

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Шолохова Алена Алексеевна

Выпускная квалификационная работа аспиранта на тему

**РАЗРАБОТКА МЕТОДОВ ОБНАРУЖЕНИЯ
АНОМАЛИЙ МАНЕВРИРОВАНИЯ
И ПОТРЕБЛЕНИЯ ТОПЛИВА ДЛЯ МОРСКИХ СУДОВ**

Образовательная программа МК.3005.2015 «Математическая кибернетика»

Направление 02.06.01 «Компьютерные и информационные науки»

Научный руководитель,
доктор физ.-мат. наук,
профессор

Андрианов С. Н.

Рецензент,
кандидат тех. наук,
старший научный сотрудник

Юдин И. П.

Санкт-Петербург

2018

Оглавление

Введение	3
1 Постановка задачи	7
1.1 Обнаружение аномалий как область анализа данных.....	7
1.2 Методы машинного обучения	21
1.3 Модельный подход в анализе маневров судна	30
2 Математическое моделирование и численные методы обнаружения аномалий	32
2.1 Построение признакового пространства входных данных	34
2.2 Метод изолирующего леса.....	38
2.3 Матричный формализм для решения задачи идентификации ...	44
3 Программная реализация и вычислительный эксперимент	49
3.1 Архитектура и логика работы программных компонент.....	49
3.2 Валидация программного обеспечения.....	52
3.3 Вычислительный эксперимент	55
3.3.1 Обнаружение аномалий маневрирования	55
3.3.2 Обнаружение аномалий потребления топлива	58
Выводы	61
Заключение	63
Список литературы и источников	65
Приложение А. Формат входных данных	74
Приложение В. Интерфейс программы Полифит	76

Введение

В настоящее время морская область переходит к этапу цифровой трансформации [12]. Появляются и внедряются системы интернета вещей (Internet of Things, IoT), больших данных (Big Data), машинного обучения. На современном судне установлено большое количество датчиков, ведущих непрерывную запись показателей функционирования его различных подсистем. В случае организации сбора и хранения подобной информации открываются возможности применения интеллектуальных систем анализа данных. Установленные на судне датчики могут генерировать числовые данные с интервалами, варьирующимися от десяти секунд для данных телеметрии до одной минуты для навигационных параметров. Учитывая большой объем получаемой информации, становится очевидной необходимость развития автоматических систем анализа и обработки данных.

Внедрение подобных автоматических систем поддержки принятия решения в морской области в основном направлено на две группы задач. К первой относятся вопросы повышения безопасности судовождения, анализа и предупреждения опасных маневров, обнаружения нештатных режимов движения судна. Ко второй группе следует отнести экономические эффекты от оптимизации режимов потребления топлива, прогнозирование возможных поломок. Особенностью последней задачи является то, что момент обнаружения поломки не всегда соответствует её возникновению, то есть поломка может быть обнаружена не вовремя, и разность во времени между возникновением проблемы и её обнаружением может достигать нескольких недель. Решение указанных задач требует как развития инфраструктуры по сбору и хранению данных, так и разработки новых алгоритмических подходов анализа данных.

Обнаружение аномалий в данных было изучено в статистическом сообществе уже в XIX веке [40]; со временем в нескольких исследовательских группах были разработаны методы обнаружения аномалий, часть из которых

имела узкие области применения [13, 60], в то время как прочие подходы были междисциплинарными [31, 32, 54].

Обнаружение аномалий (Anomaly Detection) относится к проблеме поиска образцов данных, которые не соответствуют определенному понятию нормального поведения и называются аномалиями, выбросами, противоречивыми наблюдениями, исключениями, особенностями или загрязнениями в разных областях применения. Задачи обнаружения аномалий не имеют единой формулировки и часто интерпретируются различным образом в зависимости от поставленной цели и характера данных [19, 45]. Важность обнаружения аномалий обусловлена тем фактом, что они преобразуются в значительную, часто критически важную информацию с использованием в разных областях применения. Становится актуальной необходимостью разработки методов, способных на основе анализа поведения процесса прогнозировать возникновение нештатных ситуаций в динамических системах.

Обнаружение новизны в данных (Novelty Detection) является другой проблемой, связанной с детектированием аномалий и направленной на обнаружение ранее не наблюдаемых шаблонов в данных, которые в отличие от выбросов (Outlier Detection), обычно включаются в нормальную модель после их обнаружения [71].

Целью работы является разработка методов поиска аномалий в сенсорных данных, а также исследование свойств и апробация алгоритмов на задаче анализа движения морских судов.

Для достижения указанной цели необходимо решить ряд **задач**.

1. Выявление актуальных направлений в области прогнозирования аномальных событий применительно к анализу движения морских судов.
2. Построение математической модели обнаружения аномалий маневрирования и потребления топлива для морских судов.
3. Реализация программного инструментария для обработки и анализа данных на основе построенной математической модели.

4. Проведение вычислительного эксперимента на основе сенсорных данных с морских судов и анализ полученных результатов.

Методы исследования в данной работе основываются на фундаментальных положениях интеллектуального анализа данных, теории временных рядов и теории машинного обучения. **Объектом исследования** являются методы теории интеллектуального анализа данных. **Предметом исследования** являются модели, методы и алгоритмы автоматического обнаружения и прогнозирования аномалий в сенсорных данных. **Научная новизна** исследования заключается в решении актуальной задачи обнаружения и прогнозирования аномалий в сенсорных данных.

Публикации. Основные результаты исследования отражены в двух статьях в изданиях, рекомендованных ВАК РФ; опубликованы 1 работа в издании, индексируемом Scopus, и 1 работа в сборнике конференции, индексируемой в РИНЦ. Также получено свидетельство о регистрации программы на ЭВМ «Полифит: полиномиальная нейронная сеть».

Структура и содержание работы. Выпускная квалификационная работа состоит из введения, трех глав, выводов, заключения, списка литературных источников из 89 наименований и двух приложений. Общий объем работы составляет 78 страниц.

Во введении отражена актуальность рассматриваемых проблем, приводятся формулировки цели и задач исследования, представлено описание структуры работы. **Первая глава** посвящена теоретическому описанию задачи поиска аномалий как проблемы интеллектуального анализа данных. Представлен обзор численных методов, применяемых в данной предметной области. **Вторая глава** отражает построение математической модели предметной области: формализован алгоритм изолирующего леса, использованный в качестве основного метода поиска аномалий, представлена модель прогнозирования потребления топлива, построенная на основе полиномиальных нейронных сетей. **В третьей главе** рассматривается реализованный программный инструментарий обработки и анализа данных на основе построенной математической модели, приведены

результаты тестирования предложенного численного метода на хорошо изученных модельных задачах. **В заключении** приведены основные теоретические и практические результаты, выносимые на защиту выпускной квалификационной работы, указаны направления дальнейшего развития исследования. Формат входных данных и интерфейс программы Полифит представлены в **приложениях**.

1 Постановка задачи

1.1 Обнаружение аномалий как область анализа данных

Интеллектуальный анализ данных (Data Mining) является областью исследований, позволяющей выделить некоторую новую значимую информацию в большом объеме данных [58]. Главной задачей данной области является извлечение явно не видимых связей и непредвиденных тенденций с помощью различных методов обработки данных.

Обнаружение аномалий относится к проблеме поиска образцов данных, которые не соответствуют определенному понятию нормального, ожидаемого поведения и называются выбросами. Процесс поиска аномалий заключается в определении области, представляющей собой нормальное поведение, и в последующем отнесении к аномалиям тех наблюдений в данных, которые не относятся к этой нормальной области. Таким образом, аномалии определяют не их собственные характеристики, а сравнение с тем, что есть норма.

Аномальные значения могут быть следствием следующего.

- Наличия шумовых объектов (неверно классифицированных объектов).
- Показания сломавшегося датчика, то есть будет присутствие объектов «других» выборок.
- Ошибок в данных (к примеру, неточности измерения, округления или неверной записи).

На рис. 1.1 представлен пример задачи поиска аномальных значений с двумя признаками. Видно, что шум (noise) является аномалией в слабом смысле, поскольку он может только лишь размывать границы кластера, в то время как интерес представляют аномалии в сильном смысле, которые эти границы искажают.

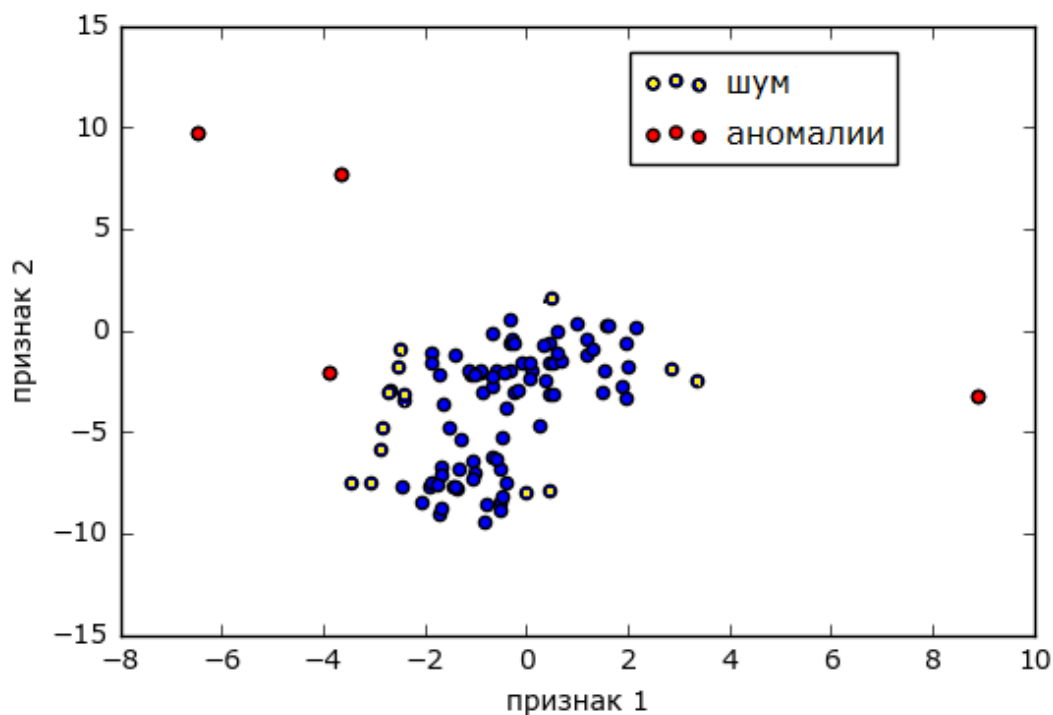


Рис. 1.1 – Задача поиска аномалий в данных с двумя признаками

Детектирование аномалий используется, к примеру, при поиске отказов в системах безопасности; для контроля движения поездов с целью повышения безопасности железнодорожного движения; при обнаружении мошенничества при проведении банковских транзакций и вторжений в компьютерной безопасности, для чего проводится анализ трафика на аномальность; при выявлении раковых опухолей, которые могут быть обнаружены с помощью детектирования аномалий на снимках магнитно-резонансной томографии.

При обнаружении аномалий могут возникнуть следующие сложности.

- 1) Определение нормальной области, охватывающей всё возможное нормальное поведение, затруднено, поскольку границы между нормальным и аномальным поведением зачастую не точны, и аномальное наблюдение, лежащее вблизи границы, может быть нормальным и наоборот.
- 2) Когда аномалии являются результатом злонамеренных действий, злоумышленники часто приспосабливаются таким образом, чтобы аномальные наблюдения казались нормальными, что затрудняет задачу определения аномального поведения.

- 3) Зачастую данные содержат шум, похожий на фактические аномалии и по этой причине трудно от неё отличимый.
- 4) Наличие промаркированных, используемых для обнаружения аномалий, данных для обучения и проверки моделей.
- 5) Поскольку понятие нормального поведения развивается, тем самым изменяясь, текущее понятие такого поведения может быть недостаточно репрезентативным в будущем.

Из-за вышеперечисленных пунктов проблему обнаружения аномалий в ее наиболее общей форме решить непросто. Фактически, большинство существующих в настоящее время методов детектирования аномалий решают конкретную формулировку проблемы. Разработка индуцируется различными факторами, например, характером данных, доступностью промаркированных данных, типом и характером аномалий, подлежащих обнаружению, и прочее. Зачастую эти факторы определяются областью применения, в которой необходимо выявлять аномалии.

На рис. 1.2 представлены примеры аномалий $a1$, $a2$ и A к группам нормальных областей данных $N1$ и $N2$. Здесь аномалии $a1$, $a2$ имеют схожий характер, в то время как аномалия A от них отличается.

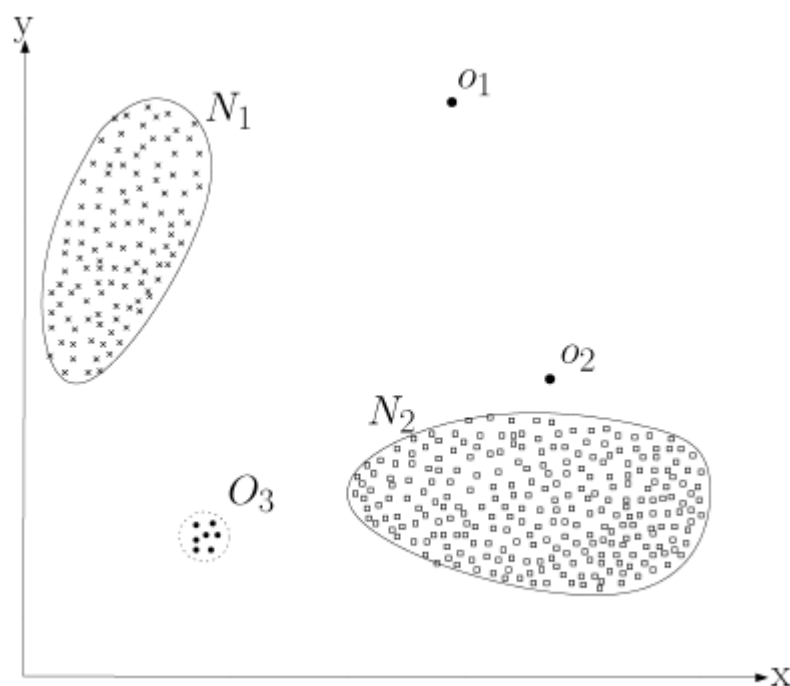


Рис. 1.2 – Пример аномальных значений в двумерном пространстве

Важнейшим аспектом решения задачи обнаружения аномалий является определение вида аномалий [31].

1. Если аномалия может быть классифицирована таковой по отношению к остальному набору данных, то такую аномалию называют **одионой** или **точечной**. На рис. 1.2 $a1$, $a2$ и A являются одиоными аномалиями к нормальным множествам объектов $N1$ и $N2$.

Примером применения методов детектирования одионых аномалий может служить выявление злоумышленных действий, например, когда данные могут быть для упрощения представлены одним-двумя признаками – скоростью движения судна и потреблением топлива. В таком случае одионой аномалией будет являться точечное повышенное потребление топлива по сравнению с нормальными потреблением для данной скорости движения судна.

2. Если данные аномальны только в определенном контексте, то такая аномалия будет являться **контекстуальной** или **условной** [84]. Контекст определяется структурой набора данных и формулировкой проблемы. На рис. 1.3 изображен пример контекстуальной аномалии применительно к задаче анализа среднемесячной температуры воды в море. В данном случае только $t2$ является контекстуальной аномалией, в то время как $t1$ ею не является, несмотря на равенство по значению $t2$.

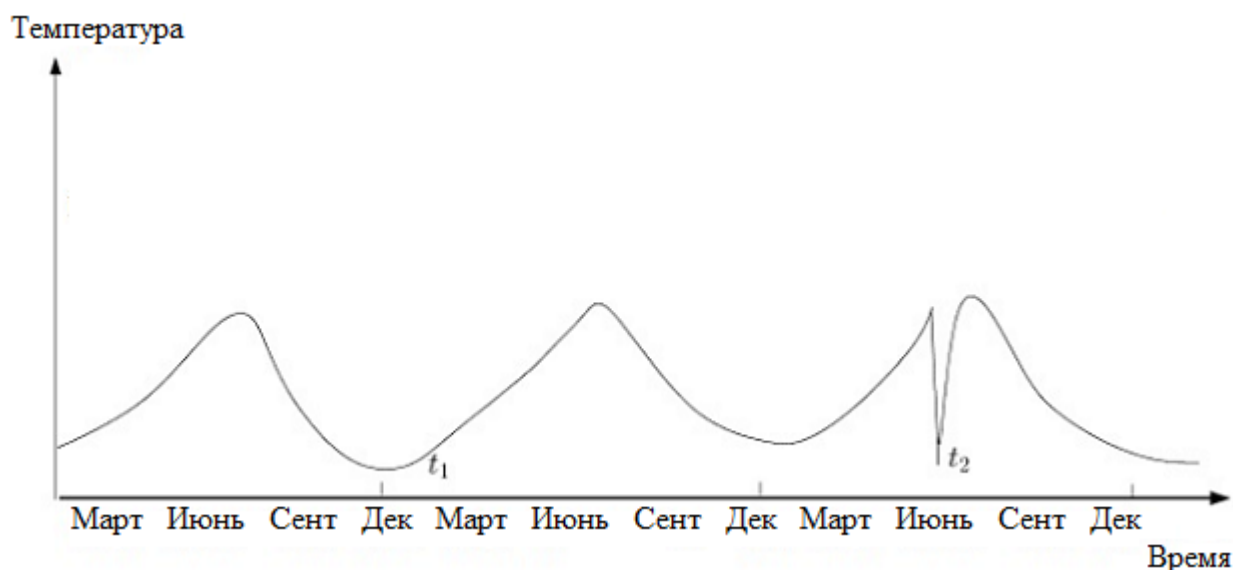


Рис. 1.3 – Пример контекстуальных аномалий

Для данных могут быть определены следующие признаки.

- Контекстуальные признаки необходимы при определении соседних примеров, представляющих собой контекст, для текущего: например, таким признаком является момент времени во временных рядах.
- Не контекстуальные характеристики примера определяются поведенческими признаками. Например, не контекстуальной характеристикой при описании средней мировой скорости ветра скорость ветра в определенном месте.

Среди подходов к обнаружению контекстуальных аномалий могут быть выделены два направления.

- Использование методов детектирования одиночных аномалий применительно к контексту, когда производится преобразование в пространство состояний, позволяющее обнаруживать аномалии без использования информации о контексте.
- При обработке данных в контексте применяются методы моделирования временных рядов. Тогда для обнаружения контекстуальных аномалий используются следующие регрессионные методы: устойчивая регрессия [75], авторегрессионные модели [21], авторегрессионные модели скользящего среднего [87].

В работе [41], посвященной моделированию авторегрессионной моделью, новые данные проверяются на аномальность путем сравнения с ковариационной матрицей авторегрессионного процесса. При этом если данное состояние находится вне границ допустимой ошибки, оно будет признано классификатором как аномальное. Несмотря на большую вычислительную сложность на этапе обучения этап проверки на аномальность вычислительно менее затратен, чем при преобразовании контекстуальных данных для поиска одиночных аномалий.

3. Если связанные между собой данные аномальны по отношению к остальному набору данных, то они называются **групповой** аномалией, при этом её отдельные примеры могут не являться одиночными аномалиями. Групповая аномалия представлена на фрагменте кардиограммы (рис. 1.4). Здесь групповой аномалией является запись преждевременного сокращения сердца, которое в единичном случае является допустимым.

