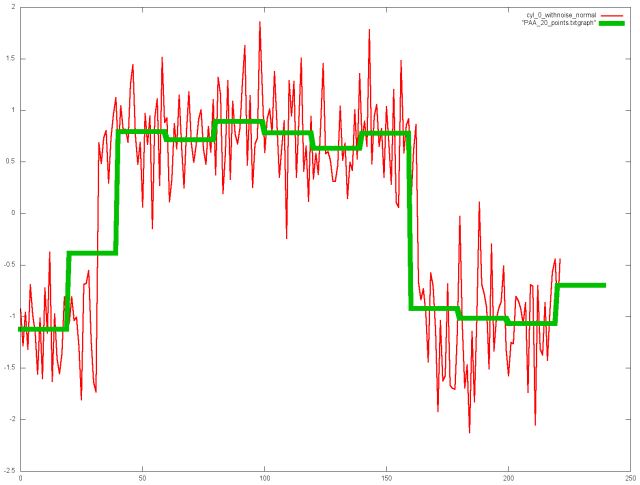


Рисунок 2.43 — «Сжатие» в 20 раз

Рисунок 2.44 — «Сжатие» в 30 раз

2.7 Постановка задачи обнаружения аномалий

Задача обнаружения аномалий для набора временных рядов ставится следующим образом. Пусть имеется набор объектов, где каждый объект есть временной ряд: $TS\_STUDY = \{ts\_study_1, ts\_study_2, .., ts\_study_{m_1}\}$. Назовем TS_STUDY обучающей выборкой. Каждый из временных рядов $ts\_study_i, 1 \leq i \leq m_1$ в обучающей выборке является примером «нормального» протекания некоторого процесса.

Множество $TS\_TEST = \{ts\_test_1, ts\_test_2, .., ts\_test_{m_2}\}$ назовем экзаменационной выборкой. На основании анализа временных рядов из TS_STUDY необходимо построить модель, позволяющую относить временные ряды из экзаменационной выборки TS_TEST к «нормальным рядам» или к «аномалиям» на основании некоторого критерия.

Рассмотрим набор ситуаций из табл. 2.5. Предположим, что данные ситуации описывают нормальное протекание процессов на сложном техническом объекте и принадлежат одному классу – «норма». На основании этих ситуаций необходимо построить такую модель, которая описывала бы «нормальное» протекание процессов и позволяла бы относить ситуации, возникающие на объекте, к «нормальным» или «аномальным». В данном случае перед нами задача обнаружения аномалий в наборах временных рядов, когда в обучающем множестве содержатся примеры единственного класса («норма»).

В общем случае для решения задачи определения аномалий в наборах временных рядов с одним классом распространены подходы, основанные на методе опорных векторов (и его модификациях) [92], нейронных сетях [75], использовании дискриминанта Фишера [93], продукционных правилах и др.

Рассмотрим данную задачу на простом примере. Пусть обучающая выборка TS_STUDY состоит из трех временных рядов - рис. 2.45, рис. 2.46, рис. 2.47. Экзаменационная выборка TS_TEST состоит также из трех временных рядов - рис. 2.48, рис. 2.49, рис. 2.50.

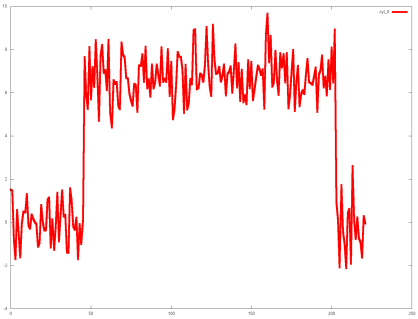


Рисунок 2.45 — Ряд 1 обуч. мн-ва

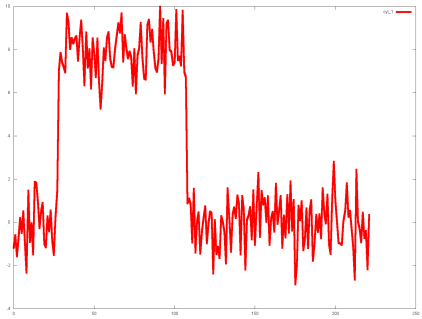


Рисунок 2.46 — Ряд 2 обуч. мн-ва

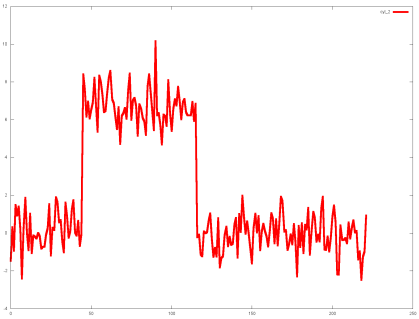


Рисунок 2.47 — Ряд 3 обуч. мн-ва

Исходя из приведенной выше постановки задачи обнаружения аномалий, видно, что временные ряды на рис. 2.49, 2.50 из экзаменационного множества значительно отличаются (в данном случае – по форме) от временных рядов из обучающего множества и, следовательно, будут являться аномалиями для данного обучающего множества. При этом можно предположить, что механизм, или

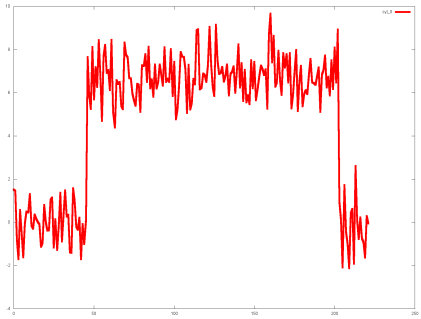


Рисунок 2.48 — Ряд 1 экз. мн-ва

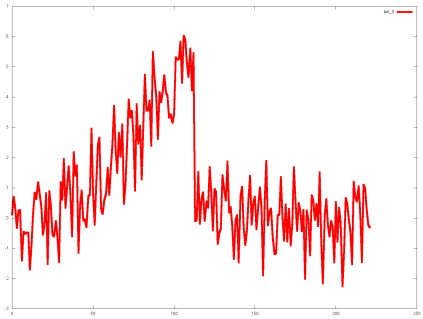


Рисунок 2.49 — Ряд 2 экз. мн-ва

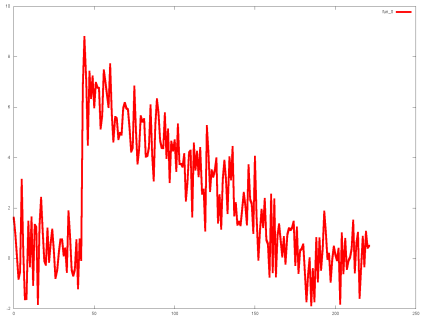


Рисунок 2.50 — Ряд 3 экз. мн-ва

закон, по которому были получены временные ряды, представленные на этих рисунках, отличается от механизма, с помощью которого были получены временные ряды из обучающего множества. Напротив, временной ряд на рис. 2.48 из экзаменационного множества не будет являться аномалией, так как по форме очень «похож» на временные ряды из обучающего множества.

Рассмотрим теперь набор ситуаций, представленных в табл. 2.7. Ситуации в таблице описывают нормальное протекание процессов и идентичны ситуациям из табл. 2.5, единственным отличием является последний столбец – класс ситуации. Каждая ситуация *Cum1-Cum9* теперь относится к одному из классов, взятых для примера из описанного ранее набора данных «цилиндр-колокол-воронка». Для краткости обозначим классы CY, BE и FU (**cy**linder – цилиндр, **be**ll – колокол, **fu**nnel – воронка [81]). Аналогично случаю с одним классом на основании ситуаций из табл. 2.7 необходимо построить модель, описывающую «нормальное» протекание процессов и позволяющую для каждой ситуации определить, относится ли она к «нормальным» или «аномальным». В случае, если ситуация была отнесена к «нормальным», требуется определить, к какому классу она относится. Класс обычно соответствует режиму функционирования объекта. В данном случае перед нами задача обнаружения аномалий в наборах временных рядов, когда в обучающем множестве содержатся примеры нескольких классов, объявленных «нормальными ситуациями». Это могут быть, например, временные ряды, относящиеся к двум классам: «цилиндр» и «колокол».

В общем случае для решения задачи определения аномалий в наборах временных рядов с несколькими классами распространены байесовский подход [76], подходы, основанные на использовании нейронных сетей [74], продукционных правил.

Таблица 2.7 — Описание ситуаций на объекте для случая 1 датчика

t	0	1	2	3	4	5	6	7	8	9	КС
Сит1	-1.07	-0.13	0.85	0.96	0.81	0.84	-0.08	-1.01	-0.90	-1.13	CY
Сит2	-0.72	-0.70	1.25	1.23	1.27	0.03	-0.76	-0.71	-0.71	-0.74	CY
Сит3	-0.94	-0.84	1.06	0.97	1.01	1.04	-0.35	-0.92	-0.83	-0.80	CY
Сит4	-0.56	-0.62	-0.19	0.64	1.45	1.39	-0.69	-0.61	-0.66	-0.62	BE
Сит5	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99	BE
Сит6	-0.54	-0.44	-0.28	0.75	1.61	0.40	-0.45	-0.53	-0.38	-0.61	BE
Сит7	-0.45	1.05	1.25	0.61	-0.35	-0.50	-0.39	-0.27	-0.89	-0.28	FU
Сит8	-0.68	-0.67	1.63	1.07	0.69	0.01	-0.59	-0.70	-0.64	-0.53	FU
Сит9	-1.01	0.50	1.35	0.89	0.33	0.18	-0.34	-0.75	-0.98	-0.65	FU

Рассмотрим данную задачу на простом примере. Пусть обучающая выборка TS_STUDY состоит из шести временных рядов: рис. 2.51–рис. 2.56. Экзаменационная выборка состоит из трёх временных рядов: рис. 2.57–рис. 2.59.

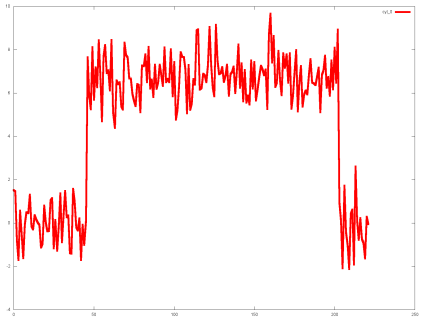


Рисунок 2.51 — Ряд 1 обуч. мн-ва

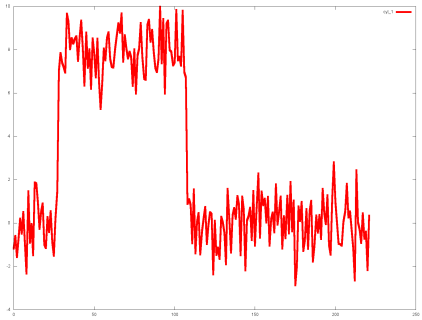


Рисунок 2.52 — Ряд 2 обуч. мн-ва

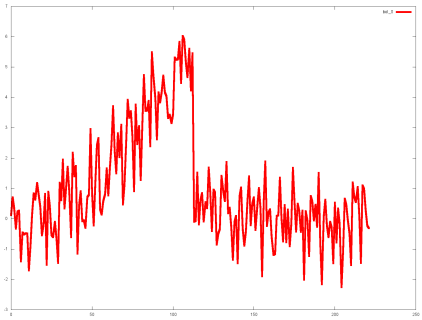


Рисунок 2.53 — Ряд 3 обуч. мн-ва

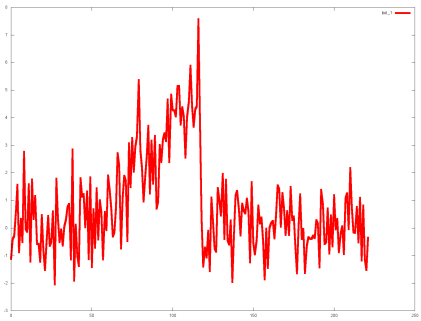


Рисунок 2.54 — Ряд 4 обуч. мн-ва

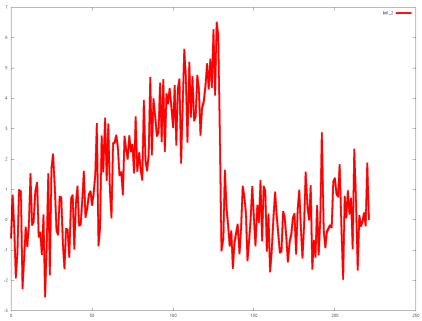


Рисунок 2.55 — Ряд 5 обуч. мн-ва

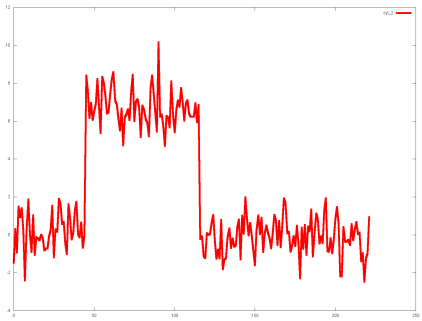


Рисунок 2.56 — Ряд 6 обуч. мн-ва

Исходя из приведенной выше постановки задачи, видно, что временные ряды на рис. 2.51, рис. 2.52 и рис. 2.56 сильно схожи между собой, а значит,

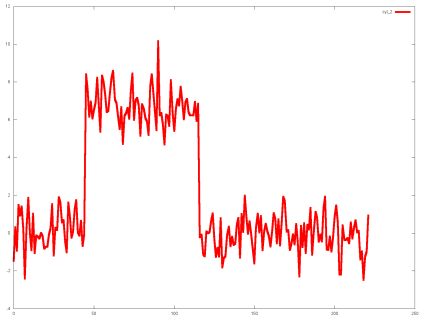


Рисунок 2.57 — Ряд 1 экз. мн-ва

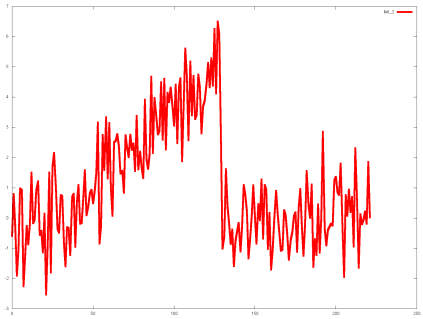


Рисунок 2.58 — Ряд 2 экз. мн-ва

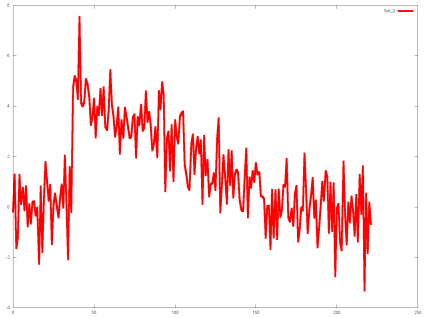


Рисунок 2.59 — Ряд 3 экз. мн-ва

принадлежат одному классу – назовем его *класс1*. Временные ряды рис. 2.53, рис. 2.54 и рис. 2.55 также схожи, но принадлежат другому классу – назовем его *класс2*. Из экзаменационного множества (рис. 2.57–рис. 2.59) видно, что временной ряд на рис. 2.57, скорее всего, принадлежит классу *класс1*, временной ряд на рис. 2.58 – классу *класс2*. Третий же временной ряд (рис. 2.59) значительно отличается от двух предыдущих и, очевидно, «не похож» ни на один ряд из обучающего множества. При этом можно предположить, что механизм, или закон, по которому был получен этот временной ряд экзаменационной выборки, отличается от механизма, с помощью которого были получены временные ряды из обучающего множества. Напротив, временные ряды на рис. 2.57 и рис. 2.58 из экзаменационного множества не будут являться аномалиями, так как по форме очень «похожи» на отдельные временные ряды из обучающего множества.

2.8 Задача обнаружения аномалий в наборах временных рядов с одним классом

2.8.1 Разработка метода обнаружения аномалий

В данной работе предлагается метод обнаружения аномалий в наборах временных рядов, который является модификацией метода, основанного на «точном описании исключения» [94].

Исходная постановка задачи, данная в [94], следующая: для заданного конечного множества объектов I необходимо получить множество-исключение I_x. Для этого на множестве I вводятся:

1. функция неподобия (dissimilarity) $D(I_j)$, $I_j \in I$, определенная на *P(I)* – множестве всех подмножеств I и принимающая положительные вещественные значения;

2. функция мощности (cardinality) $C(I_j)$, $I_j \in I$, определенная на *P(I)* – множестве всех подмножеств I и принимающая положительные вещественные значения, такая, что для любых $I_1 \subset I$, $I_2 \subset I$ выполняется $I_1 \subset I_2 \Rightarrow C(I_1) < C(I_2)$;
3. «фактор сглаживания» (smoothing factor) $SF(I_j) = C(I \setminus I_j) * (D(I) - D(I \setminus I_j))$, который вычисляется для каждого $I_j \subseteq I$.

Тогда $I_x \subset I$ будет считаться множеством-исключением для I относительно D и C, если его фактор сглаживания $SF(I_x)$ максимален [94].

Неформально, множество-исключение – это наименьшее подмножество из I, которое вносит наибольший вклад в его неподобие. Фактор сглаживания показывает, насколько может быть уменьшено неподобие множества I, если из него исключить подмножество I_j.

На основании метода, описанного в [94], автором был разработан алгоритм $TS - ADEEP$ [3], предназначенный для обнаружения аномалий в наборах временных рядов. В качестве множества I рассматриваются множества $TS\_STUDY \cup \{ts\_test_j\}$ для каждого $ts\_test_j \in TS\_TEST$.

Функция неподобия для временных рядов будет задана следующим образом: $D(I_j) = \frac{1}{|I_j|} * \sum\limits_{i \in I_j} |i - \overline{I_j}|^2$, где $\overline{I_j} = \sum\limits_{i \in I_j} \frac{i}{|I_j|}$.

Сначала вычисляется $\overline{I_j}$ – среднее для временных рядов из I_j. В данном случае это эквивалентно вычислению среднего для обычных векторов: i – временной ряд из подмножества I_j, $|I_j|$ – число элементов в $|I_j|$.

Функция неподобия вычисляется как сумма квадратов расстояний (используется евклидова метрика) между средним и векторами из I_j, которая затем нормализуется – делится на число элементов во множестве I_j.

Функция мощности задаётся формулой $C(I \setminus I_j) = \frac{1}{|I_j|+1}$.

Формула для вычисления фактора сглаживания имеет прежний вид $SF(I_j) = C(I \setminus I_j) * (D(I) - D(I \setminus I_j))$.

Если множество-исключение I_x, полученное для $I = TS\_STUDY \cup \{ts\_test_j\}$ содержит ts_test_j, $1 \leq j \leq |TS\_TEST|$, то ts_test_j является аномалией.

2.8.2 Алгоритм *«TS-ADEEP»*

На основании описанного выше метода реализован непараметрический [95] алгоритм $TS-ADEEP$ [3] для определения аномалий в наборах временных рядов для обучающего множества с одним классом.

В табл. 2.8 приведен псевдокод алгоритма TS-ADEEP.

Таблица 2.8 — Псевдокод алгоритма TS-ADEEP

Алгоритм TS-ADEEP (TS_STUDY: обучающее множество, TS_TEST: экзаменационное множество)
Результат: TS_ANOM – набор временных рядов-аномалий
начало
$TS\_ANOM = \emptyset$
Для j от 1 до $|TS\_TEST|$
нц
  выбрать $ts\_test_j \in TS\_TEST$
  $I = TS\_STUDY \cup \{ts\_test_j\}$
  Найти множество-исключение I_x в I
  Если $ts\_test_j \in I_x$, то $TS\_ANOM = TS\_ANOM \cup \{ts\_test_j\}$
кц
вывести TS_ANOM
конец

Рассмотрим работу алгоритма $TS-ADEEP$ на примере. Пусть в обучающем множестве три временных ряда – рис. 2.60-2.62 (обозначим их для удобства $cyl1, cyl2, cyl3$).

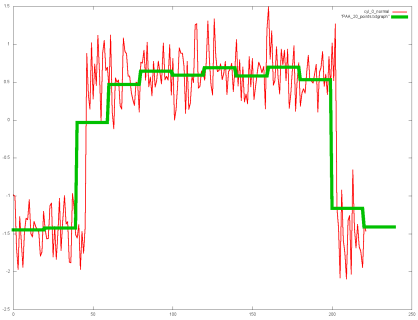


Рисунок 2.60 — Ряд 1 обуч. мн-ва

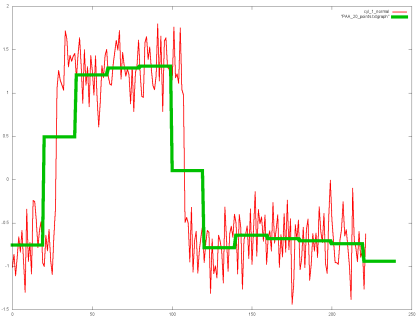


Рисунок 2.61 — Ряд 2 обуч. мн-ва

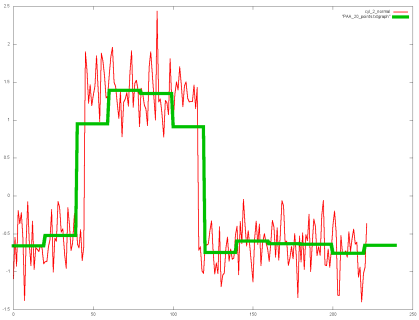


Рисунок 2.62 — Ряд 3 обуч. мн-ва

Нужно определить, является ли временной ряд, представленный на рис. 2.63 (обозначим его *bel*), аномалией.

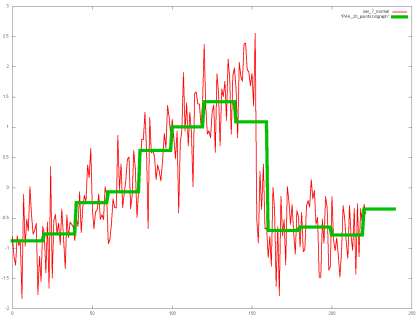


Рисунок 2.63 — bel

Таблица 2.9 — Результаты вычисления фактора сглаживания для подмножеств I

Подмножество I_j	Фактор сглаживания
cyl2, cyl3, bel	0.370713
cyl1, cyl3, bel	0.370713
cyl3, bel	0.0677333
cyl1, cyl2, bel	0.370713
cyl2, bel	0.205667
cyl1, bel	**0.45465**
bel	0.136392
cyl1, cyl2, cyl3	0.370713
cyl2, cyl3	0.362783
cyl1, cyl3	-0.00448333
cyl3	-0.128708
cyl1, cyl2	0.03515
cyl2	0.0194417
cyl1	0.0941917

В соответствии с алгоритмом множество I будет состоять из указанных четырех временных рядов: $I = \{cyl1, cyl2, cyl3, bel\}$. Рассматриваются все возможные подмножества I_j из I (за исключением пустого множества и самого множества I). Таких подмножеств $2^{|I|} - 2 = 14$: $\{\{cyl1\}, \{cyl2\}, \{cyl3\}, \{bel\}, \{cyl1, cyl2\}, \{cyl1, cyl3\}, \{cyl1, bel\}, \{cyl2, cyl3\}, \{cyl2, bel\}, \{cyl3, bel\}, \{cyl1, cyl2, cyl3\}, \{cyl2, cyl3, bel\}, \{cyl1, cyl3, bel\}, \{cyl1, cyl2, bel\}\}$. Для каждого из подмножеств по указанным формулам вычисляется фактор сглаживания. Результаты вычислений приведены в таблице 2.9.

В данном случае максимальный фактор сглаживания (0.45465) имеет множество, состоящее из временных рядов $I_x = \{cyl1, bel\}$, следовательно, оно и

является множеством-исключением. А так как временной ряд bel попал в множество-исключение ($\{bel\} \in I_x$), то он является аномалией.

2.8.2.1 Вычислительная сложность алгоритма *«TS-ADEEP»*

Нами была рассчитана вычислительная сложность алгоритма $TS-ADEEP$. Пусть N – число временных рядов в рассматриваемом множестве $TSset$. Для поиска множества-исключения надо рассмотреть булеан $TSset$, за исключением пустого множества и самого $TSset$. Общее число подмножеств I_j за исключением пустого и самого $TSset$ равно равно $2^N - 2$ (не превосходит 2^N). Таким образом, сложность алгоритма – $O(2^N)$, в связи с чем не рекомендуется использовать в качестве обучающих множеств большое число временных рядов (более 20). Однако если учесть тот факт, что множество-исключение – это *наименьшее* подмножество из I, которое вносит наибольший вклад в его неподобие, то можно ограничиться рассмотрением подмножеств не более некоторого заданного размера, что позволяет значительно сократить перебор без снижения точности обнаружения аномалий в большинстве экспериментов, проведенных в работе.

