Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Томский государственный университет систем управления и
радиоэлектроники» (ТУСУР)

На правах рукописи

Бардамова Марина Борисовна

АЛГОРИТМЫ ПОСТРОЕНИЯ НЕЧЕТКИХ КЛАССИФИКАТОРОВ НЕСБАЛАНСИРОВАННЫХ ДАННЫХ НА ОСНОВЕ МЕТАЭВРИСТИК «ГРАВИТАЦИОННЫЙ ПОИСК» И «ПРЫГАЮЩИЕ ЛЯГУШКИ»

Специальность 05.13.17
«Теоретические основы информатики»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Ходашинский Илья Александрович

Томск 2021

Оглавление

Введение

Машинное обучение применяется для создания автоматических систем анализа данных, которые позволяют ускорить и облегчить работу специалистов в сферах, в которых «ручной» анализ требует существенных затрат времени и ресурсов: экономике, информационной безопасности, медицине и других. Важным условием эффективного взаимодействия между интеллектуальной системой и её пользователем является доверие. Доверие пользователя достигается не только уверенностью в правильности результата автоматического анализа, но и в понимании, какие процессы внутри системы привели к этому результату. Системы нечеткого вывода отличаются от прочих методов машинного обучения тем, что в их основе лежат принципы человеческого мышления и логики. Нечеткие правила и функции принадлежности легко поддаются интерпретации, позволяя обеспечить понимание пользователем закономерностей вывода системы без глубокого погружения в специфику машинного обучения.

Однако при построении систем нечеткого вывода с целью решения задач классификации могут возникнуть трудности при работе с данными, отличающимися несбалансированным характером. Нечеткие классификаторы подвержены переобучению на классах с наибольшим числом экземпляров, что ведет к получению высокой общей точности при низкой доле правильной классификации объектов, принадлежащих наименьшим классам [1]. Так как классы с меньшим числом экземпляров зачастую оказываются наиболее важными для прогноза, требуются инструменты, способные улучшить качество их распознавания. Задача создания алгоритмов построения нечетких классификаторов, позволяющих построить точные, компактные и интерпретируемые модели на несбалансированных данных, является актуальной.

Анализ существующих подходов по улучшению точности нечетких классификаторов несбалансированных данных показывает, что основным методом преодоления дисбаланса является применение алгоритмов предобработки данных. Классы меньшинства дополняются путем генерации искусственных экземпляров, что облегчает процесс обучения классификатора. Однако употребление таких алгоритмов при наличии шумов в данных ведет к многократному повторению ошибок в новых образцах [2]. Кроме того, генерация данных затруднительна при количестве классов, большем двух, или при рассредоточении экземпляров наименьшего класса, так как создание новых образцов приводит к перемешиванию областей различающихся классов.

Использование для повышения точности нечетких классификаторов таких этапов обучения, как формирование структуры, настройка параметров и отбор признаков, является устоявшейся практикой. Их эффективность многократно подтверждена публикациями P. Angelov, V. Bolon-Canedo, S.L. Chiu, O. Cordon, A. Fernandez, H. Hagras, F. Herrera, H. Ishibuchi, M.J. del Jesus, V. Lopez, M. Sugeno, T. Takagi, L. Xu, R.R. Yager. Внесение модификаций в эти

этапы может позволить нечеткому классификатору достигать высокого качества на несбалансированных данных, то есть получать и высокую общую точность, и большую долю распознавания экземпляров наименьшего класса по сравнению со стандартными методами.

Перечисленные задачи обучения классификатора могут быть решены с помощью метаэвристических алгоритмов [3, 4, 5]. Метаэвристики – это класс алгоритмов, осуществляющих поиск удовлетворительных решений разнообразных задач оптимизации без доказательства оптимальности найденных вариантов. Качество решения может быть выражено через некоторую метрику, например точность, стабильность, время. В отличие от традиционных способов оптимизации, основанных на вычислении производных, метаэвристики, как правило, реже попадают в локальные оптимумы и предусматривают способы преодоления таких ситуаций, а также имеют более широкую применимость. Использование метаэвристик с соответствующей задаче фитнес-функцией позволит достигнуть улучшения качества классификации несбалансированных данных с помощью нечетких систем без использования этапа редактирования данных. В качестве такой функции выбрана средняя геометрическая точность, рассчитывающаяся на основе процента правильной классификации каждого класса.

Кроме упомянутых выше ученых, наиболее значимых результатов в изучении нечетких систем достигли А.Н. Аверкин, И.З. Батыршин, М.В. Бобырь, М.И. Дли, Ю.Н. Золотухин, А.С. Катасёв, С.М. Ковалев, Л.Г. Комарцова, В.В. Круглов, Ю.И. Кудинов, А.О. Недосекин, Ф.Ф. Пащенко, Д.А. Поспелов, Ю.П. Пытьев, Е.С. Семенкин, А.В. Язенин, Н.Г. Ярушкина, Г.Э. Яхъева, R. Babuska, A. Bastian, J.C. Bezdek, J. Casillas, J.L. Castro, D. Dubois, D. Filev, J. Gonzalez, S. Guillaume, U. Kaymak, B. Kosko, R. Krishnapuram, R. Kruse, E.H. Mamdani, S. Oh, W. Pedrycz, H. Prade, H. Tanaka, I. B. Turksen, T. Yasukawa, L. Zadeh.

Целью диссертационной работы является повышение средней геометрической точности нечетких классификаторов несбалансированных данных за счет использования метаэвристических алгоритмов на различных этапах построения классификатора.

Для достижения поставленной цели поставлены следующие **задачи**:

1) обзор существующих методов обработки несбалансированных данных и методов построения систем нечеткого вывода;

2) разработка и исследование алгоритма формирования структуры нечеткого классификатора, позволяющего улучшить среднюю геометрическую точность;

3) разработка и исследование гибридного алгоритма оптимизации параметров нечеткого классификатора несбалансированных данных;

4) разработка и исследование алгоритма настройки весовых коэффициентов, учитывающих важность признаков в базе нечетких правил;

5) проверка разработанных алгоритмов на контрольных примерах и сравнение с аналогами.

Объектом исследования является процесс построения нечетких классификаторов несбалансированных данных.

Предметом исследования являются алгоритмы построения и оптимизации нечетких классификаторов для несбалансированных данных.

Методы исследования. В диссертационной работе применялись методы оптимизации, анализа данных и теории информации, теория нечетких множеств и нечеткой логики.

Достоверность результатов обеспечивается корректностью применения математических методов, результатами проведенных экспериментов, статистически сопоставимых с результатами, полученными исследователями других научных групп.

Научная новизна полученных результатов.

В диссертации получены следующие новые научные результаты.

1. Разработан авторский алгоритм формирования базы правил нечеткого классификатора несбалансированных данных, отличительной особенностью которого является применение метаэвристики "прыгающие лягушки" для итеративной процедуры генерации и настройки дополнительного правила для класса с наименьшим процентом правильной классификации.

2. Разработан новый гибридный алгоритм оптимизации параметров нечетких классификаторов несбалансированных данных, особенность которого заключается в дополнении метаэвристики «гравитационный поиск» локальным поиском из метаэвристики «прыгающие лягушки» для улучшения эффективности оптимизации.

3. Разработан авторский алгоритм настройки весовых коэффициентов признаков при классификации несбалансированных данных, отличительной особенностью которого является применение гибридного метаэвристического алгоритма для поиска оптимального вектора весов признаков в базе нечетких правил.

Теоретическая значимость работы заключается в развитии технологии построения нечетких систем интеллектуального анализа несбалансированных данных. Алгоритм формирования базы правил нечеткого классификатора и алгоритм настройки весов признаков могут использовать любые аналогичные метаэвристики вместо предложенных. Гибридный алгоритм оптимизации может применяться для решения других задач параметрической оптимизации.

Практическая значимость работы подтверждается применением полученных в ней результатов для решения практической задачи оценки свертываемости крови у беременных женщин. Результаты внедрены в ОГАУЗ «Родильный дом №1» города Томска.

Разработанные алгоритмы использованы при выполнении следующих проектов:

– научный проект при поддержке РФФИ «Методы и инструментальные средства построения самообучающихся систем, основанных на нечетких правилах» (№16-07-00034-а), 2016-2018 гг. (№ госрегистрации АААА-А16-116021210312-4);

– научный проект при поддержке РФФИ «Методы построения нечетких классификаторов несбалансированных данных на основе алгоритма гравитационного поиска» (№19-37-90064-аспиранты), 2019-2021 гг. (№ госрегистрации АААА-А19-119101790046-5);

– государственное задание Министерства образования и науки Российской Федерации на 2017–2019 гг., проект № 2.8172.2017/БЧ «Методы и модели определения уровня защищенности информационных систем» (№ госрегистрации АААА-А17-117073110015-3);

– государственное задание Министерства образования и науки Российской Федерации на 2017-2019 гг., проект № 8.9628.2017/8.9 «Теоретические основы человеко-машинных интерфейсов» (№ госрегистрации АААА-А17-117073110013-9).

Разработанные алгоритмы применимы при построении нечетких классификаторов для решения практических задач классификации и в научно-исследовательских целях при анализе данных.

На защиту выносятся следующие положения.

1. Разработанный алгоритм формирования базы нечетких правил, основанный на итеративном процессе генерации и настройки правила метаэвристикой «прыгающие лягушки», в комбинации с алгоритмом генерации структуры на основе экстремумов признаков классов позволяет создавать классификатор, демонстрирующий при меньшем числе правил большую среднюю геометрическую точность по сравнению с классификаторами, полученными общеизвестными алгоритмами генерации структуры Ishibuchi+SMOTE и E-алгоритмом, а также сопоставимую точность при сравнении с комбинациями Chi+SMOTE и HFRBCS+SMOTE [6]. На исследуемых несбалансированных наборах данных средняя геометрическая точность возросла в среднем на 23 процента относительно точности, полученной при использовании только алгоритма экстремальных значений признаков классов.

Соответствует пункту 5 паспорта специальности: Разработка и исследование моделей и алгоритмов анализа данных, обнаружения закономерностей в данных и их извлечениях.

2. Разработанный гибридный алгоритм настройки параметров нечеткого классификатора на основе комбинации метаэвристик «гравитационный поиск» и «прыгающие лягушки» позволил увеличить среднюю геометрическую точность классификации на исследуемых несбалансированных наборах данных в среднем на 24 процента по сравнению с точностью до оптимизации. Статистическое сравнение подтвердило существование значимой разницы в точности по сравнению с исходными метаэвристиками при оптимизации параметров нечетких классификаторов несбалансированных данных. Построенные нечеткие классификаторы

продемонстрировали большую среднюю геометрическую точность по сравнению с Chi+SMOTE, Ishibuchi+SMOTE и E-алгоритмом, и сопоставимое качество классификации при сравнении с HFRBCS+SMOTE.

Соответствует пункту 13 паспорта специальности: Применение бионических принципов, методов и моделей в информационных технологиях.

3. Разработанный алгоритм настройки весовых коэффициентов признаков позволил увеличить среднюю геометрическую точность классификации в среднем на 16 процентов относительно точности до введения весов. При существенно меньшем количестве используемых правил алгоритм позволил продемонстрировать сопоставимую точность с комбинациями Chi+SMOTE и Ishibuchi+SMOTE и большую точность по сравнению с E-алгоритмом.