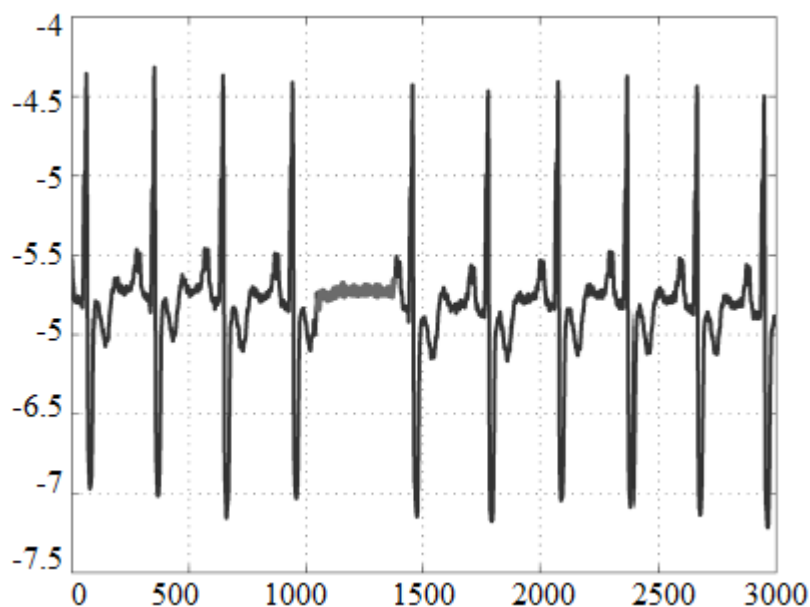


Рис. 1.4 – Пример групповой аномалии на электрокардиограмме

И одиночные, и групповые аномалии могут быть контекстуальными. В таком случае задача поиска одиночных или групповых аномалий может быть сведена к задаче обнаружения контекстуальных аномалий.

Проблема обнаружения аномалий была предметом ряда обзоров статей и книг. В [47] представлен обширный обзор подходов обнаружения аномалий на основе статистических моделей и машинного обучения. Согласно отображенной в работе классификации, существуют следующие подходы к детектированию аномалий.

- Определение аномалий с помощью кластеризации без учителя: данные обрабатываются по статистическому распределению, а наиболее отдаленные точки маркируются как потенциальные выбросы (рис. 1.5).

Кластеризация без учителя предполагает два возможных действия с обнаруженными аномалиями [75]: в результате диагностики аномальные значения могут быть удалены из выборки, в то время как при внедрении они могут быть использованы в модели распределения.

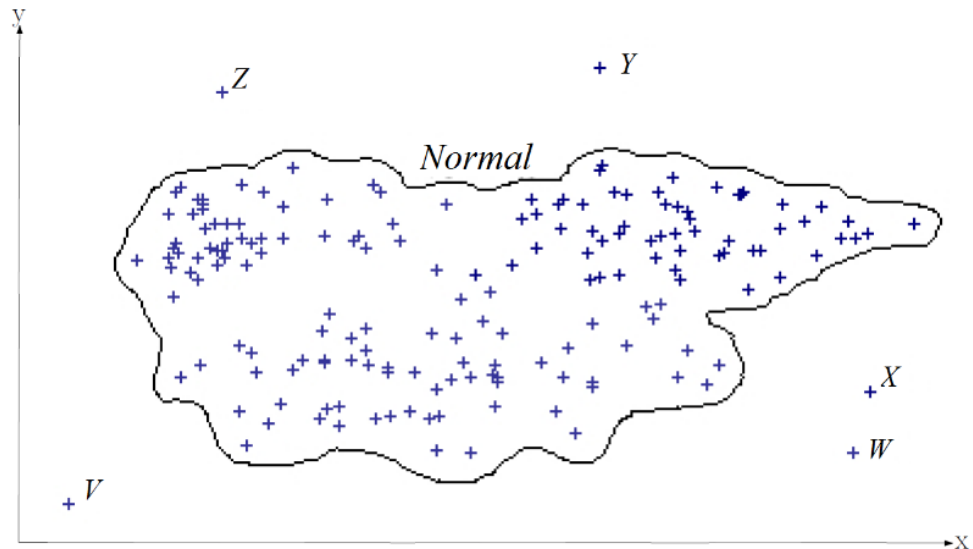


Рис. 1.5 – Пример применения обнаружения аномалий без учителя

- Детектирование аномалий с помощью классификации с учителем, что требует маркированных в соответствии с принадлежностью к аномальным значениям данных [36]. В результате маркировки могут быть определены несколько классов, содержащих нормальные данные (рис. 1.6). Потенциальной проблемой такого подхода может стать изменение статистических характеристик данных, влекущее за собой перестройку классификатора. Эта проблема решена с помощью инкрементальных классификаторов, используемых, к примеру, в эволюционных нейронных сетях в режиме реального времени, где построенная модель сравнивается с поступающими на вход примерами по заданному алгоритму [26,82].

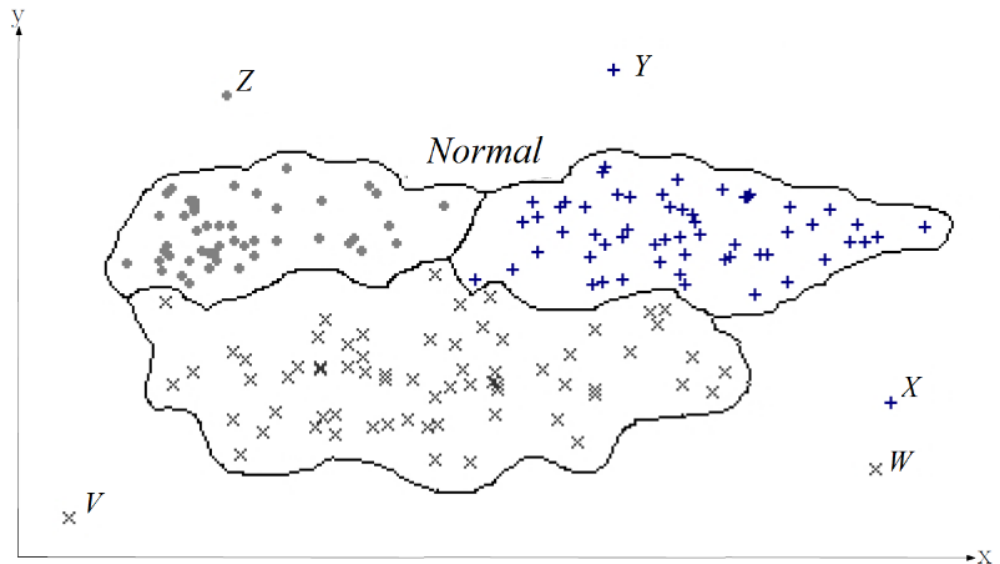


Рис. 1.6 – Пример применения обнаружения аномалий с учителем

- Для поиска новизны используют обучение частично с учителем [33], обучаясь только на одном классе и предполагая распознавание алгоритмом противоположного класса [51]. Таким образом, новый пример классифицируется как нормальный, если он сравним с обучающей выборкой, и распознается как новизна в обратном случае. Рис. 1.7 иллюстрирует нормальную область, необходимую для обучения частично с учителем и подобную информации на рис. 1.5. При сравнении точек V , W , X , Y (рис. 1.5) с обученной моделью на основе данных, представленных на рис. 1.7, они будут классифицированы как новизна в связи с их удаленностью от обучающей выборки.

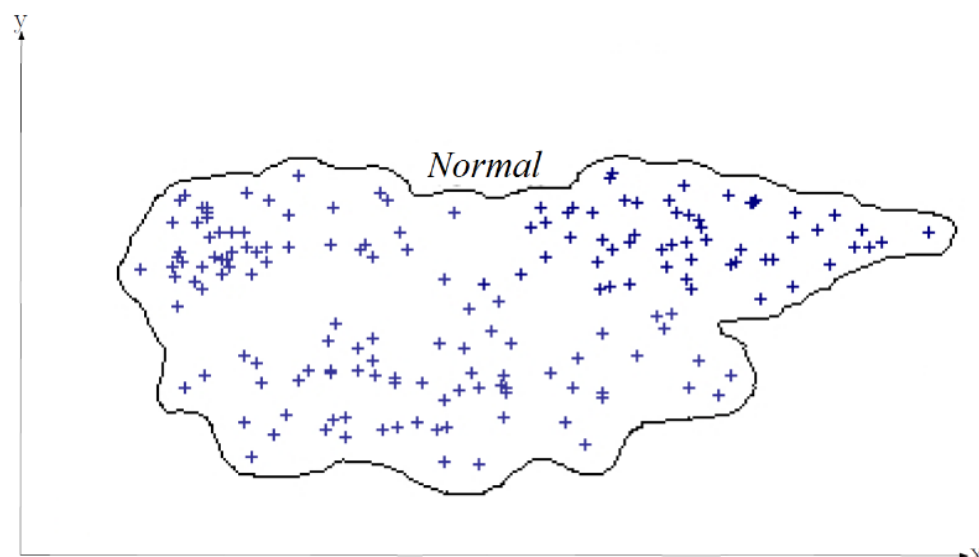


Рис. 1.7 – Пример применения обнаружения аномалий частично с учителем

Классификация, основанная на способах вычислений, представлена в [70]. При поиске аномалий на основе **статистического анализа** выполняется построение профилей, представляющих реальное и идеальное поведения. Статистическая система обнаружения аномалий после обработки процесса обновляет реальный профиль и путем сравнения с идеальным профилем с помощью предопределенной функции аномальности вычисляет его степень аномальности. К преимуществу такого подхода относится отсутствие требования априорной информации о признаках аномалий, что позволяет детектировать уязвимости нулевого дня, против которых ещё не разработаны защитные механизмы [25]. К недостаткам относится сложность определения порога для оптимального детектирования аномалий, невозможность идентификации аномалий в результате злонамеренных действий, схожих с привычными действиями или работой и требование наличия статистических распределений при недоступности всех элементов исследуемого процесса.

При поиске аномалий на основе методов **машинного обучения** предполагается построение системы, которая совершенствуется на основе предыдущих знаний [69]. Высокие вычислительные затраты, а также сложность адаптации к предметной области относятся к недостаткам методов машинного обучения.

Поиск аномалий на основе *искусственных нейронных сетей* относится к классу статистических алгоритмов, идея которых взята из принципа организации биологических нейронных сетей [26]. Достоинством подхода является устойчивость к неточно вводимой информации, а также независимость от наличия информации о зависимостях примеров входных данных. Недостатком является сложное и долгое обучение нейронных сетей, а также требовательность к объему обучающей выборки.

В основе *генетических алгоритмов* поиска аномалий лежит поиск приближительных решений для оптимизации или решения поставленной проблемы [3]. Преимуществом таких алгоритмов является устойчивость к случайным шумовым изменениям выборки данных; недостатком — сложность подбора правил для выбора лучших решений.

Поскольку множество процессов представляется только лишь нечеткими величинами, становится возможным поиск аномалий на основе *нечеткой логики*, концепция «размытости» данных которой позволяет учесть границу между нормальными и аномальными данными [88]. При данном методе возникает сложность с генерацией нечетких правил, сложнореализуемых на современных, работающих с дискретной логикой устройствах.

Системы детектирования аномалий на основе *индуктивного вывода*, предполагающие введение набора ассоциативных правил, используют события Y , стараясь найти такие X , при которых Y прогнозируется наилучшим образом. В качестве недостатка можно указать сложность описания некоторых типов данных, а также их обработки. Достоинством же является то, что система стремится быть интуитивной и гибкой.

В отдельный класс выделены *гибридные* методы, включающие в себя два и более алгоритма, относящиеся к разным классам поиска аномалий. Такой подход представляется наиболее гибким, поскольку позволяет уменьшить влияние от недостатков одного метода за счет достоинств другого.

Обзор различных методов детектирования аномалий, используемых для обнаружения вторжений в компьютерные сети, был представлен в [56].

Широкий обзор методов обнаружения аномалий как числовых, так и символических данных, представлен в работе [20]. Подробный обзор методов обнаружения новизны с использованием нейронных сетей и статистических подходов был представлен в [64] и [65] соответственно. Значительная часть исследований по обнаружению выбросов была сделана в области статистики и была рассмотрена в нескольких книгах [23, 44, 75], а также в следующих обзорных статьях [22, 24].

Традиционно в задаче поиска аномалий выделяют несколько наиболее используемых подходов. К одной группе методов можно отнести подходы, основанные на **статистических тестах**. Также подобные алгоритмы применяются в анализе сенсорных данных при моделировании их временными рядами. Эту группу методов применяют для отдельных признаков с целью выявления экстремальных значений. Такой подход применен, например, в работе [29], где сенсорные данные моделируются авторегрессионной моделью, а для обнаружения аномалий используется теория проверки гипотез (Z – оценка: $Z_i = \frac{|x_i - \mu|}{\sigma}$ [55], коэффициент эксцесса $\frac{1}{n} \sum_{i=1}^n Z_i^4$ [55]) (рис. 1.8).

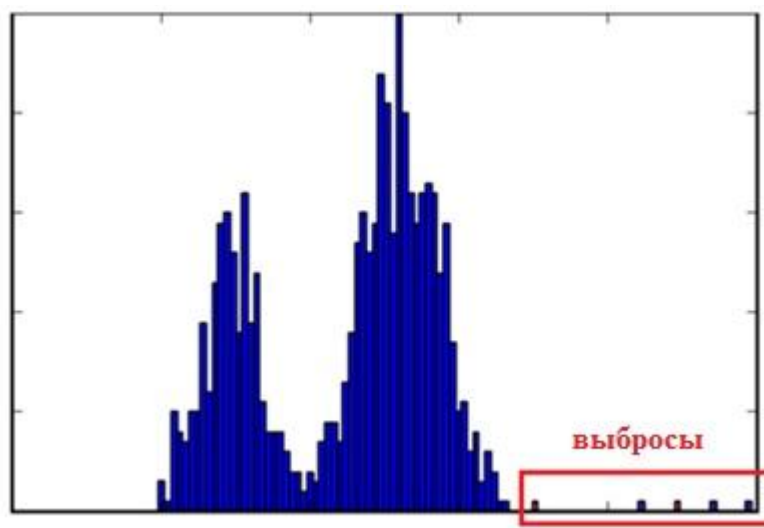


Рис. 1.8 – Пример применения статистического метода

Другую группу составляют тесты, заключающиеся в **построении модели**, описывающей данные, и выявлении точек, отклоняющейся от неё, — **модельный подход** (рис. 1.9). При выборе модели мы можем учесть природу задачи, ввести

функционал качества и прочее. Такие методы лучше подходят для определения новизны, чем для поиска выбросов, так как при настройке модели используются данные, в которых есть выбросы.

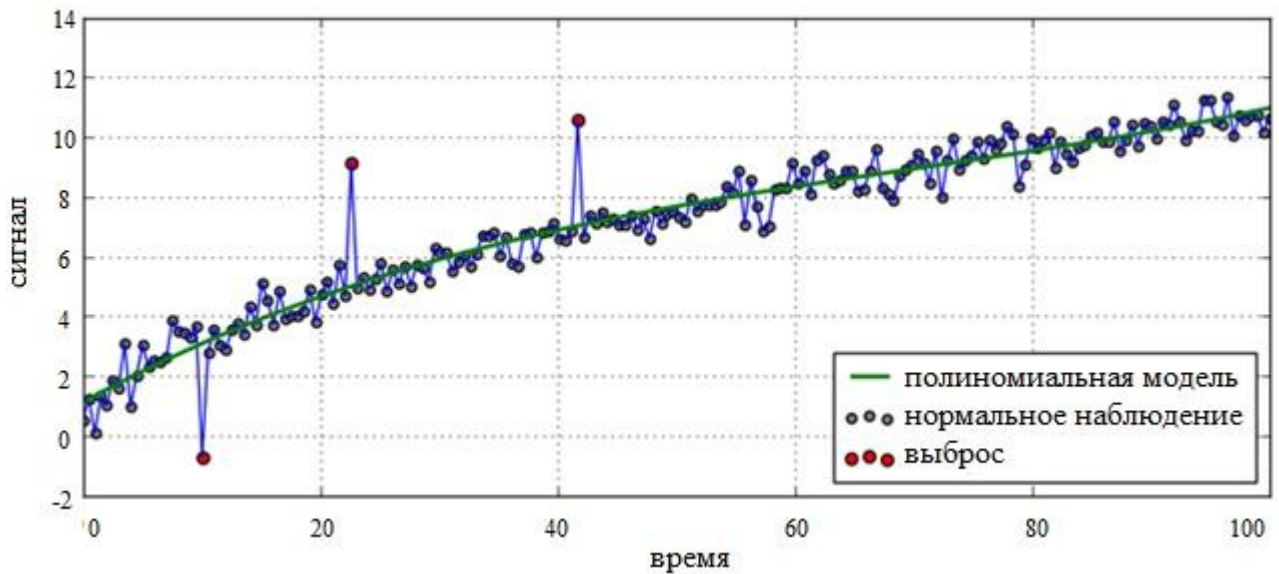


Рис. 1.9 – Пример применения модельного подхода

Итерационные методы заключаются в построении оболочек в многомерном пространстве в предположении, что выбросы в данных будут лежать на ней (рис. 1.10). Недостатком группы методов является их трудоемкость.

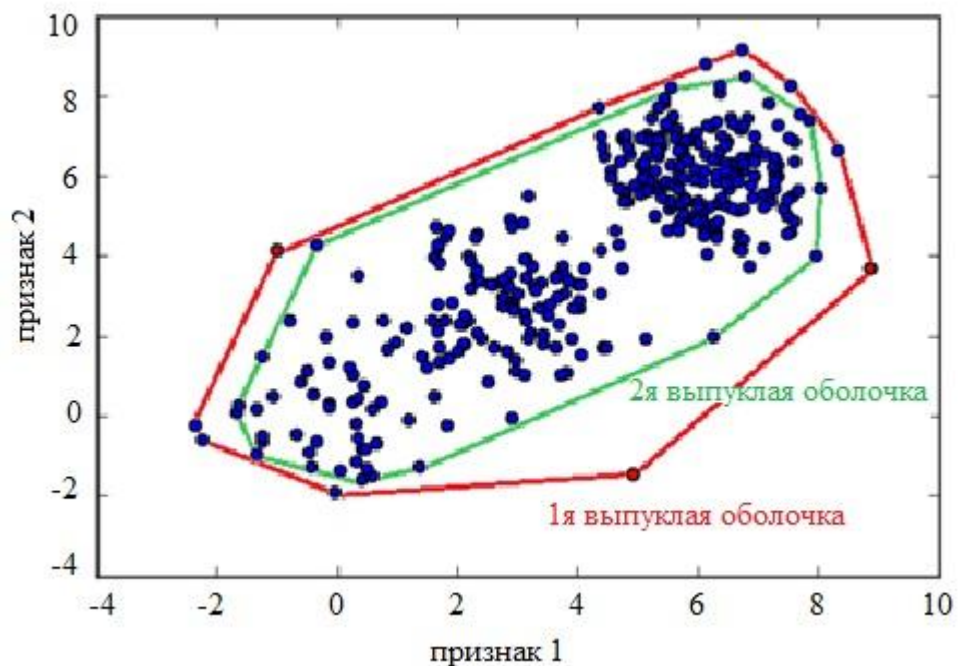


Рис. 1.10 – Сведение задачи поиска аномалий к построению выпуклых оболочек

Следующую группу составляют **метрические** алгоритмы обнаружения выбросов, которые являются одними из самых популярных методов [19] и заключаются в задании некоторой метрики в пространстве объектов, помогающей детектировать аномалии. В качестве примера можно привести метод k ближайших соседей (k -nearest neighbors algorithm) [53, 73], в котором применительно к аномалиям предполагается, что выбросы сильно удалены соседних точек пространства (рис. 1.11).