2.9 Задача обнаружения аномалий в наборах временных рядов с несколькими классами

2.9.1 Разработка метода обнаружения аномалий

В данной работе предлагается метод обнаружения аномалий в наборах временных рядов с несколькими классами, который является обобщением метода обнаружения аномалий для случая обучающего множества, содержащего примеры одного класса.

Обобщение является достаточно очевидным: разделив обучающее множество на подмножества, содержащие примеры только одного класса и последовательно применив к ним и каждому из временных рядов экзаменационного множества метод обнаружения аномалий в наборах временных рядов с одним классом, можно определить, является ли рассматриваемый временной ряд аномалией. Если временной ряд является аномалией для каждого подмножества, то он является аномалией и для всего обучающего множества.

Таблица 2.10 — Псевдокод алгоритма TS-ADEEP-Multi

Алгоритм TS-ADEEP-Multi
(TS_STUDY: обучающее множество, содержащее примеры нескольких классов; TS_TEST: экзаменационное множество)
Результат: TS_ANOM – набор временных рядов-аномалий
НАЧАЛО
$TS\_ANOM = \emptyset$
Пусть N – число классов, содержащихся в обучающем множестве
$TS\_STUDY\_C = \{TS\_STUDY\_C_1, TS\_STUDY\_C_2, .., TS\_STUDY\_C_N\}$ – разбиение множества TS_STUDY такое, что $TS\_STUDY\_C_k$ содержит только примеры класса $k, k = 1..N$
Для j от 1 до $|TS\_TEST|$
нц
  выбрать ts_test_j из TS_TEST
  Для k от 1 до N
  нц
    $I = TS\_STUDY\_C_k \cup ts\_test_j$
    Найти множество-исключение I_x в I
    Если $ts\_test_j \in I_x$, то ts_test_j является аномалией для класса k (то есть не принадлежит ему)
  кц
  Если ts_test_j не принадлежит ни одному из классов $TS\_STUDY\_C_k, k = 1..N$,то $TS\_ANOM = TS\_ANOM \cup ts\_test_j$
кц
вывести TS_ANOM
КОНЕЦ

2.9.2 Алгоритм *«TS-ADEEP-Multi»*

На основании описанного выше метода реализован непараметрический [95] алгоритм $TS-ADEEP-Multi$, который является обобщением алгоритма $TS-ADEEP$ для случая обучающего множества, содержащего примеры нескольких классов временных рядов.

В табл. 2.10 приведен псевдокод алгоритма TS-ADEEP-Multi.

Рассмотрим работу алгоритма $TS-ADEEP-Multi$ на примере. Пусть в обучающем множестве шесть временных рядов: три временных ряда из предыдущего примера, рис. 2.60-2.62 (обозначим их для удобства $cyl1, cyl2, cyl3$), и

три временных ряда, изображенных на рис. 2.64-2.66 (обозначим их для удобства $fun1, fun2, fun3$).

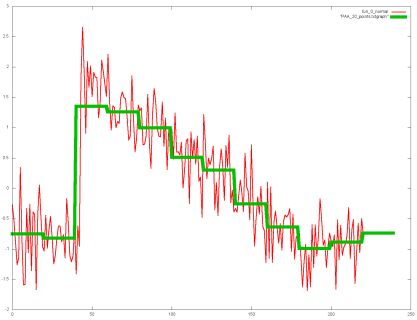


Рисунок 2.64 — Ряд 1 обуч. мн-ва

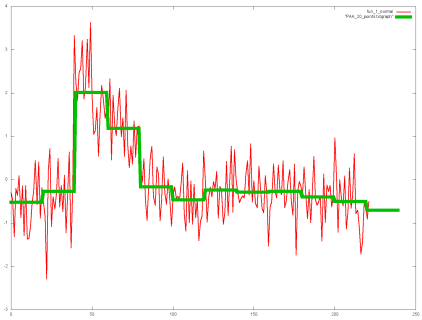


Рисунок 2.65 — Ряд 2 обуч. мн-ва

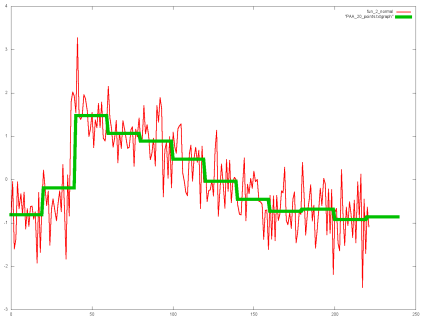


Рисунок 2.66 — Ряд 3 обуч. мн-ва

Нужно определить, является ли временной ряд на рис. 2.67 (обозначим его bel) аномалией.

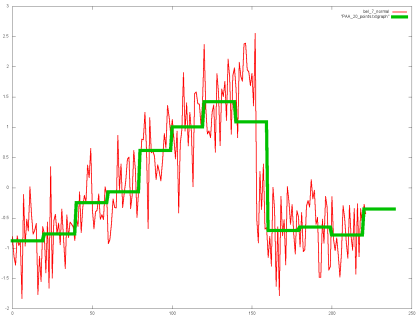


Рисунок 2.67 — bel

В соответствии с алгоритмом будут рассмотрены два множества – $I1 = \{cyl1, cyl2, cyl3, bel\}$, $I2 = \{fun1, fun2, fun3, bel\}$. Аналогично алгоритму $TS-ADEEP$ рассматриваются все возможные подмножества из $I1$ и $I2$ и для каждого их них по указанным формулам вычисляется фактор сглаживания. Для $I1$ множеством-исключением стало $\{cyl1, bel\}$ с фактором сглаживания 0.45465, для $I2$ - $\{bel\}$ с фактором сглаживания 0.385212. Так как временной ряд bel является аномалией для обоих классов из обучающего множества, этот временной ряд является аномалией по отношению и ко всему обучающему множеству.

2.9.2.1 Вычислительная сложность алгоритма *«TS-ADEEP-Multi»*

Нами была рассчитана вычислительная сложность алгоритма $TS-ADEEP-Multi$. Пусть N – число временных рядов в рассматриваемом множестве $TSset$, $N1 < N$ – максимальное число временных рядов, принадлежащих

одному и тому же классу, k – число классов. Для поиска множества-исключения надо рассмотреть булеан $TSset$, за исключением пустого множества и самого $TSset$. Общее число подмножеств I_j за исключением пустого и самого $TSset$ не превосходит 2^N. Таким образом, сложность алгоритма – $O(k * 2^N)$, в связи с чем не рекомендуется использовать в качестве обучающих множеств большое число временных рядов одного и того же класса (более 20). Аналогично алгоритму *«TS-ADEEP»*, можно сократить перебор, ограничившись рассмотрением подмножеств не более некоторого заданного размера.

2.9.3 Использование деревьев решений для обнаружения аномалий в наборах временных рядов с несколькими классами

В главе 1 были рассмотрены наиболее успешные для индуктивного формирования понятий модели представления знаний – деревья решений. Эта модель используется в ряде алгоритмов, относящихся к категории «обучение с учителем». В соответствии с этой стратегией на основе обучающей выборки, содержащей примеры и контрпримеры объектов определённого класса, строится дерево решений, представляющее собой особую форму теста, позволяющего в дальнейшем успешно классифицировать новые примеры, не вошедшие первоначально в обучающую выборку.

Известен ряд алгоритмов, результатом работы которых будет построенное в определённой форме дерево решений: алгоритм ДРЕВ [38], алгоритм, основанный на метрике Хэмминга [38], *ID3* [35] и различные модификации этих алгоритмов – *C4.5* [96], *ID5R* [97] и другие – обрели широкое распространение и зарекомендовали себя в широком спектре приложений.

Формально дерево решений – это взвешенный ориентированный граф $T = (V, E)$. В множестве вершин V выделим вершину $v_0 \in V$ – корень дерева. Все вершины разделим на два класса: $V_i \subset V$– множество внутренних вершины (узлов) дерева; V_i включает в себя такие вершины, из которых выходят дуги; $V_l \subset V$ – множество внешних, конечных, вершин дерева (листьев); V_l включает в себя такие вершины, из которых дуги не выходят; V_i и V_l образуют разбиение множества вершин V дерева решений T: $V_i \cap V_l = \emptyset, V_i \cup V_l = V$.

Внутренние вершины V_i дерева взвешены (помечены) именами атрибутов, используемых при признаковом описании объектов. Вершины-листья V_l взвешены (помечены) именами классов.

Каждая дуга e дерева решений взвешена условием «атрибут = значение атрибута» (для качественных значений атрибутов) либо «атрибут σ значение атрибута» (для количественных значений атрибутов, $\sigma \in \{\geq, >, <, \leq\}$), где «атрибут» – имя атрибута в вершине, из которой исходит дуга e, «значение атрибута» – одно из возможных значений (количественное или качественное) признака «атрибут».

Рассмотрим возможность применения алгоритмов построения деревьев решений для работы с такими объектами, как временные ряды. Из перечисленных выше алгоритмов возьмём алгоритм *ID3*, поскольку он позволяет строить деревья решений, отличные от бинарных, и успешно работает с символьными данными. Этот алгоритм также позволяет классифицировать примеры в случае их принадлежности к нескольким классам (2, 3 и более). Исходными данными для алгоритма построения дерева решений является обучающее множество, представленное в виде таблицы. Каждая строка таблицы содержит описание одного из примеров с указанием того, к какому классу относится данный пример. Описание примера представляет собой строку значений атрибутов (признаков), характеризующих свойства данного объекта.

Для рассмотренного ранее алгоритма *«TS-ADEEP-Multi»* использовались исходные данные, представленные в виде таблицы 2.7. Каждая строка таблицы 2.7 представляет описание одного из временных рядов, причём явно указано, к какому классу принадлежит этот ряд. Чтобы иметь возможность использовать эти данные как обучающую выборку для построения дерева решений алгоритмом *ID3* применим к числовым данным преобразование в символьную форму. Преобразование выполняется с помощью алгоритма SAX, описанного ранее. Результат преобразования представлен в таблице 2.11 (использовался алфавит из 10 символов). Таблица 2.11 с формальной точки зрения может быть использована как исходные данные (обучающая выборка) для алгоритма *ID3*: каждая строка представляет описание одного объекта – временного ряда; известно, к какому классу относится объект (*CY*, *BE*, *FU*), атрибутами являются моменты времени (0, 1, 2, . . . 9), а их значениями – показания датчиков в дискретном символьном представлении в соответствующие моменты времени.

На рисунке 2.68 представлено дерево решений, полученное алгоритмом *ID3* по обучающей выборке, данной в табл. 2.11.

Таблица 2.11 — Описание ситуаций на объекте для случая 1 датчика - символьное представление

t	0	1	2	3	4	5	6	7	8	9	КС
Сит1	B	E	I	I	H	I	E	B	B	B	CY
Сит2	C	C	I	I	I	F	C	C	C	C	CY
Сит3	B	B	I	I	I	I	D	B	C	C	CY
Сит4	C	C	E	H	J	J	C	C	C	C	BE
Сит5	B	B	C	E	G	H	I	J	B	B	BE
Сит6	C	D	D	H	J	G	D	C	D	C	BE
Сит7	D	I	I	H	D	D	D	D	B	D	FU
Сит8	C	C	J	I	H	F	C	C	C	C	FU
Сит9	B	G	J	I	G	F	D	C	B	C	FU

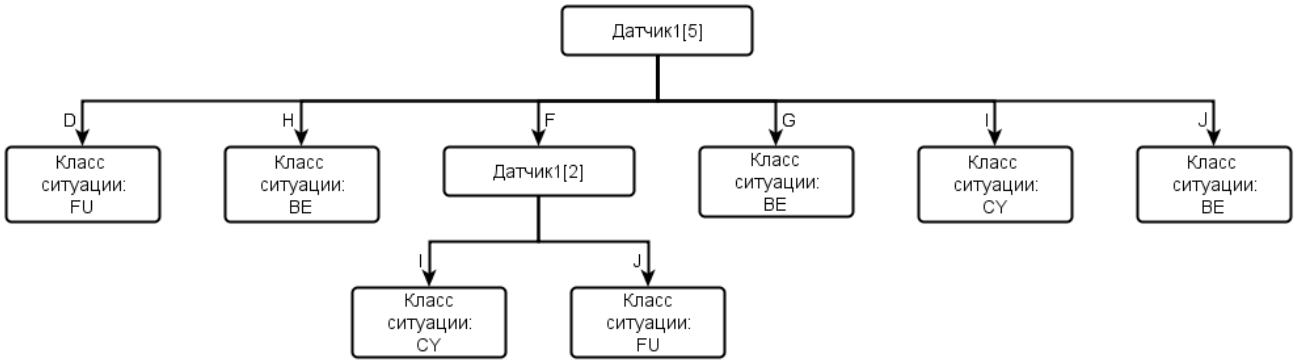


Рисунок 2.68 — Дерево решений

Полученное дерево решений можно использовать для обнаружения аномалий в наборах временных рядов: если оно относит некоторый временной ряд ts к одному из классов CY, BE или FU, то рассматриваемый временной ряд не является аномалией. В противном случае временной ряд ts является аномалией.

Несмотря на возможность использовать описанные выше методы для обнаружения аномалий, данные методы обладают недостатком: они могут быть использованы для решения лишь частной задачи – обнаружения аномалий для набора динамических объектов с одним атрибутом. В связи с этим необходим метод, позволяющий работать с динамическими объектами общего вида: необходимо учитывать более одного признака (что соответствует наличию нескольких датчиков), а также тот факт, что ситуации могут развиваться за разные интервалы времени. Кроме того, при значительном количестве ситуаций, используемых как исходные данные для задачи диагностики/обнаружения аномалий, модель

может быть неэффективной, в связи с чем необходимо использовать лишь «существенные» данные из набора исходных ситуаций.

2.10 Выводы ко второй главе

Во второй главе было сделано следующее:

1. Рассмотрено понятие динамического объекта, представляющего собой временной ряд. Дано определение временного ряда, характеристики временных рядов и способы их представления.
2. Приведены методы и алгоритмы предварительного преобразования временных рядов: сведение числового временного ряда к нормализованной форме, снижение уровня шума в данных, преобразование нормализованного числового ряда в символьный ряд.
3. Рассмотрена проблема обнаружения аномалий в случае, когда ситуация представима временным рядом. Дана постановка задачи обнаружения аномалий в наборах временных рядов с одним и несколькими классами и выполнен обзор существующих методов решения данных задач.
4. Предложен подход к решению задач обнаружения аномалий в наборах временных рядов, сводящий данную задачу к задаче обобщения понятий и классификации.
5. Приведен обзор наборов данных, используемых в работе.
6. Предложен метод и разработан алгоритм **TS-ADEEP** обнаружения аномалий для наборов временных рядов с одним классом, рассчитана его вычислительная сложность.
7. Предложен метод и разработан алгоритм **TS-ADEEP-Multi** обнаружения аномалий для наборов временных рядов, относящихся к нескольким классам, рассчитана его вычислительная сложность.

Глава 3. Задача обобщения для динамических объектов. Общий случай.

В главе 2 рассматривалась лишь частная задача и методы ее решения – для наиболее простго случая, когда динамический объект обобщения представляет собой временной ряд. При этом время рассматривалось лишь формально, неявно, тогда как в реальных системах поддержки принятия решений требуется явно учитывать фактор времени.

Общий случай представляется гораздо более сложным: динамический объект обобщения можно рассматривать как *набор* временных рядов (см. представление в табл. 1.2), причем длина временных рядов может быть различной.

В данной главе описано применение аппарата темпоральных деревьев решений, позволяющее решать задачу обобщения для динамических объектов с произвольным числом атрибутов при условии использования символьного описания такого атрибута (временного ряда). Методы описания временного ряда набором символов были подробно рассмотрены в главе 2.

Нашей целью является исследование случая, когда динамический объект обобщения характеризуется q>1 признаками. Предположим (для примера), что q=3. Тогда в представлении, введенном в главе 1 (см. табл. 1.2) динамический объект обобщения, или динамическую ситуацию, можно рассматривать как набор из трех временных рядов. Зададим некоторое $r = t^*$ – максимальный интервал времени, на котором будем рассматривать ситуацию – такой промежуток времени соответствует максимальной длине временного ряда. Пример динамического объекта обобщения приведен в табл. 3.1: здесь длина временного интервала $r = 10$, каждая строка таблицы представляет собой значения одного из параметров на рассматриваемом интервале. Отметим, что представление в табл. 3.1 в точности соответствует представлению динамического объекта, данному в табл. 1.2.

Таблица 3.1 — Пример динамического объекта обобщения для случая получения наблюдений от трёх датчиков

t	0	1	2	3	4	5	6	7	8	9
Датчик$_1$	-0.56	-0.62	-0.19	0.64	1.45	1.39	-0.69	-0.61	-0.66	-0.62
Датчик$_2$	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99
Датчик$_3$	-0.54	-0.44	-0.28	0.75	1.61	0.40	-0.45	-0.53	-0.38	-0.61

Пример набора динамически изменяющихся ситуаций (Сит1-Сит4) для случая, когда поведение сложной системы контролируется показаниями нескольких датчиков (q=3), приведён в табл. 3.2: здесь длина временного интервала, на котором ведутся наблюдения за ситуацией, $r = 10$, для описания каждой ситуации используются показания трёх датчиков на заданном интервале, каждая ситуация относится к классу $NORM$ (соответствует нормальному состоянию системы, поведение которой следует контролировать). Заданный таким образом набор ситуаций предлагается использовать как исходные данные для решения задачи обобщения.

Таблица 3.2 — Набор ситуаций на объекте для случая 3 датчиков

	t	0	1	2	3	4	5	6	7	8	9	КС
Сит1	$Датчик_1$	-1.07	-0.13	0.85	0.96	0.81	0.84	-0.08	-1.01	-0.90	-1.13	NORM
	$Датчик_2$	-0.72	-0.70	1.25	1.23	1.27	0.03	-0.76	-0.71	-0.71	-0.74	
	$Датчик_3$	-0.94	-0.84	1.06	0.97	1.01	1.04	-0.35	-0.92	-0.83	-0.80	
Сит2	$Датчик_1$	-0.56	-0.62	-0.19	0.64	1.45	1.39	-0.69	-0.61	-0.66	-0.62	NORM
	$Датчик_2$	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99	
	$Датчик_3$	-0.54	-0.44	-0.28	0.75	1.61	0.40	-0.45	-0.53	-0.38	-0.61	
Сит3	$Датчик_1$	-0.45	1.05	1.25	0.61	-0.35	-0.50	-0.39	-0.27	-0.89	-0.28	NORM
	$Датчик_2$	-0.68	-0.67	1.63	1.07	0.69	0.01	-0.59	-0.70	-0.64	-0.53	
	$Датчик_3$	-1.01	0.50	1.35	0.89	0.33	0.18	-0.34	-0.75	-0.98	-0.65	
Сит4	$Датчик_1$	-0.72	-0.70	1.25	1.23	1.27	0.03	-0.76	-0.71	-0.71	-0.74	NORM
	$Датчик_2$	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99	
	$Датчик_3$	-0.68	-0.67	1.63	1.07	0.69	0.01	-0.59	-0.70	-0.64	-0.53	

Набор динамических объектов обобщения, или динамических ситуаций, может описывать различные состояния сложного технического объекта или системы, причем объекты могут описывать как нормальное состояние системы, так и ненормальное, или аномальное, то есть соответствующее неисправности. В таком виде набор динамических объектов может использоваться как исходные данные для решения задачи диагностики – определения неисправности системы и указания причин, вызвавших неисправность. Рассмотрим задачу диагностики более подробно.

3.1 О технической диагностике

Техническая диагностика [98–100] – научно-техническая дисциплина, изучающая и устанавливающая признаки дефектов технических объектов, а также методы и средства обнаружения и поиска (указания местоположения) дефектов. Основной предмет технической диагностики – организация эффективной проверки исправности, работоспособности, правильности функционирования технических объектов (деталей, элементов, узлов, блоков, заготовок, устройств, изделий, агрегатов, систем, а также процессов передачи, обработки и хранения материи, энергии и информации), то есть организация процессов диагностирования технического состояния объектов при их изготовлении и эксплуатации, в том числе во время, до и после применения по назначению, при профилактике, ремонте и хранении. Диагностирование — одна из важных мер обеспечения и поддержания надёжности технических объектов.

Диагностирование как раздел искусственного интеллекта занимается разработкой методов и алгоритмов, способных определить корректность работы изучаемого объекта (системы). Если система работает некорректно, нужно как можно более точно определить, в какой части системы произошел отказ и какая ошибка произошла. Определение ошибки происходит на основе наблюдений, которые дают информацию о поведении системы.

Термин «диагностирование» также относится к определению неисправности системы.

Исходными данными для задачи диагностики обычно бывают описания некорректных состояний объекта (или ситуаций, которые могут возникнуть на объекте) и причины, к этому приведшие. При обучении на таких данных диагностическая система должна построить некоторую обобщенную модель, которая в дальнейшем смогла бы распознавать подобные (или схожие) ситуации на объекте и указывать причину неполадки.

Математическая модель объекта диагностирования (детерминированная или вероятностная) представляет собой описание объекта в исправном и в неисправном его состояниях в виде формальных зависимостей между возможными воздействиями на объект и его реакциями на эти воздействия. Модели (даже исправных объектов), используемые при диагностировании, могут отличаться от моделей, используемых при проектировании тех же объектов. В случае обнаружения неисправности может быть произведено некоторое корректирующее

воздействие на диагностируемый объект с целью перевода его в исправное состояние.

Алгоритм диагностирования предусматривает выполнение некоторой условной или безусловной последовательности определённых экспериментов с объектом. Эксперимент характеризуется тестовым или рабочим воздействием и составом контролируемых признаков, определяющих реакцию объекта на воздействие.

3.1.1 Диагностика на основе использования модели объекта

В данной работе предлагается использовать методы диагностики на основе использования модели объекта [101–103]. Для имитации поведения объекта и выявления неисправностей может быть использована модель специального вида, которая описывает структуру и поведение сложного технического объекта; данная модель представляет собой четверку $< O, E, S, B >$, где:

- O – множество компонент сложного технического объекта;
- E – функциональные связи между компонентами;
- S – множество переменных, описывающих состояние системы (в технической диагностике это чаще всего измерения, получаемые от датчиков, установленных в системе, или результаты вычислений выполненные над полученными измерениями);
- B – множество управляющих действий, допустимых в системе.

Для описания отдельного компонента $o \in O$ также используется модель, представляющая из себя тройку $< S', M, R >$, где:

- $S' \subseteq S$ – подмножество переменных, описывающих состояние данного компонента;
- M – набор режимов работы, включающих в себя состояние «норма» (корректное поведение) и состояния «неисправность» (некорректное поведение);
- R – набор отношений, связывающих множество переменных S, описывающих состояние системы, и набор режимов работы M.

Введем понятие C_n – множество состояний сложного технического объекта, которые диагностируются как нормальные; C_f – множество состояний, в которых наблюдаются неисправности на объекте. Для использования методов искусственного интеллекта в диагностике предлагается сформировать описание

понятия C_n и C_f в рамках введенной модели. На основе полученных обобщенных описаний классов C_n и C_f необходимо определить, какая из неисправностей произошла на объекте и – в более сложном случае – выработать рекомендации по выбору восстанавливающего действия на сложном техническом объекте; такое действие должно переводить систему из состояния «неисправно» в состояние «норма».

Для оценки эффективности построенной модели, необходимо сравнить поведение, предсказанное моделью, и наблюдаемое поведение объекта. Результаты наблюдений за поведением системы представлены в виде S – множества показаний, поступающих с датчиков, установленных на сложном техническом объекте. Чтобы построенная модель была полезной и для бортовой диагностики, она должна включать в себя действия, которые должны производиться в случае обнаружения неисправности/отказа. В общем случае действия (восстановительные действия) характеризуются некоторой стоимостью, которая чаще всего выражается в уменьшении функциональности системы. Таким образом, основная цель процедуры бортовой диагностики заключается в выборе оптимального действия в аварийном режиме.

Диагностирование на основе модели представляет собой частный случай абдуктивного вывода.