Соответствует пункту 5 паспорта специальности: Разработка и исследование моделей и алгоритмов анализа данных, обнаружения закономерностей в данных и их извлечениях.

Внедрение результатов диссертационного исследования. Результаты исследовательской работы легли в основу программного обеспечения для оценки состояния свертывающей системы крови у беременных женщин, применяемого в ОГАУЗ «Родильный дом №1».

Разработанные алгоритмы были использованы в ФГБОУ ВО «ТУСУР» при выполнении проекта № 8.9628.2017/8.9 «Теоретические основы человеко-машинных интерфейсов» в рамках государственного задания Министерства науки и высшего образования РФ, а также при выполнении проекта № 2.8172.2017/8.9 «Методы и модели определения уровня защищенности информационных систем» в процессе исполнения государственного задания ТУСУР.

Результаты диссертационного исследования используются при изучении дисциплины «Информатика» на кафедре комплексной информационной безопасности электронно-вычислительных систем ТУСУР.

Апробация работы. Основные положения работы докладывались и обсуждались на конференциях различного уровня. Среди них:

– международная конференция IEEE Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT) (2021, онлайн, IEEE);

– международные научно-практические конференции «Электронные средства и системы управления» (2015, 2017-2020 гг., Томск, ТУСУР);

– международные научно-технические конференции студентов, аспирантов и молодых ученых «Научная сессия ТУСУР» (2015-2021 гг., Томск, ТУСУР);

– международные конференции студентов, аспирантов и молодых ученых «Перспективы развития фундаментальных наук» (2018-2020 гг., Томск, ТУСУР);

– всероссийские молодежные научные форума «Наука будущего – наука молодых» (12-14 сентября 2017 г., Нижний Новгород; 2019 гг.; 14-17 мая 2019, Сочи, Министерство науки и высшего образования РФ)

– всероссийский конкурс-конференция студентов и аспирантов по информационной безопасности «SIBINFO-2018» (19 апреля 2018 г., Томск, ТУСУР);

– всероссийский форум молодых ученых (27-28 апреля 2017 г., Екатеринбург, УрФУ);

– всероссийская научно-практическая конференция «Нечеткие системы, мягкие вычисления и интеллектуальные технологии» (3–7 июля 2017 г., Санкт-Петербург, СПИИРАН);

– международная летняя школа-семинар по искусственному интеллекту для студентов, аспирантов, молодых ученых и специалистов «Интеллектуальные системы и технологии: современное состояние и перспективы» (30 июня – 3 июля 2017 г., Санкт- Петербург, СПИИРАН);

– международной научно-практической конференции «Молодежь и современные информационные технологии» (7-11 ноября 2016 г., Томск, ТПУ);

– всероссийская научно-практическая конференция в рамках конгресса «Здравоохранение России. Технологии опережающего развития» (4-7 ноября 2015 г., Томск, Министерство здравоохранения РФ).

Публикации по теме диссертации. По результатам исследований опубликовано 28 печатных работ, из которых в рекомендованных ВАК РФ периодических изданиях – 6. Десять работ проиндексированы в международной базе SCOPUS, четыре – в Web of Science. Получены 4 свидетельства о государственной регистрации программ для ЭВМ.

Личный вклад автора. Постановка цели и задач научного исследования, интерпретация экспериментальных данных, подготовка публикаций по выполненной работе проводилась совместно с научным руководителем. Автором самостоятельно разработаны и реализованы алгоритмы формирования структуры нечеткого классификатора несбалансированных данных, настройки весовых коэффициентов признаков, настройки параметров термов на основе комбинации двух метаэвристик; получены результаты экспериментов, проведена апробация разработанных алгоритмов. Разработка программного обеспечения для ОГАУЗ «Родильный дом №1» проведена автором совместно с сотрудниками родильного дома.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав основной части, заключения, списка литературы из 155 наименований и 4 приложений. Основная часть работы содержит 116 страниц, в том числе 14 рисунков и 39 таблиц.

Во введении описана актуальность работы, сформулированы цель и задачи исследования, изложены основные результаты, их теоретическая и практическая значимость, приведена новизна исследования и защищаемые положения.

В первой главе содержится обзор проблемы построения интеллектуальных систем при несбалансированном характере исследуемых данных. Приведен и проанализирован перечень типовых методов преодоления проблемы дисбаланса данных. Представлен обзор основных методик формирования структуры и вывода результата в нечетких системах классификации.

Во второй главе приведено описание разработанного алгоритма формирования структуры нечеткого классификатора несбалансированных данных, алгоритма нечеткого вывода с учетом весовых коэффициентов признаков, а также гибридного алгоритма настройки параметров термов. Объясняется выбор целевой функции для разработанных алгоритмов.

Третья глава посвящена экспериментальной проверке эффективности разработанных алгоритмов и статистическому сравнению полученных результатов с аналогами.

В четвертой главе представлено описание применение разработанных алгоритмов для построения системы оценки системы свертывания крови у беременных женщин.

Диссертант благодарит за помощь и поддержку в работе над диссертацией научного руководителя, д.т.н., профессора Илью Александровича Ходашинского, а также выражает признательность за ценные замечания и рекомендации к.т.н., доцента Константина Сергеевича Сарина.

Глава 1. Задача построения нечетких классификаторов несбалансированных данных

1.1 Несбалансированные данные

Задачи обучения с учителем построены на анализе ретроспективных данных для восстановления зависимости между объектами и выходными переменными. Результативность анализа зависит не только от эффективности алгоритмов анализа, но и от качества самих данных. Одним из возможных существенных недостатков данных является дисбаланс, осложняющий построение моделей классификации из-за превосходства в количестве экземпляров (образцов) одних классов над другими. Поиск закономерностей в несбалансированных данных является сложной задачей для специалистов по интеллектуальному анализу данных, машинному обучению, распознаванию образов, статистике [7]. Основной проблемой построения классификаторов несбалансированных данных является плохая приспособленность стандартных алгоритмов обучения, что приводит к значительному ухудшению результатов классификации. Из-за дисбаланса между классами классификаторы определяют экземпляры классов меньшинства неправильно, поскольку модель переобучается на экземплярах классов большинства [1].

Не существует четкого критерия, разграничивающего сбалансированные данные от несбалансированных. Устоявшейся в научной среде мерой, характеризующей дисбаланс, является коэффициент дисбаланса (imbalance ratio, *IR*) – отношение числа экземпляров самого большого класса к количеству образцов самого мелкого. В общем случае, чем больше коэффициент, тем сложнее задача правильного распознавания наименьшего класса. Идеально сбалансированным набором является известный набор данных iris [8], насчитывающий по 50 экземпляров для каждого класса. Такие наборы являются редкостью; несбалансированное распределение классов, при котором количество экземпляров одних классов превосходит число экземпляров других, характерно для большинства реальных задач. Например, в известном репозитории «Knowledge Extraction based on Evolutionary Learning» (KEEL) в разделе «стандартные данные для классификации» [9], насчитывающем 75 наборов данных из различных сфер деятельности, только 23 имеют коэффициент дисбаланса ниже 1,5. Для остальных 52 наборов он варьируется от 1,6 до 140395.

Наличие количественного превосходства экземпляров одних классов над другими наблюдается в задачах классификации из разнообразных сфер деятельности. По данным международной базы научных изданий Scopus в период между 2010 и 2020 годами только 36 процентов публикаций, посвященных несбалансированным данным, принадлежат к области «Компьютерные науки». Еще около 12 процентов относятся к «Математике» и примерно 5

процентов к «Науке о принятии решений». Чаще всего публикации из этих областей посвящены способам преодоления проблемы дисбаланса и анализу эффективности этих подходов. Самая высокая численность публикаций, касающихся несбалансированных данных, соответствует таким прикладным областям, как «Инженерия» (16 процентов), «Медицина» (5,5 процента), «Химия» (3,4 процента). Диаграмма распределения публикаций по областям науки представлена на рисунке 1.1.

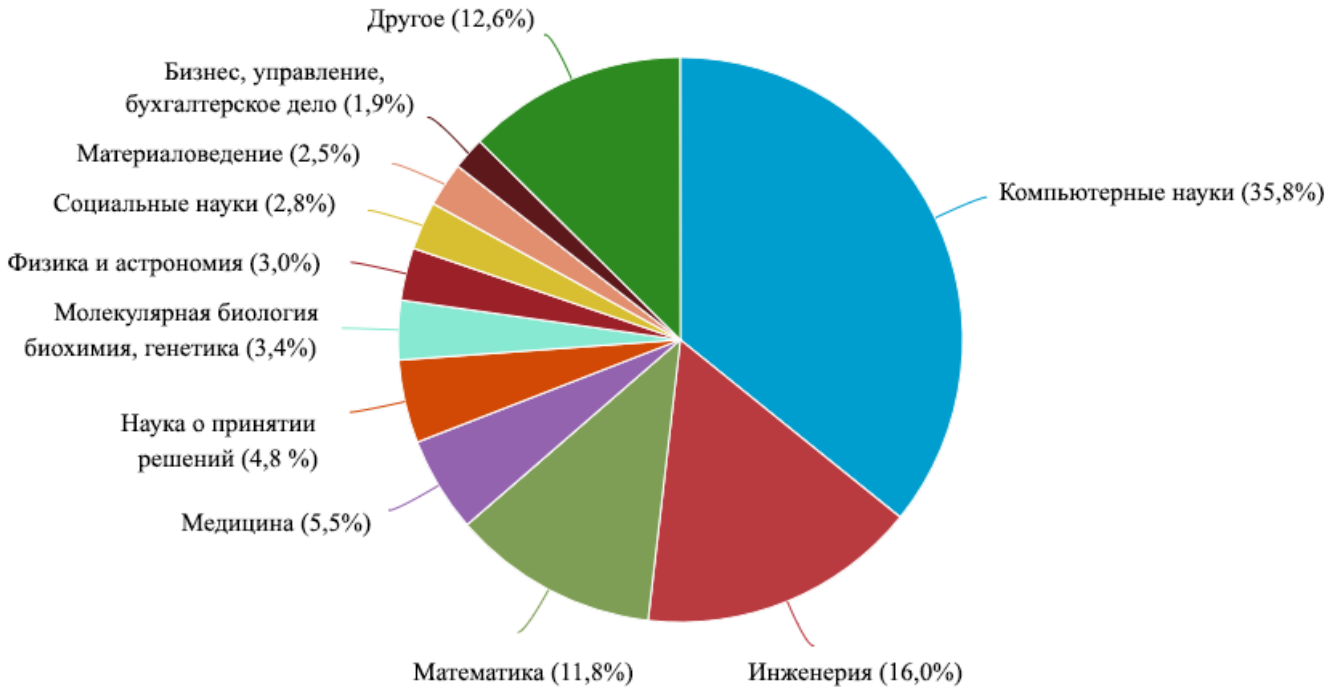

Рисунок 1.1 – Распределение публикаций о несбалансированных данных по отраслям науки за последнее десятилетие (2010 – 2020 гг.)