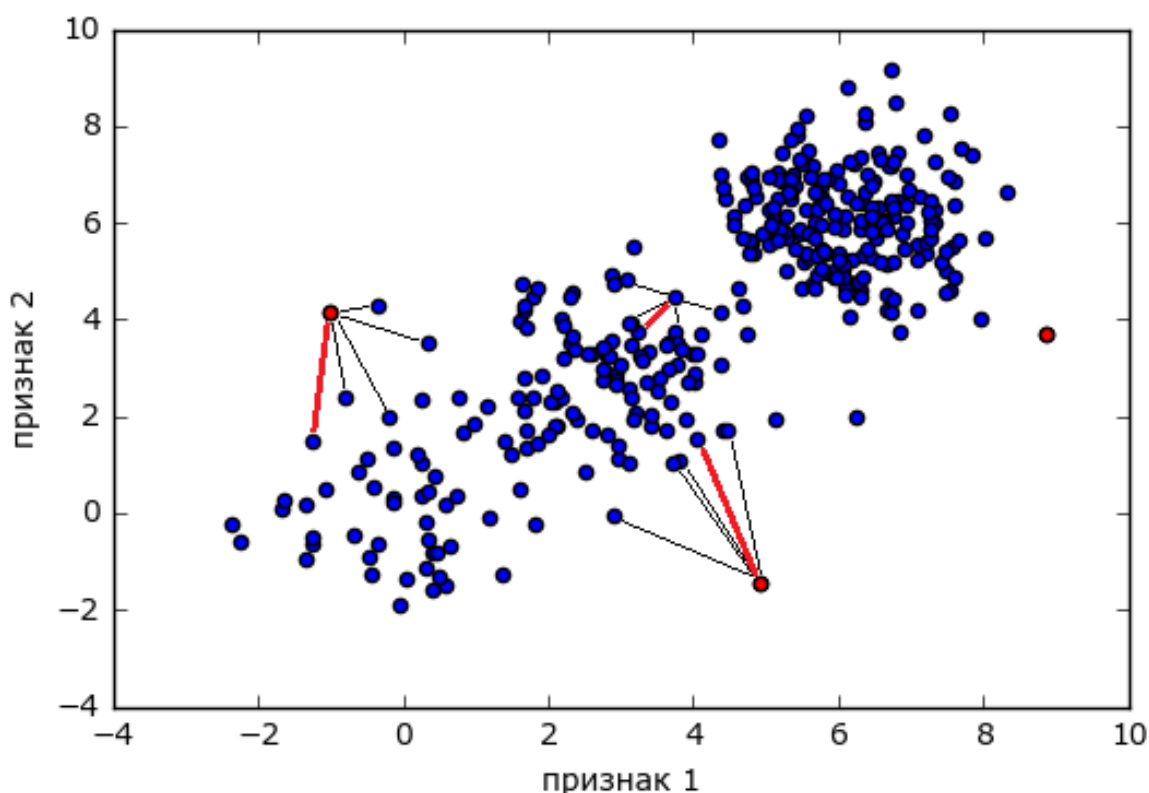


Рис. 1.11 – Поиск аномалий по расстоянию до ближайших соседей

Применение подобных алгоритмов для задач поиска аномалий рассмотрено в работах [28, 61]. К недостаткам таких подходов относятся вычислительная трудоемкость и сложность выбора подходящей метрики. Причем если производительность метода можно повысить при помощи современных технологий параллельных и распределенных вычислений, то выбор метрики, удачно отражающей структуру многомерных данных, остается открытым

вопросом и требует отдельного рассмотрения. Более подробный обзор указанных подходов может быть найден, например, в работах [6, 14].

В методах **машинного обучения** задача обнаружения аномалий рассматривается как отдельная задача обучения без учителя. Этот метод решения может быть отнесен к модельному подходу. Наиболее популярные алгоритмы, такие, как метод опорных векторов для одного класса, метод изолирующего леса и эллипсоидальная аппроксимация данных, рассмотрены в части 2 главы 1.

В данной работе применяется концептуальный подход к детектированию и прогнозированию аномалий, основанный на метрических итерационных алгоритмах, а также модельном подходе. Параметры системы дифференциальных уравнений в модельном подходе определяются с помощью решения задачи идентификации по измеренным данным, выступающим в роли векторов состояния системы, взятых в дискретные интервалы времени. Для решения задачи идентификации применяются численные методы интегрирования систем дифференциальных уравнений, сводящиеся к оптимизационной задаче в пространстве параметров модели.

1.2 Методы машинного обучения

Машинное обучение (Machine Learning) представляет собой класс методов искусственного интеллекта, для которого характерно не прямое решение задачи, а обучение в процессе применения решений множества сходных задач [78]. В процессе построения таких методов могут быть использованы средства математической статистики, численных методов, методов оптимизации, интеллектуального анализа данных, теории вероятностей, теории графов. Теоретические разделы машинного обучения были выделены в теорию вычислительного обучения (Computational Learning Theory, COLT). Машинное обучение используется в том случае, когда явное программирование высокопроизводительных алгоритмов является трудной или вовсе не осуществимой задачей. Однако качество настройки системы обучения напрямую зависит от качества и полноты обучающей выборки.

С помощью машинного обучения возможно изучение закономерностей в данных, которые впоследствии используются для выявления аномального поведения. Задачи машинного обучения обычно подразделяются на следующие категории в зависимости от наличия обучающего «сигнала» или «обратной связи», доступного для системы обучения.

- Обучение с учителем (supervised learning) заключается в обучении системы с помощью примеров «стимул-реакция» [77]. В качестве частных случаев выделяют способы машинного обучения, при которых входной сигнал может быть доступен частично или же ограничен специальной обратной связью:
 - обучение с частичным привлечением учителя (semi-supervised learning), когда для части прецедентов задается только тренировочный набор без выходных значений (например, методы на основе графов) [33, 89];
 - активное обучение (active learning) является частным случаем обучения с частичным привлечением учителя, особенностью которого

является способность интерактивного запрашивания эксперта [38, 81]. Используется в том случае, когда в выборке преобладают немаркированные данные, а ручная маркировка представляется трудозатратной. Некоторые алгоритмы активного обучения основаны на методе опорных векторов (support vector machines).

- В ходе обучения с подкреплением (reinforcement learning) система обучается, взаимодействуя с некоторой средой и получая в ходе обучения сигналы подкрепления от системы. Обучение модели представимо виде выполнения наборов правил, на основании которых с течением времени изменяется матрица взаимодействия модели и среды (например, алгоритм Q-learning [85]).
- Обучение без учителя (unsupervised learning) предполагает обучение системы без вмешательства со стороны эксперта [50]. Подходит для задач, объекты в которых подробно описаны, и требуется установить внутренние взаимосвязи между объектами. Методы решения – метод ближайших соседей (k-means) [53, 73], графовые алгоритмы кластеризации (иерархическая кластеризация) [74].

Далее в качестве методов, использованных для детектирования аномалий, будут рассмотрены метод опорных векторов, изолирующий лес и эллипсоидная аппроксимация данных.

Метод опорных векторов для одного класса

Как было сказано выше, классический метод опорных векторов относится к алгоритмам обучения с учителем, называемым также методом классификатора с максимальным зазором. Основная идея метода заключается в переводе исходных векторов в пространство более высокой размерности и построении разделяющей гиперплоскости в многомерном пространстве [4, 7, 11]. Если обучающие данные линейно разделяемые, то дополнительно строят две параллельные разделяющей гиперплоскости таким образом, чтобы точки классов, ближайšie к разделяющей гиперплоскости, принадлежали этим

гиперплоскостям. Эти точки являются опорными векторами. Построенные гиперплоскости образуют пространство, свободное от объектов классификации, и дальнейшая задача состоит в том, чтобы достичь максимального значения ширины этого пространства. Разделяющую гиперплоскость в этом случае называют оптимальной. Размерность пространства соответствует количеству классифицирующих признаков, а их значение определяет положение точек в пространстве. Таким образом, метод выполняет разделение объектов, а также в нем подбирается гиперплоскость таким образом, чтобы та была максимально удалена от ближайшего элемента каждой из групп (рис. 1.12). Следует отметить, что такой подход может быть использован для решения задачи классификации, регрессии и поиска аномалий.

В случае если множество не является линейно разделимым, применяют увеличение размерности пространства и затем в уже измененном пространстве находят линейный разделитель. Построение разделяющей гиперплоскости в этом случае требует применения нелинейной функции ядра [7, 10].

Среди преимуществ подхода выделяют следующие пункты.

- a) является одним из самых быстрых методов нахождения решающих функций;
- b) сводится к решению задачи квадратичного программирования в выпуклой области, всегда имеющей единственное решение;
- c) находит разделяющее пространство максимальной ширины, что способствует более точной классификации.

В свою очередь принято выделять недостатки метода.

- a) неустойчив к шуму в данных;
- b) отсутствие общего подхода к выбору ядра в случае линейной неразделимости классов.

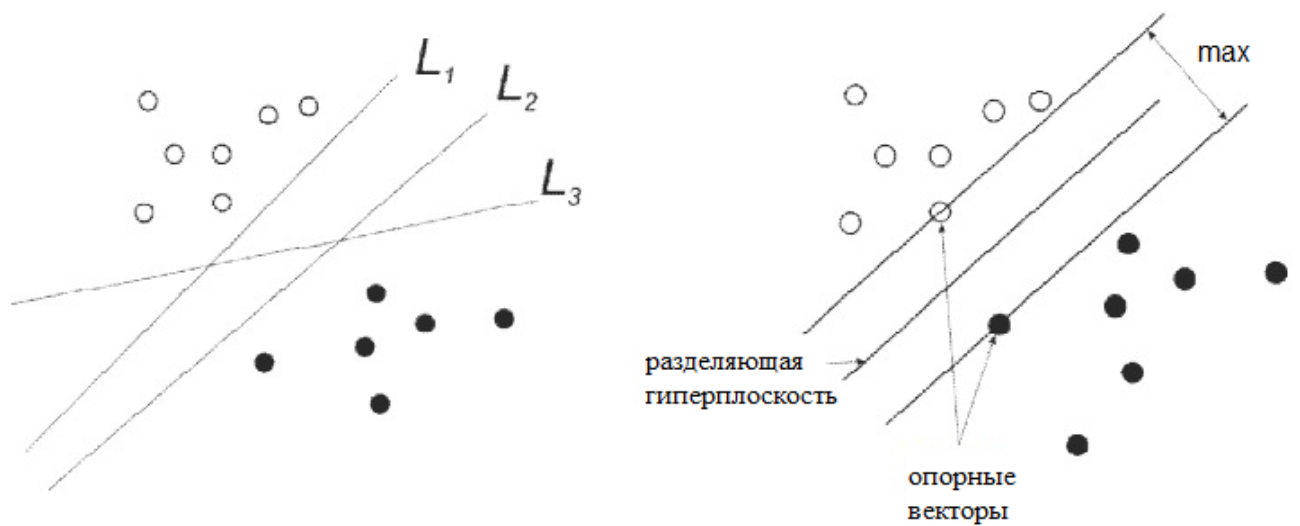


Рис. 1.12 – Примеры классификации элементов:
произвольная (слева) и оптимальная (справа)

Метод опорных векторов для одного класса (one-class SVM) является модификацией классического алгоритма метода опорных векторов [80], который отделяет выборку от начала координат. Несмотря на простоту идеи, она оказалась довольно работоспособной и применяется в различных областях. Следует отметить, что в отличие от стандартного метода опорных векторов, в данном случае в качестве ядра подойдёт лишь радиальные базисные функции, поскольку остальные нелинейные ядра показывают худшие результаты. Данный метод больше подходит для поиска новизны, а не выбросов, поскольку «затачивается» под обучающую выборку.

Изолирующий лес

Дерево решений (Decision tree) представляет собой инструмент поддержки принятия решений, в котором правила представлены в иерархической структуре древовидного графа, где каждый объект соответствует узлу, дающему решение (рис. 1.13) [30]. Корнем дерева является пространство признаков; в очередном узле случайным образом выбираются признак и порог разбиения, выбранный из равномерного распределения на отрезке от минимального до максимального значений выбранного признака. Критерием остановки служит тождественное

совпадение всех объектов в узле. Ответом в листе, который также соответствует оценке алгоритма, объявляется его глубина.

Аномальным значениям свойственно оказываться в листьях с небольшой глубиной, тогда как для разбиения кластера нормальных данных гиперплоскостями дереву требуется построить ещё несколько уровней, количество которых пропорционально размеру кластера, следовательно, пропорциональна и оценка для лежащих в нём точек. Таким образом, кластеры малых размеров, потенциально являющиеся аномальными, будут состоять из объектов с более низкой оценкой, чем объекты из кластеров нормальных данных.

В результате получается модель принятия решений, расчета вероятности наступления события, затраты ресурсов. Несмотря на существующие проблемы переобучения [67] и получения оптимального дерева решений, метод не требует подготовки данных, использует модель «белого ящика», объясняя ситуации при помощи булевой логики, позволяет оценить надежность модели при помощи статистических тестов.

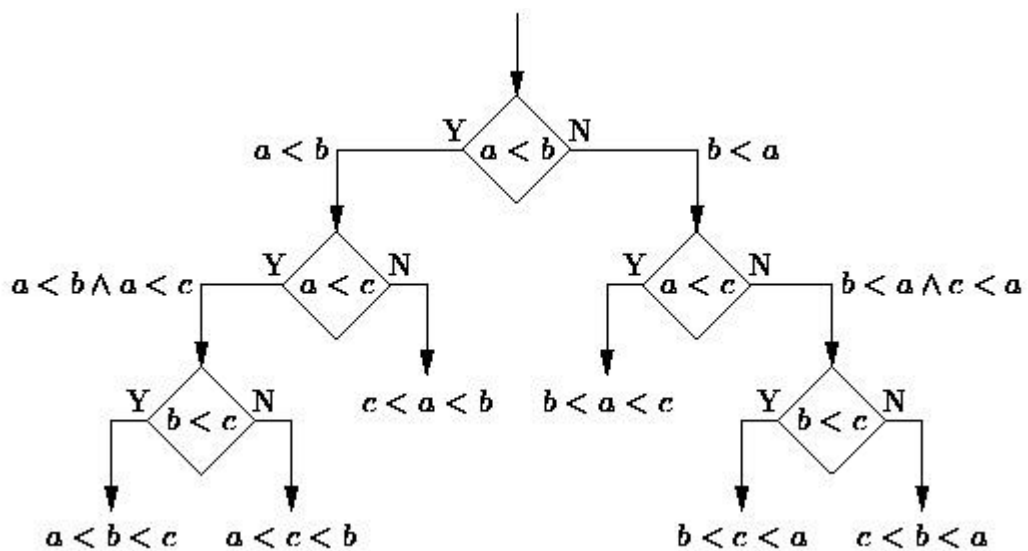


Рис. 1.13 – Пример дерева принятия решений

Изолирующий лес (Isolation forest) является ансамблевым методом обучения для классификации, регрессии и других задач, выполняющихся путем построения множества решающих деревьев во время обучения и вывода класса, который

является средним прогнозом (регрессия) отдельных деревьев или модой класса (классификация) (рис. 1.14) [46, 62]. При использовании большого количества решающих деревьев результат получается точным, в то время как каждое дерево в отдельности дает невысокое качество детектирования аномалий.

Основная идея метода основана на принципе Монте-Карло, поскольку производится случайное разбиение пространства признаков так, что в среднем изолированные точки становятся отсеченными от кластеризованных данных. Результат усредняется с помощью стохастического алгоритма..

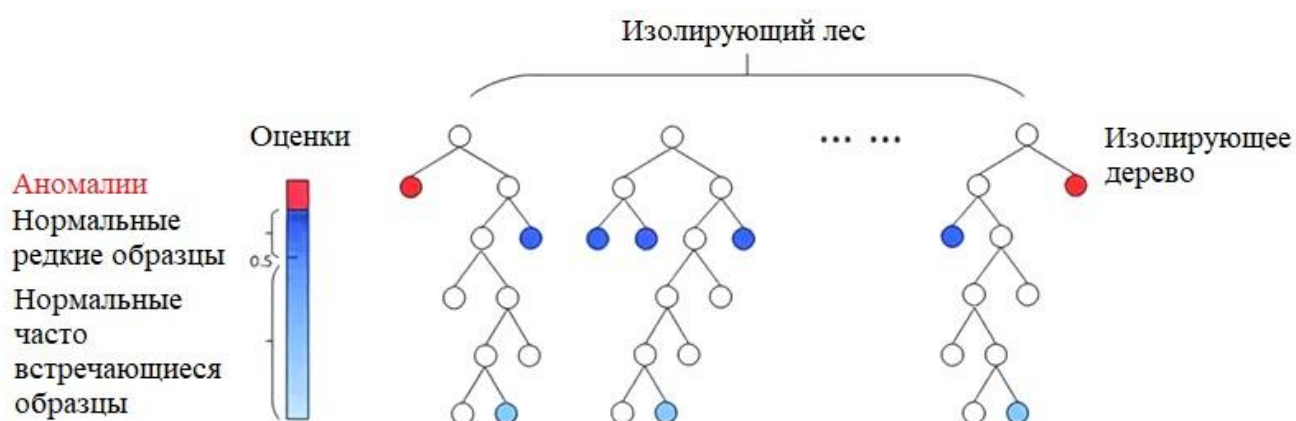


Рис. 1.14 – Схематическое представление изолирующего леса

Алгоритм изолирующего леса имеет ряд достоинств.

- Детектирование аномальных значений различного вида от изолированных точек с низкой локальной плотностью до кластеров аномалий малых размеров.
- Эффективность по сравнению с большинством прочих алгоритмов ввиду сложности $O(n \log n)$.
- При реализации метода отсутствуют параметры, требующие подбора.
- Инвариантность к масштабированию признаков.
- Не требует задания метрики и прочей информации об устройстве данных.
- Не требует значительных затрат памяти, например, в отличие от метрических методов, требующих построения матрицы попарных расстояний.

- Устойчивость к проклятию размерности, связанному с экспоненциальным возрастанием количества данных из-за увеличения размерности пространства (решением может выступать, например, метод главных компонент, осуществление отбора признаков, использование алгоритма вычисления оценок).

Эллипсоидальная аппроксимация данных

Одним из распространенных способов обнаружения аномалий появляется в предположении, что данные распределены каким-то известным способом, например, по Гауссу. В таком случае задача заключается в определении вида этого распределения и выделении тех объектов, которые не удовлетворяют найденному распределению.

В методе эллипсоидальной аппроксимации данных (Elliptic envelope) множество точек моделируется как внутренность эллипсоида [76]. Метод успешно работает только на нормально распределенных одномодальных данных. Степень аномальности объекта определяется по расстоянию Махаланобиса, в математической статистике являющимся мерой расстояния между векторами случайных величин и обобщающим понятие евклидова расстояния (рис 1.15).

В задаче определения принадлежности точки одному из классов необходимо найти матрицы ковариации всех классов, что, обычно, делается на основе известных выборок из каждого класса. При вычислении расстояния Махаланобиса до каждого класса выбирается тот класс, для которого это расстояние оказалось минимальным, что эквивалентно методу максимального правдоподобия (Maximum likelihood estimation, MLE).

Точка, имеющая наибольшее расстояние Махаланобиса до остального множества точек, считается аномалией. Такая точка имеет наибольшее влияние на кривизну и на коэффициенты уравнения регрессии. Также расстояние Махаланобиса может быть использовано в задаче определения многомерных выбросов.