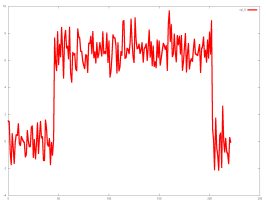
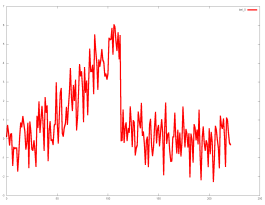
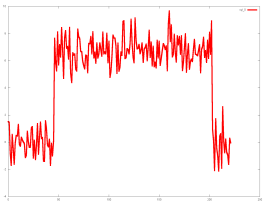
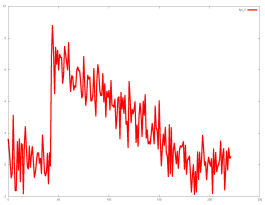
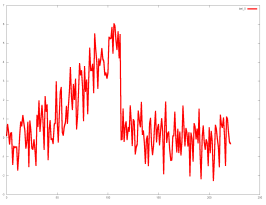
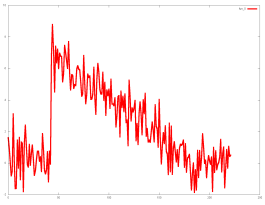
Успешность функционирования моделей, используемых в диагностике, зависит от выбора способа описаний классов ситуаций C_n и C_f. Для описания классов ситуаций можно использовать различные методы, такие как продукционные модели [104], нечеткие множества [105], приближенные множества [106; 107], деревья решений [35].

3.1.2 Исходные данные для задачи диагностики

В общем случае динамический объект обобщения имеет более одного параметра, то есть представляет собой набор из нескольких временных рядов. В таблице 3.3 приведен пример трёх классов динамических объектов, каждый из которых определяется тем, как меняются во времени сразу две величины (обозначим их далее **Параметр 1**, **Параметр 2**).

Из таблицы 3.3 следует, что на основе информации, поступающей только с одного датчика, невозможно точно различать классы *Класс 1*, *Класс 2* и *Класс 3*. Так, если в нашем распоряжении будет только величина **Параметр 1**, можно

Таблица 3.3 — Пример трёх классов динамических объектов с двумя параметрами

Параметр 1	Параметр 2	Класс
		Класс 1
		Класс 2
		Класс 3

однозначно распознать ситуации, соответствующие *Классу 3*, но *Класс 1* и *Класс 2* окажутся неразличимыми.

Используя 2 датчика (2 параметра), можно различить уже все 3 класса: *Класс 2* и *Класс 3* – на основании показаний первого датчика (**Параметр 1**), *Класс 1* и *Класс 2* – на основании показаний второго датчика (**Параметр 2**). Таким образом, предполагается, что наличие большего числа параметров может повысить различающую способность алгоритмов, использующих такие данные.

Рассмотрим более подробно пример для набора данных «контрольные карты», содержащего 6 классов, соответствующих различным образцам поведения параметра (рис. 3.1-3.6).

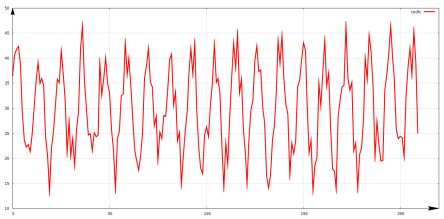


Рисунок 3.1 — «Цикличность»

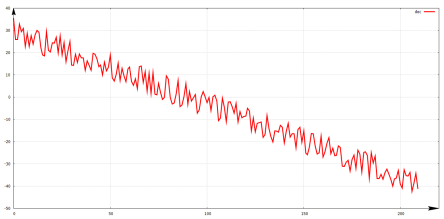


Рисунок 3.2 — «Уменьшение значения»

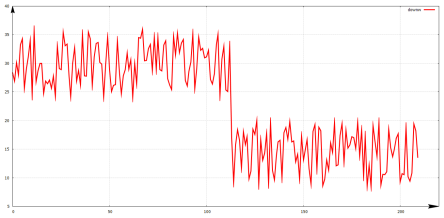


Рисунок 3.3 — «Резкий спад»

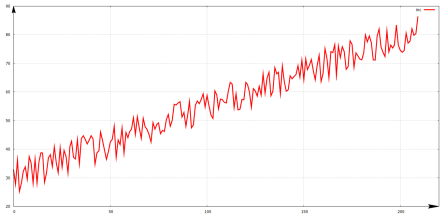


Рисунок 3.4 — «Увеличение значения»

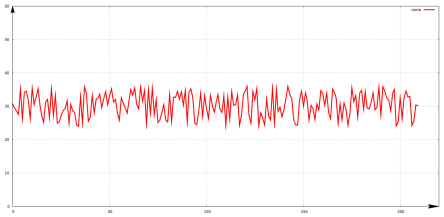


Рисунок 3.5 — «Нормальное значение»

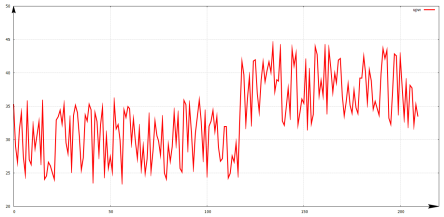


Рисунок 3.6 — «Резкое возрастание»

Сформируем из исходного набора данных новый набор следующим образом: динамические объекты обобщения будут иметь пять параметров, то есть представлять собой набор из пяти временных рядов. Каждый новый класс будет содержать 5 временных рядов из следующего упорядоченного перечисления:

1. «цикличность»
2. «уменьшение значения»
3. «резкий спад»
4. «увеличение значения»
5. «нормальное значение»
6. «резкое возрастание»

В таблице 3.4 приведен пример шести классов динамических объектов, каждый из которых определяется тем, как меняются во времени сразу пять величин (обозначим их далее **Параметр 1**, **Параметр 2**, **Параметр 3**, **Параметр 4**, **Параметр 5**). Первым классом в новом наборе будет являться динамический

объект с пятью параметрами, представляющими из собой следующие временные ряды:

1. «уменьшение значения»
2. «резкий спад»
3. «увеличение значения»
4. «нормальное значение»
5. «резкое возрастание»

(все, кроме «цикличности»);

Вторым классом в новом наборе будет являться динамический объект с пятью параметрами, представляющими из собой следующие временные ряды:

1. «цикличность»
2. «резкий спад»
3. «увеличение значения»
4. «нормальное значение»
5. «резкое возрастание»

(все, кроме «уменьшения значения»);

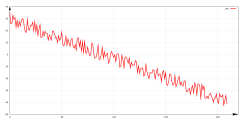
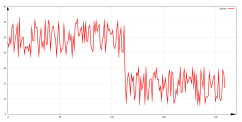
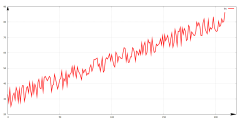
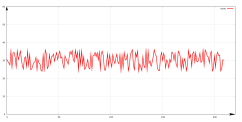
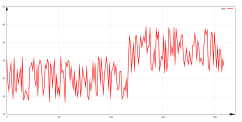
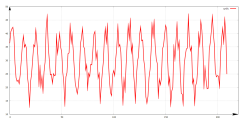
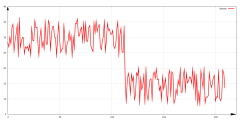
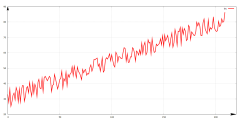
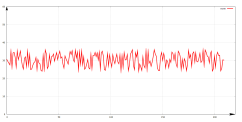
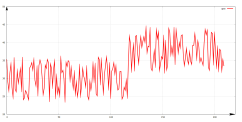
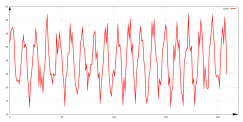
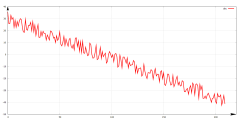
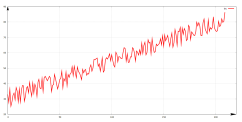
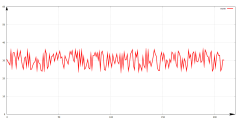
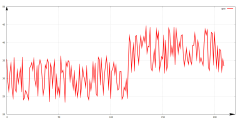
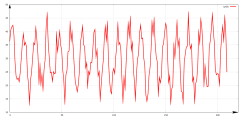
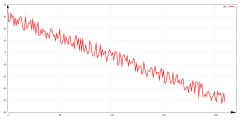
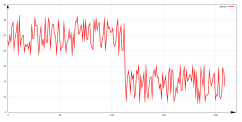
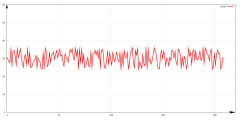
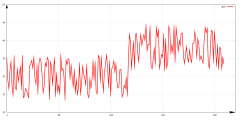
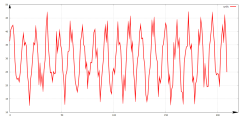
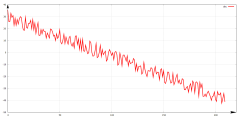
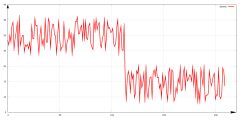
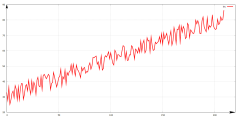
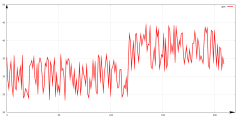
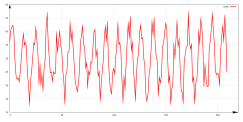
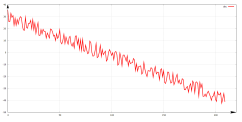
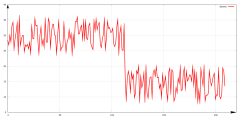
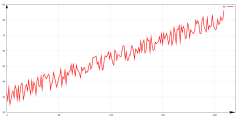
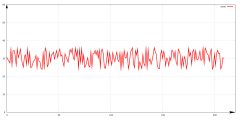
И так далее.

Из таблицы 3.4 следует, что на основе информации, поступающей только с одного датчика, невозможно точно различать все 6 классов. Так, если в нашем распоряжении будет только величина **Параметр 1**, можно однозначно распознать ситуации, соответствующие *Классу 1*, но *Класс 2 .. Класс 6* окажутся неразличимыми. Используя 2 датчика (2 параметра), можно распознать ситуации, соответстующие *Классу 1* и *Классу 2*, остальные 4 класса остаются неразличимыми. И только используя все 5 параметров, можно будет выделить 6 классов из имеющегося набора данных.

Таким образом, ни один из параметров не является наиболее информативным и каждый из параметров вносит свой вклад в процессе разделения объектов на классы. Предполагается, что наличие большего числа параметров может повысить различающую способность алгоритмов, использующих такие данные.

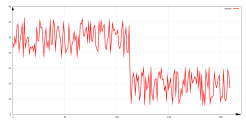
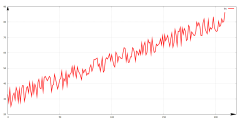
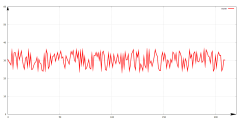
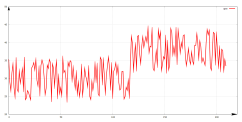
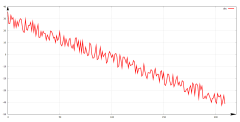
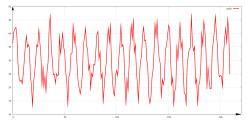
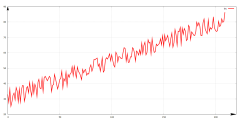
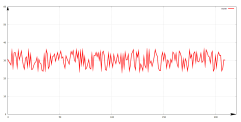
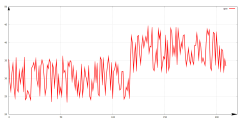
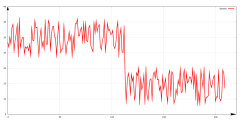
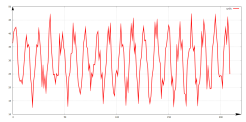
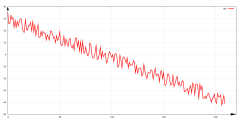
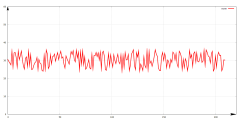
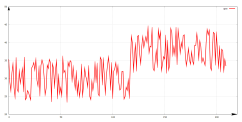
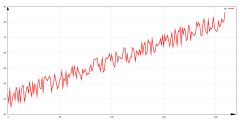
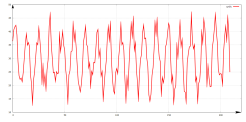
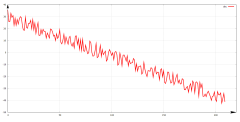
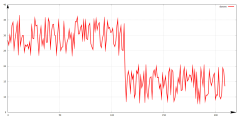
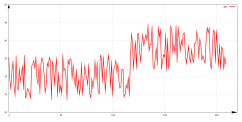
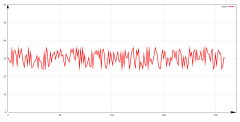
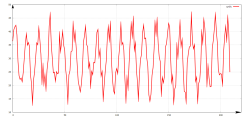
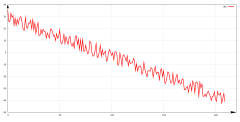
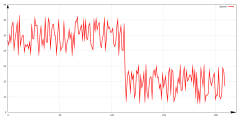
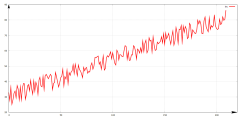
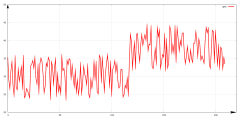
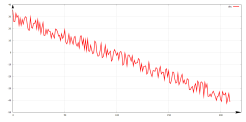
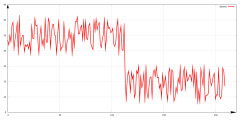
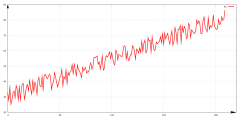
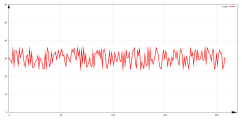
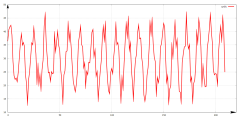
Рассмотрим теперь следующий пример, приведённый в табл. 3.5. Очевидно, что наиболее информативным является **Параметр 5**, так как значения этого параметра (форма временного ряда) различны для каждого класса. Предполагается, что значения параметров **Параметр 1**.. **Параметр 4** должны несколько улучшить точность классификации за счет дополнительной информации.

Таблица 3.4 — Пример шести классов динамических объектов с пятью параметрами

Параметр 1	Параметр 2	Параметр 3	Параметр 4	Параметр 5
Класс 1 – все временные ряды кроме «цикличности»				
				
Класс 2 – все временные ряды кроме «уменьшения значения»				
				
Класс 3 – все временные ряды кроме «резкого спада»				
				
Класс 4 – все временные ряды кроме «увеличения значения»				
				
Класс 5 – все временные ряды кроме «нормы»				
				
Класс 6 – все временные ряды кроме «резкого роста»				
				

Итак, в нашем случае исходные данные для задачи диагностики – это набор динамических объектов (динамических ситуаций). Представление для динамических объектов введено в главе 1, табл. 1.2. Пример набора ситуаций, ис-

Таблица 3.5 — Пример шести классов динамических объектов с пятью параметрами. «Параметр 5» – наиболее информативный.

Параметр 1	Параметр 2	Параметр 3	Параметр 4	Параметр 5
Класс 1 – все временные ряды кроме «цикличности»				
				
Класс 2 – все временные ряды кроме «уменьшения значения»				
				
Класс 3 – все временные ряды кроме «резкого спада»				
				
Класс 4 – все временные ряды кроме «увеличения значения»				
				
Класс 5 – все временные ряды кроме «нормы»				
				
Класс 6 – все временные ряды кроме «резкого роста»				
				

пользуемых как исходные данные для задачи диагностики, приведён в табл. 3.6. Предположим, что для контроля за состоянием сложного технического объекта используются $q = 3$ датчика. Зададим некоторое $r = t^*$ – эта величина определя-

ет максимальную длину временных рядов, которую будем рассматривать. Назовем набор временных рядов, значения которых получены с каждого из датчиков за период времени t^*, ситуацией на объекте. Пример ситуации для данного случая приведен в табл. 3.1. Задача усложняется тем, что время развития ситуации

Таблица 3.6 — Описание ситуаций на объекте для случая 3 датчиков. Время для принятия решения меньше *t**

	t	0	1	2	3	4	5	6	7	8	9	КС
Сит1	$Датчик_1$	-1.07	-0.13	0.85	0.96	0.81	0.84	-0.08	-1.01	-0.90	-1.13	
	$Датчик_2$	-0.72	-0.70	1.25	1.23	1.27	0.03	-0.76	-0.71	-0.71	-0.74	CY
	$Датчик_3$	-0.94	-0.84	1.06	0.97	1.01	1.04	-0.35	-0.92	-0.83	-0.80	
Сит2	$Датчик_1$	-0.56	-0.62	-0.19	0.64	1.45	1.39	-0.69	-0.61	-0.66	-0.62	
	$Датчик_2$	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99	BE
	$Датчик_3$	-0.54	-0.44	-0.28	0.75	1.61	0.40	-0.45	-0.53	-0.38	-0.61	
Сит3	$Датчик_1$	-1.03	0.39	0.97	0.82	0.84	-0.63	-0.92	-1.06			
	$Датчик_2$	-0.73	0.10	1.23	1.27	-0.15	-0.68	-0.71	-0.81			CY
	$Датчик_3$	-0.94	-0.04	0.95	1.04	1.01	-0.86	-0.88	-0.71			
Сит4	$Датчик_1$	-0.45	1.05	1.25	0.61	-0.35	-0.50	-0.39	-0.27	-0.89	-0.28	
	$Датчик_2$	-0.68	-0.67	1.63	1.07	0.69	0.01	-0.59	-0.70	-0.64	-0.53	FU
	$Датчик_3$	-1.01	0.50	1.35	0.89	0.33	0.18	-0.34	-0.75	-0.98	-0.65	
Сит5	$Датчик_1$	-0.72	-0.70	1.25	1.23	1.27	0.03	-0.76	-0.71	-0.71	-0.74	
	$Датчик_2$	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99	CY
	$Датчик_3$	-0.68	-0.67	1.63	1.07	0.69	0.01	-0.59	-0.70	-0.64	-0.53	

на объекте может быть различным. В табл. 3.6 приведён пример, когда ситуации рассматриваются на временном интервале длиной $t^* = 10$, тогда как ситуация *Сит3* развивается за время $\hat{t} = 8, \hat{t} < t^*$. В таком случае время принятия решения меньше времени, на котором рассматриваются ситуации.

Рассмотрим теперь возможность явно ввести время как один из параметров в описание состояния сложного технического объекта. Расширим признаковое описание объектов – введем понятие «время» как один из атрибутов, используемых явно при построении дерева решений. Будем использовать дискретное время: $t = 0, 1, 2, \ldots$.

Вернемся к описанию модели системы, приведенной в главе 1. Расширим табл. 3.6, в которой приведён набор ситуаций, на основе который надо построить модель, еще одним параметром – крайним сроком принятия решения Tm. В

Таблица 3.7 — Набор ситуаций на объекте

	t	0	1	2	3	4	5	6	7	8	9	КС	Tm
Сит1	Датчик$_1$	-1.07	-0.13	0.85	0.96	0.81	0.84	-0.08	-1.01	-0.90	-1.13	CY	9
	Датчик$_2$	-0.72	-0.70	1.25	1.23	1.27	0.03	-0.76	-0.71	-0.71	-0.74		
	Датчик$_3$	-0.94	-0.84	1.06	0.97	1.01	1.04	-0.35	-0.92	-0.83	-0.80		
Сит2	Датчик$_1$	-0.56	-0.62	-0.19	0.64	1.45	1.39	-0.69	-0.61	-0.66	-0.62	BE	9
	Датчик$_2$	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99		
	Датчик$_3$	-0.54	-0.44	-0.28	0.75	1.61	0.40	-0.45	-0.53	-0.38	-0.61		
Сит3	Датчик$_1$	-1.03	0.39	0.97	0.82	0.84	-0.63	-0.92	-1.06			CY	7
	Датчик$_2$	-0.73	0.10	1.23	1.27	-0.15	-0.68	-0.71	-0.81				
	Датчик$_3$	-0.94	-0.04	0.95	1.04	1.01	-0.86	-0.88	-0.71				
Сит4	Датчик$_1$	-0.45	1.05	1.25	0.61	-0.35	-0.50	-0.39	-0.27	-0.89	-0.28	FU	9
	Датчик$_2$	-0.68	-0.67	1.63	1.07	0.69	0.01	-0.59	-0.70	-0.64	-0.53		
	Датчик$_3$	-1.01	0.50	1.35	0.89	0.33	0.18	-0.34	-0.75	-0.98	-0.65		
Сит5	Датчик$_1$	-0.72	-0.70	1.25	1.23	1.27	0.03	-0.76	-0.71	-0.71	-0.74	CY	9
	Датчик$_2$	-0.98	-0.91	-0.59	-0.53	0.30	0.80	1.25	1.41	-0.98	-0.99		
	Датчик$_3$	-0.68	-0.67	1.63	1.07	0.69	0.01	-0.59	-0.70	-0.64	-0.53		

табл. 3.7 представлены динамические объекты (динамические ситуации), для которых явно указан крайний срок принятия решения.

Используя способ представления динамических объектов (временных рядов), описанный в главе 2, можно получить символьное представление для динамических объектов из табл. 3.7. После дискретизации временных рядов набор динамических объектов будет выглядеть следующим образом (табл. 3.8).

Одним из способов, удобных для работы с темпоральной или временн**о**й, информацией при обобщении понятий являются темпоральные продукционные правила [108–111]. Однако часто темпоральные продукционные правила трудны для восприятия и интерпретации человеком. Другим способом, который и будет рассмотрен в данной работе, являются темпоральные деревья решений [108; 112].

3.2 Темпоральные деревья решений

Введем теперь понятие темпорального, или временн**о**го, дерева решений T_{temp}.

Неформально темпоральное дерево решений T_{temp} – это дерево, внутренние вершины которого помечены именами атрибутов и временной меткой, а вер-

Таблица 3.8 — Набор ситуаций на объекте - символьное представление

	t	0	1	2	3	4	5	6	7	8	9	КС	Tm
Сит1	Датчик$_1$	B	E	I	I	H	I	E	B	B	B	CY	9
	Датчик$_2$	C	C	I	I	I	F	C	C	C	C		
	Датчик$_3$	B	B	I	I	I	I	D	B	C	C		
Сит2	Датчик$_1$	C	C	E	H	J	J	C	C	C	C	BE	9
	Датчик$_2$	B	B	C	E	G	H	I	J	B	B		
	Датчик$_3$	C	D	D	H	J	G	D	C	D	C		
Сит3	Датчик$_1$	B	G	I	H	I	C	B	B			CY	7
	Датчик$_2$	C	F	I	I	E	C	C	C				
	Датчик$_3$	B	E	I	I	I	B	B	C				
Сит4	Датчик$_1$	D	I	I	H	D	D	D	D	B	D	BE	9
	Датчик$_2$	C	C	J	I	H	F	C	C	C	C		
	Датчик$_3$	B	G	J	I	G	F	D	C	B	C		
Сит5	Датчик$_1$	C	C	I	I	I	F	C	C	C	C	CY	9
	Датчик$_2$	B	B	C	E	G	H	I	J	B	B		
	Датчик$_3$	C	C	J	I	H	F	C	C	C	C		

шины-листья содержат названия классов – в задачах диагностики это обычно вид неисправности и, возможно, предлагаемое в данной ситуации восстановительное действие. Дуги темпорального дерева решений помечены проверками значений атрибутов в определенный момент времени.