Банковские данные часто являются несбалансированными. В статье [10] описана задача построения классификатора для прогнозирования оттока клиентов банка. Своевременное получение банком информации о снижении лояльности клиента необходимо для осуществления попытки удержать клиента предложениями более выгодных условий или специальных услуг. Авторы данной работы располагали базой данных реального китайского банка, состоящей из записей о двадцати тысяч пациентов. Данные включали информацию о личности клиента (возраст, образование, занятость, семейное положение и т.п.), сведения о счете (тип счета, данные о кредитах) и сведения о поведении клиента (кредитный статус, частота задолженностей и т.п.). Среди всего объема записей случайным образом, но с сохранением пропорции классов, было отобрано 1524 образца. Среди них только 73 относилось к клиентам с низкой лояльностью, остальные 1451 принадлежали к множеству клиентов с нормальной лояльностью. Для построения классификатора были использованы случайные леса с внедрением функции штрафов за неправильное определение экземпляров наименьшего класса и формированием дубликатов данных меньшинства. Разработанный классификатор показывал лучшие результаты по

распознаванию наименьшего класса, чем классический алгоритм построения дерева решений и многослойный персептрон.

В задачах медицинской сферы часто требуется с высокой точностью определить наличие редкого заболевания или разделить пациентов на менее и более тяжелые случаи, но массив данных для обучения по менее важным примерам оказывается доминирующим [11]. Например, авторы исследования [12] при разработке модели прогнозирования риска серьезных осложнений после бариатрической операции столкнулись с ситуацией нехватки данных, так как среди 44061 изучаемых пациентов только у трех процентов наблюдались серьезные осложнения. Для преодоления проблемы дисбаланса в этом случае был использован алгоритм, генерирующий искусственные экземпляры наименьшего класса на основе уже существующих образцов. Авторы использовали различные ансамбли алгоритмов классификации, однако им так и не удалось получить модель, которую можно было бы применять в реальной практике.

В работе [13] проводился анализ данных, представляющих собой сведения о производственных процессах, собираемых устройствами контроля качества, с целью автоматизации прогнозирования неисправностей. Данные обладали несбалансированным характером, так как образцов нормального функционирования оборудования больше, чем экземпляров, указывающих на ошибки и дефекты. Авторы анализировали эффективность трех методов построения ансамблей деревьев решений, а также различные алгоритмы увеличения количества экземпляров наименьшего класса, в том числе основанные на использовании нейронных сетей.

Задача построения классификатора сетевых атак всегда связана с обработкой несбалансированных данных, так как образцов нормального трафика и экземпляров простых атак, связанных с отказом в обслуживании, всегда больше, чем примеров атак более сложных. Ярким примером является известный набор данных о сетевом трафике KDD Cup 1999, который часто применяется для проверки эффективности решающих алгоритмов. Он состоит из 4,9 миллиона экземпляров и 23 классов [14]. Среди них на три класса – нормальное соединение, атаку neptune и атаку smurf – суммарно приходится 99 процентов данных (20, 22 и 57 процентов соответственно). Семь классов насчитывают не больше десяти образцов. Даже при объединении атак в четыре группы (DoS, R2L, U2R, Probe) сохраняется крупный дисбаланс, так как и neptune, и smurf относятся к одной группе DoS атак. Аналогичная ситуация наблюдается и в более новых наборах для анализа сетевого трафика [15, 16]. В работе [17] был исследован потенциал улучшения качества классификации набора KDD Cup 1999 при использовании инструментов генерации дополнительных данных наименьших классов. Исследователи пришли к выводу, что прибегать только к предобработке данных неправильно, так как экземпляры групп атак R2L и

U2R перекрываются другими классами, и генерация новых экземпляров приводит к сильному перемешиванию данных.

В работе [18] указаны пять основных причин необходимости учета особенностей классификации несбалансированных данных:

1) стандартные классификаторы, такие как логистическая регрессия, машина опорных векторов, дерево решений хорошо работают на сбалансированных обучающих наборах данных и непригодны для работы с несбалансированными данными;

2) процесс обучения, ориентированный на такой показатель эффективности как обобщенная точность, показывает высокую общую точность, однако неправильно классифицирует при этом экземпляры класса меньшинства;

3) экземпляры класса меньшинства при обучении могут классифицироваться как шум, а шумы могут ошибочно идентифицированы как экземпляры меньшинства, поскольку обе эти категории являются редкими образцами в наборе данных;

4) экземпляры класса меньшинства часто пересекаются с областями классов большинства [19].

5) небольшого размера выборки, недостаточное для обучения количество экземпляров наименьшего класса, слабая разделимость – проблемы несбалансированного обучения, не позволяющие обнаружить экземпляры класса меньшинства.

Актуальность исследования определяется необходимостью развития методов классификации на основе индуктивных методов обучения, которые в дальнейшем будут использованы в системах, основанных на знаниях. Повышенный интерес к обучению на несбалансированных данных находит свое отражение в значительном увеличении числа публикаций на эту тему, а также в организации специальных выпусков журналов, семинаров, конференций, симпозиумов [1, 7]. За десять лет число публикаций, ежегодно индексируемых в базе данных Scopus по теме «Imbalanced Data», возросло более чем в семь раз, с 32 до 232. На рисунке 1.2 приведен график, иллюстрирующий увеличение количества индексируемых в Scopus публикаций, затрагивающих тему несбалансированных данных, в период с 2010 (257 публикаций) по 2020 год (1688 публикаций).

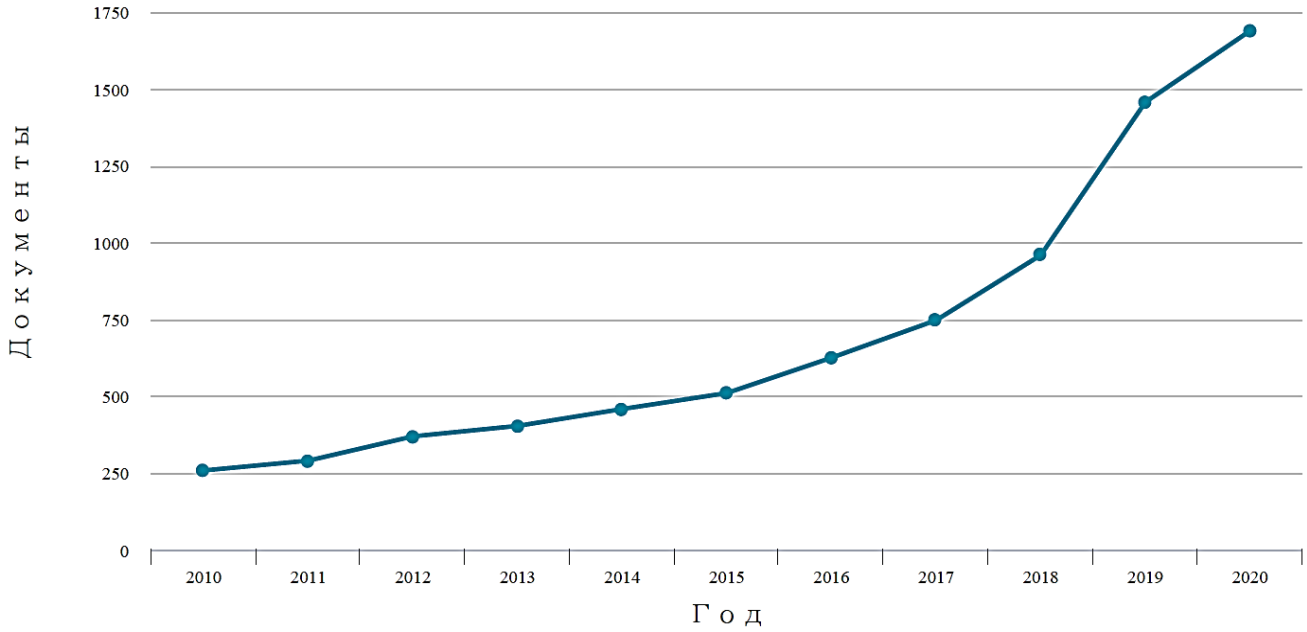

Рисунок 1.2 – Количество публикаций на тему «imbalanced data», проиндексированных в международной базе Scopus с 2010 по 2020 год

Так как дисбаланс данных наблюдается в большинстве реальных задач классификации, необходимость построения качественных моделей классификации данных привела к созданию методов и алгоритмов, позволяющих справиться с этой проблемой или вовсе устранить её. Выделяют три уровня стратегий преодоления дисбаланса данных: 1) связанные с определением показателей качества классификации, 2) связанные с обучающими данными, 3) связанные с алгоритмом обучения [20].

1.1.1 Выбор меры качества классификации при наличии дисбаланса в данных. При работе с несбалансированными данными недостаточно оценивать построенную модель с помощью обобщенной точности (accuracy), представляющей собой процент правильной классификации [21]. Например, для набора данных, включающего 99 образцов одного класса и 1 экземпляр второго класса, классификатор, относящий все объекты к первому классу, получит значение обобщенной точности, равное 99 процентам. Несмотря на полное игнорирование одного из классов, модель заработала высокую оценку качества. Понимание того, что обобщенная точность является плохим показателем при классификации несбалансированных данных, привело к применению новых мер, таких как AUC (область под ROC-кривой) [22], средняя геометрическая точность, сбалансированная точность, Fβ-мера и других [23]. Для оценки эффективности работы классификаторов в [23] предложены 18 показателей, которые классифицируются по трем категориям:

1) пороговые метрики, направленные на минимизацию числа ошибок: обобщенная точность, усредненная точность (арифметическая и геометрическая), Fβ-мера и Каппа-статистика;

2) метрики, основанные на вероятностном понимании ошибки, применяемые для оценки надежности классификаторов, например, средняя абсолютная ошибка, средняя квадратичная ошибка, кросс-энтропия;

3) метрики, основанные на оценивании разделимости экземпляров, например, AUC, которая для двух классов эквивалентна статистике Манна-Уитни-Вилкоксона [23].

Проведя анализ этих категорий, авторы [23] делают вывод о том, что выбор метрик для несбалансированных данных имеет первостепенное значение.

Положительные классы (с наименьшим числом экземпляров) обычно более важны, чем отрицательные классы (с наибольшим числом экземпляров). Уменьшение ошибочной классификации экземпляров классов меньшинства обычно имеет решающее значение в реальных приложениях [24, 25], но в некоторых случаях необходимо максимально точно определять все классы вне зависимости от их редкости. Например, при диагностике наличия у человека диабета (набор данных pima), ложноотрицательный результат оставит больного человека без лечения, а ложноположительный заставит здорового потратиться на дополнительные анализы. В то же время повышение качества классификации положительных классов может привести к ухудшению распознавания экземпляров отрицательных классов, так как экземпляры разных классов часто пересекаются между собой [19].

Таким образом, в каждой задаче классификации данных разработчику системы анализа данных необходимо расставить приоритеты: или сосредоточиться на повышении общей точности, или стараться правильно определять положительные экземпляры при потере в качестве определения отрицательных, или искать некоторый компромиссный вариант со сбалансированными ошибками. В конечном итоге выбор метрики зависит от цели создания модели и требований, предъявляемых к ней заказчиком, а также способности конкретного алгоритма классификации работать с выбранной мерой.

Оценка качества классификации зачастую происходит не только на конечном этапе построения модели, но и при обучении. Например, приведенные выше метрики могут применяться в качестве фитнес-функций алгоритмов формирования и оптимизации классификаторов. При использовании адекватной фитнес-функции алгоритмы оптимизации способны принимать во внимание несбалансированный характер данных и обучать классификатор на распознавание не только отрицательных классов. Например, в работе [26] проводился отбор признаков генетическим алгоритмом с помощью фитнес-функции, основанной на доле выбранных признаков и средней геометрической точности. После проведения эксперимента по отбору признаков на несбалансированных данных с использованием классификатора SVM (support vector machine – метод опорных векторов), авторы заключили, что применение в фитнес-функции средней геометрической точности вместо процента правильной

классификации позволило выбрать признаки, способствующие улучшению распознавания положительного класса.