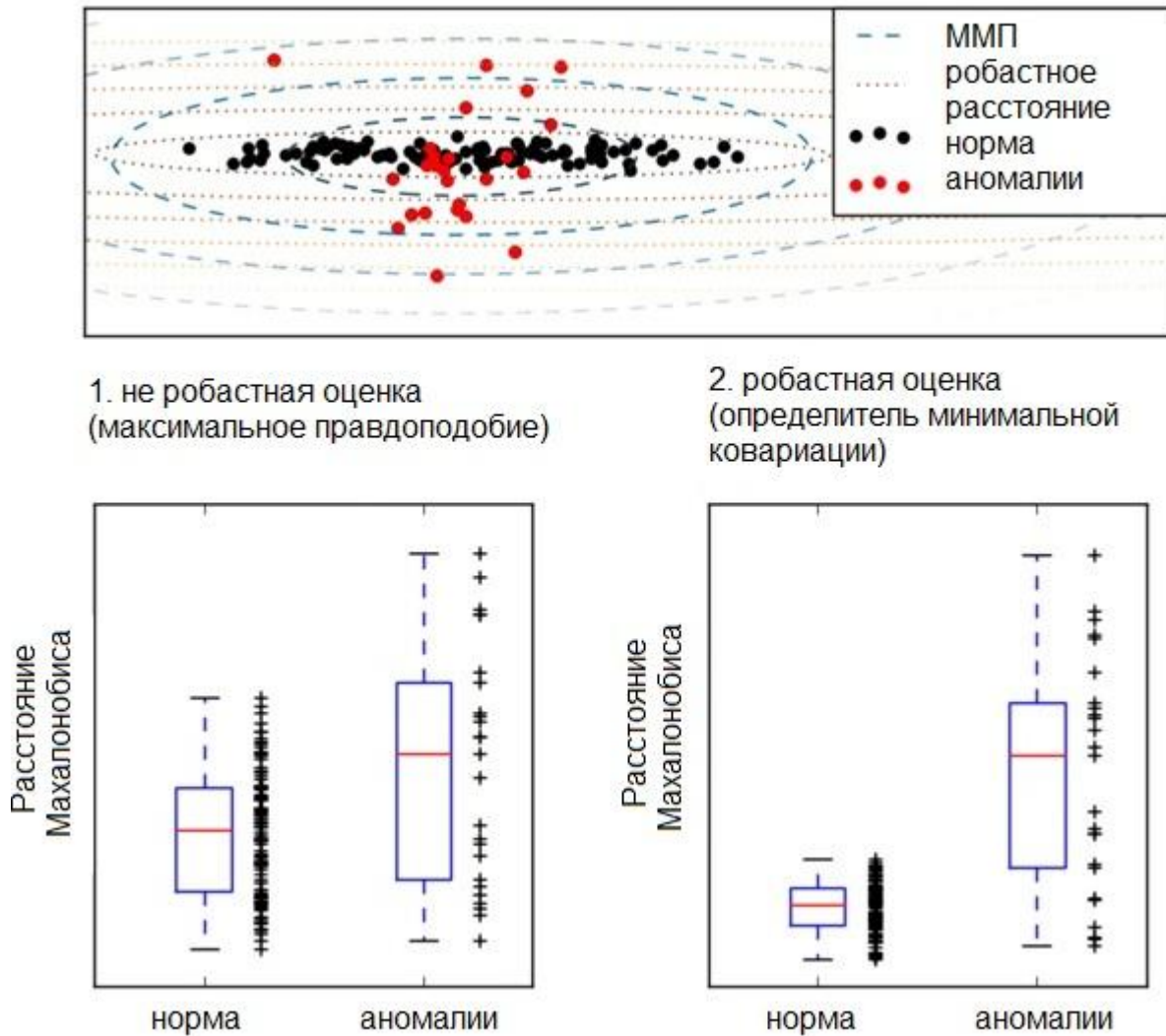


Рис. 1.15 – Расстояние Махаланобиса для набора исходных данных

В случае наличия нескольких кластеров в данных можно использовать модели гауссовых смесей (Gaussian mixture models, GMM) [79]. Гауссова смесь распределений представляет собой статистическую модель для описания нормально распределенных подгрупп данных внутри общей группы и параметризуется смесью весов и средних компонентов или ковариаций для многомерного случая. Если число компонентов (кластеров данных) известно, для оценки параметров смеси распределения используется EM-алгоритм (Expectation-maximization algorithm) [43]. В результате вычисления центроидов, матрицы ковариации, весов и логарифмических функций правдоподобия для каждой гауссовой компоненты получают вероятностные кластера, описывающие распределение данных. На основе вероятностной функции

правдоподобия возможно детектирование тем аномальных значений, которые не были кластеризованы, то есть с малой вероятностью принадлежат всем возможным гауссовым смесям.

1.3 Модельный подход в анализе маневров судна

Как было описано в части 1 главы 1, один из методов выявления аномальных значений заключается в построении модели, описывающей данные, и выявлении точек, отклоняющейся от неё. Преимуществами таких подходов являются возможность учета предметной области и задание требуемого функционала качества [52, 83].

Движение судна с некоторой точностью может быть описано динамической системой [86]. При этом сенсорные данные будут являться измерениями вектора состояния этой системы. В общем виде модель можно описать системой уравнений:

$$\begin{aligned}v_y' &= \beta_1 \frac{v_y}{v_x} + \left(\frac{\beta_2}{v_x} - v_x \right) \omega_z + \beta_3 \delta(t), \\ \omega_z' &= \beta_4 \frac{v_y}{v_x} + \beta_5 \frac{\omega_z}{v_x} + \beta_6 \delta(t), \\ x' &= v_x \cos(\varphi), \\ y' &= v_x \sin(\varphi), \\ \varphi' &= \omega_z,\end{aligned}\tag{1}$$

где v , ω – линейная и угловая скорости, функции β_i зависят от параметров судна (масса, размер, моменты инерция и т.п.), $\delta(t)$ – управление. В общем случае точный вид уравнений, с заданной точностью описывающий динамику конкретного судна, не известен. Задачей идентификации является определение параметров, входящих в описанную выше систему, по измеренным данным. Измеренные данные в данном случае выступают в роли векторов состояния системы, взятых в дискретные интервалы времени.

Для решения задачи идентификации применяются численные методы интегрирования систем дифференциальных уравнений. Существует два класса алгоритмов численного решения. К первому относятся пошаговые методы интегрирования (например, схемы Рунге – Кутты) [5]. Ко второму – методы построения отображений [8]. Последний заметно выигрывает у пошаговых

методов по производительности, так как отсутствует необходимость пошагового деления интервала интегрирования, а также в виду независимости коэффициентов матричного отображения от начальных условий интегрирования.

В обоих случаях решение задачи идентификации динамических систем сводится к оптимизационной задаче в пространстве параметров модели. Отметим, что в случае применения пошаговых методов параметрами являются переменные, явно входящие в систему дифференциальных уравнение. В случае применения отображений параметрами являются коэффициенты этого отображения, что позволяет решать задачу идентификации, не задавая вид дифференциальных уравнений. Далее будем подразумевать, что движение судна описывается системой дифференциальных уравнений

$$\frac{dX}{dt} = F(t, X),$$

правая часть которого раскладывается в степенной ряд. Данное требование выполняется в реальной ситуации в виду вида системы дифференциальных уравнений (1) и условия, что данные с судна приходят с достаточно частым интервалом. В нашем случае интервал между измерениями варьируется в диапазоне от 10 секунд до одной минуты, что в масштабах судна является достаточно маленькой величиной.

2 Математическое моделирование и численные методы обнаружения аномалий

Для верификации разработанных методов были использованы тестовые данные с судна, на котором в режиме тестовой эксплуатации была установлена система сбора данных с большого количества датчиков.

В рамках данной работы используются значения таких параметров, как [2]:

- скорость относительно воды (STW – Speed through water),
- скорость относительно грунта (SOG – Speed over ground),
- курс относительно грунта (COG – Course over ground),
- угловая скорость, скорость поворота (ROT – Rate of turn),
- истинный курс (HDG – Heading),
- глубина (Depth),
- дистанция кратчайшего сближения с другим судном (CPA – Closest point of approach),
- время кратчайшего сближения (TCPA – Time of closest point of approach).

Для моделирования потребления судном топлива используются исторические данные по его расходу.

В рамках тестирования алгоритмов использованы данные, собранные за месяц и представимые в виде множества:

$$X = \left\{ \left(t_i, x_{1,i}, x_{2,i}, \dots, x_{9,i} \right) \right\}_{i=1}^N, \quad (2)$$

где t_i обозначает время поступления данных, а $x_{1,i}, x_{2,i}, \dots, x_{9,i}$ – значения параметров, указанных выше в той же последовательности в момент времени t_i . Величина N определяет общее число сделанных измерений. Поскольку параметры измеряются датчиками с различной частотой, для нормализации данных используется простая линейная интерполяция по времени:

$$\frac{x_{k,j} - x_{k,i}}{x_{k,i+1} - x_{k,i}} = \frac{t_j - t_i}{t_{i+1} - t_i}, k = \overline{1,9},$$

такая, что соседние интервалы по времени равны постоянной величине $\Delta t = t_j - t_{j-1} = 1$ мин. Таким образом, в каждый момент времени рассматривается девятимерный вектор состояния судна.

Следует отметить, что для более эффективной работы алгоритмов, а также для наглядной визуализации применен метод главных компонент, основная идея которого заключается в поиске ортогональных проекций, на которые разброс данных максимален [42]. В этом случае множество (2) преобразуется к

$$X = \left\{ (t_i, PC_{1,i}, PC_{2,i}, \dots, PC_{9,i}) \right\}_{i=1}^N, \quad (3)$$

где $PC_{k,i}$ обозначает k -ую главную компоненту вектора состояния X в момент времени i .

В главе 2 описаны этапы предварительной обработки полученных с судна данных, формализован алгоритм изолирующего леса, использованный в качестве основного метода поиска аномалий, представлена модель прогнозирования потребления топлива, построенная на основе полиномиальных нейронных сетей.

2.1 Построение признакового пространства входных данных

Для подготовки данных к дальнейшей работе производилась их предварительная обработка, состоящая из трех этапов.

1. Фильтрация и удаление шума в данных.
2. Интерполяция данных для приведения данных к одним и тем же временным отсчетам.
3. Обезразмеривание данных для устойчивости работы алгоритмов детектирования аномалий.

Фильтрация данных

В качестве первичной обработки сенсорных данных производится удаление шума, для чего применяется усреднение значений методом скользящего среднего. В качестве размера окна выбирается время реакции судна [35], которое можно вычислить как временной интервал, за которое судно проходит свою длину.

Замеры навигационных данных могут производиться с интервалами от 0,2 секунды до минуты. Так как данные приходят с разных сенсоров в разных временных шкалах, для их объединения необходимо привести их одним временным отсчетам. Это достигается путем интерполяции значений для достижения фиксированного временного интервала $\Delta T = 10$ сек.

На рис. 2.1 приведена схема предварительной обработки данных. После фильтрации данные представляются в табличном виде с фиксированным интервалом отсчетов и заданным набором параметров.



Рис. 2.1 – Обработка и объединение данных из различных источников

Обезразмеривание переменных

Для достижения устойчивости работы алгоритмов поиска аномалий и масштабирования на разные типы судов произведено обезразмеривание входных данных, цель которого заключается в переходе от абсолютных значений к относительным [35].

Линейная скорость

При движении судна у носа и кормы создаются волны, увеличивающиеся с увеличением скорости движения, поскольку в таком случае в кормовой части возникает значительное разрежение, а в носовой – зона повышенного давления. Волновое сопротивление является энергией, израсходованной на образование волн и определяемой скоростью и длиной судна.

Число Фруда является безразмерной скоростью, позволяющей сопоставлять скорости разных судов с учетом их размера путем сравнения условия

волнообразования для этих судов [68]. Число Фруда может быть вычислено по формуле:

$$F = \frac{v}{\sqrt{gL}},$$

где v – линейная скорость судна, L – длина судна, g – ускорение свободного падения. Пример соотношений чисел Фруда с разными скоростями и длинами судна представлен в таблице 2.1.

Табл. 2.1 – Соотношений чисел Фруда с разными скоростями и длинами судна

Число Фруда	Скорость, узлы	Длина судна, метры
0.16	7	50
0.16	10	100
0.16	17	300
0.21	9	50
0.21	13	100
0.21	22	300
0.26	11	50
0.26	16	100
0.26	27	300

Угловая скорость

Резким считается поворот по дуге окружности, радиус которой меньше трех длин судна. Учитывая это, можно ввести безразмерную величину:

$$r = \frac{v}{3\omega L},$$

характеризующую степень кривизны поворота. Здесь под ω подразумевается угловая скорость судна. Чем меньше величина r , тем более резко осуществляется

поворот. Значение $r = 1$ соответствует пороговому значению допустимого радиуса поворота.

Линейное ускорение

Допустимым торможением считается такое замедление, при котором судно, двигаясь на максимальной скорости, полностью останавливается, пройдя путь в 15 длин судна. Исходя из этого положения:

$$15L = v_{\max} t + \frac{at^2}{2},$$
$$0 = v_{\max} + at,$$

можно вывести нормировочный коэффициент для ускорения:

$$15L = -v_{\max} \frac{v_{\max}}{a} + \frac{av_{\max}^2}{2a^2} = \frac{-v_{\max}}{2a} \Rightarrow a = \frac{-v_{\max}}{30L}.$$

Все полученные данные преобразуются в соответствии с формулами, приведенными в этой части работы, и после этого используются алгоритмами машинного обучения.

2.2 Метод изолирующего леса

Изолирующий лес является методом обнаружения аномалий, основной идеей которого является построение множества решающих деревьев во время обучения и вывода класса, который является средним прогнозом отдельных деревьев (регрессия) или модой класса (классификация) [46]. Необходимость использования ансамбля деревьев заключается в том, что при использовании большого количества решающих деревьев результат получается точным, в то время как каждое дерево в отдельности дает невысокое качество детектирования аномалий.

Алгоритм обучения заключается в следующем: пусть обучающая выборка состоит из N образцов, размерность пространства признаков равна M и задан параметр m как некоторое количество признаков для обучения. Наиболее распространённым способом построения деревьев комитета является нижепредставленный.

1. Генерация из обучающей выборки случайной подвыборки с повторениями размером N , при этом в среднем каждый образец войдет в подвыборку $N * (1 - 1/N)^N$ раз, N/e образцов не войдут в неё вообще.
2. Построение решающего дерева, классифицирующего образцы данной подвыборки: в ходе создания очередного узла дерева случайно выбираются m признаков, на основе которых производится разбиение. Для выбора наилучшего признака используется критерий Джини [72] или критерий прироста информации.
3. Классификация объектов путём голосования: каждое дерево комитета относит классифицируемый объект к одному из классов, в результате выявляется класс, за который проголосовало наибольшее число деревьев.

Оптимальное число деревьев подбирается с помощью минимизации ошибки классификатора на тестовой выборке или оценки ошибки неотобранных элементов. Мерой нормальности объекта является среднее арифметическое

глубин листьев, в которые он попал. Таким образом, аномальные значения попадают в листья на небольшой глубине дерева и тем самым детектируются. Достоинством этого метода являются возможность качественной обработки как непрерывных, так и дискретных данных с большим числом признаков и классов, в том числе с пропущенными значениями признаков; недостатком – объем памяти для хранения модели $O(K)$, где K – число деревьев.

Основную идею поиска опасных маневров можно продемонстрировать на примере анализа соотношений линейной и угловой скоростей судна. В нормальном режиме при значительных линейных скоростях судно должно двигаться прямолинейно. Случаи, когда у судна на высокой скорости наблюдается большая угловая скорость, соответствуют резким маневрам. На рис. 2.1 отображено подмножество множества точек $\{(x_{1,i}, x_{4,i})\}$.

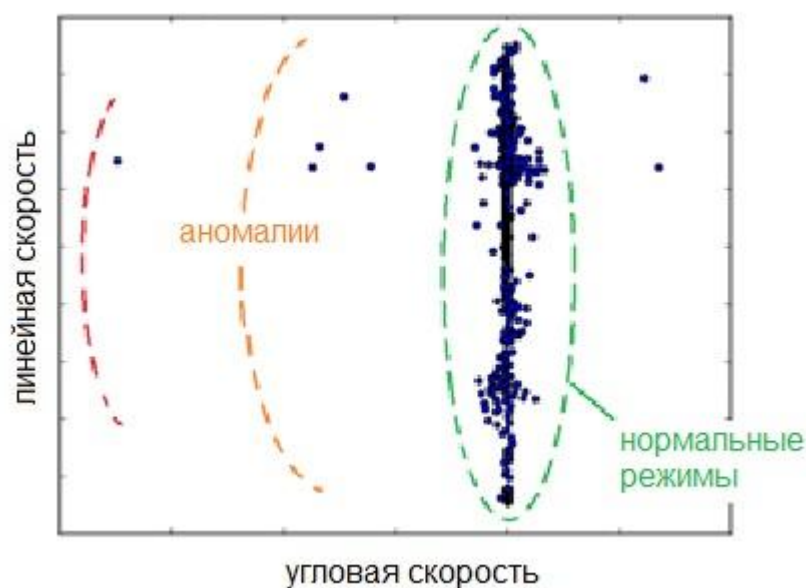


Рис.2.1 – Аномальные маневры в пространстве линейной и угловой скоростей

В общем случае на маневры судна оказывают влияние как минимум все параметры вектора X . Так, например, с точки зрения безопасности не желательно движение с большой скоростью на малых глубинах. Также на маневры оказывает влияние наличие рядом с судном других судов, которое может быть определено по величинам СРА и ТСРА. На рис. 2 представлен результат работы алгоритма поиска аномалий на всем

наборе входных данных, преобразованных методом главных компонент (2). Для визуализации использованы только три первые главные компоненты $\{PC_{1,i}, PC_{2,i}, PC_{3,i}\}$.

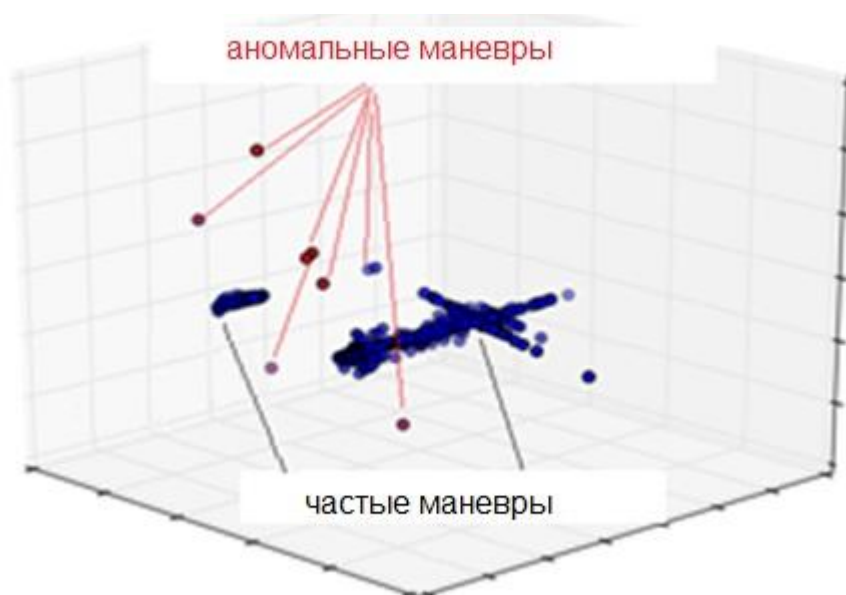


Рис.2.2 – Аномальные маневры в пространстве главных компонент

На рис. 2.2 могут быть выделены несколько кластеров, описывающих маневры. Возможным объяснением таких групп может служить наличие различных режимов движения судна, к примеру, на открытой воде, в порту, при поворотах.

Найденные алгоритмом изолирующего леса аномалии могут быть ранжированы по близости к разделяющей гиперплоскости либо при помощи использования метрических методов. Данное значение можно интерпретировать как вероятность аномалии и использовать в качестве параметра для настройки количества обнаруженных аномалий за определенный промежуток времени. В данной работе использовалась встроенная в библиотеку `scikit-learn` [48, 66] функция $f(x) = decision_function(x)$, вычисляющая для каждой точки множества (2) “меру нормальности” как глубину листа, содержащего эту точку, что эквивалентно количеству разветвлений, необходимых для ее изоляции:

$$F(X) = \{t_i, f(x_{1,i}, x_{2,i}, \dots, x_{9,i})\}_{i=1}^N, \quad (4)$$

Несмотря на то, что методы поиска аномалий изначально рассматриваются как полностью автоматические, предметная область может накладывать определенные ограничения. Так, например, при анализе маневров с точки зрения безопасности движения особый интерес представляют только относительно большие скорости судна. В данном случае практически невозможно добиться от метода автоматического извлечения подобной информации из данных. Требуется либо тонкая настройка параметров алгоритма, либо предварительная трансформация данных. Так, на рис. 2.3 представлен результат работы алгоритма изолирующего леса на множестве точек $\left\{ \left(x_{1,i}, \frac{x_{4,i}}{x_{1,i}} \right) \right\}$. Здесь величина $\frac{x_{4,i}}{x_{1,i}}$ равна отношению угловой и линейной скоростей и определяет кривизну траектории. Движение с нулевой кривизной соответствует движению по прямой. Красным цветом обозначены точки, в которых значение нормированной в интервале $[0; 1]$ функции f больше 0,95. Из рисунка видно, что аномалии обнаруживаются в том числе в области низких скоростей, что не представляет интереса.