Дадим формальное определение [112]. Пусть P – процесс принятия решений, где A – набор возможных решений, $\mathbb{O}$ – набор проверок, которые могут быть проведены (соответствуют параметрам динамического объекта обобщения в определенные моменты времени), $out(o_i) = v_1, .., v_{k_i}$ – возможные результаты проверки $o_i \in \mathbb{O}$ (соответствуют значениям параметров динамического объекта обобщения в определенные моменты времени). Темпоральное дерево решений для P – это помеченная древовидная структура $T_{temp} =< v0_{temp}, V_{temp}, E_{temp}, \Lambda_V, \Lambda_E, \tau >$, где:

- $< v0_{temp}, V_{temp}, E_{temp} >$ – древовидная структура с корнем $v0_{temp}$, набором вершин V_{temp}, и набором дуг $E_{temp} \subset V_{temp} \times V_{temp}$; $V_{temp} = V_{temp\ I} \cup V_{temp\ L}$: V_{temp} разделено на множество внутренних вершин $V_{temp\ I}$ и множество вершин-листьев дерева $V_{temp\ L}$;
- Λ_V – маркирующая функция, определенная на V_{temp};
- Λ_E – маркирующая функция, определенная на E_{temp};

– если $v \in V_{temp\ I}$, то $\Lambda_V(v) \in \mathbb{O}$ – каждая внутренняя вершина дерева помечена названием проверки;
– если $(v, c) \in E_{temp}$, то $\Lambda_E(v, c) \in out(\Lambda_V(v))$ – дуга из v в c помечена одним из возможных результатов проверки, связанной с вершиной v;
– более того, если $(v, c1), (v, c2) \in E_{temp}$ и $\Lambda_E((v, c1)) = \Lambda_E((v, c2))$, то $c1 = c2$ и для каждого $n \in out(\Lambda_V(v))$ существует с такая, что $(v, c) \in E_{temp}$ и $\Lambda_E((v, c)) = n$ – из вершины v выходит в точности одна дуга, соответствующая каждому возможному результату проверки $\Lambda_V(v)$;
– если $l \in V_{temp\ L}$, то $\Lambda_V(l) \in A$ – каждый лист дерева помечен одним из возможных решений;
– $\tau(v)$ – временн**а**я метка;
– * дополнительно может присутствовать следующее ограничение: если $v' \in V_{temp\ I}$ и существует v такая, что $(v, v') \in E_{temp}$, то $\tau(v') \geq \tau(v)$ – неубывание временн**о**й метки при следовании от корня дерева к листьям.

Таким образом, темпоральное дерево решений – это взвешенный ориентированный граф $T_{temp} = (V_{temp}, E_{temp})$. В множестве вершин V_{temp} выделена вершина $v0_{temp} \in V_{temp}$ – корень дерева. Все вершины разделены на два класса: $V_{temp\ I} \subset V_{temp}$ – множество внутренних вершины (узлов) дерева; $V_{temp\ I}$ включает в себя такие вершины, из которых выходят дуги; $V_{temp\ L} \subset V_{temp}$ – множество внешних, конечных, вершин дерева (листьев); $V_{temp\ L}$ включает в себя такие вершины, из которых дуги не выходят; $V_{temp\ I}$ и $V_{temp\ L}$ образуют разбиение множества вершин V_{temp} темпорального дерева решений T_{temp}: $V_{temp\ I} \cap V_{temp\ L} = \emptyset, V_{temp\ I} \cup V_{temp\ L} = V_{temp}$.

Внутренние вершины $V_{temp\ I}$ дерева взвешены (помечены) названием проверки и временн**о**й меткой, определяющей, когда надо эту проверку производить.

Вершины-листья $V_{temp\ L}$ взвешены (помечены) названием или номером ситуации из $C_n \cup C_f$ и, в более сложном случае – предлагаемым восстановительным действием, если ситуация относится к классу C_f.

Каждая дуга e темпорального дерева решений взвешена результатом проверки, проводимой в вершине, из которой она исходит.

Таким образом, основными отличиями темпоральных деревьев решений от обычных деревьев решений является наличие метки времени в каждом внутреннем узле дерева. Проверка значения атрибута во внутреннем узле дерева про-

изводится только в том случае, если момент времени, которым помечен набор значений датчиков, совпадает с временной меткой в этом узле.

3.3 Алгоритмы построения темпоральных деревьев решений

Дополнительная метка времени в каждом узле темпорального дерева решений приводит к существенным отличиям в представлении данных для алгоритма построения темпорального дерева решений по сравнению с алгоритмом, построения обычных деревьев решений, и к существенным различиям в самих алгоритмах.

Появление временной переменной и необходимость учитывать крайние сроки принятия решения об отнесении ситуации к определенному классу приводят к тому, что алгоритм построения дерева решений – теперь уже темпорального – будет несколько модифицирован. Общая схема алгоритма построения темпорального дерева решений приведена в табл. 3.9.

На вход алгоритма подаются:

1. таблица с ситуациями;
2. наблюдения в виде множества пар «датчик, временная метка»;
3. модель восстановительных действий (при наличии).

3.3.1 Алгоритм *«CPD»*

В работе [112] было дано формальное определение темпоральных деревьев решений и был предложен базовый алгоритм построения темпоральных деревьев решений (назовем его *«CPD»* как сокращение от фамилий авторов L. Console, C. Picardi, D. Dupre).

Алгоритм *«CPD»* представляет интерес как один из первых алгоритмов, в котором реализована процедура построения темпорального дерева решений. Рассмотрим далее подробно характерные особенности этого алгоритма. Этот алгоритм предлагалось использовать для бортовой, или онлайн-диагностики, в связи с чем исходным ограничением на темпоральное дерево решений было неубывание временных меток при движении от корня дерева к листьям. Помимо этого предлагалось использовать определенную модель «восстановительных действий», позволяющих при обнаружении неисправности в работе объекта произвести некоторое управляющее воздействие, которое должно по возможности вернуть объект или систему в корректное состояние. В общем случае восстано-

Таблица 3.9 — Псевдокод алгоритма – построение темпорального дерева решений

Алгоритм Построение_темпорального_дерева_решений
(S: Таблица с ситуациями,
O: Наблюдения,
M: Модель восстановительных действий)
Результат: Темпоральное дерево решений T_{temp}
НАЧАЛО
Если для всех ситуаций из S восстановительные действия совпадают, то вернуть Лист(S, M)
Пусть D – минимальный крайний срок для ситуаций из S.
Если ситуации из S неразличимы на основе показаний датчиков c меткой времени $t \leq D$, то вернуть Лист(S, M).

Выбрать наблюдение $< s ,t' >$, которое будет проверяться в данном узле дерева.*

Пусть $s_1^*, s_2^*, .., s_n^*$ – различающиеся показания датчика s^* в момент времени t', а S_j^*, $j = 1, 2, .., n$ – подмножества ситуаций из S, состоящие из ситуаций с показанием s_j^* датчика s^* в момент времени t'.
Вернуть темпоральное дерево решений T_{temp} с корнем, помеченным выбранным наблюдением $< s^*,t' >$, и дугами, помеченными $s_1^*, s_2^*, .., s_n^*$, соединяющими корень соответственно с деревьями:
Построение_темпорального_дерева_решений(S_1^*, $O\backslash\{< s^*,t' >\}$, M)
Построение_темпорального_дерева_решений(S_2^*, $O\backslash\{< s^*,t' >\}$, M)
...
Построение_темпорального_дерева_решений(S_n^*, $O\backslash\{< s^*,t' >\}$, M)
КОНЕЦ

вительное действие имеет некоторую «стоимость», которая выражается в том, что функциональность объекта или системы уменьшается – например, уменьшение скорости или полная остановка автомобиля, отключение каких-либо модулей системы и т. п. Поэтому рассматривается функция ожидаемой стоимости темпорального дерева решений [112] (табл. 3.10) – это ожидаемая стоимость восстановительного действия, выбранного при помощи темпорального дерева решений, относительно распределения вероятностей неисправностей.

В связи с этим построение каждого узла темпорального дерева решений состоит из двух этапов – на первом шаге принимается во внимание стоимость,

Таблица 3.10 — Ожидаемая стоимость темпорального дерева решений

$\chi(node)$ = стоимость действий в $node$, если $node$ – лист дерева; $\chi(node) = \sum_{c \in V_{next}} P$(перейти из $node$ в c) $* \chi(c)$, если $node$ – внутренний узел.

Таблица 3.11 — Ситуации для построения темпорального дерева решений

	Датчик$_1$								Датчик$_2$								Датчик$_3$								Д	Tm
t	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7		
Сит1	n	n	n	n	h	h			l	v	v	v	v	v			n	n	n	l	l	l			a	5
Сит2	h	h	h						h	n	n						n	n	n						b	2
Сит3	n	n	n	n	h	h	h	h	l	l	l	l	v	v	v	v	n	n	n	l	l	l	v	v	b	7
Сит4	n	n	n	h	h	h	h		l	l	l	l	l	v	v		n	n	h	h	h	h	h		c	6
Сит5	h	h	h	h					h	n	n	n					n	n	n	l					c	3
Сит6	n	n	n	h	h	h			l	v	v	z	z				n	n	n	l	l	v			d	5
Сит7	h	h	h	h	h	h			l	l	n	n	l	v			n	n	n	l	l	v			b	5
Сит8	h	h	h	h	h	h			h	h	n	n	l	l			n	n	n	l	v	z			c	5

выбираются только те наблюдения, смогут обеспечить минимальную ожидаемую стоимость дерева. На втором из них выбирается наиболее информативное наблюдение.

3.3.2 Пример работы алгоритма *«CPD»*

Рассмотрим работу алгоритма «CPD» на примере. Пусть задано обучающее множество следующего вида (табл. 3.11), где каждой строке соответствует некоторая ситуация $Сит_i$, которая определяется: показаниями датчиков $Датчик_j, j = 1,2,3$ в моменты времени *t=0, 1,..7* (здесь z, n, l, v, h соответствуют качественным показаниям датчиков: n – норма, l и h соответственно низкое и высокое значения, v – очень низкое значение, z – нуль); восстановительным действием $Д_i$ для каждой ситуации, обозначенным как a, b, c, d и соответствующим некоторым управляющим действиям, которые надо произвести в случае обнаружения соответствующей ситуации; крайним сроком $К_i$ выполнения соответствующего действия.

Поясним работу алгоритма «CPD» по построению темпорального дерева решений на примере данных из табл. 3.11. В корне дерева необходимо разместить проверку: пару вида $< s_i, t' >$, где s_i – датчик, t' – момент времени. Опре-

деляем t_{up} – максимальное время наблюдения, до наступления которого можно не принимать никаких решений. Из табл. 3.11 находим минимальный крайний срок для все ситуаций $t_{up} = 2$. Анализируем поведение объекта в наблюдаемых ситуациях в моменты времени $t = 0, 1, 2$. Для каждого t выполняем разбиение ситуаций на классы по принципу совпадения значений атрибутов (значениями атрибутов являются показания датчиков h, l, n, v, z). По формулам из табл. 3.10 для полученных разбиений вычисляются оценки стоимости выбора управляющего действия; выбирается разбиение с наименьшей оценкой стоимости. В примере было выбрано разбиение *{{Cum1}, {Cum6}, {Cum2, Cum5}, {Cum7}, {Cum8}, {Cum3}, {Cum4}}*, полученное в момент времени $t = 1$. На основе данного разбиения выполняется вычисление информативности наблюдений [35] и выбирается наблюдение с наилучшей оценкой. В нашем случае это $s2$ – наблюдения, основанные на показаниях *Датчик*$_2$ в момент времени $t = 1$.

Рисунок 3.7 — Дерево решений, построенное с использованием алгоритма CPD

Дальнейшие шаги алгоритма позволяют достроить и уточнить дерево решений. Так, вершина, к которой ведет дуга с меткой h, является листом: ей сопоставлена единственная ситуация *Cum8* с единственно возможным управляющим действием c. Вершина, к которой ведет дуга с меткой v, не является конечной; две ситуации, связанные с данной вершиной *(Cum1, Cum6)*, требуют различных управляющих действий (a, d), поэтому в данном узле вводится дополнительная проверка. Вершина, к которой ведет дуга с меткой n, связана с ситуациями *Cum2* и *Cum5*. В данном случае эта вершина конечная, в ней возможны два управляющих действия: b, c; проблема в том, что показания датчиков в ситуациях 2 и 5 полностью совпадают в моменты времени $t = 0, 1, 2$, а при $t = 2$ решение

по выбору действия обязательно должно быть принято. В связи с этим вершина помечается сразу двумя действиями - b и c.

Пример темпорального дерева решений, построенного с использованием алгоритма *«CPD»*, приведен на рис. 3.7.

3.3.3 Алгоритм *«Темпоральный ID3»*

В работе предлагается оригинальный алгоритм (назовем его «Темпоральным ID3»), который является расширением алгоритма ID3 [35], учитывающим фактор времени. Псевдокод алгоритма представлен в табл. 3.12. По сравнению с алгоритмом «CPD», также позволяющим учитывать фактор времени, на темпоральное дерево решений не накладывается никаких ограничений по временным меткам в узлах, однако не учитывается стоимость восстановительных действий при выборе наблюдения на каждом шаге.

Таблица 3.12 — Псевдокод алгоритма «Темпоральный ID3»

Алгоритм «Темпоральный_ID3» (S: таблица с ситуациями,
O: наблюдения)
Результат: Темпоральное дерево решений T_{temp}
НАЧАЛО
Если для всех ситуаций из S классы ситуаций совпадают,
то вернуть Лист(S)
Если множество наблюдений O пусто, то вернуть Лист(S)
Пусть D – минимальный крайний срок для ситуаций из S
Если ситуации из S неразличимы на основе показаний датчиков
с меткой времени $t \leq D$, то вернуть Лист(S)
$< s^*, t' >=$ Выбор_наблюдения_для_разбиения_Темпоральный_$ID3(S, O)$
Пусть $s_1^*, s_2^*, .., s_n^*$ – различающиеся показания датчика s^* в момент
времени t', а $S_j^*, j = 1, 2, .., n$ – подмножества ситуаций из S, состоящие
из ситуаций с показанием s_j^* датчика s^* в момент времени t'.
Вернуть темпоральное дерево решений T_{temp} с корнем, помеченным
выбранным наблюдением $< s^*, t' >$, и дугами, помеченными $s_1^*, s_2^*, .., s_n^*$,
соединяющими корень соответственно с деревьями
«Темпоральный_ID3»($S_1^*, O \backslash \{< s^*, t' >\}$)
«Темпоральный_ID3»($S_2^*, O \backslash \{< s^*, t' >\}$)
...
«Темпоральный_ID3»($S_n^*, O \backslash \{< s^*, t' >\}$)
КОНЕЦ

При выборе наблюдения для разбиения используется критерий «прирост информативности» Куинлана [35]. Величина $Gain(< s,t >, S) = Info(S) - Info(< s,t >, S)$ показывает количество информации, которое мы получаем благодаря наблюдению $< s, t >$. Алгоритм использует эту величину для оценки информативности наблюдения при построении дерева решений, что позволяет получать деревья минимальной высоты [38]. Процедура выбора наблюдения с использованием данного критерия представлена в табл. 3.13. Так как ограничение на неубывание временных меток при движении от корня дерева к листьям отсутствует, при построении темпорального дерева решений показания датчиков будем рассматривать как обычные атрибуты [108] – например, показание датчика s_1 в момент времени $t = 0$ будет атрибутом $< s1, 0 >$, в момент времени $t = 1$ – атрибутом $< s1, 1 >$ и т. д. Алгоритм строит такое дерево, в котором с каждым узлом ассоциирован атрибут, являющийся наиболее информативным среди всех атрибутов, еще не рассмотренных на пути от корня дерева.

Таблица 3.13 — Псевдокод алгоритма – выбор наблюдения для Темпорального ID3

Алгоритм Выбор_наблюдения_для_разбиения_Темпоральный_ID3 (S: Таблица с ситуациями, O: Наблюдения) Результат: $o*$ - наиболее информативное наблюдение **НАЧАЛО** Для всех наблюдений $< s,t >$ из O вычисляем количество информации, которое получаем благодаря этому наблюдению: $Gain(< s,t >, S) = Info(S) - Info(< s,t >, S)$, где $Info(S)$ – энтропия для ситуаций из S (распределение восстановительных действий) $Info(< s,t >, S)$ – взвешенное среднее информации, необходимой для идентификации класса ситуации в каждом подмножестве, полученном при разбиении множества ситуаций из S на основе значений $< s,t >$ Вернуть $< s*, t' >$ – наблюдение с наибольшим значением $Gain(< s, t >, S)$ **КОНЕЦ**

При выборе наблюдения для разбиения следует рассматривать только те наблюдения, для которых временные метки не превосходят крайнего срока для рассматриваемой в данный момент времени таблицы ситуаций S.

3.3.4 Вычислительная сложность алгоритма *«Темпоральный ID3»*

Рассчитаем вычислительную сложность алгоритма *«Темпоральный ID3»* по аналогии с алгоритмом *ID3* [38].

Размер областей определения для всех атрибутов равен размеру используемого алфавита $|A|$. Количество рекурсивных вызовов составит в худшем случае $1 + |A| + |A|^2 + ... + |A|^{k-1}$, где k – количество атрибутов, равное $q * r$ (q – число параметров динамического объекта обобщения, r – длина рассматриваемого временного ряда), общая сложность составит $C(TID3) = \sum_{i=0}^{k} C(i) * |A|^i$, где $C(i)$ – сложность одного шага алгоритма. Трудоемкой операцией является вычисление информативности атрибутов и определение решающего атрибута на каждом шаге, при этом количество альтернатив и размеры подмножества обучающих примеров сужаются с увеличением глубины рекурсивной вложенности. На уровне i они составляют соответственно $k - i$ и $\frac{n}{b^i}$, где n – количество примеров. Для вычисления информативности атрибутов достаточно одного просмотра множества примеров, поэтому общее число операций составит $C(i) = \frac{n}{b^i} * (k - i)$. Подставив это выражение в сумму, определяющую общую сложность алгоритма, получим $C(TID3) = O(k^2 * n) = O((q * r)^2 * n)$.

3.3.5 Пример работы алгоритма *«Темпоральный ID3»*

Рассмотрим теперь подробнее процесс формирования темпорального дерева решений с использованием алгоритма *«Темпоральный ID3»*. Исходные данные, представленные в табл. 3.11, будем рассматривать как массив, в котором имеются 24 информативных атрибута; решающим является атрибут *Д* – выбор управляющего действия. Атрибут K является вспомогательным, на основании значений этого параметра – критического времени принятия решений – будем осуществлять выборку данных из таблицы.

На первом шаге построения темпорального дерева решений надо выбрать ситуации, требующие наиболее быстрой реакции. В примере такие ситуации определяются значением $K = 2$. В соответствии с этим выберем из исходной таблицы атрибуты с временными метками $t = 0, 1, 2$. Полученная таблица будет содержать 9 наблюдений ($s1(0)$, $s1(1)$, $s1(2)$, $s2(0)$, $s2(1)$, $s2(2)$, $s3(0)$, $s3(1)$, $s3(2)$); поиск среди них наиболее информативного для размещения его в корне

дерева решений проводится на основании вычисления оценок прироста информативности [35].

В корне дерева размещается атрибут $s2(1)$, который был выбран как наиболее информативный. Четыре дуги, взвешенные значениями h, n, v, l, ведут к вершинам следующего уровня. Из этих вершин одна является конечной (взвешена действием c), в остальных случаях требуются дополнительные проверки условий. Для формирования ветвей в вершине, связанной с корнем дугой v, необходимо провести новую выборку из таблицы данных. В случае $s2(1) = v$ для ситуаций *Cum1* и *Cum6* требуются разные управляющие действия: a и d; при этом в обеих ситуациях $K = 5$. В соответствии с этим включаем в выборку атрибуты со значениями t от 0 до 5 (за исключением атрибута $s2(1)$), и проводим поиск наиболее информативного атрибута в таблице, содержащей 2 строки (*Cum1, Cum6*) и 17 атрибутов.

Как показано на рис. 3.8, для размещения в узле второго уровня был выбран наиболее информативный атрибут $s1(3)$; ветвление по остальным вершинам выполняется аналогично. Вершина становится листом, если с ней связано единственное восстанавливающее действие, либо рассмотрены все информативные атрибуты.

Целью построения темпоральных деревьев решений является их использование для диагностики сложного технического объекта. Далее будет приведено описание процесса диагностики с использованием темпоральных деревьев решений, проведено моделирование процесса диагностики.

3.4 Моделирование процесса диагностики

Эксперимент по моделированию процесса диагностики разделим на 2 части. Первую часть назовем «апостериорной диагностикой»: для ситуаций известны все значения датчиков на рассматриваемом временном интервале. Эта часть эксперимента соответствует тому, что со сложного технического объекта, работавшего в течение некоторого времени, сняли показания датчиков. Показания можно использовать, чтобы апостериорно, то есть «после опыта», проверить, как построенные темпоральные деревья решений смогут определить корректность или некорректность ситуаций.

Вторую часть эксперимента назовем диагностикой в псевдореальном времени. Эта часть эксперимента соответствует тому, что на объекте установлена

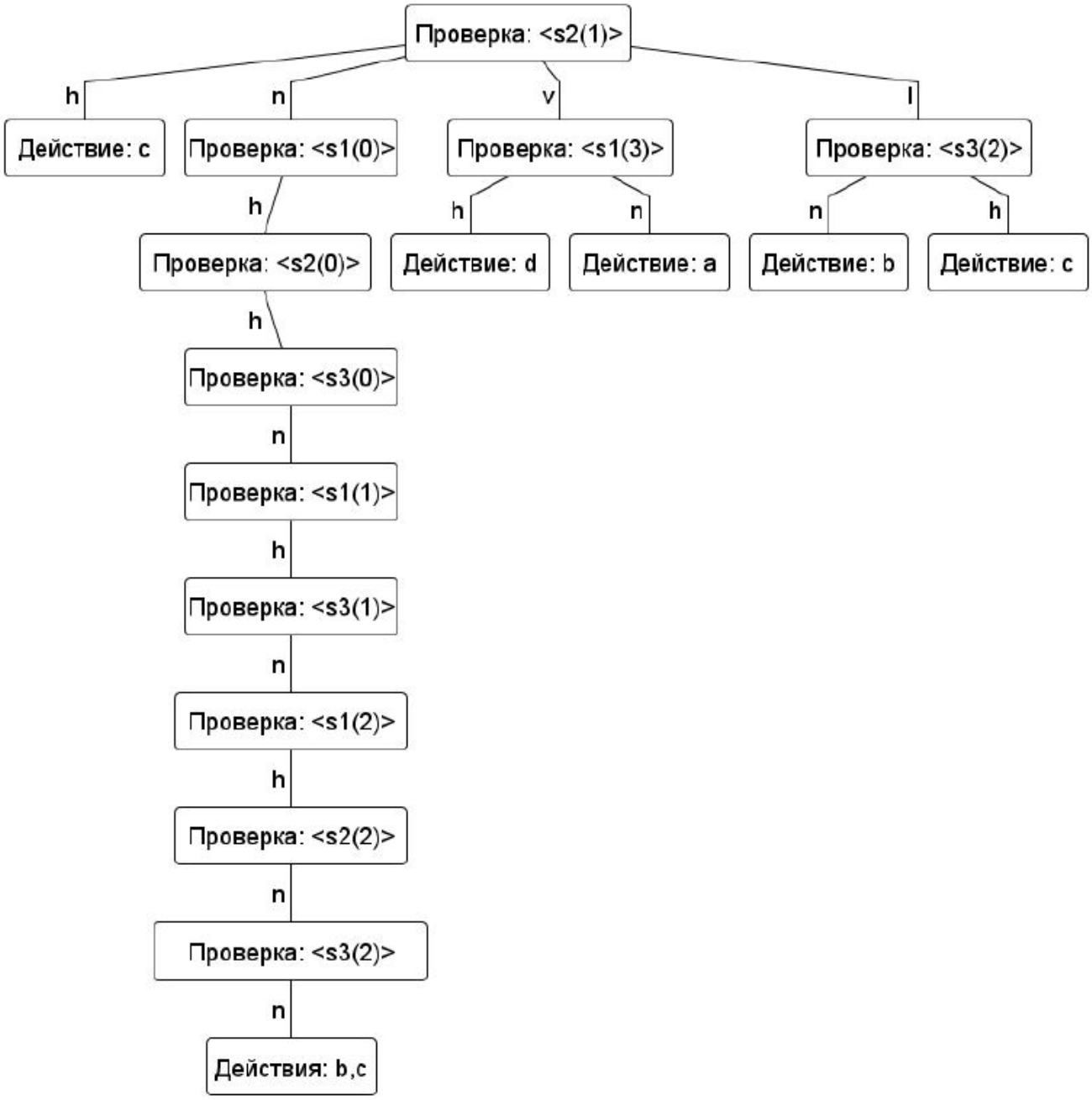


Рисунок 3.8 — Дерево решений, построенное с использованием алгоритма «Темпоральный ID3»

некоторая диагностическая система, которая использует темпоральные деревья решений. На вход этой системы последовательно поступают показания датчиков, установленных на объекте. Диагностическая система, использующая темпоральное дерево решений, должна «на лету» обрабатывать поступающие показания датчиков, определять некорректные ситуации и, в случае, если указаны управляющие воздействия, которые позволят перевести сложный технический объект из состояние «неисправно» в состояние «норма», указать на необходимость их выполнения.