Таким образом, грамотный выбор метрики для обучения позволяет улучшить качество классификации на несбалансированных данных.

1.1.2 Исправление данных для устранения дисбаланса. Данные играют ключевую роль в исследованиях по машинному обучению и интеллектуальному анализу. Для исправления дисбаланса исследователями разработан ряд алгоритмов предобработки данных, применение которых позволяет упростить задачу построения классификаторов, неспособных напрямую обращать внимание на наличие несбалансированности. Существуют три группы инструментов предобработки данных: удаления экземпляров, добавления образцов данных и их комбинации. Для достижения численного равновесия между классами, алгоритмы, направленные на увеличение количества экземпляров меньшего класса (over-sampling), генерируют дополнительные экземпляры класса меньшинства, в то время как алгоритмы удаления (under-sampling) сокращают количество классов большинства.

Простейшим алгоритмом удаления данных является Random under-sampling (RUS) – неэвристический метод, который направлен на устранение дисбаланса по классам путем случайного исключения экземпляров класса большинства. Недостатком RUS является то, что он теряет информацию о данных класса большинства [18, 27]. Помимо алгоритма случайного удаления экземпляров разработаны методы управляемого сокращения данных. Первые из них были предназначены для определения и удаления экземпляров, которые могут считаться избыточными или шумовыми. Так, алгоритм «сокращающего правила ближайшего соседа» (Condensed Nearest Neighbor Rule) избавляется от экземпляра, если у ближайшего соседа в заранее составленной подвыборке данных такой же класс [28]. Другой подход, Tomek links, удаляет экземпляр отрицательного класса, если его ближайший сосед принадлежит к положительному классу [29]. В своем оригинальном варианте эти алгоритмы не имели параметров для управления количеством удаляемых экземпляров и также, как и RUS, могли привести к излишнему уменьшению объема данных. Более современные методы применяют более глубокий анализ влияния экземпляров на обучение. Например, авторы [30] предлагают оценивать твердость экземпляра на основе достоверности прогнозов нескольких различных классификаторов. Такой подход позволяет детально проанализировать каждый экземпляр, но весь процесс является довольно трудоемким и занимает продолжительное время.

В качестве способа усечения данных может использоваться кластеризация. В [31] экземпляры большинства объединяются в кластеры алгоритмом k-средних; в результирующую выборку данных включаются экземпляры классов меньшинства и несколько представителей классов большинства из каждого кластера. Авторами [32] предложен другой подход: в кластеры

объединяются экземпляры положительного класса. Те образцы отрицательного класса, расстояние до ближайшего кластера которых больше некоторого заданного значения, исключаются.

Среди алгоритмов, направленных на увеличение количества образцов наименьших классов, также есть более и менее сложные. Чаще всего оригинальные версии алгоритмов направлены на работу с данными, насчитывающими только два класса; но для большинства из них существуют способы расширения на большее число классов.

Простейший алгоритм случайного добавления данных (Random over-sampling, ROS) на каждом шаге генерирует дубликат экземпляра положительного класса, выбранного случайным образом, до тех пор, пока количество образцов не сравняется. Несмотря на простоту метода его применения бывает достаточно для данных с небольшим дисбалансом и без существенного перекрытия классов [33].

Наиболее известным алгоритмом добавления данных и одним из самых распространенных инструментов преодоления дисбаланса данных в целом является SMOTE (Synthetic Minority Oversampling Technique) [18, 25, 34]. Оригинальная работа [34] с описанием SMOTE является самой цитируемой статьей (8704 цитирований) среди публикаций на тему несбалансированных данных, индексируемых в международной базе Scopus. В исходной версии алгоритма генерация нового синтетического образца положительного класса происходит путем осуществления интерполяции между некоторым экземпляром положительного класса и случайно выбранным экземпляром этого же класса среди k ближайших соседей. Количество соседей k задается пользователем. Пусть x_i – случайно выбранный на некотором шаге экземпляр наименьшего класса, x_{ij} – ближайший сосед экземпляра x_i. Тогда новый образец положительного класса r_j будет создан следующим образом:

$$r_j = x_i + \text{rand} \times (x_{ij} - x_i),$$

где rand – функция, генерирующая случайное число из промежутка [0;1].

На рисунке 1.3 продемонстрирован пример применения SMOTE для экземпляра x_1 при количестве ближайших соседей k, равном шести. Экземпляры наибольшего класса изображены в форме круга, исходные экземпляры наименьшего класса представлены в виде четырехконечной звезды. Пятиконечные звезды представляют собой варианты положительных образцов, сгенерированных алгоритмом SMOTE. Как видно из рисунка, число ближайших соседей выбрано неверно; при создании экземпляра между x_1 и x_{15}, а также между x_1 и x_{16} происходит смешение кластеров разных классов.

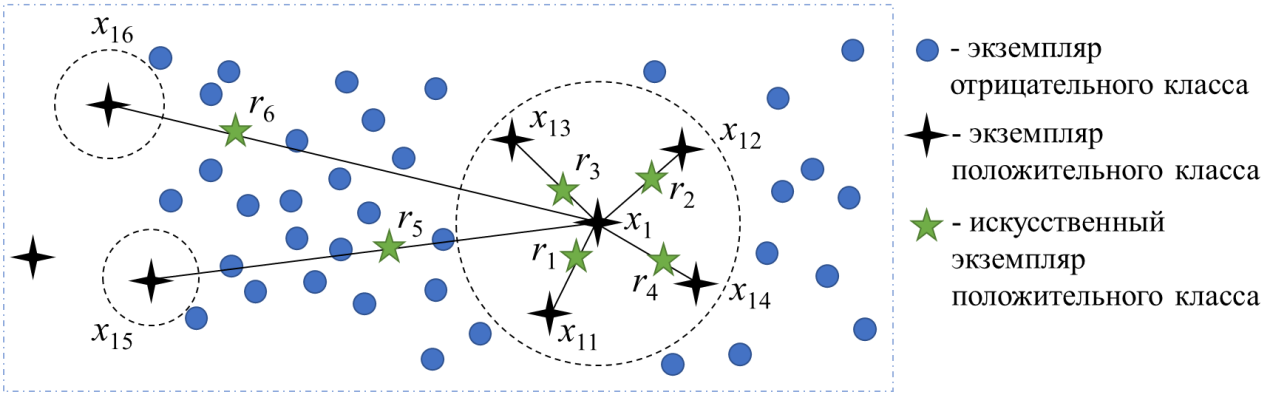

Рисунок 1.3 – Генерация новых образцов положительного класса для шести ближайших соседей экземпляра x_1

Использование SMOTE является простым и быстрым способом устранения дисбаланса в данных. Однако добавление дополнительных экземпляров ведет к естественному увеличению временных затрат на обучение и может стать причиной воспроизведения ошибочной информации, если она присутствует в данных положительного класса [2]. Оригинальный SMOTE не пользуется информацией о распределении объектов класса большинства, что может привести к синтезу экземпляров, перекрывающих области отрицательного класса [35].

Для исправления недостатков оригинального алгоритма разработан ряд модификаций SMOTE. Создатели модификаций вводят механизмы, позволяющие учитывать при создании искусственных данных распределение экземпляров различных классов. Пограничный SMOTE (Borderline SMOTE) генерирует новые образцы только для тех экземпляров, среди ближайших соседей которого есть как минимум один экземпляр отрицательного класса [36]. В Safe-Level-SMOTE выбранный для генерации на текущей итерации экземпляр сначала проверяется на наличие образцов положительного класса среди его k ближайших соседей; если их нет, экземпляр рассматривается как выброс и генерация не проводится. Если положительные образцы есть, один из них выбирается для генерации случайным образом. Далее рассчитывается отношение между количеством положительных экземпляров среди k ближайших соседей исходного экземпляра и числом положительных экземпляров среди k ближайших соседей у выбранного соседа. Если полученное значение больше единицы, то генерация нового образца происходит ближе к исходному к экземпляру; если меньше единицы, то ближе к соседу [37]. Авторы [38] предлагают сначала формировать из образцов положительного класса кластеры, а затем использовать SMOTE в пределах каждого кластера таким образом, чтобы сбалансировать количество экземпляров в кластерах. Еще одним потомком SMOTE является алгоритм ADASYN (Adaptive Synthetic Sampling Approach for Imbalanced Learning). Генерация новых образцов в ADASYN происходит в первую очередь для самых сложных экземпляров; сложность экземпляра тем выше, чем больше среди его k ближайших соседей образцов противоположного класса. Количество

генерируемых экземпляров для рассматриваемого экземпляра пропорционально его сложности [39].

Ряд методов увеличения количества данных построен на использовании генеративно-состязательных сетей (GAN, Generative adversarial network) [13]. Генеративно-состязательные сети состоят из двух нейронных сетей, чаще всего выполненных в виде многослойных персептронов и функционирующих в противоположном ключе [40]. Первая сеть, генератор, создает на основе имеющихся экземпляров данных новые образцы. Вторая – дискриминатор – получает на вход дополненную выборку данных и определяет для каждого экземпляра вероятность, что он был создан искусственным путем. Цель генератора заключается в подборе таких параметров, которые бы позволили создавать наиболее достоверные образцы, а задача дискриминатора состоит в улучшения точности распознавания искусственных данных. Для применения генеративно-состязательных сетей необходимо значительно больше вычислительных ресурсов, чем для работы с алгоритмами на основе SMOTE, но в некоторых случаях GAN позволяет добиться большего улучшения качества классификации [13].

Гибридные методы объединяют в себе обе стратегии добавления и удаления экземпляров данных. Типичными примерами этой группы методов является совмещение SMOTE с алгоритмами «сокращенного правила ближайшего соседа», Tomek lines и другими способами проверки искусственных образцов на увеличение уровня шума в данных [41].

Пример работы алгоритмов исправления дисбаланса на абстрактном наборе данных приведен на рисунке 1.4. Отрицательные образцы представлены в форме круга, положительные изображены в виде четырехконечной звезды.

Методы предобработки универсальны и просты в употреблении, но имеют невысокую эффективность и не могут использоваться как единственный инструмент решения проблемы дисбаланса в классах. Кроме того, в некоторых задачах классификации создание новых экземпляров данных неприемлемо. К примеру, искусственное создание записей о пациентах может привести к ошибкам в диагностике заболеваний.