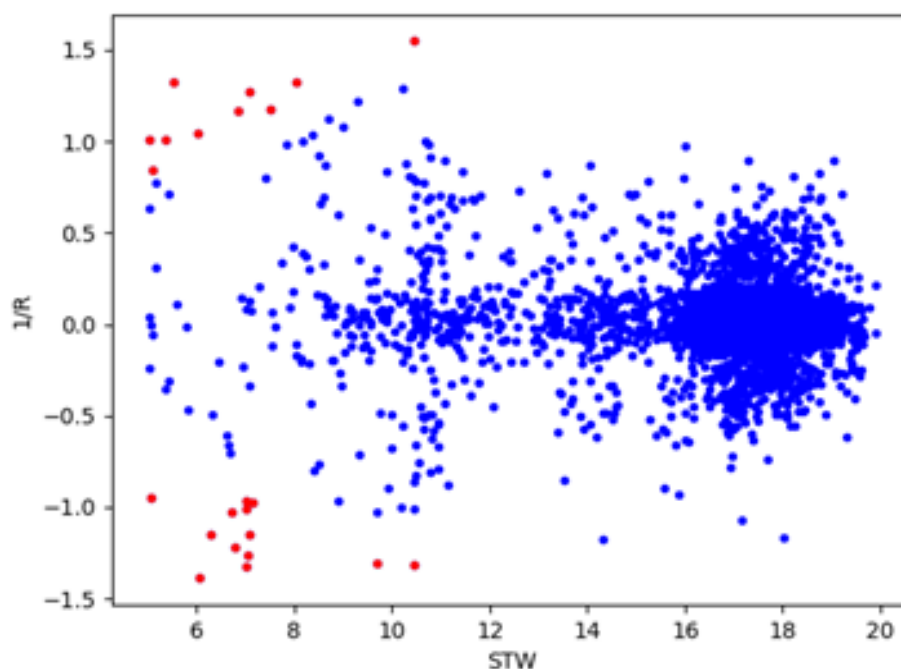


Рис. 2.3 – Аномальные маневры в пространстве скорости (STW) и кривизны траектории (R – мгновенный радиус поворота)

Для решения подобного рода проблем требуется предварительная экспертная фильтрация данных. Например, на рис. 2.4 приведена работа алгоритма изолирующего леса на данных, первоначально подвергнутых такой экспертной фильтрации, поскольку учитывались только те скорости, что превышали 12 узлов:

$$\tilde{X} = \left\{ (t_i, x_{1,i}, x_{2,i}, \dots, x_{9,i}) \mid x_{1,i} > 12 \right\}_{i=1}^N.$$

На рис. 2.4 зеленым цветом обозначены точки, которые не интересны в задаче обнаружения аномалий ввиду невысокой скорости движения судна. Синим – нормальные ситуации маневрирования судна. Красным и оранжевым отмечены подозреваемые в аномальности объекты с высоким значением функции f .

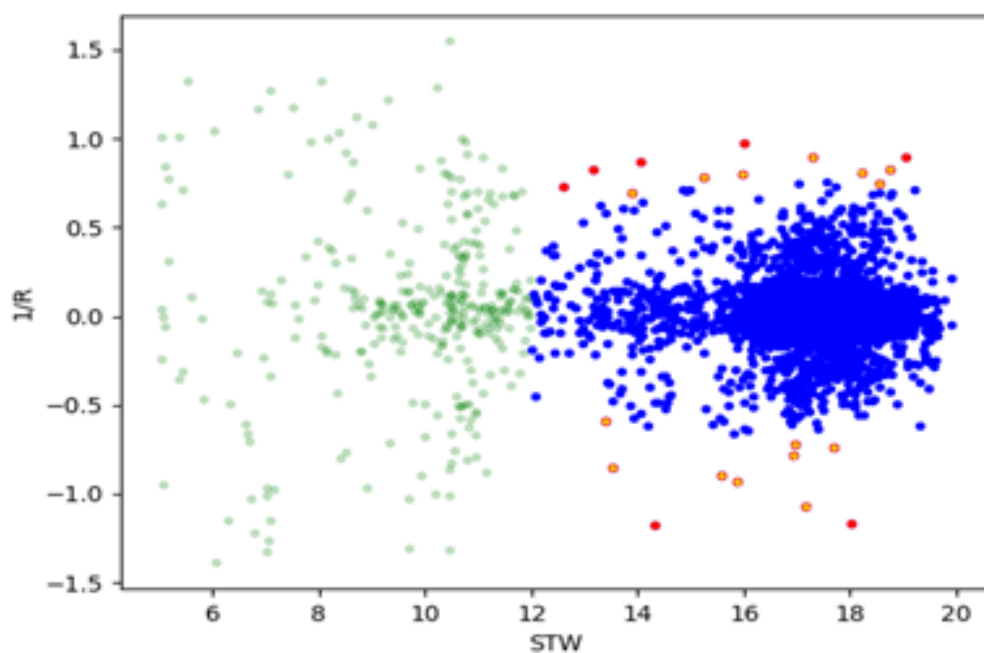


Рис. 2.4 – Аномальные маневры в пространстве скорости (STW) и кривизны траектории (R – мгновенный радиус поворота) с предварительной фильтрацией

Детектированные аномалии могут быть отображены и на маршруте судна в географических координатах. На рис. 2.5 представлен маршрут с обнаруженными выше аномалиями маневров.

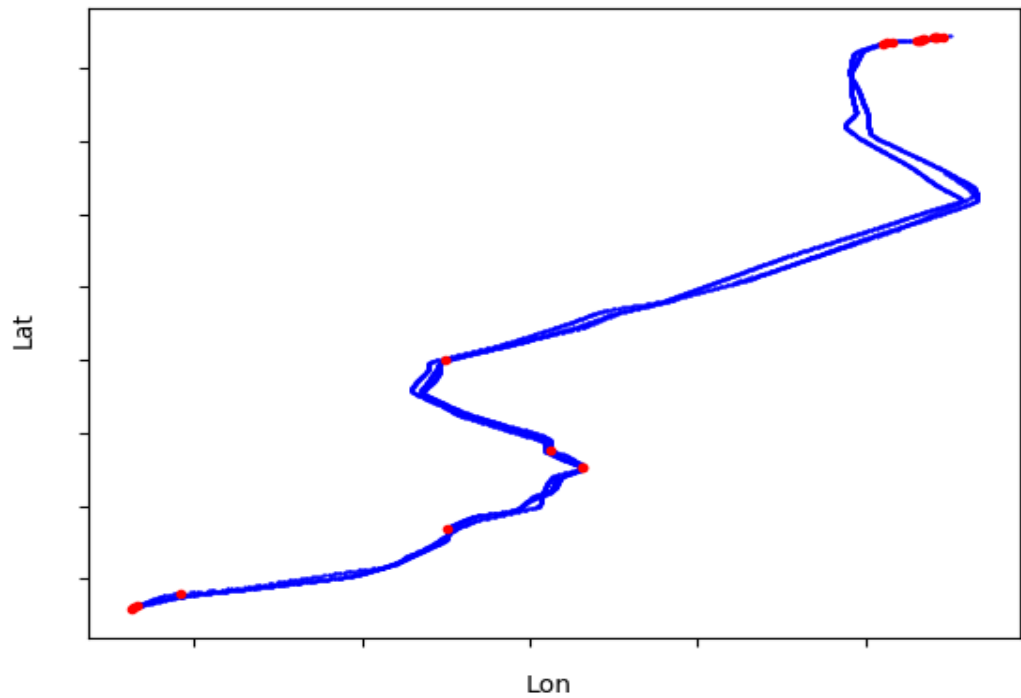


Рис. 2.5 – Аномальные маневры в пространстве скорости (STW) и кривизны траектории (R – мгновенный радиус поворота) с привязкой к географическим координатам (Lon – долгота, Lat – широта)

Стоит отметить, что резкие маневры могут быть вызваны разными причинами, такими как уход от столкновения, корректировка маршрута и др. Кроме того, возможны и выбросы в данных, когда аномалии являются случайным шумом. Предполагается, что анализ причин аномальных ситуаций в дальнейшем будет производиться дополнительно с использованием статистических тестов и экспертной оценки.

2.3 Матричный формализм для решения задачи идентификации

При моделировании и идентификации сложных динамических систем часто возникают задачи учета эффектов нелинейности и неопределённости. Кроме того, при практической реализации подходов требуется принимать во внимание вычислительную сложность алгоритмов, которая должна быть достаточно простой для обработки большого количества данных.

Нейронные сети как инструмент моделирования систем могут выступать одним из вариантов решения указанных проблем. Применение нейронных систем в области исследования динамических систем отражено в различных приложениях и предметных областях, таких как решение обыкновенных дифференциальных уравнений [56, 63], обработка сигналов [57], системы управления [59], теория моделирования и идентификации динамических систем [34] и прочих. В указанных работах подходы к применению нейронных сетей можно разделить на две группы. К одной группе следует отнести методы решения заранее известных систем дифференциальных уравнений при помощи нейронных сетей. В другой группе рассматривается использование аппарата искусственных нейронных сетей для идентификации динамических систем и решения обратных задач. В этом случае вид уравнений не задан, либо задан с точностью до параметров, а обученная нейронная сеть является моделью рассматриваемого процесса или динамического объекта.

В обоих случаях применимость нейронных сетей объясняется лишь универсальной теоремой аппроксимации [37] без строгих математических доказательств, почему именно предлагаемая нейронная сеть применима в рассматриваемой проблеме. Кроме того, такое моделирование на основе черного ящика не может гарантировать сохранение физических инвариантов, которые часто возникают в динамических системах. Такие инварианты играют ключевую роль в компьютерном моделировании и позволяют сохранять физические свойства реальных систем, однако практически не рассматриваются в теории искусственных нейронных сетей.

В данной работе рассматривается архитектура нейронной сети, которая решает указанные проблемы для достаточно широкого класса динамических систем, которые описываются системой нелинейных дифференциальных уравнений:

$$\frac{dX}{dt} = F(t, X), \quad (5)$$

где t является независимой переменной, $X \in \mathfrak{R}^n$ задает вектор состояния системы. На функцию F накладывается ограничение о разложимости в ряд Тейлора относительно вектора состояния X , переменная t может входить в уравнение как произвольная нелинейная функция [39]. Подобные системы описывают процессы в различных областях, таких как автоматическое управление, робототехника, механические и биологические системы, химические реакции, разработка лекарств, молекулярная динамика.

Рассматриваемая архитектура полиномиальной нейронной сети является матричным оператором Ли [1, 39, 49], применяемым для описания динамических систем. Такое представление оператора Ли позволяет объединить вычислительную параллельную архитектуру и гибкую возможность учета неопределенностей, присущих нейронным сетям, со строгой теорией динамических систем на основе обыкновенных дифференциальных уравнений.

Динамика вектора X в уравнении (5) может быть описана при помощи хронологически упорядоченной экспоненты [39]

$$M(t|t_0) = T \exp\left(\int_{t_0}^t \mathfrak{F}_F(\tau) d\tau\right),$$

задаваемой оператором Ли $\mathfrak{F}_F(\tau)$. В предположении, что функция F в уравнении (5) может быть представлена в виде степенного ряда

$$F = \sum_{k=0}^{\infty} P_k(t) X^{[k]},$$

искомое решение внутри области своей сходимости также может быть представлено в виде полиномиального преобразования

$$X(t) = M(t) \circ X_0 = \sum_{k=0}^{\infty} R_k(t) X_0^{[k]}. \quad (6)$$

Здесь под $X^{[k]}$ понимается k -ая кронекеровская степень вектора X . В работе [1] показано, как можно вычислять матрицы $R_k(t)$ либо в символьном, либо в численном виде. Идея алгоритмов заключается в замене дифференциального уравнения (5) относительно вектора состояния X системой дифференциальных уравнений относительно коэффициентов матриц $R_k(t)$ [9]. Интегрируя уравнение (6) по времени

$$\frac{dX}{dt} = \sum_{k=0}^n \frac{d}{dt} R_k(t) X_0^{[k]},$$

и сопоставляя его с уравнением (5), в котором подставлены степенные разложения векторов $X^{[k]}$

$$\begin{aligned} \frac{dX}{dt} &= P_0(t) + P_1(t)X + P_2(t)X^{[2]} + \dots + P_p(t)X^{[p]} = \\ &= P_0(t) + \\ &+ P_1(t)(R_0(t) + R_1(t)X_0 + \dots + R_k(t)X_0^{[k]}) + \\ &+ P_2(t)(R_0(t) + R_1(t)X_0 + \dots + R_k(t)X_0^{[k]})^{[2]} + \\ &+ \dots + \\ &+ P_p(t)(R_0(t) + R_1(t)X_0 + \dots + R_k(t)X_0^{[k]})^{[p]}, \end{aligned}$$

можно получить новую систему дифференциальных уравнений

$$\begin{aligned} \frac{dR_0(t)}{dt} &= \sum_{k=0}^n P_k(t)(R_k(t))^{[k]}, \\ \frac{dR_j(t)}{dt} &= \sum_{k=0}^p P_k(t) \frac{\partial (X(t))^{[k]}}{\partial (X_0^{[j]})^T}, \end{aligned} \tag{7}$$

где вектор X задается равенством (6), а под $(\cdot)^T$ понимается операция транспонирования. Данную систему уравнений (7) можно решить подходящим численным методом интегрирования дифференциальных уравнений с нулевыми начальными условиями $R_1(t_0) = E, R_k(t_0) = \Theta$ при $k \neq 1$.

Предлагаемая архитектура нейронной сети реализует полиномиальное преобразование (6) до заданного порядка нелинейности k

$$Y = W_0 + W_1X + W_2X^{[2]} + \dots + W_kX^{[k]}, \tag{8}$$

в котором $X \in \mathcal{R}, Y \in \mathcal{R}^m, W_i$ – матрицы весовых коэффициентов соответствующих размерностей. Для примера, на рис. 2.6 отображено графическое представление такой нейронной сети до третьего порядка нелинейности. На схеме входной вектор $X = (x_1, x_2)$ последовательно преобразуется в кронекеровские степени $X^{[2]} = (x_1^2, x_1 x_2, x_2 x_1, x_2^2)$ и $X^{[3]} = (x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3)$, которые затем умножаются на матрицы весовых коэффициентов.

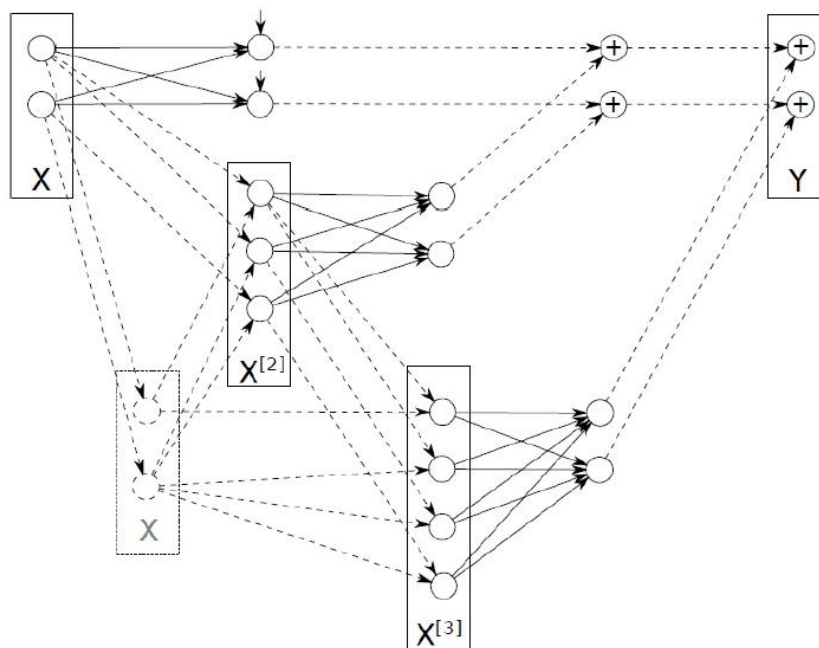


Рис. 2.6 – Нейросетевое представление матричного преобразования Ли

Такая архитектура нейронной сети описывает динамическую систему (5) и может быть использована для решения задачи идентификации. В зависимости от сложности искомой системы можно выбрать отображение требуемого порядка. При этом задача идентификации сводится к поиску коэффициентов матриц W_i . Отметим также, что наложение физических свойств и инвариантов на рассматриваемую модель приводит лишь к дополнительным ограничениям, накладываемым на искомые коэффициенты. Например, в работе [1] представлен алгоритм симплектификации отображения (6), который приводит к тому, что некоторые из весовых коэффициентов нейронной сети будут связаны между собой.

Основным преимуществом предлагаемой архитектуры является тот факт, что веса нейронной сети связаны с динамикой системы в целом и интерпретируются как соответствующие нелинейные члены в уравнениях. Следует также отметить, что если система дифференциальных уравнений полностью определена, веса можно вычислить с заданной точностью вплоть до требуемого порядка нелинейности. Однако если исходное уравнение неизвестно или задано с некоторой точностью, указанный подход можно применять для решения задачи идентификации системы по измеренным векторам состояния.

3 Программная реализация и вычислительный эксперимент

3.1 Архитектура и логика работы программных компонент

Для реализации поиска аномалий на основе модельного подхода была разработана программа Полифит, которая может применяться как в научных исследованиях при изучении нелинейных систем обыкновенных дифференциальных уравнений, так и при решении практических задач машинного обучения. Программа состоит из двух частей [16]. Первая часть позволяет решать задачу Коши на основе построения матричного отображения Ли. Вторая часть реализована в виде полиномиальной нейронной сети и нацелена на решение задачи идентификации динамической системы.

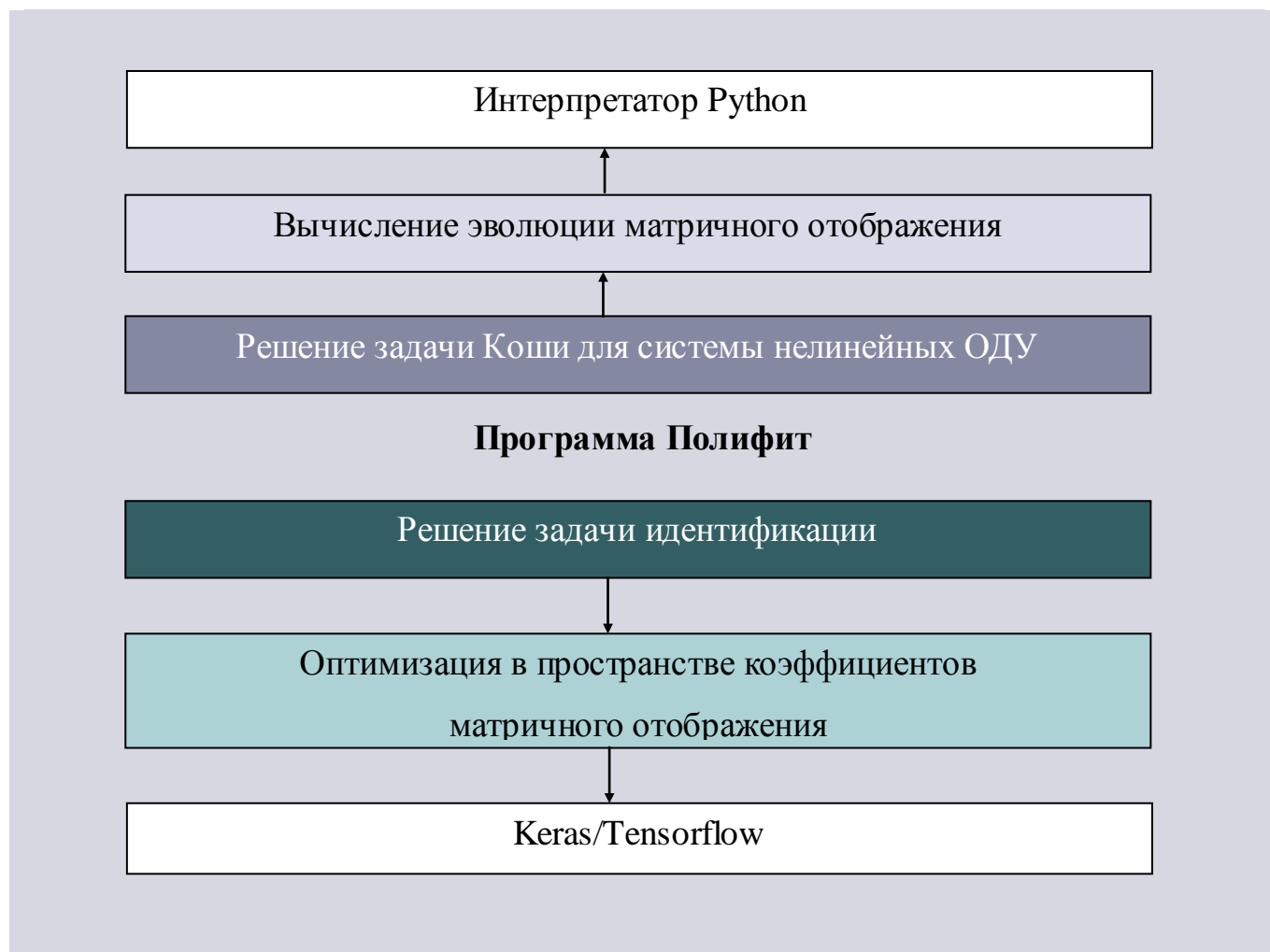


Рис. 3.1 – Реализация полиномиальной нейронной сети на Keras\Tensorflow

Программа Полифит реализует алгоритмы, описанные в части 3 главы 2. Реализация осуществлена на языке Python, программный интерфейс приведен в приложении В.