3.4.1 Апостериорная диагностика

Для проведения апостериорной диагностики (то есть такой диагностики, когда уже известны значения датчиков на всем рассматриваемом временном интервале) достаточно одного темпорального дерева решений. При апостериорной диагностике известно множество ситуаций, темпоральное дерево решений должно определить некорректные ситуации и выдать рекомендации по управляющему воздействию, которое могло бы перевести сложный технический объект из состояния «неисправно» в состояние «норма». Ситуации задаются такой же таблицей наблюдений, как и при построении темпорального дерева решений.

3.4.2 Диагностика в псевдореальном времени

В нашей задаче процесс диагностики в псевдореальном времени можно рассматривать как обработку последовательно поступающих значений датчиков системой диагностики и выявление некорректного поведения на основе изначально заданных ситуаций.

При проведении диагностики в псевдореальном времени, когда значения датчиков для последующих временных отсчетов еще неизвестны, одного дерева решений недостаточно.

Кроме того, при построении темпорального дерева решений мы рассматривали временной интервал r, на котором задавались изначальные некорректные ситуации.

Одно темпоральное дерево решений само по себе не может работать с данными, поступающими последовательно от датчиков, установленных на объекте. В связи с этим предлагается использовать многоагентный подход для реализации системы диагностики.

При построении темпорального дерева решений одним из параметров была величина временного интервала для ситуаций r. Поэтому для проведения диагностики в псевдореальном времени потребуется r рабочих агентов, каждый из которых будет использовать темпоральное дерево решений для определения некорректных ситуаций. Работа этих агентов организована следующим образом: первый агент начинает работу в момент времени $t = 0$, второй — в момент времени $t = 1$, и т. д. Последний, $(r - 1)$-ый агент начинает работу в момент времени $t = r - 1$. В каждый момент времени рабочий агент владеет следующей информацией:

– текущая локальная временная метка: момент времени из интервала $[0,t^* - 1]$;
– текущая вершина дерева.

Получая показания датчиков, агент обрабатывает их и увеличивает локальную временную метку. Если локальная временная метка превышает $t^* - 1$, то она сбрасывается в 0, текущим узлом дерева становится корень дерева и диагностика для рабочего агента начинается заново. Кроме того, для организации совместной работы этих агентов потребуется агент-координатор, который будет получать информацию с датчиков, рассылать ее рабочим агентам и получать от них сведения о некорректной работе объекта или системы.

3.5 Выводы к третьей главе

В третьей главе:

1. Рассмотрен подход к формированию динамических объектов для решения задачи обобщения в случае, когда для описания ситуаций на сложном техническом объекте используются показания множества датчиков за определенные промежутки времени.
2. Рассмотрена задача технической диагностики, проводимой на основе анализа состояний сложного объекта. Обоснована необходимость использования методов обобщения динамических объектов при формировании моделей, используемых при решении задачи диагностики. Рассмотрены основные способы представления данных в интеллектуальных системах, решающих задачи диагностики.
3. Для решения поставленной задачи предложен подход с использованием темпоральных деревьев решений, приведено понятие темпорального дерева решений.
4. Предложен новый алгоритм **Темпоральный ID3**, позволяющий построить темпоральное дерево решений для динамических объектов обобщения, которые представляют собой *наборы* временных рядов. На темпоральное дерево решений не накладывается ограничений на временные метки в узлах, временные ряды при этом могут быть различной длины.

Глава 4. Программная реализация и результаты моделирования

4.1 Описание реализованного программного комплекса

С целью проверки эффективности предложенных в работе методов был спроектирован и разработан программный комплекс, работающий в операционной системе *Windows*. Архитектура программного комплекса представлена на рис. 4.1.

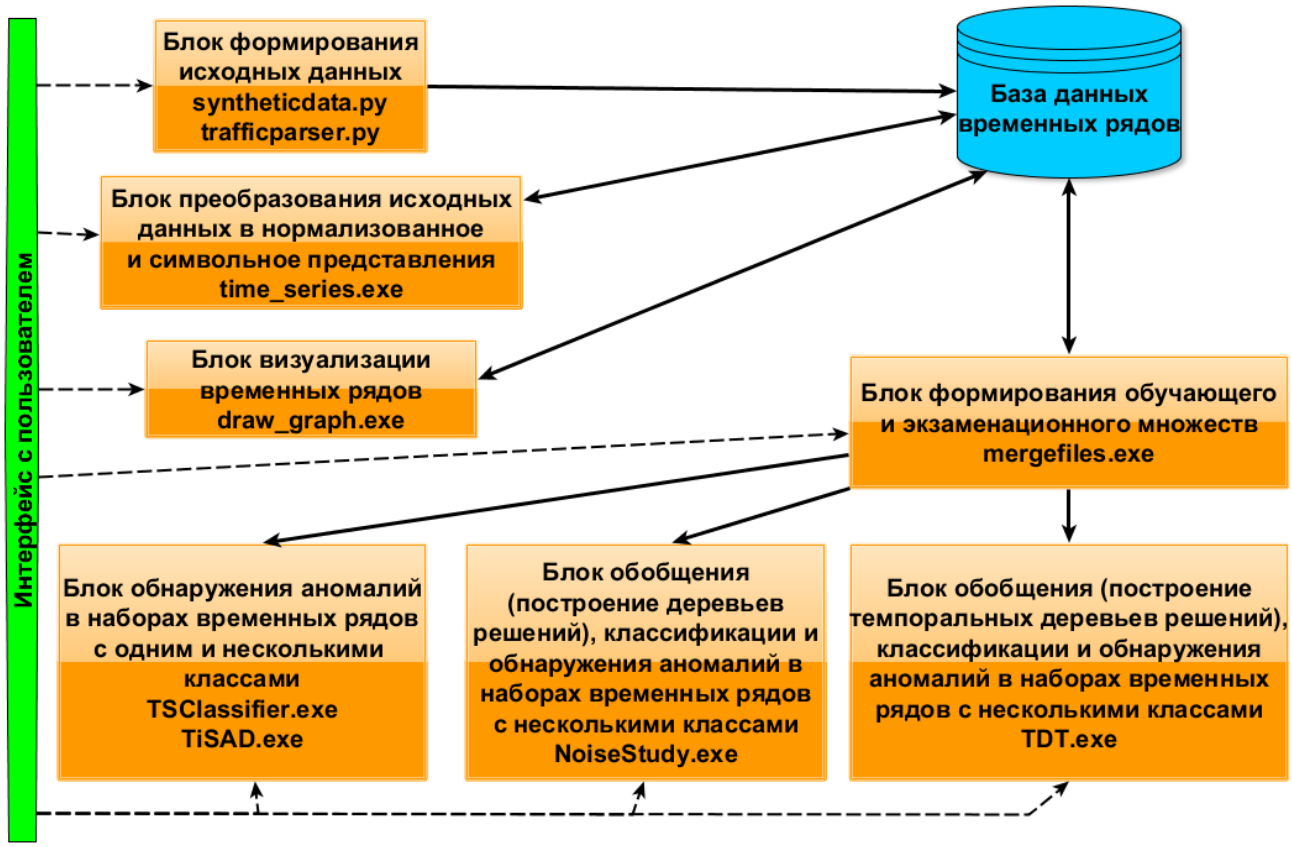


Рисунок 4.1 — Архитектура программного комплекса

Программный комплекс состоит из:

- набора утилит [113] для генерации, предварительной обработки и визуального отображения данных;
- приложения *Noise Study – Изучение шума* (свидетельство о государственной регистрации программы для ЭВМ №2012611444 от 07.02.2012 (Б.1));
- приложения *Time Series Anomaly Detection (TiSAD) – Обнаружение аномалий в наборах временных рядов* (свидетельство о государственной регистрации программы для ЭВМ №2013618587 от 12.09.2013 (Б.2));
- приложения *TSClassifier (Time series classifier)* – консольной версии *TiSAD*;

– приложения *Temporal Decision Trees (TDT) – Темпоральные деревья решений* (свидетельство о государственной регистрации программы для ЭВМ №2013618586 от 12.09.2013(Б.3)).

Набор утилит позволяет:

– сгенерировать наборы данных «цилиндр-колокол-воронка», «контрольные карты» (*syntheticdata.py*) с различными параметрами;
– выделить из предоставленных для эксперимента логов сервера данные для набора «трафик» *trafficparser.py*;
– преобразовать данные из репозиториев UC Irvine Repository и UCR Time Series Classification Archive в используемый в программном комплексе формат;
– на основании заданного размера временного ряда получить нормализованное представление для каждого временного ряда из указанного набора данных (*time_series.exe*);
– на основании заданного размера временного ряда и размера алфавита получить символьное представление для заданных временных рядов с помощью алгоритма Symbolic Aggregate approXimation [58] (*time_series.exe*);
– построить графики исходных и нормализованных временных рядов (*drawgraph.exe*);
– сформировать обучающую и экзаменационную выборки для рассмотренных в работе алгоритмов (*mergefiles.exe, createStudySets.py*).

Приложение *Noise Study–Изучение шума* позволяет:

– строить деревья решения для наборов временных рядов с несколькими классами в символьном представлении;
– проводить классификацию временных рядов;
– на основании набора временных рядов – обучающего множества и набора временных рядов – экзаменационного множества решать задачу обнаружения аномалий в наборах временных рядов с несколькими классами с использованием деревьев решений.

Приложения $TiSAD$ и $TSClassifier$ позволяют:

– на основании набора временных рядов – обучающего множества и набора временных рядов – экзаменационного множества обнаружить аномалии среди временных рядов:

- с использованием алгоритма *«TS-ADEEP»* для обучающего множества с одним классом;
- с использованием алгоритма *«TS-ADEEP-Multi»* для случая обучающего множества с несколькими классами.

Приложение TDT позволяет:

- на основании набора ситуаций, представленных в виде табл. 3.8, строить темпоральные деревья решений с использованием алгоритма *«CPD»* и предложенного в работе алгоритма *«Темпоральный ID3»*;
- моделировать апостериорную диагностику (классифицировать ситуации, возникшие на объекте);
- моделировать диагностику в псевдореальном времени.

Примеры работы программного комплекса приведены в приложении.

Утилиты для предварительной обработки данных реализованы с помощью языка python, расширения py2exe, bat-файлов. Приложение *Noise Study – Изучение шума* реализовано на языке программирования C# [114] в среде Microsoft Visual Studio Express 2010. Приложения $TiSAD$ и $TSClassifier$ реализованы на языке программирования C++ [115] в среде Microsoft Visual Studio Express 2010. Приложение TDT реализовано на языке C# [114] в среде Microsoft Visual Studio Express 2010.

4.2 Результаты обнаружения аномалий для обучающего множества с одним классом

4.2.1 Алгоритм *«TS-ADEEP»*

Для того чтобы определить, насколько хорошо предложенный алгоритм справляется с обнаружением аномалий в наборах временных рядов, было проведено его программное моделирование.

При моделировании на этапе предварительной обработки данных можно, во-первых, снизить размерность временных рядов: или указать «коэффициент сжатия» для временных рядов, что позволяло сократить размерность в заданное число раз; или задать желаемый новый размер временного ряда. Далее, при переходе к символьному представлению, можно задать желаемый размер алфавита. Размер временного ряда и размер используемого алфавита оказывали существенное влияние на точность обнаружения аномалий, что в том числе являлось предметом исследований.

Рассмотрим процесс моделирования на примере набора данных «цилиндр-колокол-воронка». Сначала в качестве обучающего множества TS_STUDY генерируется набор временных рядов, принадлежащих первому из классов, «цилиндр». В качестве экзаменационного множества TS_TEST генерируются временные ряды, принадлежащие всем трем классам — «цилиндр», «колокол», «воронка». Временной ряд ts_test_j является «нормальным», если он принадлежит классу «цилиндр» и «аномалией», если принадлежит классу «колокол» или «воронка». Соответственно, алгоритм корректно находит аномалии, если он относит временные ряды класса «колокол» и «воронка» из TS_TEST к аномалиям, а временные ряды класса «цилиндр» аномалиями не считает. При этом были рассмотрены как численное представление временных рядов, так и символьное с разным размером алфавита. Аналогично моделирование проводилось для классов «колокол» и «воронка».

В таблице 4.1, представлены результаты обнаружения аномалий с использованием алгоритма $TS - ADEEP$. Аномалиями являются временные ряды тех классов, которые не вошли в обучающую выборку.

Таблица 4.1 — Точность обнаружения аномалий для различных наборов данных. Символьное представление. Алгоритм $TS - ADEEP$

Класс рядов в обучающем множестве	Точность обнаружения аномалий, %	
	без шума	с шумом
«трафик»		
normal	-	100.00
«цилиндр-колокол-воронка»		
bell	67.00	70.67
cylinder	67.00	76.00
funnel	89.00	90.00
Среднее	74.33	78.89
«контрольные карты»		
cyclic	93.33	93.50
dec	83.33	99.67
downw	99.67	98.67
inc	83.33	98.83

norm	83.33	97.67
upw	98.83	99.83
Среднее	90.30	98.03
«beef»		
1	-	90.00
2	-	80.00
3	-	80.00
4	-	76.67
5	-	80.00
Среднее	-	81.33
«coffee»		
0	-	85.71
1	-	78.57
Среднее	-	82.14
«Face(four)»		
1	-	87.50
2	-	85.23
3	-	92.05
4	-	89.77
Среднее	-	88.64
«Olive oil»		
1	-	83.33
2	-	70.00
3	-	86.67
4	-	86.67
Среднее	-	81.67

Алгоритм TS-ADEEP справляется с задачей обнаружения аномалий в наборах временных рядов с одним классом: показаны высокие результаты (близкие к 100%) на использованных в работе реальных данных (анализ трафика). Для проверки работы алгоритма также использовались искусственные данные: на наборе временных рядов *цилиндр-колокол-воронка* выбор оптимальных пара-

метров представления временных рядов позволяет достичь точности до 90% для класса *funnel* в обучающем множестве, 76% для класса *cylinder*, 70% для класса *bell*. Для набора *контрольные карты* выбор оптимальных параметров представления временных рядов позволяет достичь точности до 99.83% правильно определенных аномалий на некоторых наборах данных. В большинстве случаев точность обнаружения аномалий для «зашумленных» данных выше (в среднем 91.65%), чем для данных без шума (в среднем 84.98%).

Чтобы оценить эффективность алгоритма $TS-ADEEP$, можно исходить из следующего предположения: обнаружение аномалий с помощью алгоритма является по сути отнесением рассматриваемых объектов к одному из классов – «нормальный» или «аномальный», при этом, с одной стороны, задача облегчается тем, что не нужно в точности определить, к какому из «нормальных» или «аномальных» классов (если таковых несколько) относится объект. С другой стороны, этот же факт усложняет задачу тем, что при наличии нескольких «нормальных» или «аномальных» классов этим невозможно воспользоваться, так как алгоритм предназначен для обнаружения аномалий в наборах с единственным классом. Таким образом, сравнение точности обнаружения аномалий с точностью классификации на таких же наборах данных может в некотором приближении позволить оценить эффективность алгоритма.

Для сравнения будем использовать классические алгоритмы:

– метод K ближайших соседей (Knn);
– алгоритм C4_5 [116];
– байесовские сети
– многослойный персептрон, логистическая регрессия (MLP);
– алгоритм Random Forest(RF) [117];
– логистическая регрессия+деревья решений(LMT);
– метод опорных векторов (SVM);

Сравнение результатов для предложенного алгоритма *«TS-ADEEP»* приведено в таблице 4.2.

Как видно из таблицы, на двух из пяти рассмотренных наборах данных предложенный алгоритм *«TS-ADEEP»* показал результаты лучше, чем остальные алгоритмы.

В среднем на рассмотренных наборах данныъ точность обнаружения аномалий с помощью алгоритма *«TS-ADEEP»* выше, чем у 4 из 7 сравниваемых с

Таблица 4.2 — Точность классификации временных рядов классическими алгоритмами [118] и точность обнаружения аномалий в наборах временных рядов с одним классом алгоритмом *«TS-ADEEP»*

	Knn	NB	C4_5	MLP	RF	LMT	SVM	*TS-ADEEP (среднее)*
Coffee	75.00	67.86	57.14	96.43	75.00	**100.00**	96.43	82.14
CBF	85.00	**89.67**	67.33	85.33	83.56	77.00	87.67	78.89
Olive oil	76.67	76.67	73.33	**86.67**	**86.67**	83.33	**86.67**	81.67
CC	88.00	96.00	81.00	91.33	86.00	92.00	92.33	**98.03**
Beef	60.00	50.00	56.67	73.33	50.00	80.00	66.67	**81.33**
Среднее	76.93 (5)	76.04 (7)	67.09 (8)	**86.61 (1)**	76.25 (6)	86.47 (2)	85.95 (3)	84.41 (4)

ним алгоритмов. В связи с этим можно говорить об эффективности алгоритма *«TS-ADEEP»* обнаружения аномалий в наборах временных рядов с одним классом.

4.3 Результаты обнаружения аномалий для обучающего множества с несколькими классами

4.3.1 Алгоритм *«TS-ADEEP-Multi»*

Для того чтобы определить, насколько хорошо предложенный алгоритм справляется с обнаружением аномалий в наборах временных рядов, было проведено его программное моделирование. При этом были рассмотрены как численное представление временных рядов различной размерности, так и символьное с разным размером алфавита.

Для временных рядов «цилиндр-колокол-воронка» и «трафик» в качестве обучающего множества использовались различные возможные комбинации из двух классов. Для временных рядов «контрольные карты» рассматривались все возможные комбинации из двух, трех, четырех и пяти классов.

В таблице 4.3 представлены результаты обнаружения аномалий с использованием алгоритма $TS-ADEEP-Multi$ для некоторых наборов данных. В скобках для алгоритма $TS-ADEEP-Multi$ указано число классов в обуча-

ющем множестве. Аномалиями являются временные ряды тех классов, которые не вошли в обучающую выборку.

Таблица 4.3 — Точность обнаружения аномалий для различных наборов данных. Символьное представление. Алгоритм $TS - ADEEP - Multi$

Классы рядов в обучающем множестве	Точность обнаружения аномалий, %	
	без шума	с шумом
«цилиндр-колокол-воронка» (2)		
bell, cylinder	76.33	79.33
cylinder, funnel	88.33	85.33
bell, funnel	69.00	69.00
Среднее	77.89	77.89
«трафик» (2)		
normal, normal_1	-	100.00
«контрольные карты» (2)		
cyclic, dec	76.67	92.33
cyclic, downw	92.33	91.17
dec, downw	100.00	100.00
cyclic, inc	76.67	92.00
dec, inc	66.67	97.00
downw, inc	83.00	97.33
cyclic, norm	100.00	100.00
dec, norm	66.67	85.17
downw, norm	83.00	85.00
inc, norm	66.67	83.83
cyclic, upw	79.83	86.50
dec, upw	93.33	98.67
downw, upw	97.00	97.16
inc, upw	100.00	99.67
norm, upw	82.17	88.17
Среднее	84.27	92.33
«контрольные карты» (3)		
cyclic, dec, downw	93.33	91.17

cyclic, dec, inc	60.00	89.67
cyclic, downw, inc	75.67	89.50
dec, downw, inc	83.33	96.00
cyclic, dec, norm	83.33	97.83
cyclic, downw, norm	98.50	98.17
dec, downw, norm	98.33	96.50
cyclic, inc, norm	83.33	97.00
dec, inc, norm	81.33	82.00
downw, inc, norm	66.33	81.83
cyclic, dec, upw	63.17	85.00
cyclic, downw, upw	77.67	84.83
dec, downw, upw	92.83	95.17
cyclic, inc, upw	91.50	91.50
dec, inc, upw	83.33	95.83
downw, inc, upw	98.50	97.50
cyclic, norm, upw	96.50	98.83
dec, norm, upw	65.50	84.17
downw, norm, upw	80.33	84.67
inc, norm, upw	94.33	92.50
Среднее	83.36	91.48
«контрольные карты» (4)		
cyclic, dec, downw, inc	76.67	89.17
cyclic, dec, downw, norm	100.00	98.50
cyclic, dec, inc, norm	66.67	93.00
cyclic, downw, inc, norm	81.83	81.00
dec, downw, inc, norm	81.67	80.33
cyclic, dec, downw, upw	75.00	84.00
cyclic, dec, inc, upw	74.83	88.50
cyclic, downw, inc, upw	90.83	88.17
dec, downw, inc, upw	98.17	97.50
cyclic, dec, norm, upw	79.83	97.67
cyclic, downw, norm, upw	94.33	95.33
dec, downw, norm, upw	77.83	94.33

cyclic, inc, norm, upw	100.00	98.83
dec, inc, norm, upw	77.67	80.83
downw, inc, norm, upw	81.83	82.33
Среднее	83.81	89.97
«контрольные карты» (5)		
cyclic, dec, downw, inc, norm	83.33	89.33
cyclic, dec, downw, inc, upw	91.00	89.33
cyclic, dec, downw, norm, upw	91.67	94.67
cyclic, dec, inc, norm, upw	83.33	92.33
cyclic, downw, inc, norm, upw	98.50	96.17
dec, downw, inc, norm, upw	92.67	91.33
Среднее	90.08	92.19
FaceFour (2)		
1, 2	-	72.73
1, 3	-	72.73
2, 3	-	79.55
1, 4	-	88.63
2, 4	-	86.36
3, 4	-	70.45
Среднее		78.41
FaceFour (3)		
1, 2, 3	-	76.14
1, 2, 4	-	90.91
1, 3, 4	-	76.14
2, 3, 4	-	86.36
Среднее		82.39

Алгоритм TS-ADEEP-Multi справляется с задачей обнаружения аномалий в наборах временных рядов с несколькими классами. Показаны высокие результаты (близкие к 100%) на использованных в работе реальных данных (анализ трафика) и в некоторых экспериментах для наборов данных «контрольные карты». Для других наборов данных точность обнаружения аномалий в среднем

Таблица 4.4 — Точность классификации временных рядов классическими алгоритмами [118] и точность обнаружения аномалий в наборах временных рядов с несколькими классами алгоритмом *«TS-ADEEP-Multi»*

	Knn	NB	C4_5	MLP	RF	LMT	SVM	*TS-ADEEP-Multi (среднее)*
CBF	85.00	**89.67**	67.33	85.33	83.56	77.00	87.67	77.89
CC	88.00	**96.00**	81.00	91.33	86.00	92.00	92.33	91.49
Face(Four)	87.50	84.09	71.59	87.50	78.41	77.27	**88.64**	80.40
Среднее	86.83 (4)	**89.92 (1)**	73.31 (8)	88.05 (3)	82.66 (6)	82.09 (7)	89.55 (2)	83.26 (5)

составляет от 77.89 до 98.50%, что свидетельствует об эффективности алгоритма TS-ADEEP-Multi.

На зашумленных данных точность обнаружения аномалий в среднем выше, максимальная разница составляет 8.12%. Поскольку реальные данные в большинстве случаев содержат шум, это является преимуществом предложенного в работе алгоритма.

Для оценки эффективности алгоритма $TS-ADEEP-Multi$, можно исходить из тех же предположений, что и с алгоритмом $TS-ADEEP$. Обнаружение аномалий с помощью алгоритма является по сути отнесением рассматриваемых объектов к одному из «нормальных» или «аномальных» классов, при этом не все из «нормальных» или «аномальных» классов известны заранее. Это с одной стороны, облегчает задачу тем, что разделение нужно провести на меньшее число классов, чем их существует в рассматриваемых наборах данных; с другой стороны, этот же факт усложняет задачу тем, что невозможно воспользоваться информацией, касающейся *всех* классов рассматриваемой предметной области. Оценка эффективности алгоритма $TS-ADEEP-Multi$ приведена в таблице 4.4.

Алгоритм *«TS-ADEEP-Multi»* обнаружения аномалий в наборах временных рядов с несколькими классами показывает удовлетворительные результаты по сравнению с классическими алгоритмами классификации, что свидетельствует о его эффективности.