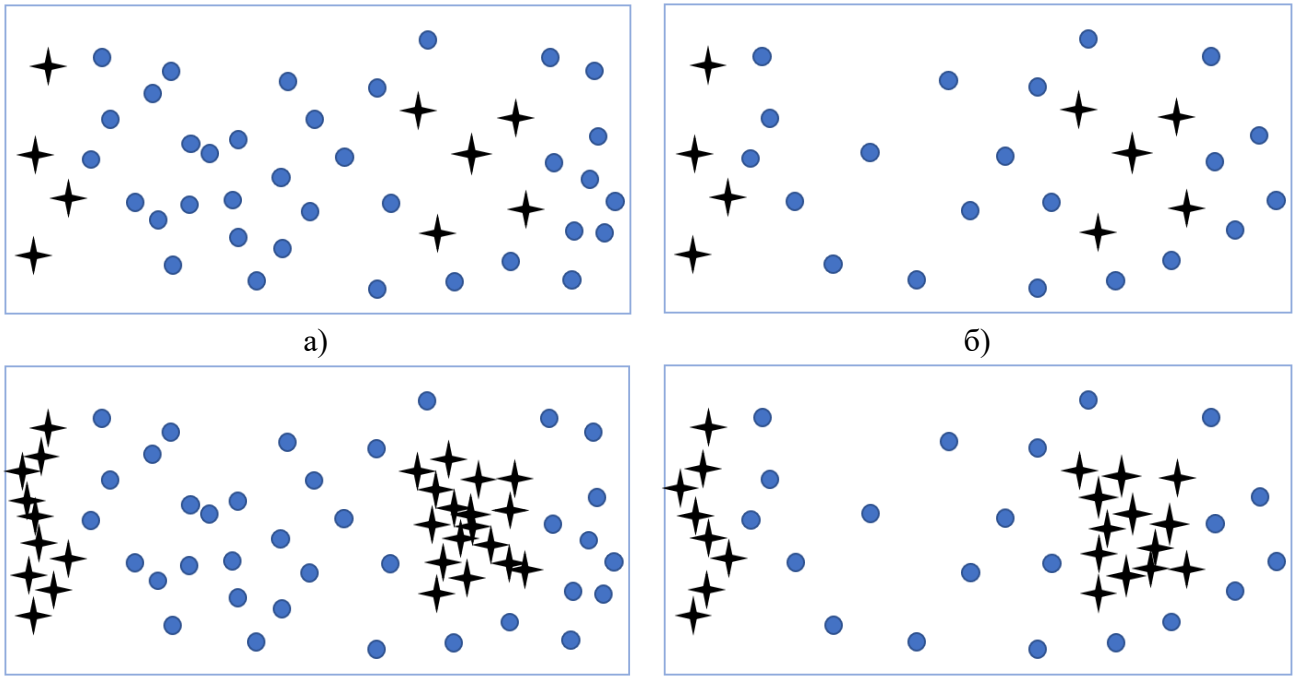

Рисунок 1.4 – Пример предобработки данных: а) исходное распределение экземпляров набора данных, б) распределение экземпляров после применения алгоритма удаления отрицательных образцов, в) распределение экземпляров после использования алгоритма добавления положительных образцов, г) распределение после гибридной предобработки

1.1.3 Модификация алгоритмов классификации. Для уменьшения влияния дисбаланса данных на качество обучения моделей разрабатываются модификации решающих алгоритмов. Изменения могут носить специфический характер, если они применимы только к конкретным классам алгоритмов классификации. Например, для классификаторов, основанных на правилах, может быть введена процедура взвешивания правил [42]. Чем большее количество экземпляров положительного класса классифицирует правило, тем больший вес оно может иметь. Веса применяются и в модификациях алгоритма k-ближайших соседей. Как правило, соседи, расположенные ближе к рассматриваемому объекту, получают набольший вес. В работе [43] было предложено рассчитывать вес соседей в зависимости не только от их расположения, но и от редкости их класса.

Распространенным способом модификации алгоритмов классификации является «обучение с учетом затрат» (cost-sensitive learning), основанные на применении штрафных функций. Методы, учитывающие затраты на неправильное классифицирование, основаны на изменении алгоритма классификации таким образом, чтобы издержки за неправильную классификацию экземпляров класса меньшинства были более велики по отношению к экземплярам класса большинства. Типичным решением здесь является использование матрицы штрафов за отнесение экземпляра, принадлежащего одному классу, к другому [44].

Существуют и универсальные модификации, которые относятся не столько к самим решающим алгоритмам, сколько к способу построения модели классификации. Например, как в контексте работы с несбалансированными данными, так и при наличии проблемы пересечения классов, часто применяется объединение алгоритмов в ансамбль. Если данные насчитывают больше двух классов, то на распознавание каждого из них можно обучить отдельный алгоритм ансамбля, затем провести агрегирование результатов в итоговый выход [45]. При этом в рамках ансамбля не обязательно прибегать к одному и тому же решающему алгоритму. Авторы [46] предложили использовать для распознавания отрицательного класса более простые и быстрые инструменты, например, SVM или деревья решений, а для работы с положительным классом применять нейронные сети, которые отличаются более высокой вычислительной сложностью.

В современных исследованиях разработчики часто комбинируют ансамбли и методы предобработки данных. Автор [47] предлагает использовать схему бэггинга вместе с генетическим алгоритмом, отбирающим для построения модели подмножества экземпляров отрицательных классов. В работе [48] применяется анализ плотности экземпляров для устранения избыточных образцов и обучение ансамбля с учетом затрат.

Использование ансамблей решающих алгоритмов – эффективный способ решения проблемы пересечения классов. Однако процесс построения и обучения ансамбля, а также проектирование схемы итогового вывода всей модели требует существенных затрат ресурсов и времени разработчика.

Выбор способа преодоления проблемы несбалансированности зависит от многих параметров: величины дисбаланса, наличия иных недостатков данных (перекрытия классов, недостатка данных, зашумленности и других), особенностей алгоритма классификации, требований к точности и сложности разрабатываемой модели. Нечеткие классификаторы также имеют склонность к переобучению на классах большинства, поэтому разработка алгоритмов, способных исправить этот недостаток, является актуальной задачей.

1.2 Нечеткие системы, основанные на правилах

Концепция нечетких множеств была предложена ученым Лотфи Заде в 1965 году [49]. В противовес четким множествам, в которых объект либо принадлежит, либо не принадлежит определенной группе объектов, нечеткие множества предполагают, что объект может принадлежать множеству с некоторой долей принадлежности. Л. Заде предложил использовать функции принадлежности (термы) в качестве субъективной оценки принадлежности объекта некоторому множеству [50]. Термы могут быть заданы функциями принадлежности треугольного, трапециевидного, гауссова типа и другими (рисунок 1.5) [51].

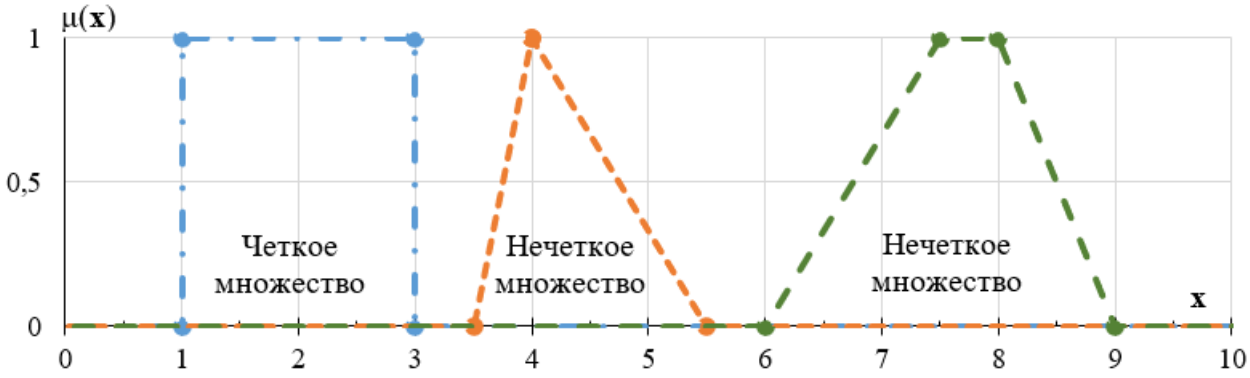

Рисунок 1.5 – Пример четкого множества и нечетких множеств треугольного и трапециевидного типа

Нечеткие системы, основанные на правилах, применяются для решения задач классификации и аппроксимации. Они опираются на базу правил вида «если – то». В антецедентной части («если»-части) содержится комбинация функций принадлежности входных переменных, а в консеквентной («то»-части) указывается выход правила. Существуют два основных типа нечетких правил, отличающихся консеквентной частью: Мамдани и Такаги–Сугено [50]. При использовании первого типа консеквент представляет собой нечеткий терм. В качестве примера ниже приведено правило типа Мамдани для двух входных переменных x^1 *и* x^2, которым соответствуют термы T_{11} и T_{21} соответственно, и выходной переменной y с соответствующим термом T_{y1}:

$$\text{ЕСЛИ } x^1 = T_{11} \text{ И } x^2 = T_{21}, \text{ ТО } y = T_{y1}. \quad (1.1)$$

В частном случае выходной терм может быть вырожден в некоторое число или метку класса, в зависимости от имеющегося типа шкалы, на которой он определяется [52]. При построении системы Такаги-Сугено на выходе правила помещается полиномиальная функция [53]:

$$\text{ЕСЛИ } x^1 = T_{11} \text{ И } x^2 = T_{21}, \text{ТО } y = d_0 + d_1x^1 + d_2x^2, \quad (1.2)$$

где d_0, d_1 и d_2 – параметры функции. Выбор типа правила зависит от решаемой задачи: для классификации стоит воспользоваться первым типом, для аппроксимации – вторым.

В [54] выделяют пять концепций проектирования систем нечеткого вывода: генетические нечеткие системы (genetic fuzzy systems), нейро-нечеткие системы (neuro-fuzzy systems), иерархические нечеткие системы (hierarchical fuzzy systems), самообучающиеся нечеткие системы (evolving fuzzy systems) и многокритериальные нечеткие системы (multiobjective fuzzy systems).

В генетических нечетких системах применяются эвристические алгоритмы для формирования структуры и дальнейшей оптимизации её компонентов [55]. Объекты, подвергающиеся оптимизации, кодируются в виде вектора, на основе которого создается

популяция (или популяции при одновременном использовании нескольких алгоритмов). Полученная популяция подается на вход выбранному алгоритму, осуществляющего поиск оптимального решения. В процессе поиска система неоднократно перестраивается для оценки качества исследуемых решений. В форме закодированных векторов могут быть представлены параметры функций принадлежности, правила, целые базы правил и другие элементы систем [54].

Нейро-нечеткие системы совмещают принципы нечеткой логики и нейронных сетей для исправления недостатков отдельных алгоритмов. Нейронные сети, отличающиеся лучшей способностью к обучению, могут применяться для настройки параметров систем нечеткого вывода с целью повышения прогностических способностей модели [56, 57]. Принципы нечеткой логики, в свою очередь, могут способствовать улучшению интерпретируемости нейронных сетей. Широкое применение получил алгоритм ANFIS (adaptive neuro-fuzzy inference system – адаптивная нейро-нечеткая система), совмещающий нейронные сети и нечеткую систему типа Такаги-Сугено [58]. Сеть ANFIS включает пять слоев: первый соответствует функциям принадлежности, второй отвечает за вычисление t-норм правил, третий нормирует полученные значения t-норм, четвертый осуществляет расчет консеквентов правил на основе линейных функций, последний слой представляет собой единственный узел, вычисляющий сумму всех своих входов для определения итогового выхода системы [59]. Для работы сети необходима уже сформированная база правил. Параметры сети могут быть настроены путем оптимизации; для этой задачи обычно используют метод наименьших квадратов, градиентный спуск, генетический алгоритм и другие инструменты.