Решение задачи Коши для системы нелинейных ОДУ

В отличие от пошаговых методов интегрирования, примером которых может служить семейство методов Рунге – Кутта, применение отображения не требует итеративного пересчета каких-либо коэффициентов. Основной задачей является вычисление этого отображения. В соответствии с приведенными в части 3 главы 2 алгоритмами процесс построения отображения выглядит следующим образом.

1. **Разложение правых частей** системы уравнений в степенной ряд по векторам состояния. Данный этап может быть выполнен в любой системе символьных вычислений (например, Maple или Maxima). На вход программе Полифит требуется подать вычисленные степенные ряды (коэффициенты при мономах компонент вектора состояния).
2. **Задание метода** численного интегрирования и **временного интервала**, на котором оценивается эволюция отображения. В данный момент программа Полифит поддерживает методы Эйлера, Рунге – Кутта 4-го порядка и его симплектическую версию на основе неявной схемы.
3. **Сохранение вычисленных коэффициентов матричного отображения** в текстовом виде, в скрипте на языке MATLAB или в коде C++.
4. **Решение задачи Коши** в применении матричного отображения к начальному вектору состояния.

Полиномиальная нейронная сеть для решения задачи идентификации систем

В виду того, что предлагаемый метод в программе Полифит реализован в виде нейронной сети на платформе Keras\Tensorflow, алгоритм обучения модели полностью соответствует парадигме этой платформы.

1. Подготовка **данных для обучения** заключается в построении временных рядов, описывающих динамику вектора состояния. Так как матричное отображение строится для фиксированного интервала времени, интервал между измерениями должен быть также фиксирован.
2. В качестве **предсказываемых** значений используется сдвинутый на единичный интервал времени временной ряд.
3. Обучение нейронной сети выполняется платформой Keras\Tensorflow, при этом можно выбрать любой доступный алгоритм оптимизации.

3.2 Валидация программного обеспечения

В этой части рассмотрено применение предлагаемой в части 3 главы 2 архитектуры нейронных сетей для идентификации модели Лотки – Вольтерра [18]. Полиномиальная нейронная сеть на основе матричного представления оператора Ли сравнивается с классическим перцептроном (полносвязная нейронная сеть с сигмоидальными функциями активации). Идентификация отображения производится путем настройки коэффициентов нейронной сети на одном частном решении. Затем проверяется способность нейронной сети оценивать другие частные решения системы, соответствующие различным начальным условиям.

Уравнение Лотки – Вольтерра может служить простейшим примером нелинейных систем

$$\begin{aligned}\dot{x} &= ax - \beta xy, \\ \dot{y} &= \delta xy - \gamma y.\end{aligned}$$

Система имеет стационарную точку, а частные решения описывают осцилляции. Например, на рис. 3.2 и рис. 3.3 представлены примеры частных решений для обучающего (*training data*) и тестового (*test data*) наборов данных.

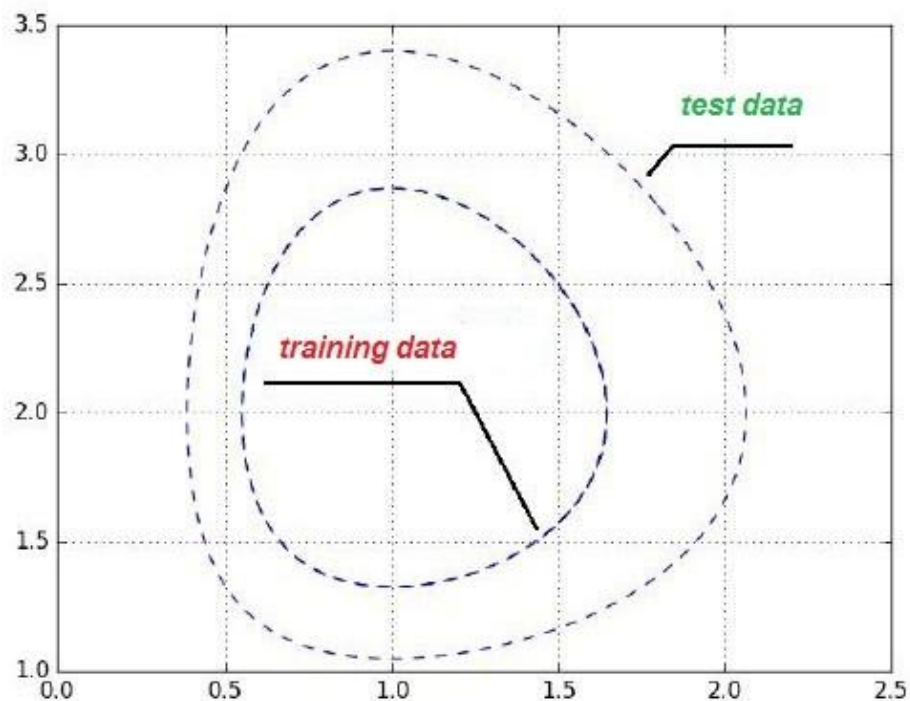


Рис. 3.2 – Визуализация решений модели Лотки – Вольтерра в фазовом пространстве

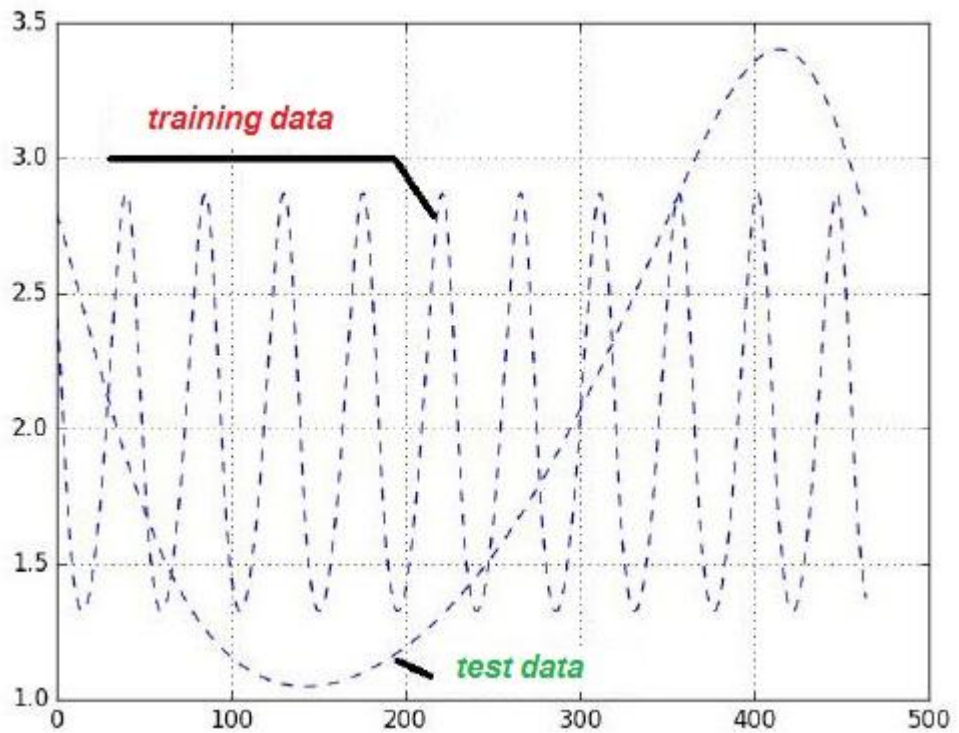


Рис. 3.3 – Визуализация решений модели Лотки – Вольтерра
во временном пространстве

Коэффициенты нейронных сетей подбираются при помощи методов численной оптимизации только по набору данных *training data*. На рис. 3.4 и рис. 3.5 представлены результаты работы обученных нейронных сетей.

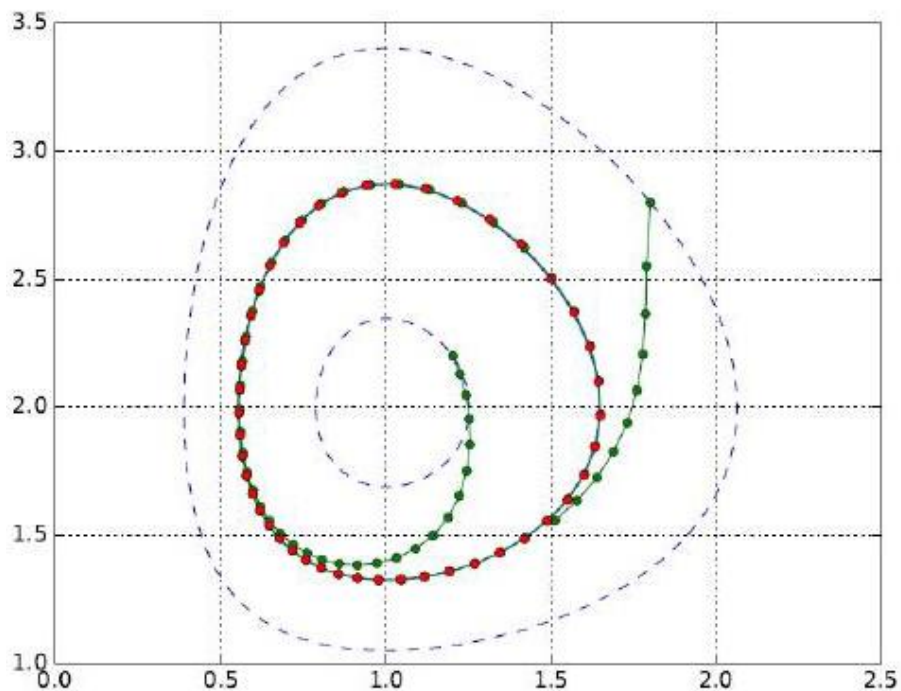


Рис. 3.4 – Оценка динамики системы при помощи обученного многослойного перцептрона

Классическая нейронная сеть оказывается неспособной оценивать траектории решений системы, отличные от той, на которой обучалась. При попытке построить частные решения, соответствующие другим начальным условиям, такая нейронная сеть стремится сойтись к траектории, предоставленной ей на этапе обучения (рис. 3.4).

Полиномиальная нейронная сеть успешно распознает характер осцилляции решений (рис. 3.5) и корректно оценивает динамику системы при других начальных условиях. Отметим, что оценка точности и сходимости предлагаемого подхода выходит за рамки данной работы и требует отдельного исследования.

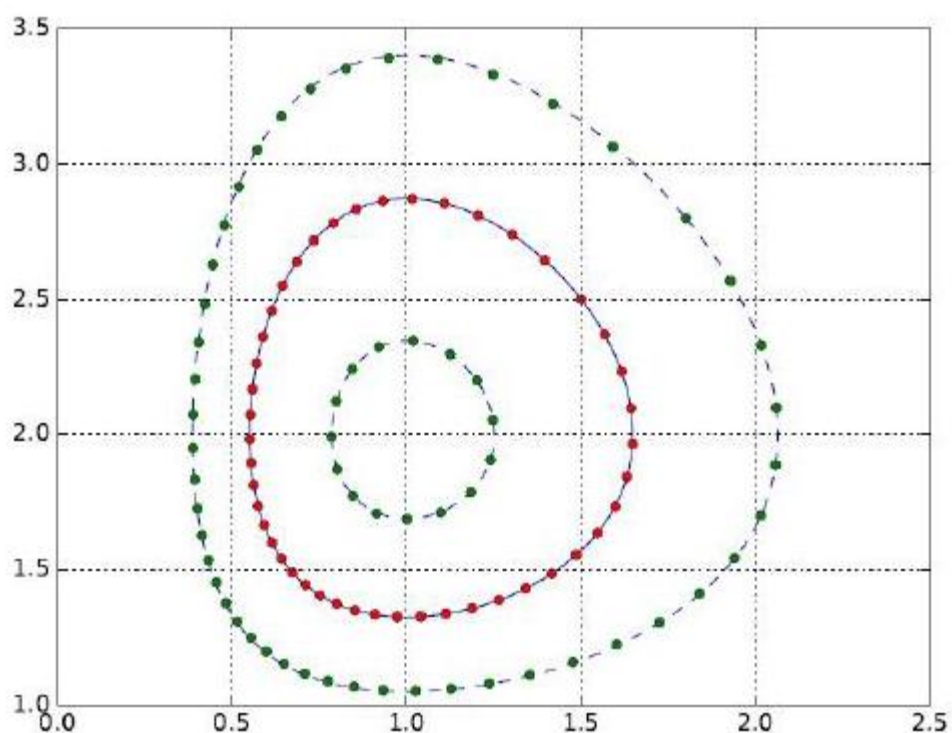


Рис. 3.5 – Оценка динамики системы при помощи полиномиальной нейронной сети

3.3 Вычислительный эксперимент

3.3.1 Обнаружение аномалий маневрирования

В качестве практического применения предлагаемой в части 3 главы 2 архитектуры нейронной сети рассмотрена задача идентификации модели движения судна. В работе [86] представлено несколько систем дифференциальных уравнений, описывающих динамику движения судна. Отметим, что уравнения могут иметь различные вектора состояния, поскольку на разных судах могут присутствовать разные датчики (направление и скорость ветра, параметры качки, характеристики работы двигателей и др.). Кроме того, коэффициенты этих уравнений отличаются от судна к судну и зависят от их физических параметров. Однако в общем виде все модели представимы в виде уравнения (5) внутри некоторого дискретного шага по времени.

При решении задач поиска аномалий в движении судна или прогнозирования потребления топлива [15] возникает необходимость наличия общей модели движения судна и ее идентификации по измеренным данным. В качестве примера используются данные о движении судна в течение месяца с дискретностью в одну минуту. Всего собрана информация о 16ти маршрутах [17]. На рис. 3.6 представлены нормализованные траектории, отмеченные разными цветами.

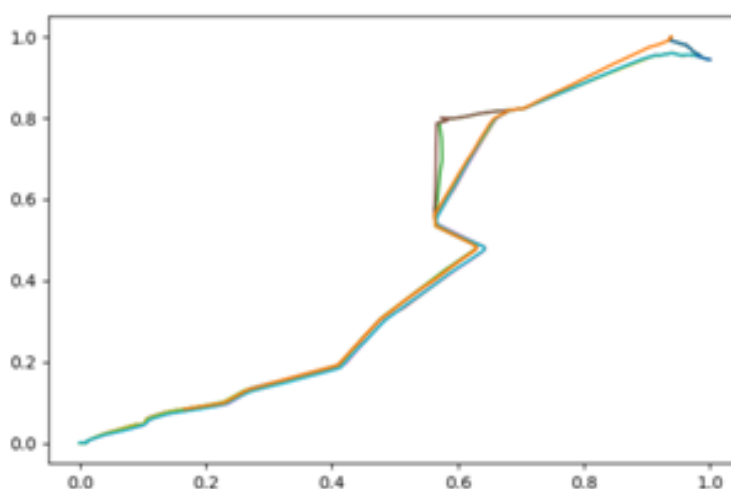


Рис. 3.6 – Нормализованные координаты траекторий движения судна, описывающих каждую траекторию

Полный вектор состояния судна описывается набором из 8ми параметров

$$X_i = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)_i,$$

где i обозначает дискретный интервал времени, а переменные x_j – географические широту и долготу, скорость судна относительно воды и земли, глубину, загрузку двигателя, направление и скорость ветра.

Задачей идентификации модели движения судна является построение такой нейросетевой модели, оптимальным образом отражающей динамику векторов состояния X_i . В такой постановке задача сводится к восстановлению регрессии (8)

$$Y \equiv (x_1, x_2)_{i+1} = W_0 + W_1 X_i + W_2 X_i^{[2]} + \dots + W_k X_i^{[k]},$$

когда географические координаты предсказываются по предыдущему вектору состояния. Для решения этой задач были использованы полиномиальные нейронные сети второго порядка нелинейности, которые обучались для каждого маршрута по отдельности.

Отметим, что наиболее значимы вклад в модель вносят первые коэффициенты, соответствующие преобразованиям координат и скорости в линейном приближении (рис. 3.7).

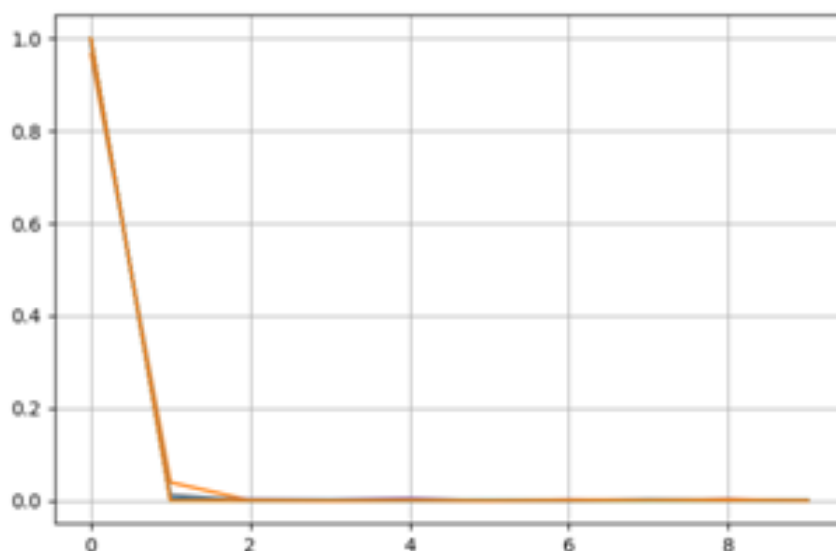


Рис. 3.7 – Коэффициенты полиномиальных нейронных сетей, описывающих каждую траекторию

Однако если проанализировать первые десять коэффициентов в увеличенном масштабе, можно заметить, что одна из построенных моделей заметно выбивается от усредненных значений. На рис. 3.8 такой маршрут отмечен красным цветом.

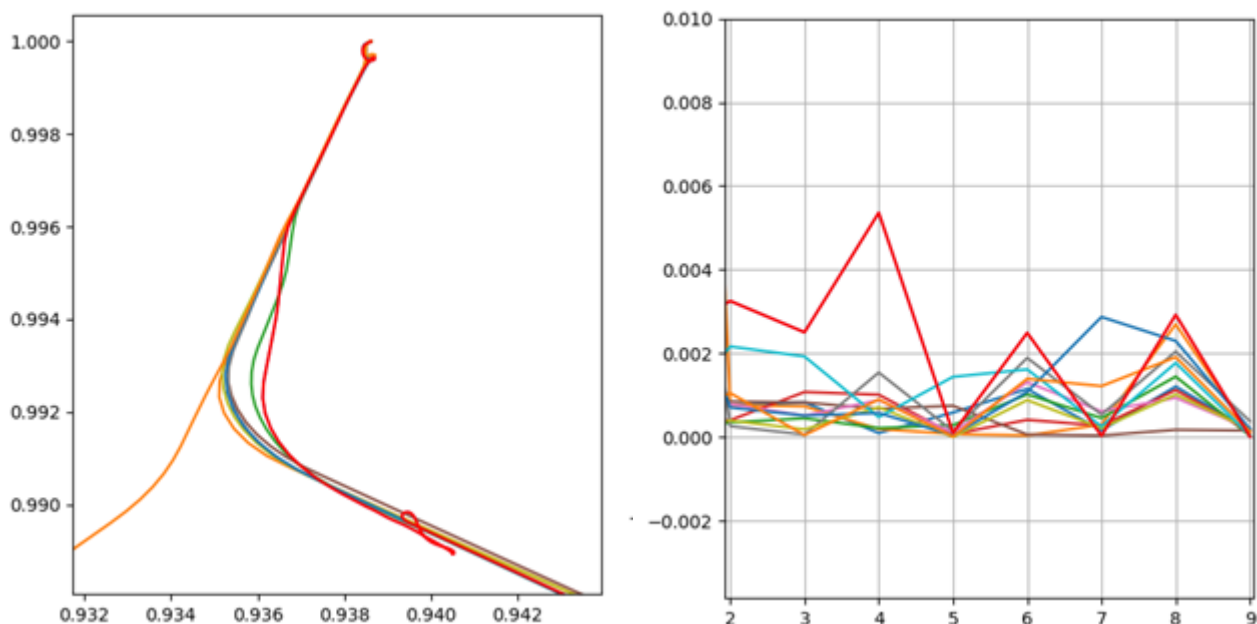


Рис. 3.8 – Определение аномального движения судна по коэффициентам нейросетевых моделей

На графике с траекториями видно, что во время движения осуществлялись резкие маневры, отсутствующие для других маршрутов, что и отражено в коэффициентах построенного отображения.