4.4 Результаты моделирования процесса диагностики с использованием темпоральных деревьев решений

4.4.1 Частный случай

Первым этапом эксперимента по исследованию результатов классификации динамических объектов обобщения была оценка точности классификации временных рядов. Временные ряды как частный случай динамических объектов обобщения были подробно рассмотрены в главе 2.

С помощью разработанного программного комплекса был проведен ряд экспериментов по классификации временных рядов с использованием аппарата темпоральных деревьев решений. Использовались два алгоритма построения темпоральных деревьев решений: *«CPD»* и предложенный в работе алгоритм *«Темпоральный ID3»*. Сравнение предложенного алгоритма проводилось как с классическими алгоритмами классификации:

- метод K ближайших соседей (Knn);
- алгоритм C4_5 [116];
- байесовские сети [119] (NB);
- многослойный персептрон, логистическая регрессия (MLP);
- алгоритм Random Forest (RF) [117];
- логистическая регрессия+деревья решений(LMT);
- метод опорных векторов (SVM);

так и со специализированными алгоритмами, созданными для работы со временными рядами:

- метод ближайшего соседа (1-NN ED);
- 1-NN Best Warping Window Dynamic Time Warping (r) (1-NN BWW DTW (r)) [120];
- 1-NN Dynamic Time Warping, no Warping Window(1-NN DTW no WW) [120].

Метод ближайшего соседа считаем специализированным, так как он позволяет без какой-либо предварительной обработки данных работать с временными рядами как с векторами в N-мерном еквлидовом пространстве.

Результаты моделирования и их сравнение со специализированными алгоритмами классификации временных рядов [64] приведены в таблице 4.5.

Таблица 4.5 — Точность классификации динамических объектов (%). Частный случай, динамический объект обобщения – временной ряд. Сравнение со специализированными алгоритмами.

	1-NN ED	1-NN BWW DTW (r)	1-NN DTW, no WW	***TID3***
wafer	**99.50**	**99.50**	98.00	98.64
Coffee	**100.00**	**100.00**	**100.00**	96.43
CBF	85.20	99.60	**99.70**	95.67
Olive oil	86.60	86.60	83.30	**93.30**
Trace	76.00	99.00	**100.00**	88.00
CC	88.00	98.30	**99.30**	83.33
ECG200	**88.00**	**88.00**	77.00	79.00
Lightning2	75.40	**86.90**	**86.90**	77.05
yoga	83.00	**84.50**	83.60	69.56
Lightning7	57.50	71.20	**72.60**	65.75
Beef	**66.60**	**66.60**	63.30	60.00
СРЕДНЕЕ	82.35 (4)	**89.11 (1)**	87.61 (2)	82.43 (3)

Сравнение показывает, что на рассмотренных наборах данных точность классификации с использованием алгоритма Темпоральный ID3 в среднем на 5.18-6.68% ниже, чем точность классификации специализированными алгоритмами, созданными для работы с временными рядами, но чуть выше (на 0.07%), чем точность классификации с использованием метода ближайшего соседа. Тем не менее на одном из наборов данных – *«Olive oil»* – алгоритм *«Темпоральный ID3»* показал точность классификации выше, чем рассмотренные специализированные алгоритмы.

Результаты моделирования и их сравнение с классическими алгоритмами [118] приведено в таблице 4.6.

На использованных наборах данных предоженный алгоритм *«Темпоральный ID3»* показывает точность классификации временных рядов в среднем на 0.34-12.76% процентов выше, чем классические алгоритмы классификации. При этом на трех наборах данных – *CBF, Olive oil, Trace* – *«Темпоральный ID3»* по точности превосходит все сравниваемые алгоритмы.

Также предложенный в работе алгоритм *«Темпоральный ID3»* показывает результаты лучше, чем наиболее близкий к нему алгоритм *«CPD»* [112].

Таблица 4.6 — Точность классификации динамических объектов (%). Частный случай, динамический объект обобщения – временной ряд. Сравнение с классическими алгоритмами.

	Knn	NB	C4_5	MLP	RF	LMT	SVM	CPD	***TID3***
wafer	**99.40**	70.83	98.20	96.28	99.32	98.09	95.96	97.12	98.64
Coffee	75.00	67.86	57.14	96.43	75.00	**100.00**	96.43	96.43	96.43
CBF	85.00	89.67	67.33	85.33	83.56	77.00	87.67	92.55	**95.67**
Olive oil	76.67	76.67	73.33	86.67	86.67	83.33	86.67	56.67	**93.30**
Trace	82.00	80.00	74.00	77.00	81.00	76.00	73.00	83.00	**88.00**
CC	88.00	**96.00**	81.00	91.33	86.00	92.00	92.33	60.67	83.33
ECG200	**89.00**	77.00	72.00	84.00	81.00	82.00	81.00	73.00	79.00
Lightning2	**80.33**	67.21	62.30	73.77	78.69	63.93	72.13	75.41	77.05
yoga	**83.30**	54.23	69.90	74.50	77.87	71.87	63.07	58.76	69.56
Lightning7	63.01	64.38	54.79	64.38	56.16	64.38	**71.23**	47.95	65.75
Beef	60.00	50.00	56.67	73.33	50.00	**80.00**	66.67	46.67	60.00
СРЕДНЕЕ	80.16 (5)	72.17 (7)	69.67 (9)	82.09 (2)	77.75 (6)	80.78 (3)	80.56 (4)	71.66 (8)	**82.43 (1)**

4.4.2 Общий случай

В общем случае, как описано в главе 3, динамический объект обобщения представляет собой набор временных рядов. С помощью разработанного программного комплекса был проведен ряд экспериментов по классификации таких объектов (ситуаций) для проверки предположения о том, что отнесение ситуаций к различным классам можно провести успешнее, если для описания такой ситуации используется несколько временных рядов. Для сравнения использовались два алгоритма, допускающих работу с динамическими объектами обобщения, содержащими несколько параметров: *«CPD»* и *«Темпоральный ID3»*. Проверка проводилась на соответствующих наборах данных, описанных ранее – *«ECG»*, *«wafer»*, *«Activities of daily living»*, а также на специально сформированных обучающих и экзаменационных выборках, составленных из временных рядов, относящихся к наборам данных «цилиндр-колокол-воронка», «контрольные карты».

Сначала были рассмотрены случаи, когда все динамически объекты обобщения относятся точно к двум классам. С помощью алгоритмов *«CPD»* и *«Темпоральный ID3»* мы исследуем, насколько успешно можно различать такие объ-

екты. В обучающей выборке *«ECG»* каждый объект описан двумя признаками (двумя временными рядами), *«Activities of daily living»* – тремя признаками, *«wafer»* – шестью признаками.

Результаты классификации для набора данных *«Activities of daily living»* представлены в таблице 4.7, для набора данных *«ECG»* – в табл. 4.8 и на графике рис. 4.2, для набора данных *«wafer»* – в табл. 4.9 и на графике рис. 4.3.

Таблица 4.7 — Точность классификации (%), набор данных *Activities of daily living*. Классификация по одному и нескольким признакам. *CPD* – алгоритм *«CPD»* [112]; *TID3* – алгоритм *«Темпоральный ID3»*

Число признаков	Число классов	Алгоритм	
		CPD	***TID3***
Классы sitdown_chair, standup_chair обучающее множество – 20% исходного набора данных			
1 (ось X)	2	99.38	99.38
1 (ось Y)	2	56.17	61.72
1 (ось Z)	2	97.53	97.53
Среднее	2	84.36	86.21
3	**2**	**99.38**	**99.38**
Классы sitdown_chair, standup_chair обучающее множество – 50% исходного набора данных			
1 (ось X)	2	99.01	99.01
1 (ось Y)	2	58.42	59.41
1 (ось Z)	2	98.02	98.02
Среднее	2	85.15	85.48
3	**2**	**99.01**	**99.01**
Классы getup_bed, liedown_bed			
1 (ось X)	2	97.03	97.03
1 (ось Y)	2	67.33	70.30
1 (ось Z)	2	91.09	93.07
Среднее	2	85.15	86.80
3	**2**	**97.03**	**97.03**

Анализ результатов позволяет сделать вывод, что предложенный в работе алгоритм *«Темпоральный ID3»* для случая, когда ситуации описываются несколькими временными рядами, показывает результаты классификации лучше, чем *«CPD»*: на 3% для набора данных *«ECG»*, на 0.49-2.01% для набора данных *«wafer»*. На наборе данных *«Activities of daily living»* результаты обоих алгорит-

Таблица 4.8 — Точность классификации (%) - набор данных ECG. Классификация по одному и нескольким признакам.
CPD – алгоритм *«CPD»* [112]; *TID3* – алгоритм *«Темпоральный ID3»*

Число признаков	Число классов	Алгоритм	
		CPD	***TID3***
1 (1ый)	2	71.00	70.00
1 (2ой)	2	75.00	76.00
Среднее	2	73.00	73.00
2	**2**	**72.00**	**75.00**

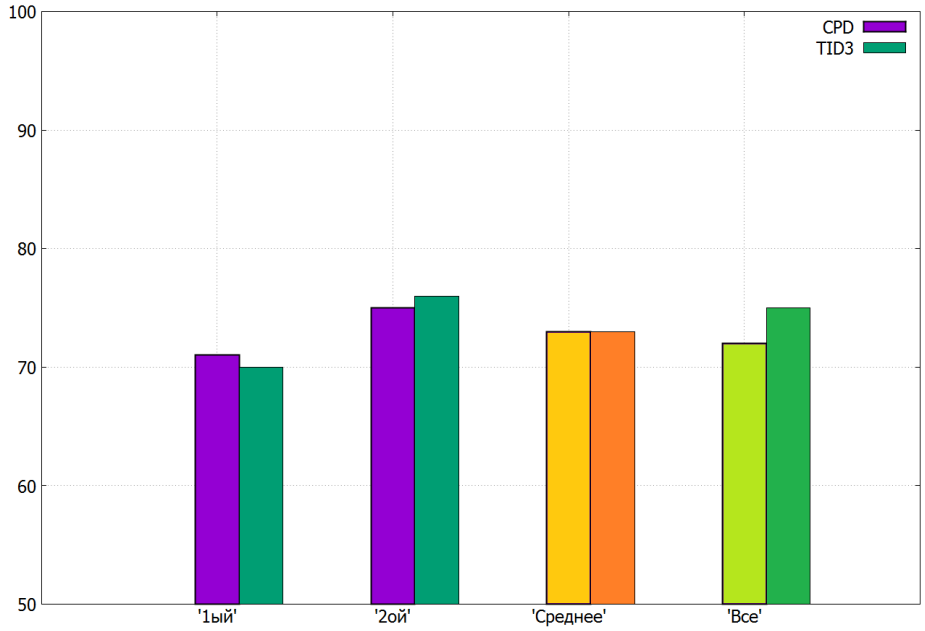


Рисунок 4.2 — Точность классификации (%) - набор данных ECG. Классификация по одному и нескольким признакам. *CPD* – алгоритм *«CPD»* [112]; *TID3* – алгоритм *«Темпоральный ID3»*

мов одинаковые. Также стоит отметить, что классификация с использованием нескольких параметров в большинстве случаев более точная, чем классификация в среднем по одному параметру.

Однако в рассмотренных наборах данных обычно присутствовал один из параметров, который являлся наиболее информативным, в связи с чем использование других параметров для классификации являлось избыточным, а иногда приводило к уменьшению точности класификации. В случае же, если такого параметра нет или он в ходе изучения предметной области еще не обнаружен, рекомендуется использовать все доступные параметры, так как это позволяет

Таблица 4.9 — Точность классификации (%) - набор данных wafer. Классификация по одному и нескольким признакам. *CPD* – алгоритм *«CPD»* [112]; *TID3* – алгоритм *«Темпоральный ID3»*

Число признаков	Число классов	Алгоритм	
		CPD	***TID3***
1 (1ый)	2	81.13	88.17
1 (2ой)	2	84.26	86.90
1 (3ий)	2	77.42	83.19
1 (4ый)	2	94.03	99.02
1 (5ый)	2	87.78	87.98
1 (6ой)	2	91.10	92.77
СРЕДНЕЕ	2	85.95	89.67
2	**2**	**96.09**	**96.58**
6	**2**	**92.47**	**96.48**

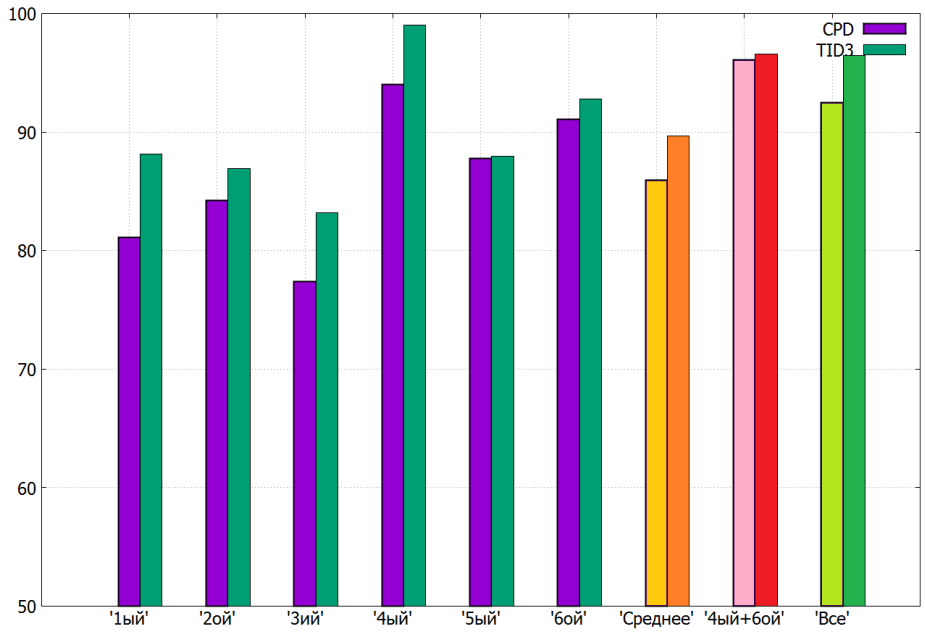


Рисунок 4.3 — Точность классификации (%) – набор данных wafer. Классификация по одному и нескольким признакам. *CPD* – алгоритм *«CPD»* [112]; *TID3* – алгоритм *«Темпоральный ID3»*

получить точность классификации в среднем большую, чем использование какого-либо одного параметра.

В главе 3 был описан принцип, по которому из временных рядов формировались динамические объекты обобщения, представленные несколькими временными рядами (табл. 3.3, 3.4). По такому же принципу были сформированы

другие обучающие и экзаменационные выборки из наборов данных «цилиндр-колокол-воронка» и «контрольные карты». Результаты для этой части эксперимента представлены в таблице 4.10.

Таблица 4.10 — Точность классификации динамических объектов (%). Наборы данных «цилиндр-коколол-воронка» (*CBF*) и «контрольные карты» (*CC*). Общий случай. Несколько признаков. *CPD* – алгоритм *«CPD»* [112]; *TID3* – алгоритм *«Темпоральный ID3»*

Набор данных	Число признаков	Число классов	Алгоритм	
			CPD	***TID3***
CBF	2	3	85.00	**89.00**
CBF	2	9	58.11	**71.44**
CC	2	6	97.50	**99.00**
CC	5	6	98.17	**99.00**

Из таблицы видно, что предложенный в работе алгоритм *«Темпоральный ID3»* на рассмотренных наборах данных показывает более высокую точность классификации, на 0.83-13.33% выше, чем алгоритм *«CPD»*. Кроме того, использование всех признаков (временных рядов) из описания ситуаций действительно позволяет с высокой точностью разделить имеющиеся объекты на соответствующие классы.

Такая ситуация – наличие нескольких динамически изменяющихся параметров, с помощью которых необходимо оценить состояние объекта управления – наиболее характерна для случая, когда речь идет об управлении сложным техническим объектом (примером такого объекта может быть, например, электростанция). При этом лицо, принимающее решения (диспетчер), должен распознать ситуацию на сложном техническом объекте и принять решение о том, что состояние объекта является нормальным или аномальным. В последнем случае крайне полезно отнести реальную ситуацию к определённому классу, в зависимости от типа неисправности. Следовательно, алгоритмы определения аномалий и классификации динамических ситуаций могут быть весьма полезными при их использовании в ИСППР РВ.

4.5 Выводы к четвёртой главе

В четвёртой главе:

1. Представлена архитектура разработанного программного комплекса, реализующего предложенные в работе методы и алгоритмы; приведено описание функционала входящих в комплекс программ.
2. Приведены результаты использования программного комплекса для решения задачи обнаружения аномалий в наборах временных рядов с одним классом (алгоритм $TS - ADEEP$). Проведено сравнение точности обнаружения аномалий с помощью алгоритма $TS-ADEEP$ с классическими алгоритмами классификации. Сделаны выводы об эффективности предложенного алгоритма.
3. Приведены результаты использования программного комплекса для решения задачи обнаружения аномалий в наборах временных рядов с несколькими классами (алгоритм $TS - ADEEP - Multi$). Проведено сравнение точности обнаружения аномалий с помощью алгоритма $TS - ADEEP - Multi$ с классическими алгоритмами классификации. Сделаны выводы об эффективности предложенного алгоритма.
4. Приведены результаты моделирования процесса диагностики с использованием темпоральных деревьев решений для частного случая и проведено сравнение предложенного алгоритма *«Темпоральный ID3»* с другими алгоритмами, решающими аналогичные задачи; сделаны выводы о том, что предложенный алгоритм *«Темпоральный ID3»* сравним со специализированными алгоритмами для классификации временных рядов и превосходит классические алгоритмы классификации в среднем на 0.34-12.76%; на трёх из одиннадцати использованных в работе наборов данных *«Темпоральный ID3»* по точности превосходит все классические алгоритмы.
5. Приведены результаты моделирования процесса диагностики с использованием темпоральных деревьев решений для общего случая и проведено сравнение предложенного алгоритма *«Темпоральный ID3»* с алгоритмом *«CPD»*. В большинстве случаев предложенный в работе алгоритм *«Темпоральный ID3»* на рассмотренных наборах данных показывает более высокую точность классификации, на 0.83-13.33% выше, чем алгоритм *«CPD»*. Сделаны выводы о том, что использование всех

признаков (временных рядов) из описания ситуаций действительно позволяет с высокой точностью разделить имеющиеся объекты на соответствующие классы. В случае, когда ситуации описываются несколькими временными рядами и нет (или не выявлен) единственный наиболее информативный параметр, алгоритм *«Темпоральный ID3»*, используя все доступные параметры, показывает точность классификации в среднем лучше, чем при использовании какого-либо одного параметра.

Заключение

В диссертационной работе получены следующие результаты:

1. Проведён обзор методов представления знаний в современных интеллектуальных системах и рассмотрена проблема работы с данными, явно зависящими от времени – темпоральными данными. Выделены основные категории таких данных, которые могут использоваться в ИСППР реального времени. Введено понятие динамического объекта обобщения – структуры, описывающей динамическое состояние сложной технической системы, одним из параметров которой является время. Дана постановка задачи обобщения для случая, когда исходными данными для обобщения являются динамические объекты.
2. Рассмотрена проблема обнаружения аномалий в случае, когда состояние сложной технической системы представимо временным рядом. Дана постановка задачи обнаружения аномалий в наборах временных рядов с одним и несколькими классами и выполнен обзор существующих методов решения данных задач.
3. На основании анализа подходов к решению задачи обнаружения аномалий в наборах временных рядов предложены методы и разработаны алгоритмы *«TS-ADEEP»*, *«TS-ADEEP-Multi»* обнаружения аномалий для наборов временных рядов с одним и несколькими классами. Рассчитана оценка вычислительной сложности разработанных алгоритмов.
4. Проведён анализ различных способов представления знаний в интеллектуальных системах. Выделен класс динамических объектов, представимых несколькими временными рядами. Для данного типа динамических объектов дана постановка задачи обобщения. Показано, что задача обобщения для динамических объектов может быть использована для решения задач диагностики состояний (ситуаций) в сложных динамических системах. На основании анализа подходов к решению такой задачи выбран подход на основе построения темпоральных деревьев решений и приведено формальное описание темпоральных деревьев решений.
5. Проведен обзор методов и алгоритмов построения темпоральных деревьев решений. Предложен новый алгоритм *«Темпоральный ID3»* построения темпоральных деревьев решений, использующий в качестве критерия выбора наблюдений для разбиения величину «прирост инфор-

мативности». Получена оценка вычислительной сложности алгоритма *«Темпоральный ID3»* и показано, что она имеет полиномиальный характер.

6. Для исследования возможностей разработанных методов и алгоритмов был спроектирован и разработан программный комплекс, позволяющий решать задачу обнаружения аномалий для наборов временных рядов с одним и несколькими классами; решать задачу обобщения для динамических объектов, представляющих собой как временные ряды, так и наборы временных рядов. На разработанные программы, являющиеся составными частями реализованного программного комплекса, получены свидетельства о государственной регистрации программ для ЭВМ №2012611444 от 13.12.2011, №2013618587 от 12.09.2013, №2013618586 от 12.09.2013.
7. Для алгоритмов *TS-ADEEP* и *TS-ADEEP-Multi* показано, что на известных наборах данных точность обнаружения аномалий сопоставима с точностью обнаружения аномалий рядом известных алгоритмов (метод опорных векторов, алгоритм *C4_5*, байесовские сети, алгоритм *Random Forest* и др.). Выявлен ряд задач (например, *control chart*, *beef*), для которых алгоритм *TS-ADEEP* показывает результаты, превосходящие результаты, показанные другими алгоритмами.
8. Проведено сравнение точности классификации временных рядов с использованием алгоритма *«Темпоральный ID3»* с известными алгоритмами классификации (метод K ближайших соседей, C4_5, байесовские сети, Random Forest и др.). Показано, что в среднем алгоритм *«Темпоральный ID3»* превосходит такие алгоритмы на 0.34-12.76%. Проведено сравнение со специализированными алгоритмами (метод ближайшего соседа (евклидова метрика), *1-NN Best Warping Window Dynamic Time Warping (r)*, *1-NN Dynamic Time Warping, no Warping Window*, предназначенными для обработки временных рядов. Показано, что *«Темпоральный ID3»* по точности классификации сопоставим с такими алгоритмами.
9. Проведено сравнение точности классификации динамических объектов, представленных наборами временных рядов, с алгоритмом *«CPD»*, наиболее «близким» к алгоритму *«Темпоральный ID3»*. Показано, что

в большинстве случаев алгоритм *«Темпоральный ID3»* превосходит *«CPD»* (в среднем на 0.83-13.33% для различных наборов данных).

10. Результаты эксперимента позволяют сделать вывод об эффективности использования алгоритма *«Темпоральный ID3»* для работы с динамическими объектами, которые представлены наборами временных рядов.

Список литературы

1. ***С. Г. Антипов, М. В. Фомина*. Метод формирования обобщенных понятий с использованием темпоральных деревьев решений** // *Искусственный интеллект и принятие решений*. — 2010. — Т. 2. — С. 64–76.
2. ***Antipov, S.G., Fomina, M.V.* A method for compiling general concepts with the use of temporal decision trees** // *Scientific and Technical Information Processing*. — 2011. — Vol. 38, no. 6. — Pp. 409–419. — URL: http://dx.doi.org/10.3103/S0147688211060025.
3. ***С. Г. Антипов, М. В. Фомина*. Проблема обнаружения аномалий в наборах временных рядов** // *Программные продукты и системы*. — 2012. — Т. 2. — С. 78–82.
4. ***Vagin, V.N., Fomina, M.V., Antipov, S.G.* Modeling of algorithms of inductive concept formation in “noisy” databases** // *Automatic Documentation and Mathematical Linguistics*. — 2013. — Vol. 47, no. 4. — Pp. 151–161. — URL: http://dx.doi.org/10.3103/S0005105513040055.
5. ***Вагин В.Н., Фомина М.В., Антипов С.Г.* Моделирование алгоритмов индуктивного формирования понятий в «зашумленных» базах данных** // *Научно-техническая информация. Информационные процессы и системы*. — 2013. — Т. 7. — С. 20–32.
6. *С. Г. Антипов, В. Н. Вагин*. Исследование алгоритмов обобщения понятий при наличии шума во входных данных // Труды XVI Международной научно-технической конференции «Информационные средства и технологии». — Т. 2. — Москва: Издательский дом МЭИ, 2008. — С. 8–13.
7. *С. Г. Антипов*. Концептуальная схема базы данных для исследования алгоритмов индуктивного формирования понятий при наличии шума в данных // Радиоэлектроника, электротехника и энергетика. Пятнадцатая международная научно-техническая конференция студентов и аспирантов: тезисы докладов в 3 частях. — Т. 1. — Москва: Издательский дом МЭИ, 2009. — С. 275–276.
8. *С. Г. Антипов*. Использование темпоральных деревьев решений для задач диагностики // XII Московская международная телекоммуникационная конференция студентов и молодых ученых "Молодёжь и Наука". Тезисы докладов в 2-х частях. — Т. 2. — Москва: МИФИ, 2009. — С. 138–139.