Иерархические нечеткие системы объединяют несколько небольших блоков нечеткого вывода в один в соответствии с различными схемами – каскадной, древообразной и другими [54]. Эта методика позволяет избавиться от большого количества правил путем подчинения одних баз правил другим, однако при этом затрудняется процесс формирования вывода, а также возможность интерпретации итоговой модели.

Самообучающиеся системы нечеткого вывода предназначены для построения онлайн-моделей машинного обучения, обрабатывающих данные в режиме реального времени. Примером систем такого типа являются системы Ангелова-Ягера, в которых нечеткие правила представляют собой облака, основанные на плотности данных [60]. Использование облаков вместо классических функций принадлежности позволяет обрабатывать данные в пакетном режиме. В [61] отмечается, что системы Ангелова-Ягера не требуют дополнительной оптимизации благодаря применению рекуррентного алгоритма построения.

Многокритериальные нечеткие системы могут быть созданы на основе любой из уже описанных видов систем; их особенность заключается в решении сразу нескольких задач.

Например, одновременного повышения точности и улучшения интерпретируемости, снижения сложности при сохранении точности, а также других вариантов. Чаще всего для работы с многокритериальными нечеткими системами подбирается функция качества, состоящая из комбинации нескольких критериев. Например, в работе [62] при построении нечетких классификаторов с возможностью отбора признаков использована функция, основанная как на ошибке классификации, так и доле отобранных признаков. Авторами [63] были предложены методы построения Парето-оптимальных нечетких классификаторов сразу для трех критериев: точности, сложности и индексу интерпретируемости. Сложность представляла собой количество правил, а индекс интерпретируемости сочетал в себе количество термов для каждого признака, различимость термов, наличие пересечений далеких по смыслу термов.

Данное исследование посвящено генетическим нечетким системам, так как среди приведенных типов систем они обладают самой простой структурой. Простота структуры позволяет не только быстро осуществлять расчет выхода системы, но и получать легко интерпретируемые модели.

1.2.1 Нечеткие классификаторы. Для решения задач классификации применяются нечеткие системы с правилами типа Мамдани, в которых консеквент вырожден в метку класса. В базовом нечетком классификаторе алгоритм вычисления выхода для некоторого входного объекта включает следующие шаги [50]:

1) фаззификация: вычисление доли принадлежности всех признаков объекта соответствующим термам;

2) нечеткий вывод: расчет степени принадлежности объекта всем правилам или группе правил;

3) дефаззификация: выбор выходной метки класса по полученным значениям степени принадлежности объекта правилам или группе правил.

На этапе фаззификации для i-го входного признака x^i, где $i = \overline{1,n}$, n – количество признаков, рассчитывается доля принадлежности $\mu_{Tj}(x^i)$, где T_j – нечеткий терм j-го правила. Этап нечеткого вывода в задаче классификации заключается в вычислении степени принадлежности объекта к правилам. В классификаторе типа Мамдани степень принадлежности рассчитывается с помощью t-нормы. Дефаззификация может быть проведена по одной из двух схем. В рамках первой схемы, «победитель получает всё», определяется правило с максимальной степенью принадлежности; его консеквент подается на выход классификатора. Согласно второй схеме, «команда получает всё», сначала вычисляется сумма степеней принадлежности правил с одинаковым консеквентом, затем определяется максимальная сумма, на выход подается

консеквент группы правил с максимальной суммой степеней принадлежности. Более подробно алгоритм нечеткого вывода для классификатора описан в параграфе 6 данной главы.

Процесс построения нечеткого классификатора может состоять из одного или двух этапов. Обязательный (базовый) этап создания системы заключается в генерации термов и формировании базы правил. Дополнительный этап оптимизации системы необходим в том случае, если построенная модель не достигает требуемого качества, и может включать различные методы: уточнение базы правил, настройку параметров термов, отбор признаков или экземпляров, расчет весов правил и аналогичные способы улучшения классификатора.

Для определения качества модели необходимо проводить оценку её эффективности. Качество может быть выражено как в некоторой мере точности (процент правильной классификации, ошибки I и II рода и прочие метрики), так и в других критериях, например, структурной сложности (количество правил, термов и т.п.) или временных затратах на осуществление вычислений.

На качество моделей влияют как особенности данных, так и эффективность алгоритмов построения нечеткого классификатора. И этап создания структуры, и этап оптимизации могут быть осуществлены разнообразными инструментами, однако немногие из них предназначены для работы с несбалансированными данными. В следующих параграфах будут описаны наиболее распространенные алгоритмы построения и оптимизации нечетких классификаторов.

1.3 Формирование структуры нечеткого классификатора

Под структурой нечеткого классификатора обозначим совокупность двух элементов – базы нечетких правил и лингвистических термов. Качество созданной структуры, например, полнота и достаточность правил, влияет как на точность и скорость вывода, так и на эффективность дальнейшей оптимизации.

Задачу создания структуры классификатора включает два этапа. На первом осуществляется нечеткое разбиение входного признакового пространства, на втором этапе формируются правила. Термы могут быть заданы экспертами или сгенерированы инструментами различной сложности.

Базовыми алгоритмами генерации структуры будем считать такие, которые учитывают только область определения признака. Например, алгоритм случайного разбиения размещает заданное количество термов на области определения признака случайным образом, а алгоритм равномерного разбиения заполняет входное пространство заданным количеством термов так, чтобы вся область определения была покрыта равномерно (рисунок 1.6).

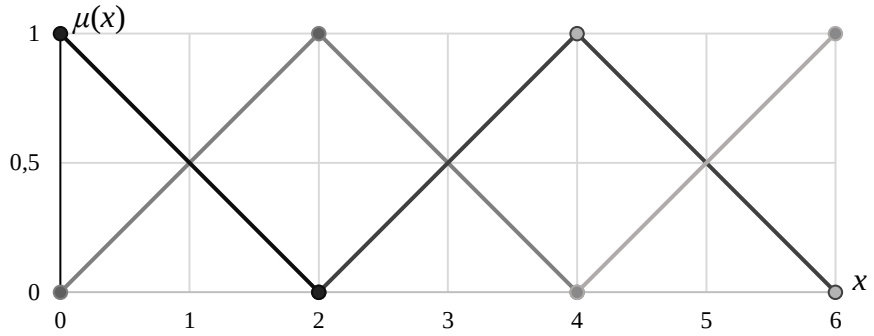

Рисунок 1.6 – Пример равномерного разбиения переменной *x*

При использовании таких подходов нечеткие правила формируются по принципу «каждый с каждым»: антецедентная часть правила составляется из возможных сочетаний термов всех признаков, а консеквент определяется по наименьшему расстоянию между вершинами термов, входящих в правило, и входной таблицей наблюдения. Такие алгоритмы просты, но, так как расположение термов никак не соотносится с входными данными, получаемые классификаторы обладают низкой точностью.

Базовые подходы применяют как основу для более продвинутых алгоритмов. Назовем такие инструменты создания структуры алгоритмами первого уровня.

Алгоритм Chi [64] расширяет алгоритм равномерного разбиения признакового пространства. После равномерного заполнения заданным количеством симметричных термов треугольного типа для каждого объекта составляется правило из тех термов, к которым объект имеет максимальную долю принадлежности. В консеквент записывается метка класса, принадлежащая данному объекту. Таким образом формируется база правил, объем которой совпадает с количеством экземпляров в обучающей выборке. Далее определяются веса правил. Веса для метода Chi обычно вычисляются с помощью меры, называемой доверием. При расчете доверия между *j*-ым правилом и консеквентом c_j вычисляется отношение между суммой степеней принадлежности к этому правилу экземпляров с меткой класса c_j и суммой степеней принадлежности к этому правилу всех экземпляров набора данных:

$$w_{R_j} = \mathrm{conf}_j = \sum_{x_p \in c_j} \beta_j(x_p) \Big/ \sum_{p=1}^{|X|} \beta_j(x_p), \qquad (1.3)$$

где w_{R_j} – вес *j*-го правила, эквивалентный значению доверия conf_j, $\beta_j(x_p)$ – степень принадлежности *j*-му правилу экземпляра с индексом *p*, $p = \overline{1,|X|}$, *X* – множество экземпляров. На последнем шаге удаляются повторяющиеся правила и правила с низкими весами. Если два правила будут иметь одинаковые антецеденты, но разные консеквенты, в базе останется правило с наибольшим весом.

Алгоритм Ishibuchi, предложенный в [65], формирует антецедентную часть правил на основе всех возможных комбинаций нечетких термов из четырех заданных вариантов разбиения признака (два, три, четыре и пять термов, равномерно распределенных между 0 и 1, рисунок 1.7), а также терм «всё равно».

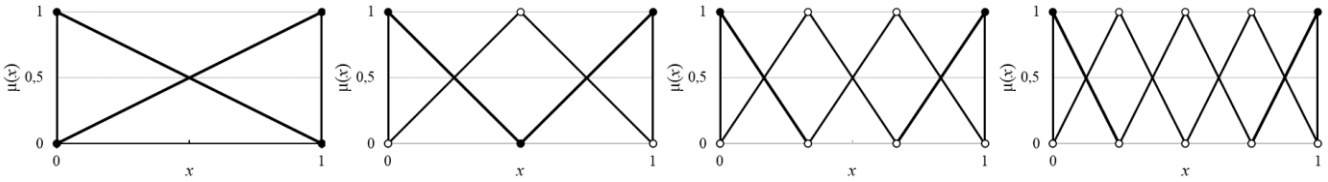

Рисунок 1.7 – Разбиение признакового пространства в методе Ishibuchi

При этом авторы рекомендуют использовать не более трех термов в правиле для уменьшения вычислительной сложности алгоритма. После генерации “Если”-части вычисляется мера доверия между правилом и каждым классом; класс с наибольшим доверием записывается в консеквент правила. Веса определяются как разность доверия и величины, представляющей собой отношение между суммой степеней принадлежности к правилу экземпляров с меткой класса, не равной c_j, и суммой степеней принадлежности к правилу всех экземпляров:

$$w_{R_j} = \left(\sum_{x_p \in c_j} \beta_j(x_p) - \sum_{x_p \notin c_j} \beta_j(x_p) \right) \Bigg/ \sum_{p=1}^{|X|} \beta_j(x_p) . \tag{1.4}$$

Для итоговой базы среди правил с одним и тем же консеквентом отбирается некоторое заданное число правил по наилучшим значениям мер доверия ($conf_j$) и поддержки. Поддержка между j-ым правилом и консеквентом c_j является отношение суммы степеней принадлежности к этому правилу всех экземпляров с классом c_j к количеству экземпляров в наборе данных:

$$\sup_j = \sum_{x_p \in c_j} \beta_j(x_p) \Big/ |X| . \tag{1.5}$$

E-алгоритм является модификацией алгоритма Ishibuchi и предназначен для работы с несбалансированными данными без их предобработки [66]. В нем меры доверия и поддержки определяются через суммы степеней принадлежности, взвешенные относительно количества экземпляров каждого класса:

$$norm - conf_j = \sum_{x_p \in c_j} \beta_j(x_p) \Bigg/ \left(\frac{|X|}{|X_{c_j}|} \times \sum_{p=1}^{|X|} \beta_j(x_p) \right), \tag{1.6}$$

$$norm - sup_j = \frac{|X_{c_j}|}{|X|^2} \times \sum_{x_p \in c_j} \beta_j(x_p) , \tag{1.7}$$

где X_{c_j} – количество экземпляров, принадлежащих классу с меткой c_j. Веса правил также рассчитываются с учетом дисбаланса:

$$w_{R_j} = \left(\sum_{x_p \in c_j} \beta_j(x_p) - \sum_{x_p \notin c_j} \beta_j(x_p) \right) \Bigg/ \left(\frac{|X|}{|X_{c_j}|} \times \sum_{p=1}^{|X|} \beta_j(x_p) \right). \quad (1.8)$$

Во всех трех предыдущих инструментах для получения точной системы необходимо большое количество правил. Авторами E-алгоритма рекомендуется использовать как минимум тридцать правил для каждого класса [66]. Следовательно, минимальный объем базы правил для данных с двумя классами равняется шестидесяти.