3.3.2 Обнаружение аномалий потребления топлива

Рассмотренный в части 2 главы 2 подход для обнаружения аномалий на основе машинного обучения без учителя может быть легко обобщен на другие случаи, связанные с опасными параметрами качки судна, неправильными маневрами или опасной глубиной. Для обнаружения аномалий потребления топлива предлагается подход обучения с учителем, когда в данных множества (2) восстанавливается зависимость расхода топлива по навигационным данным судна $\{x_{1,i}, x_{2,i}, \dots, x_{8,i}\} \rightarrow x_{9,i}$. В качестве модели используется регрессия на основе полиномиальной нейронной сети, которая в общем случае может быть записана в виде соотношения:

$$y_i = A_0 + (x_{1,i}, x_{2,i}, \dots, x_{8,i})A_1 + (x_{1,i}^2, x_{1,i}x_{2,i}, \dots, x_{2,i}^2, x_{2,i}x_{3,i}, \dots, x_{8,i}^2)A_2 + \dots, \quad (9)$$

до требуемого порядка нелинейности. Здесь A_k обозначает вектор-столбец числовых параметров с размерностью, соответствующей вектору нелинейностей порядка k . Вычисление этих параметров осуществляется методом наименьших квадратов, минимизирующим среднеквадратичную ошибку отклонения прогноза y_i от реального расхода топлива на милю $x_{9,i} / x_{1,i}$.

Использование полиномиальных нейронных сетей обусловлено хорошей интерпретируемостью модели. Так, уже даже в случае применения простой квадратичной функции ($k=2$ в формуле (8)) удастся найти, например, оптимальную скорость движения судна по отношению к потребляемому объему топлива. На рис. 3.9 представлен график зависимости расхода топлива на милю от скорости движения судна.

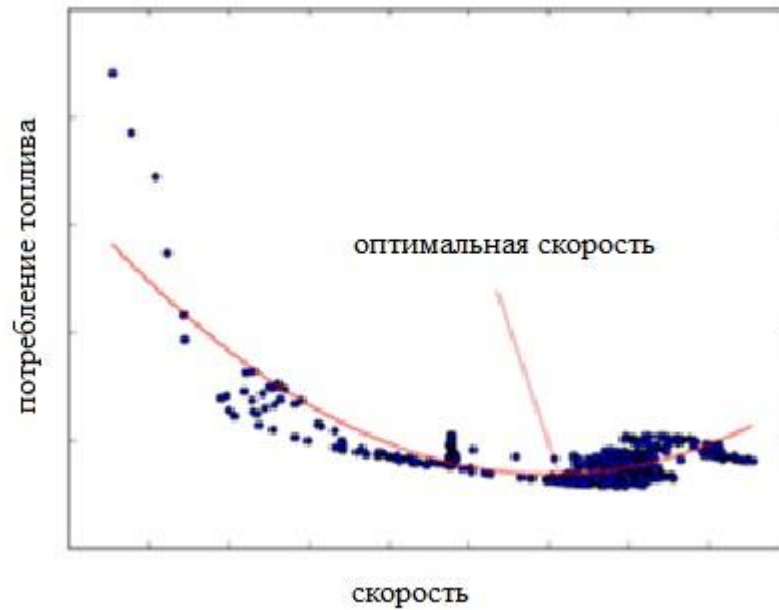


Рис. 3.9 – Модель потребления топлива на основе квадратичной зависимости от скорости движения судна

Для повышения точности модели была применена полиномиальная нейронная сеть третьего порядка нелинейности. Данная модель учитывает не только скорость движения судна, но и глубину, погодные данные (ветер, течение), угол дрейфа судна. Данные характеристики вычисляются на основе параметров $(x_{1,i}, x_{2,i}, \dots, x_{8,i})$, а затем нормируются в интервал $[0, 1]$.

Для построения модели использовалось разбиение на обучающую и тестовую выборки в соотношениях 70% и 30% соответственно. Средняя ошибка (квадратный корень из среднеквадратической ошибки) на тестовых данных составила 6%, причем 95% точек пространства имеют ошибку, меньшую 13,5%.

На рис. 3.10 синим цветом обозначены реальные данные по потреблению топлива, красным – значения, спрогнозированные моделью. Построенную модель потребления топлива можно применять и для вычисления интегральной величины потребления топлива за маршрут. Для этого прогнозные значения расхода топлива на милю y_i должны быть проинтегрированы вдоль всего маршрута

движения судна. В дискретном виде формула для вычисления расхода топлива за маршрут запишется в виде:

$$Y = \sum_{i=1}^T y_i x_{1,i} (t_i - t_{i-1}) \quad (10)$$

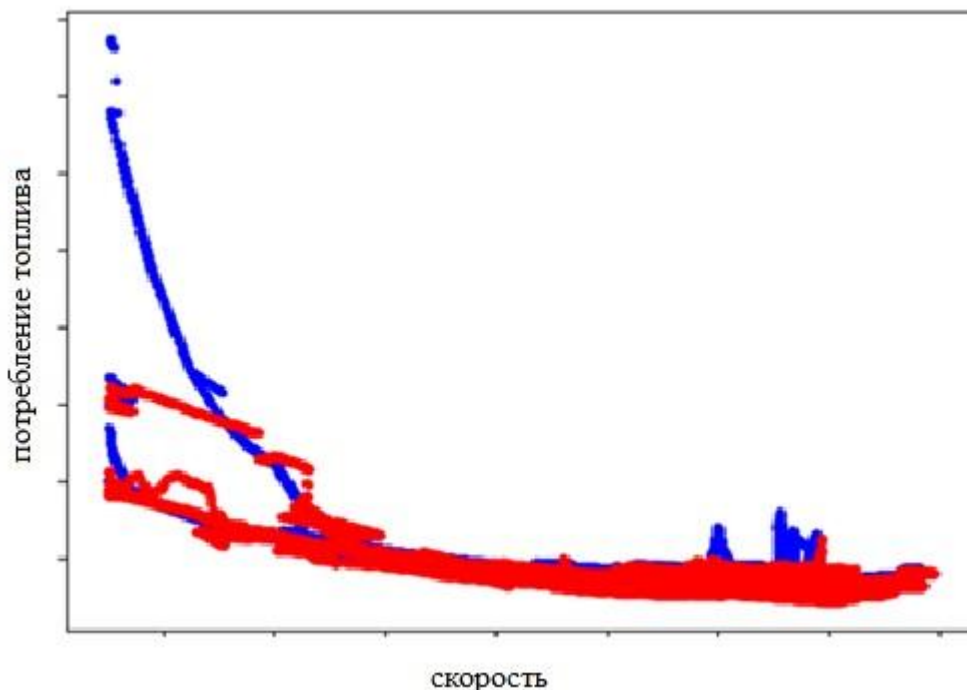


Рис. 3.10 – Аномальные маневры в пространстве скорости (STW) и кривизны траектории (R – мгновенный радиус поворота)

При использовании формулы (10) для оценки точности модели потребления топлива получено, что в 95% случаев ошибка на маршруте не превышает 6%. По рис. 3.10 можно заметить, что есть случаи, которые плохо объясняются моделью. Такое аномальное поведение требует дальнейшего рассмотрения и выяснения причин возникновения расхождения. В качестве теоретических причин можно назвать несанкционированный отбор топлива либо нештатный режим работы механизмов, приводящий к повышенному потреблению.

Выводы

Рассмотренные в работе подходы к обнаружению аномалий обладают достаточной степенью универсальности и могут быть применены в различных областях. Однако, как было продемонстрировано в данной работе, перед исследованием неизбежно возникает необходимость предварительной обработки данных и, как минимум, указания тех областей в пространстве данных, в которых следует ожидать аномальное поведение.

Для повышения точности модели прогнозирования потребления топлива можно предложить методы, основанные на градиентном бустинге моделей. Другим подходом является построение физической модели движения судна, а также учет параметров работы двигателя. Такие параметрические модели могут уточняться байесовскими методами с использованием собранных данных.

Одним из недостатков классических архитектур нейронных сетей является отсутствие математической строгости полученного результата. Обученная сеть не всегда способна корректно отображать поведение системы. Это является следствием того, что обучение по прецедентам требует большого количества данных, при этом нейронная сеть не способна спрогнозировать ситуации, которые отсутствовали в исторических данных. Данное ограничение продемонстрировано на модельном примере. Применение полиномиальной архитектуры нейронных сетей позволяет избежать указанного недостатка. Кроме того, как подтверждено математическими выводами и продемонстрировано на практической задаче по идентификации модели движения судна, коэффициенты полиномиальной нейронной сети описывают динамику системы. При этом коэффициенты (веса) могут использоваться не только как черный ящик, но и как интерпретируемая модель рассматриваемой системы.

С теоретической точки зрения основными направлениями дальнейшего развития подходов являются разработка методов интерпретации найденных аномалий и их фильтрация от случайных выбросов. Здесь могут быть выделены два метода. К первому относятся такие статистические методы, как факторный

анализ и проверка гипотез. В основе второго подхода лежит машинное обучение с учителем, требующее сбора обратной связи по выявленным аномалиям с классификацией типа каждого случая. Этот вариант более трудоемок, поскольку требует сбора маркированных данных.

С практической точки зрения следует уделить внимание оптимизации параметров рассматриваемых моделей в пространстве гиперпараметров [77]. Такие задачи направлены на уменьшение размера модели с целью повышения ее производительности и уменьшения вероятности переобучения. Стоит отметить, что увеличение быстродействия является важной прикладной задачей, позволяющей реализовывать алгоритмы обработки данных на бортовых вычислительных системах малой мощности. Решение проблемы переобучения позволит масштабировать построенные модели на разные типы судов в различных условиях.

Заключение

В выпускной квалификационной работе аспиранта проведено исследование в области выявления аномалий маневрирования и потребления топлива на примере анализа данных движения морских судов. В рамках исследования продемонстрировано применение методов изолирующего леса и полиномиальных нейронных сетей для задач обнаружения аномалий маневрирования морских судов. Также построена и протестирована модель прогнозирования потребления топлива на основе навигационных данных. Стандартная реализация алгоритма изолирующего леса взята из библиотеки `scikit-learn` и модифицирована с учетом требований предварительной фильтрации входных данных. Алгоритмы на основе полиномиальных нейронных сетей реализованы автором в виде библиотеки на основе вычислительной платформы TensorFlow.

Приведенные математические выкладки построения математической модели предоставляют строгий аппарат для построения и исследования полиномиальных нейронных сетей, задаваемых матричным преобразованием Ли. Полученные в работе результаты демонстрируют возможность применения рассмотренной нейронной сети в качестве модели машинного обучения для решения задач восстановления регрессии, а также приближенной оценки общего решения систем обыкновенных дифференциальных уравнений.

Дальнейшим развитием рассмотренного подхода являются его обобщение на теорию дифференциальных уравнений в частных производных и уравнения с запаздыванием, а также развитие специализированных эффективных методов обучения полиномиальных нейронных сетей. Также важной является необходимость теоретических оценок точности и сходимости рассмотренного подхода. Идентификация систем, описываемых уравнениями в частных производных, может вестись на основе внедрения аппарата дифференциальной алгебры. Уравнения с запаздыванием потенциально могут моделироваться рекуррентными нейронными сетями.

Таким образом, в данной работе были получены следующие результаты.

1. Проведена обработка полученных сенсорных данных для повышения устойчивости работы алгоритмов поиска аномалий и масштабирования на разные типы судов.
2. Разработана математическая модель обнаружения аномалий маневрирования и потребления топлива для морских судов.
3. На основе построенной математической модели реализован программный инструментарий.
4. Проведена апробация методов на примере решения задачи идентификации маневрирования судна.

В работе определен ряд трудностей, возникающих в процессе применения алгоритмов, приведены варианты их решения и возможные пути дальнейшего развития подходов. Реализация этих методов позволит в будущем перейти как к задачам анализа больших данных, собираемых с судов, так и к внедрению подобных алгоритмов непосредственно на борту судна с целью уменьшения передаваемого объема информации.

Список литературы и источников

1. Андрианов С. Н. Динамическое моделирование систем управления пучками частиц. — Изд-во СПбГУ, СПб. — 2004. — 368 с.
2. Вагущенко Л.Л. Современные информационные технологии в судовождении. Электронное учебное пособие. — Одесса: ОНМА. — 2013. — 135 с.
3. Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. — Х.: ОСНОВА. — 1997. — 112 с.
4. Воронцов К. В. Лекции по методу опорных векторов от 21 декабря 2007 года. URL: <http://www.ccas.ru/voron/download/svm.pdf>.
5. Демидович Б. П., Марон И. А., Шувалова Э. З. Численные методы анализа. 3-е изд. — М.: Наука. — 1967.
6. Заварзин Д. В. К вопросу поиска аномалий во временных рядах // *Инновации в науке: сб. ст. по матер. XXIX междунар. науч.-практ. конф.*, Новосибирск: СибАК. — 2014. — № 1(26). — С. 59–64.
7. Зубков Е.В., Белов В.М. Методы интеллектуального анализа данных и обнаружение вторжений // *Вестник СибГУТИ*. — 2016. — № 1.
8. Иванов А. Н. Интегрированная среда моделирования спин-орбитального движения заряженных частиц // *Вестн. С.-Петерб. ун-та. Сер. 10: прикладная математика, информатика, процессы управления*. — 2014. — № 2. — С. 49–60.
9. Иванов А. Н., Кузнецов П. М. Идентификация динамических систем на основе нелинейного матричного преобразования Ли // *Вестник УГАТУ*. — 2014. — Т. 18, № 2(63). — С. 251–256.
10. Лепский А. Е., Броневиц А. Г. Математические методы распознавания образов. Курс лекций. URL: http://window.edu.ru/resource/800/73800/files/lect_Lepskiy_Bronevich_pass.pdf.

11. *Лифшиц Ю.* Метод опорных векторов. URL: <http://logic.pdmi.ras.ru/~yura/internet/07ia.pdf>
12. *Пинский А. С.* Е-Навигация и безэкипажное судовождение // *Транспорт РФ.* — 2016. — №4 (65). — С. 50–54.
13. *Суханов А. В, Ковалев С. М.* Метод нахождения аномалий при диагностике верхнего строения пути // *Программные системы и вычислительные методы.*— 2013. — № 2(3). — С. 176–180.
14. *Шкодырев В. П., Ягафаров К. И., Баитовенко В. А., Ильина Е. Э.* Обзор методов обнаружения аномалий в потоках данных // *Proc. of the Second Conference on Software Engineering and Information Management, Санкт-Петербург, Россия.* — 2017. — Vol. 1864. — 7 с.
15. *Шолохова А. А.* Поиск аномалий в сенсорных данных на примере анализа движения морского судна // *Моделирование, оптимизация и информационные технологии.* — 2017. — №3(18). — 13 с.
16. *Шолохова А. А.* Полифит: полиномиальная нейронная сеть. Роспатент. Свидетельство №2018610385 от 10.01.2018.
17. *Шолохова А. А., Иванов А. Н.* Выполнение научно-исследовательских работ аспирантами в сотрудничестве с ИТ-компаниями // *Материалы XI (1) всероссийской научно-практической конференции «Цифровые технологии в образовании, науке, обществе».* — 2017. — С. 178-180.
18. *Шолохова А. А., Иванов А. Н.* Моделирование динамических систем на основе полиномиальных нейронных сетей // *Моделирование, оптимизация и информационные технологии.* — 2017. — №4(19). — 11 с.
19. *Aggarwal, C. C.* Outlier analysis // *Springer Science & Business Media.* — 2013. — P. 455.
20. *Agyemang, M., Barker, K., and Alhaji, R.* A comprehensive survey of numeric and symbolic outlier mining techniques // *Intelligent Data Analysis.* — 2006. — Vol. 10, no. 6. — Pp. 521–538.

21. Akaike, H. Fitting autoregressive models for prediction // *Annals of the Institute of Statistical Mathematics, Springer; The Institute of Statistical Mathematics.* — 1969. — Vol. 21, no 1. — Pp. 243–247.
22. Bakar, Z., Mohamad, R., Ahmad, A., and Andderis, M. A comparative study for outlier detection techniques in data mining // *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems.* — 2006. — Pp. 1–6.
23. Barnett, V. and Lewis, T. Outliers in Statistical Data. — J. Wiley & Sons., 3rd edition. — 1994. — XVII. — P. 582.
24. Beckman, R. J., and Cook, R. D. Outlier...s // *Technometrics.* — 1983. — Vol. 25, no 2. — Pp. 119–149.
25. Bilge, L., and Dumitraş, T. Before we knew it: an empirical study of zero-day attacks in the real world // *ACM Conference on Computer and Communications Security, Raleigh, NC.* — 2012. — Pp. 833–844.
26. Bishop, C. M. Neural networks for pattern recognition. — Oxford University Press, Oxford. — 1995. — P. 482.
27. Bishop, C. M. Pattern recognition and machine learning. — Springer Information Science and Statistics. — 2006. — P. 761.
28. Borne, K. D. Effective Outlier Detection using K-Nearest Neighbor Data Distributions: Unsupervised Exploratory Mining of Non-Stationarity in Data Streams // *Submitted to the Machine Learning Journal.* — March 2010.
29. Brandsæter, A., Manno, G., Vanem, E. & G., Ingrid, K. An application of sensor-based anomaly detection in the maritime industry // *IEEE International Conference on Prognostics and Health Management (ICPHM 2016).* — 2016. — Pp. 1–8.
30. Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. Classification and regression trees. — Wadsworth Books. — 1984. P. 358.
31. Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey // *ACM Computing Surveys (CSUR).* — 2009. — Vol. 41, №. 3, Article 15 (July 2009). — P. 58.

32. *Chandola, V., Banerjee, A., and Kumar, V.* Anomaly detection for discrete sequences: A survey // *IEEE Transactions on Knowledge and Data Engineering*. — 2012. — Vol. 24, no. 5. — Pp. 823–839.
33. *Chapelle, O., Schölkopf, B., and Zien, A.* Semi-Supervised Learning. Adaptive Computation and Machine Learning series. — The MIT Press, Cambridge. — 2006. — P. 501.
34. *Chen S., and Billings S.A.* Neural networks for nonlinear dynamic system modelling and identification // *International Journal of Control*. — 1992. — Vol. 56, no. 2. — Pp. 319–346.
35. *Clark, I.C.* Ship dynamics for mariners. — The Nautical Institute. — 2005. — P. 300.
36. *Cunningham, P., Cord, M., Delany, S. J.* Supervised Learning // *Machine Learning Techniques for Multimedia*. — Springer Berlin Heidelberg. — 2008. — Pp. 21-49.
37. *Cybenko, G. V.* Approximation by Superpositions of a Sigmoidal function // *Mathematics of Control Signals and Systems*. — 1989. — Vol. 2., issue 4. — Pp. 303–314.
38. *Das, S., Wong, W-K., Dietterich, T., Fern, A. and Emmott, A.* Incorporating Expert Feedback into Active Anomaly Discovery // *Proceedings of the IEEE 16th International Conference on Data Mining*. — 2016. — Pp. 853–858.
39. *Dragt, A. J.* Lie methods for nonlinear dynamics with applications to accelerator physics // *University of Maryland, Center for Theoretical Physics, Department of Physics*. — 2011.
40. *Edgeworth, F. Y.* On discordant observations // *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. — 1887. — Vol. 23, no. 5. — Pp. 364–375.
41. *Fox, A. J.* Outliers in time series // *Journal of the Royal Statistical Society*. — 1972. — Series B (Methodological) 34, no. 3. — Pp. 350–363.
42. *Harrou, F., Kadri, F., Chaabane, S., Tahon, C., and Sun, Y.* Improved principal component analysis for anomaly detection: Application to an emergency

- department // *Computers & Industrial Engineering*. — 2015. — Vol. 88. — Pp. 63 — 77.
43. *Hastie, T., Tibshirani, R., and Friedman, J.* The EM algorithm // *The Elements of Statistical Learning*. New York: Springer, — 2001. — Pp. 236–243.
 44. *Hawkins, D.* Identification of Outliers. — Springer Netherlands. — 1980. — P. 195.
 45. *He, Z., Deng, S., Xu, X., and Huang, J. Z.* Fp-outlier: Frequent pattern based outlier detection // *Computer Science and Information Systems*. — 2005. — Vol. 2, no. 1. — Pp. 103–118.
 46. *Ho, Tin Kam.* Random Decision Forests // *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC*. — 1995. — Pp. 278–282.
 47. *Hodge, V. and Austin, J.* A survey of outlier detection methodologies // *Artificial Intelligence Review*. — 2004. — Vol. 22, no. 2. — Pp. 85–126.
 48. <http://scikit-learn.org/stable/index.html>
 49. *Ivanov, A., Andrianov, S., Kulabukhova, N., Sholokhova, A., Krushinevskii, E., Sboeva, E.* Matrix representation of Lie transform in TensorFlow // *Proceedings of IPAC2018, Vancouver, DC, Canada*. — 2018. — P 3.
 50. *Kaelbling, L. P., Littman, M. L., and Moore, A. W.* Reinforcement Learning: A Survey // *Journal of Artificial Intelligence Research*. — 1996. — Vol. 4. — Pp. 237–285.
 51. *Khan, S. S., and Madden, M. G.* A survey of recent trends in one class classification // *Artificial Intelligence and Cognitive Science, Springer Berlin Heidelberg*. — 2010. — Pp. 188–197.
 52. *Klerx, T., Anderka, M., Büning, H. K., and Priesterjahn S.* Model-Based Anomaly Detection for Discrete Event Systems // *IEEE 26th International Conference on Tools with Artificial Intelligence, Limassol*. — 2014. — Pp. 665–672.