9. *С. Г. Антипов*. Методы представления темпоральной информации в базах знаний // Радиоэлектроника, электротехника и энергетика. Шестнадцатая международная научно-техническая конференция студентов и аспирантов: тезисы докладов в 3 частях. — Т. 1. — Москва: Издательский дом МЭИ, 2010. — С. 350–351.
10. *С. Г. Антипов, М. В. Фомина*. Метод формирования обобщенных понятий с использованием темпоральных деревьев решений // Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010 (20-24 сентября 2010 г., г. Тверь, Россия): Труды конференции. — Т. 2. — Москва: Физматлит, 2010. — С. 40–46.
11. *С. Г. Антипов, В. Н. Вагин*. Проблема обнаружения аномалий в наборах временных рядов: обучающие множества с одним и несколькими классами // Труды конгресса по интеллектуальным системам и информационным технологиям IS&IT'12. — Т. 1. — М.: Физматлит, 2012. — С. 293–300.
12. *С. Г. Антипов, В. Н. Вагин*. Проблема обнаружения аномалий в наборах временных рядов // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ 2014 (24-27 сентября 2014 г., г. Казань, Россия): Труды конференции. — Т. 2. — Казань: Изд-во РИЦ «Школа», 2014. — С. 195–203.
13. *С. Г. Антипов, Л. А. Старостина, М. В. Фомина*. Проблема формирования обобщенных понятий при наличии шума в решающих атрибутах // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ 2014 (24-27 сентября 2014 г., г. Казань, Россия): Труды конференции. — Т. 2. — Казань: Изд-во РИЦ «Школа», 2014. — С. 204–212.
14. *С. Г. Антипов, М. В. Фомина*. Метод формирования обобщенных понятий с использованием темпоральных деревьев решений // *Интеллектуальные системы. Коллективная монография. Выпуск четвертый*. — 2010. — С. 277–296.
15. *А. Н. Аверкин, М. Г. Гаазе-Рапопорт, Д. А. Поспелов*. Толковый словарь по искусственному интеллекту. — М.: Радио и связь, 1992. — Р. 256.
16. *Ю.М. Арский, В. К. Финн*. Принципы конструирования интеллектуальных систем // *Информационные технологии и вычислительные системы*. — 2008. — № 4. — С. 4–37.

17. *А. И. Башмаков, И. А. Башмаков*. Интеллектуальные информационные технологии: Учеб. пособие. — Москва: Изд. МГТУ им. Н. Э. Баумана, 2005. — 304 с.
18. *Еремеев АП., Троицкий В.В.* Модели представления временных зависимостей в интеллектуальных системах поддержки принятия решений // *Известия РАН. Теория и системы управления.* — 2003. — № 5. — С. 75–88.
19. *Thomas G. Dietterich, Ryszard S. Michalski*. Inductive Learning of Structural Descriptions: Evaluation Criteria and Comparative Review of Selected Methods // *Artif. Intell.* — 1981. — Vol. 16, no. 3. — Pp. 257–294.
20. *Джозеф Гарратано, Гари Райли*. Экспертные системы: принципы разработки и программирование. — 4 изд. — Москва: ООО «И.Д. Вильямс», 2007. — 1152 с.
21. *Д. А. Поспелов*. Моделирование рассуждений. Опыт анализа мыслительных актов. — Москва: Радио и связь, 1989. — 184 с.
22. *Newell, Allen, Simon, Herbert A.* Computer science as empirical inquiry: symbols and search // *Commun. ACM*. — 1976. — mar. — Vol. 19, no. 3. — Pp. 113–126. — URL: http://doi.acm.org/10.1145/360018.360022.
23. *Collins, A.M., Quillian, M.R.* Retrieval time from semantic memory // *Journal of Verbal Learning and Verbal Behavior*. — 1969. — Vol. 8. — Pp. 240–248.
24. *В. Н. Вагин*. Дедукция и обобщение в системах принятия решений. — Москва: Наука, 1988. — 384 pp.
25. *F. Sowa John*. Encyclopedia of Artificial Intelligence. — 2nd edition. — New York, NY, USA: John Wiley & Sons, Inc., 1992.
26. *Minsky, Marvin*. A Framework for Representing Knowledge: Tech. Rep. : Cambridge, MA, USA, 1974.
27. *Axelrod, Robert M.* Structure of decision: the cognitive maps of political elites / edited by Robert Axelrod; written under the auspices of the Institute of International Studies, University of California (Berkeley) and the Institute of Public Policy Studies, the University Michigan. — Princeton University Press, Princeton, N.J., 1976. — P. 404.
28. *TOLMAN E. C.* Cognitive maps in rats and men. // *Psychological review*. — 1948. — jul. — Vol. 55, no. 4. — Pp. 189–208.
29. *Ю. М. Плотинский*. Модели социальных процессов. — 2 изд. — М.: Логос, 2001. — 296 с.

30. *Джордж Ф. Люгер*. Искусственный интеллект: стратегии и методы решения сложных проблем. — 4-е издание изд. — Москва: Издательский дом «Вильямс», 2003. — 864 с.
31. *McCulloch, Warren S., Pitts, Walter*. A logical calculus of the ideas immanent in nervous activity // *Bulletin of Mathematical Biology*. — 1943. — dec. — Vol. 5, no. 4. — Pp. 115–133. — URL: http://dx.doi.org/10.1007/BF02478259.
32. *Colmerauer, Alain, Roussel, Philippe*. History of programming languages—II / Ed. by Thomas J. Bergin, Jr., Richard G. Gibson, Jr. — New York, NY, USA: ACM, 1996. — Pp. 331–367. — URL: http://doi.acm.org/10.1145/234286.1057820.
33. *В. К. Финн*. Об интеллектуальном анализе данных // *Новости искусственного интеллекта*. — 2004. — № 3. — С. 3–18.
34. *А. С. Нариньяни*. НЕ-факторы: state of art // Научная сессия МИФИ. — Т. 3. — 2004. — С. 26–30.
35. *Quinlan J. R.* Induction of decision trees // *Machine learning*. — 1986. — Vol. 1. — Pp. 81–106.
36. *О. И. Ларичев, А. В. Петровский*. Системы поддержки принятия решений. Современное состояние и перспективы их развития // *Итоги науки и техники. Сер. Техническая кибернетика*. — 1987. — Т. 21. — С. 131–164.
37. *Н. П. Кириллов*. Признаки класса и определения понятия «технические системы» // *Авиакосмическое приборостроение*. — 2009. — С. 1–6.
38. Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин, Е. Ю. Головина, А. А. Загорянская, М. В. Фомина; Под ред. В. Н. Вагина, Д. А. Поспелова. — Издание второе, исправленное и дополненное изд. — Москва: Физматлит, 2008. — 712 с.
39. *Dietterich, T. G., Michalski, R. S.* Machine Learning: An Artificial Intelligence Approach / Ed. by Michalski, R. S., Carbonell, J., Mitchell, T. M. — Palo Alto: Tioga, 1983. — Pp. 41–82.
40. *Langley P.* Machine learning as an experimental science // *Machine learning*. — 1988. — Vol. 3. — Pp. 5–8.
41. *Т. А. Гаврилова, К. Р. Червинская*. Извлечение и структурирование знаний для экспертных систем. — Москва: Радио и связь, 1992. — 200 с.
42. *Т. А. Гаврилова, В. Ф. Хорошевский*. Базы знаний интеллектуальных систем. — СПб: Питер, 2000. — 384 с.

43. *Е. Ю. Кандрашина, Л. В. Литвинцева, Д. А. Поспелов*. Представление знаний о времени и пространстве в интеллектуальных системах / Под ред. Д. А. Поспелова. — Москва: Наука, 1989. — 328 с.
44. *И. С. Лосев, В. В. Максимов*. Моделирование обучения и поведения. — М.: Наука, 1975. — С. 185–209.
45. *Д. А. Поспелов*. Из истории искусственного интеллекта: история искусственного интеллекта до середины 80-х годов // *Новости искусственного интеллекта*. — 1994. — № 4. — С. 70–90.
46. *Roddick John F., Spiliopoulou Myra*. A bibliography of temporal, spatial and spatio-temporal data mining research // *SIGKDD Explor. Newsl.* — 1999. — jun. — Vol. 1, no. 1. — Pp. 34–38. — URL: http://doi.acm.org/10.1145/846170.846173.
47. *Weiqiang Lin, Mehmet A. Orgun, Graham J. Williams*. An Overview of Temporal Data Mining // Proceedings of the 1st Australasian Data Mining Workshop. — 2002.
48. *C. M. Antunes, A. L. Oliveira*. Temporal data mining: an overview // Eleventh International Workshop on the Principles of Diagnosis. — 2001.
49. A Survey of Temporal Knowledge Discovery Paradigms and Methods / John F. Roddick, Ieee Computer Society, Myra Spiliopoulou, Ieee Computer Society // *IEEE Transactions on Knowledge and Data Engineering*. — 2002. — Vol. 14. — Pp. 750–767.
50. *Allen James F.* Maintaining knowledge about temporal intervals // *Communications of the ACM*. — 1983. — Vol. 26, no. 11. — Pp. 832–843. — URL: http://doi.acm.org/10.1145/182.358434.
51. *Н. Г. Ярушкина, Т. В. Афанасьева, И. Г. Перфильева*. Интеллектуальный анализ временных рядов. — Ульяновск: УлГТУ, 2010. — 315 с.
52. *Т. Андерсон*. Статистический анализ временных рядов / Под ред. Ю. К. Беляева. — Москва: Мир, 1976. — 756 с.
53. *Eamonn Keogh, Shruti Kasetty*. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration // *Data Mining and Knowledge Discovery*. — 2003. — Vol. 7. — Pp. 1–72.
54. Real-Time Classification of Streaming Sensor Data / Shashwati Kasetty, Candice, Stafford, Gregory P. Walker et al. // Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence - Volume 01.

— ICTAI '08. — Washington, DC, USA: IEEE Computer Society, 2008. — Pp. 149–156. — URL: http://dx.doi.org/10.1109/ICTAI.2008.143.

55. Locally adaptive dimensionality reduction for indexing large time series databases / Kaushik Chakrabarti, Eamonn J. Keogh, Sharad Mehrotra, Michael J. Pazzani // *ACM Trans. Database Syst.* — 2002. — Vol. 27, no. 2. — Pp. 188–228.
56. *Kin-pong Chan, Ada Wai-Chee Fu*. Efficient Time Series Matching by Wavelets // ICDE. — 1999. — Pp. 126–133.
57. *Christos Faloutsos, M. Ranganathan, Yannis Manolopoulos*. Fast Subsequence Matching in Time-Series Databases // SIGMOD Conference. — 1994. — Pp. 419–429.
58. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms / Jessica Lin, Eamonn Keogh, Stefano Lonardi, Bill Chiu // In Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. — 2003. — Pp. 2–11.
59. Experiencing SAX: a novel symbolic representation of time series / Jessica Lin, Eamonn Keogh, Li Wei, Stefano Lonardi // *Data Min. Knowl. Discov.* — 2007. — oct. — Vol. 15, no. 2. — Pp. 107–144. — URL: http://dx.doi.org/10.1007/s10618-007-0064-z.
60. *Shieh Jin, Keogh Eamonn*. iSAX: indexing and mining terabyte sized time series // Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. — KDD '08. — New York, NY, USA: ACM, 2008. — Pp. 623–631. — URL: http://doi.acm.org/10.1145/1401890.1401966.
61. *Keogh Eamonn, Lin Jessica, Fu Ada*. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence // Proceedings of the Fifth IEEE International Conference on Data Mining. — ICDM '05. — Washington, DC, USA: IEEE Computer Society, 2005. — Pp. 226–233. — URL: http://dx.doi.org/10.1109/ICDM.2005.79.
62. Time-series bitmaps: a practical visualization tool for working with large time series databases / Nitin Kumar, Nishanth Lolla, Eamonn Keogh et al. // SIAM 2005 Data Mining Conference. — SIAM, 2005. — Pp. 531–535.
63. *Keogh, E., Zhu, Q., Hu, B. et al.* — 2011. — URL: www.cs.ucr.edu/~eamonn/time_series_data.

64. *Chen, Yanping, Keogh, Eamonn, Hu, Bing et al.* The UCR Time Series Classification Archive. — 2015. — July. — www.cs.ucr.edu/~eamonn/time_series_data/.
65. *Lichman M.* UCI Machine Learning Repository. — 2013. — URL: http://archive.ics.uci.edu/ml.
66. *Varun Chandola, Arindam Banerjee, Vipin Kumar*. Anomaly Detection - A Survey // *ACM Computing Surveys*. — 2009. — Vol. 41(3). — Pp. 1–72.
67. *Anderson James P.* Computer security threat monitoring and surveillance: Tech. Rep. : James P. Anderson Co., Fort Washington, Pa., 1980.
68. *Larose Daniel T.* Discovering Knowledge in Data: An Introduction to Data Mining. — Wiley-Interscience, 2004. — 222 pp.
69. *Shyam Boriah, Varun Chandola, Vipin Kumar*. Similarity measures for categorical data: A comparative evaluation // In Proceedings of the eighth SIAM International Conference on Data Mining. — 2008. — Pp. 243–254.
70. *Jain Anil K., Dubes Richard C.* Algorithms for clustering data. — Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.
71. *А. Н. Колмогоров*. Три подхода к определению понятия «количество информации» // *Проблемы передачи информации*. — 1965. — Т. 1. — С. 3–11. — URL: http://mi.mathnet.ru/ppi68.
72. *Shannon C. E.* A mathematical theory of communication // *Bell System Technical Journal*. — 1948. — Vol. 27. — Pp. 379–423, 623–656.
73. Anomaly detection in transportation corridors using manifold embedding. / Agovic, A., Banerjee, A., Ganguly, A. R., Protopopescu, V. // First International Workshop on Knowledge Discovery from Sensor Data. — ACM Press, 2007.
74. *C. Stefano, C. Sansone, M. Vento*. To reject or not to reject: that is the question - an answer in case of neural classifiers // *IEEE Transactions on Systems, Management and Cybernetics*. — 2000. — Vol. 1. — Pp. 84–94.
75. Outlier Detection Using Replicator Neural Networks / Simon Hawkins, Hongxing He, Graham Williams, Rohan Baxter // In Proc. of the Fifth Int. Conf. and Data Warehousing and Knowledge Discovery (DaWaK02. — 2002. — Pp. 170–180.
76. *Barbara, D., Wu, N., Jajodia, S.* Detecting Novel Network Intrusions using Bayes Estimators // Proc. SIAM Intl. Conf. Data Mining. — 2001.

77. *Sebyala, A. A., Olukemi, T., Sacks, L.* Active platform security through intrusion detection using naive bayesian network for anomaly detection // In Proceedings of the 2002 London Communications Symposium. — 2002.
78. Constructing Boosting Algorithms from SVMs: An Application to One-Class Classification / Rätsch, Gunnar, Mika, Sebastian, Schölkopf, Bernhard, Müller, Klaus-Robert // *IEEE Trans. Pattern Anal. Mach. Intell.* — 2002. — sep. — Vol. 24, no. 9. — Pp. 1184–1199. — URL: http://dx.doi.org/10.1109/TPAMI.2002.1033211.
79. Using artificial anomalies to detect unknown and known network intrusions / Fan, W., Miller, M., Stolfo, S. et al. // *Knowl. Inf. Syst.* — 2004. — sep. — Vol. 6, no. 5. — Pp. 507–527. — URL: http://dx.doi.org/10.1007/s10115-003-0132-7.
80. *Agrawal, Rakesh, Srikant, Ramakrishnan*. Mining Sequential Patterns // Proceedings of the Eleventh International Conference on Data Engineering. — ICDE '95. — Washington, DC, USA: IEEE Computer Society, 1995. — Pp. 3–14. — URL: http://dl.acm.org/citation.cfm?id=645480.655281.
81. *Saito Naoki*. Local feature extraction and its application using a library of bases: Ph.D. thesis / Yale University. — 1994.
82. *Kadous Mohammed Waleed*. Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series: Ph.D. thesis / University of New South Wales. — New South Wales, Australia, Australia, 2002. — AAI0806481.
83. *Company Western Electric*. Statistical quality control handbook. — New York, USA: Mack Printing Company, Easton, Pennsylvania, 1958.
84. *D. T. Pham, A. B. Chan*. Control Chart Pattern Recognition using a New Type of Self Organizing Neural Network // *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*. — 1998. — Vol. 212(2). — Pp. 115–127.
85. *Hui-Ping Cheng, Chuen-Sheng Cheng*. Control Chart Pattern Recognition Using Wavelet Analysis and Neural Networks // *Journal of Quality*. — 2009. — Vol. 16. — Pp. 311–321.
86. *Robert T. Olszewski*. Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data: Ph.D. thesis / School of Computer Science, Carnegie Mellon University, Pittsburgh. — 2001.
87. Transformation Based Ensembles for Time Series Classification / A. Bagnall, L. Davis, J. Hills, J. Lines // Proceedings of the 12th SIAM International

Conference on Data Mining (SDM 2012). — 2012. — Pp. 307–319.

88. *Roverso Davide*. Multivariate Temporal Classification By Windowed Wavelet Decomposition And Recurrent Neural Networks // In 3 rd ANS International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface. — 2000.
89. *Roverso Davide*. Neural and Fuzzy Transient Classification Systems: General Techniques and Applications in Nuclear Power Plants // Fuzzy Systems and Soft Computing in Nuclear Engineering / Ed. by Da Ruan. — Physica-Verlag HD, 2000. — Vol. 38 of *Studies in Fuzziness and Soft Computing*. — Pp. 208–234. — URL: http://dx.doi.org/10.1007/978-3-7908-1866-6_10.
90. Genetic Algorithms and Support Vector Machines for Time Series Classification / Eads, Damian, Hill, Daniel, Davis, Sean et al. // Proc. SPIE 4787; Fifth Conference on the Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation; Signal Processing Section; Annual Meeting of SPIE. — 2002. — URL: http://www.zeus.lanl.gov/green/publications/eadsSPIE4787.pdf.
91. URL: http://code.google.com/p/lbimproved.
92. Estimating the Support of a High-Dimensional Distribution / Schölkopf, Bernhard, Platt, John C., Shawe-Taylor, John C. et al. // *Neural Comput.* — 2001. — jul. — Vol. 13, no. 7. — Pp. 1443–1471. — URL: http://dx.doi.org/10.1162/089976601750264965.
93. *Roth Volker*. Kernel Fisher Discriminants for Outlier Detection // *Neural Comput.* — 2006. — apr. — Vol. 18, no. 4. — Pp. 942–960. — URL: http://dx.doi.org/10.1162/089976606775774679.
94. *Andreas Arning, Rakesh Agrawal, Prabhakar Raghavan*. A Linear Method for Deviation Detection in Large Databases // In Proceedings of KDD'1996. — 1996. — Pp. 164–169.
95. *Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana*. Towards parameter-free data mining // Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. — KDD '04. — New York, NY, USA: ACM, 2004. — Pp. 206–215. — URL: http://doi.acm.org/10.1145/1014052.1014077.
96. *Quinlan J. R.* Improved Use of Continuous Attributes in C4.5 // *Machine learning*. — 1996. — Vol. 4. — Pp. 77–90.

97. *Utgoff Paul E.* Incremental Induction of Decision Trees // *Machine learning*. — 1989. — Vol. 4. — Pp. 161–186.
98. Большая советская энциклопедия. Т. 25. Струнино - Тихорецк. / Под ред. А. М. Прохоров. — Издание второе, исправленное и дополненное изд. — Москва, 1976. — 600 с.
99. *Селлерс Ф.* Методы обнаружения ошибок в работе ЭЦВМ, пер. с англ.,. — Москва, 1972.
100. Основы технической диагностики / В. В. Карибский, П. П. Пархоменко, Е. С. Согомонян, В. Ф. Халчев. — Москва: Энергия, 1976.
101. A Spectrum of Definitions for Temporal Model-Based Diagnosis / Vittorio Brusoni, Luca Console, Paolo Terenziani, Daniele Theseider Dupre // *Artificial Intelligence*. — 1998. — Vol. 102. — Pp. 39–79.
102. *Luca Console, Oskar Dressler*. Model-based Diagnosis in the Real World: Lessons Learned and Challenges Remaining // Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. — IJCAI '99. — San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. — Pp. 1393–1400. — URL: http://dl.acm.org/citation.cfm?id=646307.688062.
103. *Оськин П.В.* Исследование и реализация систем поддержки истинности для задач диагностики: Ph.D. thesis / Московский энергетический институт. — 2007.
104. Production system models of learning and development / Ed. by David Klahr, P. Langley, R. Neches. — Cambridge, MA: MIT Press, 1987.
105. *Zadeh L.A.* Fuzzy Sets // *Information Control*. — 1965. — Vol. 8. — Pp. 338–353.
106. *Pawlak Z.* Rough Sets - Theoretical Aspects of Reasoning about Data. — Kluwer Academic, Dordrecht, 1991.
107. *А. В. Куликов*. Исследование и разработка алгоритмов обобщения на основе теории приближенных множеств: Ph.D. thesis / Московский энергетический институт (технический университет). — 2004.
108. *Kamran Karimi, Howard J. Hamilton*. Temporal Rules and Temporal Decision trees: A C4.5 Approach: Tech. Rep. "CS-2001-02": Department of Computer Science, University of Regina, 2001.
109. *Cotofrei Paul, Stoffel Kilian*. Classification Rules + Time = Temporal Rules // Proceedings of the International Conference on Computational Science-Part I. — ICCS '02. — London, UK, UK: Springer-Verlag, 2002. — Pp. 572–581. —

URL: http://dl.acm.org/citation.cfm?id=645457.655326.

110. *Lo David, Khoo Siau-Cheng, Liu Chao*. Mining past-time temporal rules from execution traces // Proceedings of the 2008 international workshop on dynamic analysis: held in conjunction with the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2008). — WODA '08. — New York, NY, USA: ACM, 2008. — Pp. 50–56. — URL: http://doi.acm.org/10.1145/1401827.1401838.
111. *K. Karimi, Howard J. Hamilton*. Generation and Interpretation of Temporal Decision Rules // *International Journal of Computer Information Systems and Industrial Management Applications*. — 2011. — Vol. 3. — Pp. 314–323.
112. *Luca Console, Claudia Picardi, Daniele Theseider Dupre*. Temporal decision trees: model-based diagnosis of dynamic systems on-board // *Journal of Artificial Intelligence Research*. — 2003. — Vol. 19(1). — Pp. 469–512.
113. *Брайан Керниган, Роберт Пайк*. Unix. Программное окружение. — Символ-Плюс, 2003. — С. 416.
114. *Эндрю Троелсен*. Язык программирования C# 2010 и платформа .NET 4.0. — 5-е изд изд. — Москва: Вильямс, 2010. — 1392 с.
115. *Бьерн Страуструп*. Язык программирования C++: Пер. с англ. — Бином: Невский диалект, 2001.
116. *Quinlan J. R.* C4.5: programs for machine learning. — San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
117. *Breiman Leo*. Random Forests // *Machine Learning*. — 2001. — oct. — Vol. 45(1). — Pp. 5–32.
118. URL: http://www.cs.ucr.edu/~eamonn/time_series_data.
119. *Pearl Judea*. A Probabilistic Calculus of Actions // Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence. — UAI'94. — San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. — Pp. 454–462. — URL: http://dl.acm.org/citation.cfm?id=2074394.2074452.
120. Fast Time Series Classification Using Numerosity Reduction / Xi, Xiaopeng, Keogh, Eamonn, Shelton, Christian et al. // Proceedings of the 23rd International Conference on Machine Learning. — ICML '06. — New York, NY, USA: ACM, 2006. — Pp. 1033–1040. — URL: http://doi.acm.org/10.1145/1143844.1143974.