В отличие от приведенных выше способов создания структуры классифкатора, алгоритм на основе экстремальных значений признаков классов (АЭПК) позволяет создавать базу правил минимального объема [67]. Обучающее множество экземпляров разбивается на группы с одинаковым классом. Внутри групп алгоритм находит минимальное и максимальное значение для каждого признака и строит соответствующий терм, равномерно распределенный между экстремумами. Для каждой группы формируется по одному правилу: в антецедентной части перечисляются все термы группы, в консеквент записывается является её класс.

Алгоритмы первого уровня позволяют строить классификатор за один проход, так как однократно сформированные правила и термы уже не редактируются. Получаемая такими подходами точность редко оказывается достаточной. Следующий уровень алгоритмов отличается тем, что рассчитывает некоторую метрику качества в процессе формирования структуры и на основе её значений осуществляет выбор дальнейших действий.

Алгоритм перебора является простейшим представителем второго уровня инструментов формирования структуры нечетких классификаторов. На первом шаге для каждого признака генерируется один терм, покрывающий всю область определения признака. Далее на каждом шаге количество термов увеличивается на один (или на некоторое другое заданное количество термов) и осуществляется расчет метрики качества. Процесс продолжается до тех пор, пока значение метрики не окажется лучше требуемого значения. Кроме необходимости заранее иметь некоторое представление о возможных значениях метрики качества, недостатком алгоритма является экспоненциальный рост числа правил с увеличением количества термов [68].

Ко второму уровню также можно отнести методы кластерного анализа. Алгоритмы кластеризации разделяют всё множество входных объектов на подгруппы (кластеры), руководствуясь различными мерами близости или схожести. Для создания структуры нечетких классификаторов используются центроидные инструменты кластеризации, например, алгоритм k-средних [69] или c-средних [70]. На основе информации о центре кластера и его границах строится покрывающий терм. Выбор оптимального числа кластеров является непростой задачей; как правило, оно подбирается либо эмпирически, либо с помощью средств подбора. Критичным недостатком для несбалансированных данных является то, что при малом количестве

экземпляров наименьших классов они могут быть проигнорированы алгоритмами кластеризации, так как они склонны считать такие экземпляры за выбросы. В результате в базе правил может не оказаться правил для положительных классов.

В таблице 1.1 перечислены описанные выше алгоритмы, выделены их достоинства и недостатки.

Таблица 1.1 – Основные инструменты генерации структуры нечеткого классификатора

Алгоритм	Параметры	Достоинства	Недостатки
Базовый уровень			
Равномерное разбиение	Число термов	Простота вычислений	Низкая эффективность
Случайное разбиение	Число термов	Простота вычислений	Низкая эффективность
Алгоритмы первого уровня			
Алгоритм Chi	Количество термов в правиле	Высокая эффективность для классических данных	Число сгенерированных правил может достигать большого значения (вплоть до количества экземпляров). Большое количество вычислений
Алгоритм Ishibuchi	Количество термов в правиле	Высокая эффективность для классических данных	Для получения высокой точности необходимо большое число правил. Большое количество вычислений
Е-алгоритм	Количество правил для каждого класса	Учитывается несбалансированный характер данных	Для достижения высокой точности необходимо большое число правил. Большое количество вычислений
АЭПК	Нет параметров	Минимальный объем базы правил. Простота вычислений. Быстрое построение	Невысокая эффективность
Алгоритмы второго уровня			
Алгоритм перебора	Заданный уровень метрики качества		Трудность выбора желаемого уровня метрики качества. Быстрый рост числа правил. Невысокая эффективность.
Кластерный анализ	Количество кластеров	Высокая эффективность для классических данных	Трудность подбора количества кластеров; положительные классы могут быть проигнорированы при малом количестве образцов

Ни один алгоритм не имеет такого перечня достоинств, который бы позволил назвать его лучшим. Для любого алгоритма актуальны дополнения: или для улучшения точности, или для уменьшения размера формируемой базы правил, или для внедрения возможности работы с

несбалансированными данными. Существует множество способов улучшения созданной структуры нечеткого классификатора, которые можно выделить в отдельный этап оптимизации.

1.4 Оптимизация нечеткого классификатора

Оптимизация нечеткого классификатора применяется для улучшения качества построенной модели: увеличения точности, повышения согласованности правил, снижения сложности, ускорения вычислений или достижения других критериев. Оптимизация может быть реализована различными методами; важнейшие из них приведены далее.

1.4.1 Уточнение структуры нечеткого классификатора. Под уточнением структуры будем понимать процессы создания дополнительных правил, а также разбиения или редактирования уже существующих баз правил вне зависимости от первоначального способа их формирования. Ниже описаны наиболее распространенные примеры таких методов.

Эволюционное обучение для расширения базы правил классификатора применялось еще до создания нечетких систем [71]. На вход эволюционному алгоритму подаются вектора, представляющие собой либо отдельное правило (Мичиганский подход), либо целую базу правил (Питтсбургский подход). В первом случае новые вектора правил формируются путем приложения эволюционных операторов к уже существующим векторам; правила фильтруются на основе критериев, связанных с точностью классификации или полнотой базы правил, лучшие попадают в итоговую базу. Во втором случае различные операторы применяются к базам правил, перемешивая их между собой. Лучшая версия базы выбирается на основе некоторой целевой функции. Самым распространенным инструментом в эволюционном обучении является генетический алгоритм [72]: существующие правила или базы правил кодируются как хромосомы, затем к ним применяются операторы скрещивания, формирующие новые правила или базы из уже имеющихся.

Авторами [6] описан иерархический алгоритм построения нечеткого классификатора, идея которого заключается в разбиении неудачных правил на несколько более детальных. На первом этапе происходит генерация первичной структуры с помощью любого средства (в работе [6] используется алгоритм Chi), а также оценка правил на эффективность на основе точности классификатора. На втором этапе каждое правило с низкой эффективностью расширяется: термы правила разбиваются на более мелкие области, после чего на их основе формируются новые правила. Третий этап заключается в отборе и удалении избыточных и ошибочных правил генетическим алгоритмом. Несмотря на наличие этапа сокращения правил, объем итоговой базы получается большим. В исходной статье авторы не приводят конкретных сведений о сложности построенных классификаторов, но такой вывод можно сделать по их следующей публикации

[73]. В проведенном эксперименте среди 15 исследуемых несбалансированных наборов данных имелось 5 наборов, каждый из которых насчитывал только 2 класса. Для них были построены классификаторы с объемом базы от 120 до 782,6 правил (дробное число объясняется применением схемы кросс-валидации в эксперименте. Так, для разных выборок одного и того же набора данных может быть получено различное количество правил, которое впоследствии усредняется по числу выборок). Анализируя весь эксперимент целиком, можно подсчитать, что при усредненном количестве классов, равном 4,7, в среднем было построено 447,8 правил, то есть почти по 100 правил на класс. Очевидно, модели с такой объемной структурой не поддаются интерпретации, хотя и имеют высокую точность.

Близкий подход предложен в работе [74]. Алгоритм FARC-HD формирует правила путем обнаружения ассоциаций в данных, но использует не все признаки, а некоторое ограниченное подмножество для составления сокращенных правил. Далее полученные правила взвешиваются, правила с незначительными весами исключаются. Оставшиеся проходят через отбор генетическим алгоритмом; кроме того, проводится настройка параметров путем настройки бокового смещения термов. Разработанный алгоритм позиционируется авторами как инструмент построения базы правил для данных с большим числом признаков; проблема несбалансированности в работе не была рассмотрена.

Метод итеративного обучения заключается в итерационном создании новых правил и соответствующих им термов [75]. Авторы [76] предложили методологию, которая соединяет как этап создания новых правил для непокрытых правилами экземпляров, так и этап исключения избыточных правил путем использования генетического алгоритма. Критерием включения правила в базу является согласованность: число правильно классифицируемых благодаря новому правилу экземпляров должно превышать некоторое заданное значение. Метод итеративного обучения практически не используется в задачах построения нечеткого классификатора, хотя его применение позволяет получить модели с невысокой сложностью.

Распространенной практикой уточнения базы правил является введение весов правил и процедуры их настройки. В [77] изложен метод поиска оптимальных весов правил, термы которых построены на основе знаний эксперта. Первоначальные веса рассчитываются на основе нормированных степеней принадлежности обучающего набора правилам, затем применяется алгоритм роящихся частиц для оптимизации вектора весов. Настройка весов способствует улучшению точности классификаторов, но эффективность метода падает с увеличением размера базы правил. Для работы с несбалансированными данными в [78] было предложено рассчитывать вес правила с учетом затрат за ошибочную классификацию. Чтобы сделать упор на класс меньшинства, штраф за неправильную классификацию этого класса устанавливается намного более высоким, чем стоимость неправильной классификации класса большинства.

Введение весов правил осложняет интерпретацию нечетких моделей. Чтобы частично преодолеть этот недостаток, авторы [79] совместили процесс настройки весов правил и этап сокращения базы правил для классификатора, построенного алгоритмом Ishibuchi. В предлагаемом ими алгоритме вес для каждого правила настраивается отдельно, при этом веса остальных правил считаются фиксированными (если они еще не прошли через этап настройки веса, то вес будет рассчитан по методу Ishibuchi). Для рассматриваемого правила составляется подмножество экземпляров, которые правильно классифицируются только при изменении веса этого правила. Для этого осуществляется построение двух классификаторов: при равенстве веса правила нулю в первом, и при равенстве бесконечности. В целевое подмножество экземпляров включаются образцы, которые правильно классифицируются только в одном из двух классификаторов. Далее происходит балансировка веса правила уже на этом подмножестве экземпляров. Авторы алгоритма считают, что, если подмножество экземпляров оказалось пустым, то настройка веса не повлияет на качество классификации, и правило должно быть удалено из набора. Разработанный инструмент показал свою эффективность по сравнению с аналогами, но авторы отметили его склонность к переобучению модели.

Введение весов правил может быть полезно в случае наличия базы правил большого объема, когда между правилами существует высокая конкуренция. Однако исследователями обычно игнорируется вопрос интерпретируемости классификаторов при наличии весов правил.

Таким образом, добавление этапа уточнения структуры нечетких классификаторов помогает улучшить качество классификации, но приводит усложнению модели и ухудшению интерпретируемости. Разработка алгоритмов улучшения структуры классификатора, соблюдающих баланс между точностью и интерпретируемостью, является актуальной задачей. Однако задача осложняется тем, что сложность и интерпретируемость являются субъективными понятиями; единого мнения о том, как их рассчитывать, среди исследователей нечетких систем не сформировано. В данном исследовании интерпретируемость трактуется как понятие, обратное сложности – чем больше структурных элементов в модели классификатора (количества правил, термов, признаков), тем сложнее пользователю их воспринимать.