53. Knorr, E. M., Ng, R. T., and Tucakov, V. Distance-based outliers: Algorithms and applications // *The VLDB Journal the International Journal on Very Large Data Bases*. — 2000. — Vol 8, 3–4. — Pp. 237–253.
54. Kovalev, S., Sukhanov, A., and Styskala, V. Fuzzy model based intelligent prediction of objective events // *Advances in Intelligent Systems and Computing*. — 2016. — Pp. 23-33.
55. Kreyszig, E. *Advanced Engineering Mathematics*. — John Wiley & Sons Inc, 4th edition, — 1979. — P. 880.
56. Lagaris, I. E., Likas, A., and Fotiadis, D. I. Artificial neural networks for solving ordinary and partial differential equations // *IEEE Transactions on Neural Networks*. — 1998. — Vol. 9, no. 5. — Pp. 987–1000.
57. Lapedes, A., and Farber, R. Nonlinear signal processing using neural networks: Prediction and system modeling // *Technical Report LA-UR87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico*. — 1987.
58. Laxman, S. and Sastry, S. P. A survey of temporal data mining // *SADHANA, Academy Proceedings in Engineering Sciences*. — 2005. — Vol. 31, №. 2. — Pp. 173–198.
59. Lewis, F. L., and Ge, S. S. Neural Networks in Feedback Control Systems // *Mechanical Engineer's Handbook, Instrumentation, Systems, Controls, and mMEMS, John Wiley, New York*. — 2006. — Book 2, chapter 19. — P. 28.
60. Li, Y, Wang Y. A misuse intrusion detection model based on hybrid classifier algorithm // *International Journal of Digital Content Technology and its Applications, Advanced Institute of Convergence Information Technology*. — 2012. — Vol. 6, no. 5. — Pp. 25–33.
61. Liu, Y. and Ding, W. A KNNS based anomaly detection method applied for UAV flight data stream // *Prognostics and System Health Management Conference (PHM), Beijing*. — 2015. — Pp. 1–8.
62. Liu, T., Ting, K. M., and Zhou, Z. Isolation Forest // *Proceeding of the Eighth IEEE International Conference on Data Mining*. — 2008. — Pp. 413–422.

63. *Mall, S., and Chakraverty, S.* Comparison of artificial neural network architecture in solving ordinary differential equations // *Advances in Artificial Neural Systems*. — 2013. — Pp. 1–12.
64. *Markou, M. and Singh, S.* Novelty detection: A review-part 1: Statistical approaches // *Signal Processing*. — 2003a. — Vol. 83, issue 12. — Pp. 2481–2497.
65. *Markou, M. and Singh, S.* Novelty detection: A review-part 2: Neural network based approaches // *Signal Processing*. — 2003b. — Vol. 83, issue 12. — Pp. 2499–2521.
66. *Muller, A.C., and Guido, S.* Introduction to Machine Learning with Python: A Guide for Data Scientists. — O'Reilly. — 2017. — P. 396.
67. *Murthy, S. K.* Automatic construction of decision trees from data: A multidisciplinary survey // *Data Mining and Knowledge Discovery*. — 1997. — Vol. 2. — Pp. 345–389.
68. *Newman, J. N.* Marine hydrodynamics. — Cambridge, Massachusetts: MIT Press. — 1977. — P. 389.
69. *Omar, S., Ngadi, A., and Jebur, H.* Machine Learning Techniques for Anomaly Detection: An Overview // *International Journal of Computer Applications*. — 2013. — Vol. 79, no. 2. — Pp. 33–41.
70. *Patcha, A., and Park J. M.* An overview of anomaly detection techniques: Existing solutions and latest technological trends // *Computer Networks*. — 2007. — Vol. 51, no. 12. — Pp. 3448–3470.
71. *Pimentel, M.A.F., Clifton, D.A., Clifton, L., and Tarassenko, L.* A review of novelty detection // *Signal Processing* 99. — 2014. — Pp. 215–249.
72. *Raileanu, L. E., and Stoffel, K.* Theoretical Comparison between the Gini Index and Information Gain Criteria // *Annals of Mathematics and Artificial Intelligence*. — May 2004. — Vol. 41, no. 1. — Pp. 77–93.
73. *Ramaswamy S., Rastogi R., and Shim, K.* Efficient algorithms for mining outliers from large data sets // *Proceedings of the ACM SIGMOD International*

- Conference on Management of Data – SIGMOD '00.* — 2000. — ACM Press. — Pp. 427–438.
74. *Rokach, L., and Maimon, O.* Data mining and knowledge discovery handbook. — Springer US. — 2005. — Pp. 1285.
75. *Rousseeuw, P. J. and Leroy, A. M.* Robust Regression and Outlier Detection. — Wiley-Interscience, New York (Series in Applied Probability and Statistics). — 1987. — 329 pp.
76. *Rousseeuw, P.J., and Van Driessen, K.* A Fast Algorithm for the Minimum Covariance Determinant Estimator // *Technometrics.* — 1999. — Vol. 41. — Pp. 212–223.
77. *Russell, S. J., and Norvig, P.* Artificial Intelligence: A Modern Approach. — Prentice Hall. — 2009 (3rd Ed.). — P. 1152.
78. *Samuel, A. L.* Some Studies in Machine Learning Using the Game of Checkers // *IBM Journal of Research and Development.* — 1959. — Vol. 44, no. 1.2. — Pp. 335–365.
79. *Santosh, D. H. H., Venkatesh, P., Patruni, P., Rao, L. N., and Kumar, N. A.* Tracking Multiple Moving Objects Using Gaussian Mixture Model // *International Journal of Soft Computing and Engineering (IJ SCE).* — 2013. — Vol. 3, no. 2. — Pp. 114–119.
80. *Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C.* Estimating the Support of a High-dimensional Distribution // *Neural Computation.* — 2001. — Vol. 13, issue 7. — Pp. 1443–1471.
81. *Settles, B.* Active Learning Literature Survey // *Computer Sciences Technical Report 1648, University of Wisconsin–Madison.* — 2010. — P. 67.
82. *Siddique, N., and Adeli, H.* Evolutionary Neural Networks // *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing.* — 2013. — Pp. 307–355.
83. *Simon, L., and Rinehart, A.W.* A Model-Based Anomaly Detection Approach for Analyzing Streaming Aircraft Engine Measurement Data // *Technical*

- Report 2015-218454, NASA.* — 2015. URL: ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150000721.pdf
84. *Song, X, Wu, M., Jermaine, C., and Ranka, S.* Conditional anomaly detection // *IEEE Transactions on Knowledge and Data Engineering.* — 2007. — Vol. 19, no. 5. — Pp. 631–645.
85. *Tsitsiklis, J.* Asynchronous Stochastic Approximation and Q-learning // *Machine Learning.* — 1994. — Vol. 1, no. 3. — Pp. 185–202.
86. *Tu, E., Zhang, G., Rachmawati, L., Rajabally, E., and Huang, G. B.* Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology // *IEEE Transactions on Intelligent Transportation Systems.* — 2017. — Vol. 19, no. 5. — Pp. 1559–1582.
87. *Wei, W.W.S.* Time series analysis: Univariate and Multivariate Methods. — Addison-Wesley Publishing Company, Inc., New York. — 1990. — P. 624.
88. *Zadeh, L.A.* Fuzzy sets // *Information and control.* — 1965. — Vol. 8, no. 3. — Pp. 338–353.
89. *Zhu, Xiaojin.* Semi-supervised learning literature survey // *Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison.* — 2005. — P. 60.

Приложение А. Формат входных данных

Исходные данные являются измерениями, полученными с тестового судна на протяжении 3 месяцев. Данные представлены в формате csv. Файл с сырыми навигационными данными содержит 456 282 измерений, в каждом из которых 11 признаков. Файл с данные о потреблении топлива представлен 425 791 строками. В табл. 1-А приведен пример сырых данных.

Таблица 1-А. Образец исходных данных

time	Lat	Lon	Cog	Sog	Hdg	Stw	ROT	Depth	WindAngle	WindSpeed
148,75	2259,07	508,20	15,44	0,02	230,4	0,19	0,80	5,10	155,56	2,57
149,77	nan	508,20	331,94	nan	nan	0,20	0,20	nan	149,42	2,51
150,78	nan	nan	173,72	0,01	nan	0,24	-0,10	nan	153,37	2,58
151,79	nan	nan	82,13	0,03	nan	0,24	0,60	nan	153,76	2,36
152,81	nan	nan	263,34	0,04	nan	0,35	0,50	4,90	154,97	2,53
153,82	nan	nan	150,68	0,02	nan	nan	0,40	5,10	153,42	2,72
154,83	2259,07	508,20	182,91	0,03	nan	0,19	0,30	4,80	157,27	2,76
155,85	nan	nan	356,70	nan	nan	0,09	nan	nan	154,53	2,70
156,86	nan	nan	208,85	0,02	nan	0,06	nan	nan	152,23	2,68
157,88	nan	nan	307,72	0,01	nan	0,04	0,40	nan	153,37	2,93
158,89	nan	508,20	258,40	0,04	nan	0,02	0,60	nan	146,02	2,83
159,96	2259,07	508,20	1,80	0,05	nan	0,04	0,40	nan	145,72	2,84
160,92	nan	nan	179,99	0,03	nan	0,06	0,20	nan	154,26	2,62
161,93	nan	nan	197,93	0,01	nan	0,21	0,10	nan	150,33	2,58
162,95	nan	nan	327,67	nan	nan	0,15	-0,10	nan	154,40	2,57
163,96	2259,07	508,20	214,75	0,07	230,5	0,23	0,70	nan	153,84	2,69
164,98	nan	nan	143,07	0,02	nan	0,23	nan	nan	154,55	2,66
166,00	nan	nan	116,22	0,03	nan	0,17	-0,10	nan	152,73	2,37
167,00	nan	nan	297,50	0,01	nan	nan	0,30	5,10	152,44	2,43
168,02	nan	nan	nan	nan	nan	0,16	nan	nan	155,43	2,33
169,03	nan	nan	41,91	0,04	nan	0,20	0,20	nan	153,04	2,09
170,04	nan	nan	nan	nan	230,4	0,26	0,30	nan	160,00	2,13
171,06	2259,07	508,20	343,58	0,03	nan	0,42	0,40	nan	161,43	2,06
172,07	2259,07	508,20	nan	nan	nan	0,26	0,20	nan	159,44	1,92
173,09	nan	nan	260,45	nan	nan	0,35	0,60	nan	154,99	2,01
174,10	nan	nan	25,89	0,01	nan	0,25	0,10	nan	151,53	2,03
175,12	nan	nan	107,12	0,05	nan	0,28	nan	nan	161,06	1,83
176,13	nan	nan	199,63	0,03	nan	nan	0,30	nan	161,14	2,02
177,14	nan	nan	272,93	nan	nan	0,24	-0,30	nan	160,00	1,99
178,16	nan	nan	141,01	0,02	nan	0,23	0,00	nan	161,51	2,00
179,17	nan	nan	12,66	nan	nan	nan	0,40	nan	164,55	2,06

В табл. 2-А представлены данные проведения процедуры фильтрации и интерполяции, описанной в части 1 главы 2.

Таблица 2-А. Образец данных после обработки

time	Lat	Lon	Cog	Sog	Hdg	Stw	ROT	Depth	WindAngle	WindSpeed
148,75	2259,07	508,20	15,44	0,02	230,40	0,19	0,80	5,10	155,56	2,57
149,77	2259,07	508,20	331,94	0,02	230,41	0,20	0,20	5,05	149,42	2,51
150,78	2259,07	508,20	173,72	0,01	230,41	0,24	-0,10	5,00	153,37	2,58
151,79	2259,07	508,20	82,13	0,03	230,42	0,24	0,60	4,95	153,76	2,36
152,81	2259,07	508,20	263,34	0,04	230,43	0,35	0,50	4,90	154,97	2,53
153,82	2259,07	508,20	150,68	0,02	230,43	0,27	0,40	5,10	153,42	2,72
154,83	2259,07	508,20	182,91	0,03	230,44	0,19	0,30	4,80	157,27	2,76
155,85	2259,07	508,20	356,70	0,03	230,45	0,09	0,33	4,83	154,53	2,70
156,86	2259,07	508,20	208,85	0,02	230,45	0,06	0,37	4,85	152,23	2,68
157,88	2259,07	508,20	307,72	0,01	230,46	0,04	0,40	4,88	153,37	2,93
158,89	2259,07	508,20	258,40	0,04	230,47	0,02	0,60	4,90	146,02	2,83
159,96	2259,07	508,20	1,80	0,05	230,47	0,04	0,40	4,93	145,72	2,84
160,92	2259,07	508,20	179,99	0,03	230,48	0,06	0,20	4,95	154,26	2,62
161,93	2259,07	508,20	197,93	0,01	230,49	0,21	0,10	4,98	150,33	2,58
162,95	2259,07	508,20	327,67	0,04	230,49	0,15	-0,10	5,00	154,40	2,57
163,96	2259,07	508,20	214,75	0,07	230,50	0,23	0,70	5,02	153,84	2,69
164,98	2259,07	508,20	143,07	0,02	230,48	0,23	0,30	5,05	154,55	2,66
166,00	2259,07	508,20	116,22	0,03	230,47	0,17	-0,10	5,07	152,73	2,37
167,00	2259,07	508,20	297,50	0,01	230,45	0,17	0,30	5,10	152,44	2,43
168,02	2259,07	508,20	169,70	0,03	230,43	0,16	0,25	5,09	155,43	2,33
169,03	2259,07	508,20	41,91	0,04	230,42	0,20	0,20	5,07	153,04	2,09
170,04	2259,07	508,20	192,74	0,04	230,40	0,26	0,30	5,06	160,00	2,13
171,06	2259,07	508,20	343,58	0,03	230,40	0,42	0,40	5,05	161,43	2,06
172,07	2259,07	508,20	302,02	0,02	230,39	0,26	0,20	5,03	159,44	1,92
173,09	2259,07	508,20	260,45	0,02	230,39	0,35	0,60	5,02	154,99	2,01
174,10	2259,07	508,20	25,89	0,01	230,38	0,25	0,10	5,01	151,53	2,03
175,12	2259,07	508,20	107,12	0,05	230,38	0,28	0,20	4,99	161,06	1,83
176,13	2259,07	508,20	199,63	0,03	230,37	0,26	0,30	4,98	161,14	2,02
177,14	2259,07	508,20	272,93	0,03	230,37	0,24	-0,30	4,97	160,00	1,99
178,16	2259,07	508,20	141,01	0,02	230,36	0,23	0,00	4,95	161,51	2,00
179,17	2259,07	508,20	12,66	0,03	230,36	0,20	0,40	4,94	164,55	2,06

Приложение В. Интерфейс программы Полифит

Программа Полифит состоит из двух частей, написанных на языке Python.

Первая часть ориентирована на построение матричного отображения по заданной системе нелинейных дифференциальных уравнений.

Зависимости: numpy, sympy

Описание функций и классов:

Вспомогательные функции	
getKronPowers(state, order, dim_reduction = True)	<p>вычисляет кронекеровские степени вектора состояния state до заданного порядка нелинейности</p> <p>state: входной вектор типа numpy.ndarray, состоящий из элементов sympy.Symbol order: требуемый порядок нелинейности, int dim_reduction: булева переменная, указывающая на необходимость производить операцию уменьшения размерности (см. функцию <i>reduce</i>), bool</p> <p>возвращает список из векторов numpy.ndarray, состоящий из кронекеровских степеней</p>
reduce(state)	<p>производит операцию уменьшения размерности</p> <p>state: входной вектор типа numpy.ndarray, состоящий из элементов sympy.Symbol</p> <p>возвращает вектор типа numpy.ndarray, состоящий из элементов sympy.Symbol, в котором удалены повторяющиеся мономы</p>
fill_matrices(state_powers, expr, M)	<p>преобразует полином expr в матрицы коэффициентов M при соответствующих мономах state_powers</p> <p>state_powers: список из векторов numpy.ndarray, состоящий из кронекеровских степеней expr: полином sympy.Symbol M: список массивов numpy.ndarray</p>

class LieMapBuilder	
<p><code>__init__(state, right_hand_side, order=3)</code></p>	<p>инициализирует экземпляр класса, позволяющий построить матричное отображение для системы нелинейных дифференциальных уравнений</p> <p>state: вектор состояния системы типа <code>numpy.ndarray</code>, состоящий из элементов <code>sympy.Symbol</code></p> <p>right_hand_side: список правых частей дифференциального уравнения в виде полиномов <code>sympy.Symbol</code></p> <p>order: требуемый порядок нелинейности для матричного отображения, <code>int</code></p>
<p><code>right_hand_side_maps(R)</code></p>	<p>вычисляет значения правых частей для системы дифференциальных уравнений относительно коэффициентов матричного преобразования Ли</p> <p>R: текущее значение коэффициентов в виде списка массивов <code>numpy.ndarray</code></p>
<p><code>propagate(h=0.01, N=10, R=None)</code></p>	<p>численно решает систему дифференциальных уравнений относительно коэффициентов матричного отображения</p> <p>h: шаг интегрирования, <code>float</code></p> <p>N: число шагов интегрирования, <code>int</code></p> <p>R: начальное значение коэффициентов в виде списка массивов <code>numpy.ndarray</code></p>
<p><code>convert_weights_to_full_nm(R)</code></p>	<p>преобразует матрицы коэффициентов нелинейного матричного отображения, представленного с учетом уменьшения размерности, в полные матрицы коэффициентов</p> <p>R: текущее значение коэффициентов в виде списка массивов <code>numpy.ndarray</code></p> <p>возвращает значение коэффициентов в виде списка массивов <code>numpy.ndarray</code>, при этом коэффициенты, соответствующие повторяющимся мономам, равны 0</p>

Вторая часть программы реализует полиномиальную нейронную сеть для решения задачи идентификации.

Зависимости: keras, tensorflow

Описание функций и классов:

class LieLayer	
<code>__init__(output_dim, order=1)</code>	<p>инициализирует экземпляр класса, представляющий слой полиномиальной нейронной сети, эквивалентной матричному представлению оператора Ли</p> <p>output_dim: размерность вектора состояния на выходе order: порядок нелинейности отображения</p>
<code>build(input_shape)</code>	<p>создает матрицы весовых коэффициентов как настраиваемые параметры в нейронной сети</p> <p>input_shape: размерность вектора состояния на входе</p>
<code>call(x)</code>	<p>применяет нелинейное матричное отображение к входному вектору</p> <p>x: входной вектор размерности input_shape</p> <p>возвращает вектор размерности output_dim</p>