Список рисунков

Список таблиц

Приложение А

Пример работы с программным комплексом

Рассмотрим работу с программным комплексом на примере. Допустим, что требуется провести моделирование для набора данных «цилиндр-колокол-воронка». Утилита *syntheticdata.py* генерирует необходимое число временных рядов, относящихся к классам «цилиндр», «колокол», «воронка». Будем считать, что данная утилита сгенерировала файлы $cyl\_0.txt$, $cyl\_1.txt$, .. для класса «цилиндр», $bel\_0.txt$, $bel\_1.txt$, .. для класса «колокол», $fun\_0.txt$, $fun\_1.txt$, .. для класса «воронка».

Далее необходимо преобразовать исходные данные в нормализованный и символьный (SAX) вид. Для этого нужно сформировать bat-файл следующего вида:

```
process_wparams & (
for %%i in (cyl_, bel_, fun_) do for /l %%j in (0,1,9) do
   process_serie_total cbf/%%i%%j.txt
) & (
%FULLPATH%%MERGEFILEPATH% -d %FULLPATH%scripts/" -m "Sax_" -o "OUT
   "
)
```

В этом файле сначала задаются настройки среды - bat-файл $process\_wparams.bat$. Далее проводится преобразование 10 рядов каждого класса в нормализованное и символьное (SAX) представление, которое выполняется с помощью bat-файла $process\_serie\_total.bat$.

Файл $process\_wparams.bat$ может иметь следующий вид (все пути должны быть корректно указаны пользователем):

```
chcp 1251
rem ТУТ ВСЕ НАСТРОЙКИ
rem Путь к корневой директории проекта
set FULLPATH="e:/dev/asp/timeseries/
rem Путь к исполняемому файлу - преобразование из исходной в PAA и
    SAX
set TSEXEPATH=release/time_series.exe"
rem Папка с данными
rem Кто вызывает это, должен указыват путь относительно этой папки
set DATAPATH=tsdata_git/
rem Утилита для объединения данных
```

```
set MERGEFILEPATH=python/dist/mergefiles.exe"
rem Утилита для рисования графиков
set DRAWGRAPHPATH=python/dist/drawgraph.exe"
rem рисовать/нет графики
set WITHGRAPH=1
```

Файл *process_serie_total.bat* имеет следующий вид:

```
chcp 1251
rem Получение данных для одного файла
rem для поддержки связывания времени выполнения
SetLocal EnableExtensions EnableDelayedExpansion
rem Имя вызываемого файла
set FILENAME=%1
echo %FILENAME%
rem Имя нормализованного файла
set NORM_FILENAME=%FILENAME%
set NORM_FILENAME=%NORM_FILENAME:.txt=_normal.txt%"
rem Полный путь к time_series.exe
set TSEXE=%FULLPATH%%TSEXEPATH%
echo %TSEXE%
rem Полный путь к исполняемому файлу
set FULLDATAPATH=%FULLPATH%%DATAPATH%%FILENAME%"
set NORMDATAPATH=%FULLPATH%%DATAPATH%%NORM_FILENAME%
echo %DATAPATH%
rem Рисуем график исходной последовательности
if %WITHGRAPH% == 1 (
%FULLPATH%%DRAWGRAPHPATH% -d %FULLDATAPATH%
)
set SAVEPAA=1
rem внешний цикл по размеру алфавита
rem (для первого значения построим все варианты PAA, потом PAA мож
   но не сохранять)
rem внутренний цикл по размеру PAA (во сколько раз уменьшаем разме
   рность)
for %%k in (5, 10, 15, 20, 25, 30, 40, 50) do (
  for %%j in (1, 5, 10, 15, 20, 23, 25, 30, 40, 50) do (
  %TSEXE% --pointsinpaa %%j --alphabetsize %%k --timeseriespath %
     FULLDATAPATH% --savepaa !SAVEPAA!
  if %WITHGRAPH% == 1 (
    %FULLPATH%%DRAWGRAPHPATH% -d %NORMDATAPATH% -p "PAA_%%j_points
       .txtgraph"
    )
```

```
  )
  set SAVEPAA=0
)
```

На выходе получаем:

- набор файлов вида *PAA_<k>_points_all.txt*, где *<k>=1, 5, 10, 15, 20, 23, 25, 30, 40, 50*, содержащих временные ряды в нормализованном представлении, сжатые в k раз; для удобства переименуем их в файлы вида *PAA_<k>_points_all_**study**.txt* – это будут обучающие множества для нормализованного представления временных рядов;
- набор файлов вида *SAX_<k>_points_<j>_letters.txt*, где *<k>=1, 5, 10, 15, 20, 23, 25, 30, 40, 50*, содержащих временные ряды в символьном (SAX) представлении, сжатые в k раз и с размером алфавита, равным j; для удобства переименуем их в файлы вида *SAX_<k>_points_<j>_letters_**study**.txt* – это будут обучающие множества для символьного (SAX) представления временных рядов;
- набор графиков для каждого вещественного представления (если установлена опция строить графики).

Таким образом, можно считать, что были сформированы обучающие множества, состоящие из 30 временных рядов, относящихся к классам «цилиндр», «колокол», «воронка» (по 10 представителей каждого класса) в нормализованном и символьном представлениях с различными параметрами (размер ряда, размер алфавита).

Аналогично можно получить экзаменационные множества: пусть, например, экзаменационные множества будут состоять из 300 элементов (по 100 представителей каждого класса «цилиндр», «колокол», «воронка»). Для этого нужно запустить bat-файл следующего вида:

```
process_wparams & (
for %%i in (cyl_, bel_, fun_) do for /l %%j in (0,1,99) do
   process_serie_total cbf/%%i%%j.txt
) & (
%FULLPATH%%MERGEFILEPATH% -d %FULLPATH%scripts/" -m "Sax_" -o "OUT
   "
)
```

На выходе получаем:

– набор файлов вида *PAA_<k>_points_all.txt*, где *<k>=1, 5, 10, 15, 20, 23, 25, 30, 40, 50*, содержащих временные ряды в нормализованном представлении, сжатые в k раз; для удобства переименуем их в файлы вида *PAA_<k>_points_all_**test**.txt* – это будут экзаменационные множества для нормализованного представления временных рядов;
– набор файлов вида *SAX_<k>_points_<j>_letters.txt*, где *<k>=1, 5, 10, 15, 20, 23, 25, 30, 40, 50*, содержащих временные ряды в символьном (SAX) представлении, сжатые в k раз и с размером алфавита, равным j; для удобства переименуем их в файлы вида *SAX_<k>_points_<j>_letters_**test**.txt* – это будут экзаменационные множества для символьного (SAX) представления временных рядов;
– набор графиков для каждого вещественного представления (если установлена опция строить графики).

Теперь, когда имеются в наличии обучающие и экзаменационные множества, можно проводить моделирование рассмотренных в работе алгоритмов. Например, для проверки алгоритма $TS-ADEEP-Multi$ для нормализованного представления временных рядов нужно запустить bat-файл следующего вида:

```
SET MYPROGDIR="../../../Release/tsclassifier.exe"
for %%k in (1, 5, 10, 15, 20, 23, 25, 30, 40, 50) do (
  %MYPROGDIR% --classifier eepmulti --mode d --studyset "PAA_%%
     k_points_all_study.txt" --testset "PAA_%%k_points_all_test.
     txt" --lettdist "dist_10" --alphabetsize 10
)
```

Для проверки алгоритма $TS-ADEEP-Multi$ для символьного (SAX) представления временных рядов нужно запустить bat-файл следующего вида:

```
SET MYPROGDIR="../../../Release/tsclassifier.exe"
for %%j in (1, 5, 10, 15, 20, 23, 25, 30, 40, 50) do (
  for %%s in (5, 10, 15, 20, 25, 30, 40, 50) do (
        %MYPROGDIR% --classifier eepmulti --mode s --studyset "
           sax_%%j_points_%%s_letters_study.txt" --testset "sax_%%
           j_points_%%s_letters_test.txt" --lettdist "dist_%%s" --
           alphabetsize %%s
  )
)
```

В результате получим файл с результатами моделирования, в котором содержится информация об использованном алгоритме, названиях обучающего и

экзаменационного множества, параметрах временных рядов, результатах (точности) обнаружения аномалий.

С помощью *TSClassifier.exe* также можно получить данные в формате, необходимом для алгоритмов построения деревьев решений

```
SET MYPROGDIR="../../../Release/tsclassifier.exe"
for %%j in (1, 5, 10, 15, 20, 23, 25, 30, 40, 50) do (
  for %%s in (5, 10, 15, 20, 25, 30, 40, 50) do (
        %MYPROGDIR% --classifier eepmulti --mode s --studyset "
           sax_%%j_points_%%s_letters_study.txt" --testset "sax_%%
           j_points_%%s_letters_test.txt" --lettdist "dist_%%s" --
           alphabetsize %%s --dumpid3 --dumpclasses "#class:
           CYLINDER,#class:BELL,#class:FUNNEL."
  )
)
```

и темпоральных деревьев решений

```
SET MYPROGDIR="../../../Release/tsclassifier.exe"
for %%j in (1, 5, 10, 15, 20, 23, 25, 30, 40, 50) do (
  for %%s in (5, 10, 15, 20, 25, 30, 40, 50) do (
        %MYPROGDIR% --classifier eepmulti --mode s --studyset "
           sax_%%j_points_%%s_letters_study.txt" --testset "sax_%%
           j_points_%%s_letters_test.txt" --lettdist "dist_%%s" --
           alphabetsize %%s --dumptdt --dumpclasses "#class:
           CYLINDER,#class:BELL,#class:FUNNEL."
  )
)
```

Для проверки алгоритма построения деревьев решений нужно запустить bat-файл следующего вида:

```
SET MYPROGDIR="noisestudy.exe"
for %%j in (1, 5, 10, 15, 20, 23, 25, 30, 40, 50) do (
  for %%s in (5, 10, 15, 20, 25, 30, 40, 50) do (
        %MYPROGDIR% --classifier eepmulti --mode s --studyset "
           sax_%%j_points_%%s_letters_id3.data" --testset "sax_%%
           j_points_%%s_letters_id3.test" --lettdist "dist_%%s" --
           alphabetsize %%s
  )
)
```

Для проверки алгоритма построения темпоральных деревьев решений нужно запустить bat-файл следующего вида:

```
SET MYPROGDIR="../../../../TDT/tdt/bin/Release/tdt.exe"
for %%j in (1, 5, 10, 15, 20, 23, 25, 30, 40, 50) do (
  for %%s in (5, 10, 15, 20, 25, 30, 40, 50) do (
        %MYPROGDIR% --classifier eepmulti --mode s --studyset "
           sax_%%j_points_%%s_letters_tdt.dat" --testset "sax_%%
           j_points_%%s_letters_tdt.test" --metadata "sax_%%
           j_points_%%s_letters_tdtmeta.xml" --lettdist "dist_%%s"
            --alphabetsize %%s --dumpclasses "#class:CYLINDER,#
           class:BELL,#class:FUNNEL"
  )
)
```

В результате получим файл с результатами моделирования, в котором содержится информация об использованном алгоритме, названиях обучающего и экзаменационного множества, параметрах временных рядов, результатах (точности) обнаружения аномалий (классификации).

Приложение Б

Свидетельства о государственной регистрации программ для ЭВМ

Б.1 «Noise study–Изучение шума»

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2012611444

«Noise study-Изучение шума»

Правообладатель(ли): *федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Национальный исследовательский университет «МЭИ» (ФГБОУ ВПО «НИУ МЭИ») (RU)*

Автор(ы): *Антипов Сергей Геннадьевич, Фомина Марина Владимировна (RU)*

Заявка № 2011619501

Дата поступления 13 декабря 2011 г.

Зарегистрировано в Реестре программ для ЭВМ 7 февраля 2012 г.

Руководитель Федеральной службы по интеллектуальной собственности

Б.П. Симонов

Рисунок Б.1 — «Noise study–Изучение шума»

Б.2 «Time Series Anomally Detection (TiSAD)» – «Обнаружение аномалий в наборах временных рядов»

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2013618587

«Time Series Anomaly Detection (TiSAD)» - «Обнаружение аномалий в наборах временных рядов»

Правообладатель: ***федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Национальный исследовательский университет «МЭИ» (RU)***

Авторы: ***Антипов Сергей Геннадьевич (RU),***
Фомина Марина Владимировна (RU)

Заявка № **2013616058**
Дата поступления **16 июля 2013 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***12 сентября 2013 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Б.П. Симонов

Рисунок Б.2 — «Time Series Anomally Detection (TiSAD)» – «Обнаружение аномалий в наборах временных рядов»

Б.3 «Temporal Decision Trees (TDT)» – «Темпоральные деревья решений»

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2013618586

«Temporal Decision Trees (TDT)» - «Темпоральные деревья решений»

Правообладатель: *федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Национальный исследовательский университет «МЭИ» (RU)*

Авторы: *Антипов Сергей Геннадьевич (RU),*
Вагин Вадим Николаевич (RU)

Заявка № 2013616059
Дата поступления 16 июля 2013 г.
Дата государственной регистрации
в Реестре программ для ЭВМ *12 сентября 2013 г.*

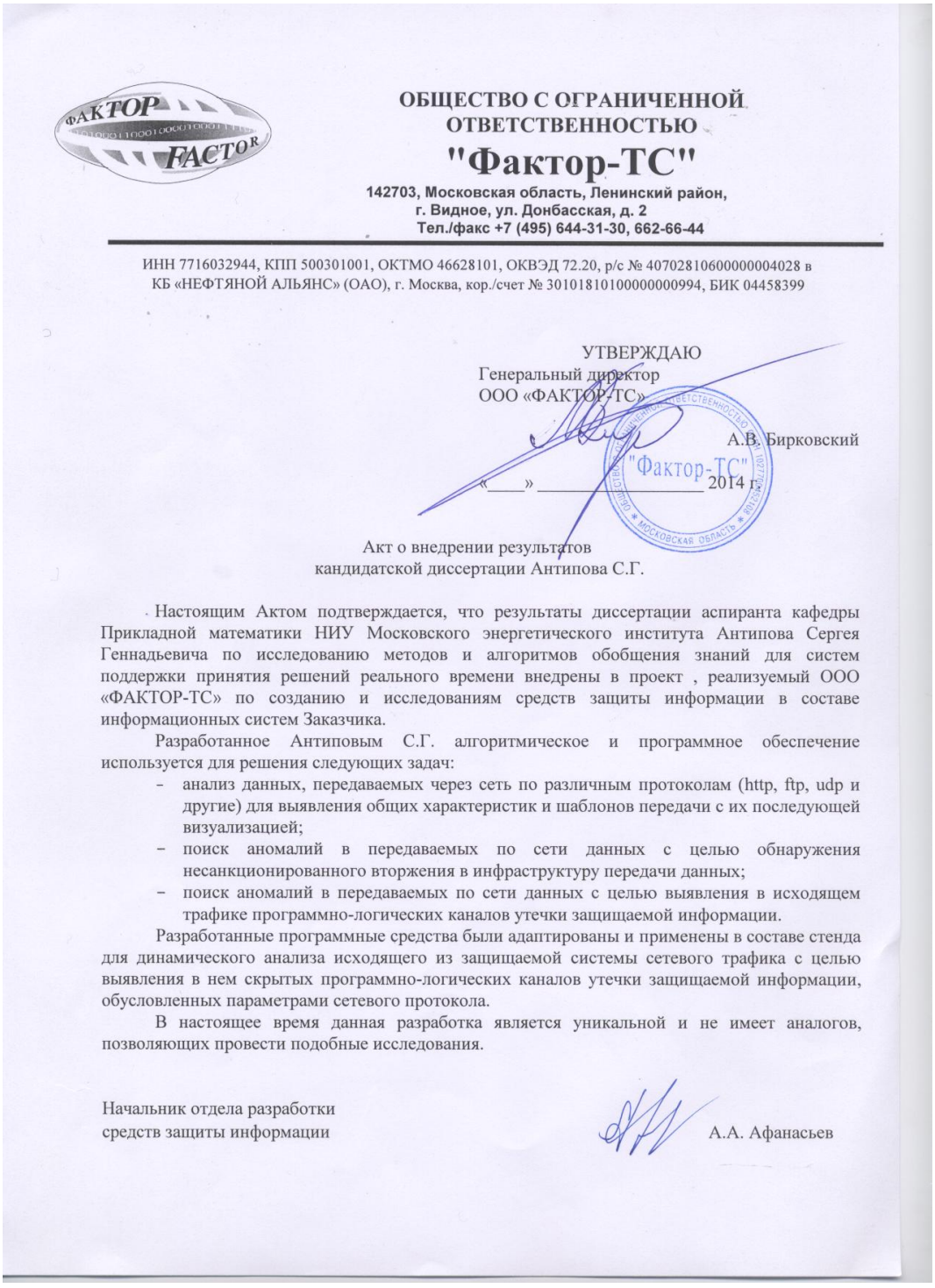
Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов

Рисунок Б.3 — «Temporal Decision Trees (TDT)» – «Темпоральные деревья решений»

Приложение В

Акт о внедрении



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
"Фактор-ТС"
142703, Московская область, Ленинский район,
г. Видное, ул. Донбасская, д. 2
Тел./факс +7 (495) 644-31-30, 662-66-44

ИНН 7716032944, КПП 500301001, ОКТМО 46628101, ОКВЭД 72.20, р/с № 40702810600000004028 в КБ «НЕФТЯНОЙ АЛЬЯНС» (ОАО), г. Москва, кор./счет № 30101810100000000994, БИК 04458399

УТВЕРЖДАЮ
Генеральный директор
ООО «ФАКТОР-ТС»
А.В. Бирковский
«___» _____ 2014 г.

Акт о внедрении результатов
кандидатской диссертации Антипова С.Г.

Настоящим Актом подтверждается, что результаты диссертации аспиранта кафедры Прикладной математики НИУ Московского энергетического института Антипова Сергея Геннадьевича по исследованию методов и алгоритмов обобщения знаний для систем поддержки принятия решений реального времени внедрены в проект , реализуемый ООО «ФАКТОР-ТС» по созданию и исследованиям средств защиты информации в составе информационных систем Заказчика.

Разработанное Антиповым С.Г. алгоритмическое и программное обеспечение используется для решения следующих задач:

- анализ данных, передаваемых через сеть по различным протоколам (http, ftp, udp и другие) для выявления общих характеристик и шаблонов передачи с их последующей визуализацией;
- поиск аномалий в передаваемых по сети данных с целью обнаружения несанкционированного вторжения в инфраструктуру передачи данных;
- поиск аномалий в передаваемых по сети данных с целью выявления в исходящем трафике программно-логических каналов утечки защищаемой информации.

Разработанные программные средства были адаптированы и применены в составе стенда для динамического анализа исходящего из защищаемой системы сетевого трафика с целью выявления в нем скрытых программно-логических каналов утечки защищаемой информации, обусловленных параметрами сетевого протокола.

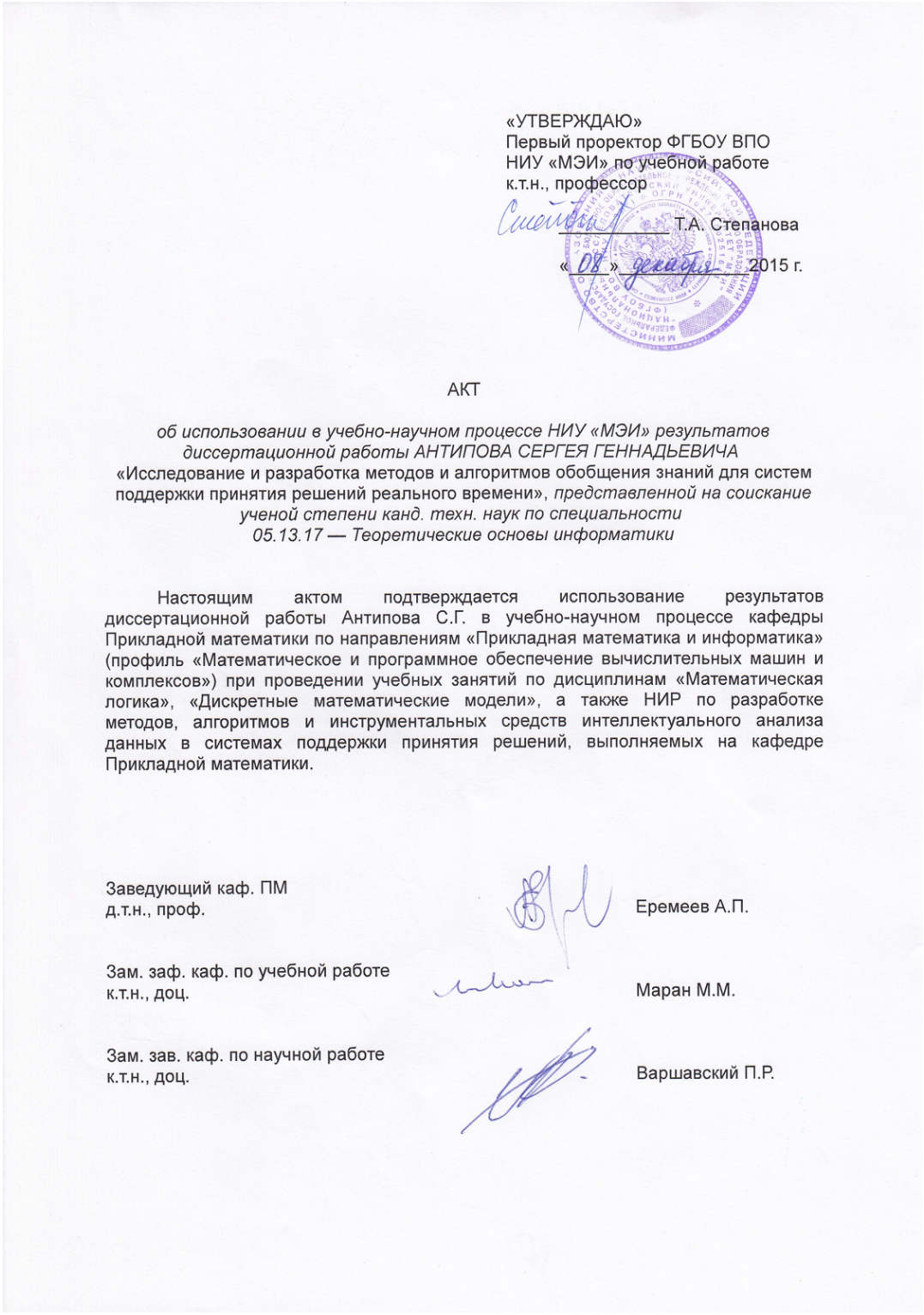
В настоящее время данная разработка является уникальной и не имеет аналогов, позволяющих провести подобные исследования.

Начальник отдела разработки
средств защиты информации А.А. Афанасьев

Рисунок В.1 — Акт о внедрении

Приложение Г

Акт об использовании в учебно-научном процессе



«УТВЕРЖДАЮ»
Первый проректор ФГБОУ ВПО
НИУ «МЭИ» по учебной работе
к.т.н., профессор
_____ Т.А. Степанова
«08» декабря 2015 г.

АКТ

об использовании в учебно-научном процессе НИУ «МЭИ» результатов диссертационной работы АНТИПОВА СЕРГЕЯ ГЕННАДЬЕВИЧА
«Исследование и разработка методов и алгоритмов обобщения знаний для систем поддержки принятия решений реального времени», *представленной на соискание ученой степени канд. техн. наук по специальности 05.13.17 — Теоретические основы информатики*

Настоящим актом подтверждается использование результатов диссертационной работы Антипова С.Г. в учебно-научном процессе кафедры Прикладной математики по направлениям «Прикладная математика и информатика» (профиль «Математическое и программное обеспечение вычислительных машин и комплексов») при проведении учебных занятий по дисциплинам «Математическая логика», «Дискретные математические модели», а также НИР по разработке методов, алгоритмов и инструментальных средств интеллектуального анализа данных в системах поддержки принятия решений, выполняемых на кафедре Прикладной математики.

Заведующий каф. ПМ
д.т.н., проф. Еремеев А.П.

Зам. заф. каф. по учебной работе
к.т.н., доц. Маран М.М.

Зам. зав. каф. по научной работе
к.т.н., доц. Варшавский П.Р.

Рисунок Г.1 — Акт об использовании в учебно-научном процессе НИУ «МЭИ»