1.4.2 Настройка термов нечеткого классификатора. Функции принадлежности, сгенерированные алгоритмом формирования структуры, не всегда оказываются близки к моделируемой предметной области, поэтому не позволяют достичь наилучшего качества классификации. Методы настройки термов направлены на подбор таких параметров функций принадлежности, которые бы позволили точнее описать признаковое пространство и, как следствие, улучшить качество классификации и интерпретируемость.

Методы настройки параметров термов подразумевают исправление функций принадлежности классификатора путем изменения их параметров. На рисунке 1.8 приведен

пример изменения треугольных термов некоторого признака *x*; каждый терм представлен тремя параметрами *a*, *b* и *c*, соответствующими координатам вершин треугольника по оси абсцисс.

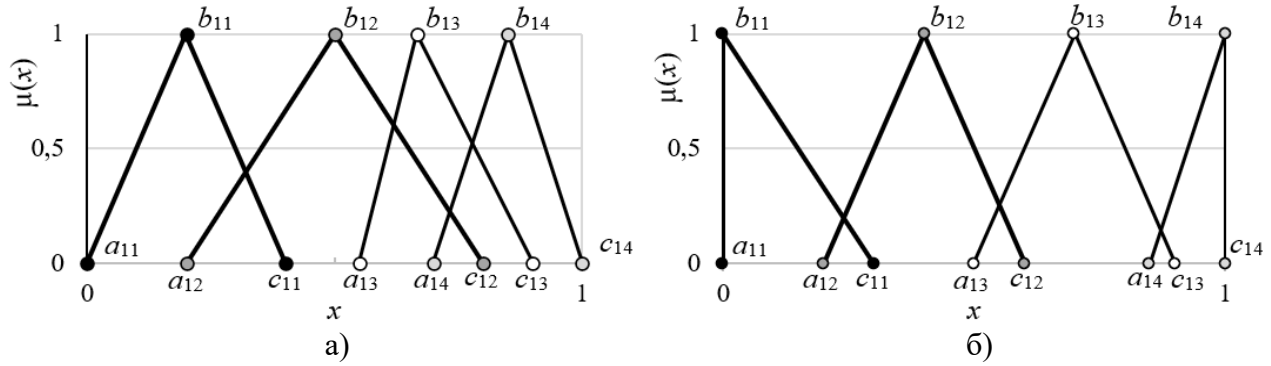

Рисунок 1.8 – Пример настройки параметров термов: а) исходные функции принадлежности; б) функции принадлежности после оптимизации

Перечень параметров термов всех признаков можно представить в виде вектора непрерывных значений (в примере выше только один признак, вектор примет следующий вид: (a_{11}, b_{11}, c_{11}, a_{12}, b_{12}, c_{12}, a_{13}, b_{13}, c_{13}, a_{14}, b_{14}, c_{14}). Поиск лучшего вектора параметров термов – задача оптимизации, для решения которой можно использовать различные оптимизационные алгоритмы, в том числе непрерывные метаэвристики. Метаэвристики осуществляют направленный случайный поиск оптимальных значений элементов вектора, руководствуясь заданной фитнес-функцией. В качестве такой функции обычно применяют точность, поэтому при получении каждой новой вариации вектора параметров термов классификатор перестраивается и оценивается заново на исследуемой выборке данных. При работе с таким способом оптимизации важно проверять, не нарушается ли форма термов. Кроме того, некоторые исследователи предпочитают избегать пустых пространств между термами. Настройка параметров термов является полезным инструментом для классификаторов, структура которых создана алгоритмами базового и первого уровня, так как она позволяет построить более тесную связь с обрабатываемыми данными.

В методе, представленном в [80], введены коэффициенты, отвечающие за боковое смещение термов. В отличие от предыдущего подхода, меняется не положение всех параметров функций принадлежности, а положение каждого терма целиком (рисунок 1.9). Коэффициенты смещения также можно настраивать инструментами оптимизации; в оригинальной работе для этой задачи применяется генетический алгоритм [80]. Достоинством такого подхода является уменьшение риска ухудшения интерпретируемости системы, который присутствует при настройке отдельных параметров термов. Однако для получения точной модели при таком подходе необходимо обеспечить высокое качество первоначальной структуры классификатора.

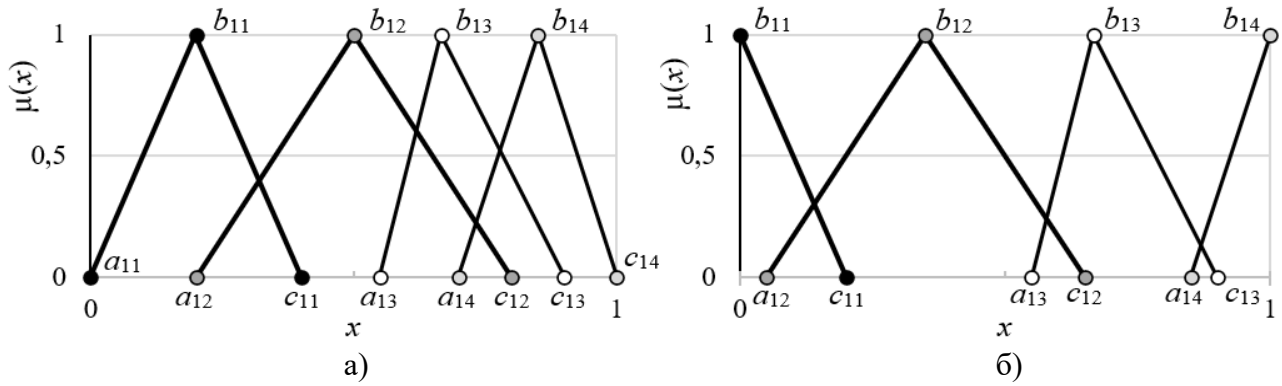

Рисунок 1.9 – Пример бокового смещения термов: а) исходные функции принадлежности; б) смещенные функции принадлежности

В некоторых работах этапы настройки параметров и уточнения базы правил совмещаются. Например, в [81] предложен модифицированный генетический алгоритм для получения оптимального набора правил и термов. В каждом векторе популяции кодируются одновременно и параметры термов в виде вещественных чисел, и правила в виде бинарной строки. Для работы с векторами, состоящими из двух типов данных, предложены специальные версии операторов мутации и кроссовера. Авторами [82] описан процесс построения нечетких классификаторов, включающий использование генетического алгоритма для создания дополнительных правил и методов целочисленного программирования для отбора наилучших среди них.

В [83] авторы представляют многоцелевой эволюционный метод, который одновременно выполняет и процесс настройки термов, и процесс выбора правил, выполняемый на исходной базе знаний классификаторов, основанных на нечетких правилах. Алгоритм нечеткой дискретизации был разработан для извлечения подходящих гранул из данных, а также для создания нечетких термов, составляющих исходную базу данных. Соответствующая база знаний была сгенерирована путем извлечения набора нечетких ассоциативных правил в соответствии с первыми двумя шагами алгоритма FARC-HD, представленного в [74].

Эффективность оптимизации термов зависит не только от качества структуры, но и от особенностей данных – наличия дисбаланса и пересечения классов, объема выборки. Например, в работе [84] после настройки функций принадлежности классификатора, предназначенного для определения наличия риска возникновения и развития сердечно-сосудистых заболеваний, на тестовой выборке была продемонстрирована точность, примерно равная 53 процентам при применении алгоритма гравитационного поиска и 61 проценту при использовании алгоритма прыгающих лягушек. Столь малые числа были получены вследствие малого объема набора данных, включавшего всего 66 экземпляров. При дальнейшем увеличении количества образцов до 168 качество классификации выросло до 90 процентов у первого алгоритма и до 95 у второго.

1.4.3 Отбор признаков. Одним из наиболее полезных способов улучшения модели классификации является отбор информативных признаков (feature selection). В рамках отбора проводится удаление избыточных и шумовых признаков, что помогает уменьшить вероятность переобучения модели, снизить её сложность, улучшить интерпретируемость, а в некоторых случаях и увеличить точность. Отбор признаков заключается в выборе из всего входного пространства такого подмножества, которое обладало бы меньшим количеством элементов при сопоставимой или большей точности классификации по сравнению с полным набором. Сформированное подмножество должно быть достаточным для адекватного представления всех классов, присутствующих в обучающих выборках. Самым результативным способом отбора, с помощью которого гарантированно находится лучшее решение, является полный перебор, однако с ростом количества переменных многократно возрастают затраты ресурсов и времени на вычисления. Поэтому создаются и исследуются такие методы отбора, которые позволяют найти оптимальное решение за меньшее время. Такие методы принято разделять на три вида: обертки, фильтры и интегрированные методы [85].

Обертками называют методы, которые оценивают каждое подмножество признаков на основе качества построенного на нем классификатора [86]. В роли инструмента отбора выступают, как правило, эвристики (случайный поиск, жадный поиск, случайный поиск с памятью и другие) и метаэвристики (генетический алгоритм, алгоритм роящихся частиц и т.п.) [87]. Так как такие алгоритмы обычно являются итерационными, то после каждой итерации требуется перестроение классификатора. Методы оберток могут требовать значительных затрат времени и ресурсов в случае наличия большого объема данных или сложной структуры классификатора. Достоинством оберток является возможность нахождения такого набора признаков, который будет оптимальным для конкретного алгоритма классификации. Недостатком данных методов является высокая вычислительная сложность, что приводит к трудности применения в задачах высокой размерности пространства поиска, то есть большого количества исходных признаков в данных.

Для отбора признаков по схеме обертки при построении нечеткого классификатора были успешно использованы бинарные версии гармонического поиска [88], прыгающих лягушек [89], алгоритма ласточек [90] и других метаэвристик. Так, в работах [91, 92] применение метаэвристик позволило выделить наиболее важные признаки для проведения процедуры аутентификации пользователей по динамическим характеристикам подписи.

Отбор признаков в режиме обертки эффективно функционирует в комбинации с настройкой параметров. В работе [93] использование двух версий алгоритма гравитационного поиска – бинарной и непрерывной – способствовало получению конкурентноспособного нечеткого классификатора для анализа сетевого трафика.

Методы фильтрации, в отличие от оберток, применяются на этапе подготовки данных до начала построения классификатора, так как принципы отбора этих методов основываются на поиске зависимостей в наблюдаемых данных. В работе [94] выделяют четыре группы фильтров. Методы первой группы, основанные на расстоянии, отбирают признаки, гарантирующие наибольшее расстояние между классами. Способы второй группы, основанные на количестве информации, оставляют такие атрибуты, которые при присоединении к имеющемуся набору уменьшают его энтропию. В третьей группе фильтров рассчитывается зависимость между признаками и классами с помощью коэффициента корреляции или количества взаимной информации. Четвертую группу представляют фильтры, которые минимизируют количество несогласованных признаков. Случай проявления несогласованности – это наличие двух экземпляров, относящихся к разным классам, но имеющих одинаковые значения одних и тех же признаков. Фильтры функционируют быстрее оберток, но, как правило, демонстрируют меньшую эффективность [95]. Кроме того, многие средства фильтрации отбирают признаки независимо друг от друга и не обладают способностью обнаруживать сложные взаимосвязи между признаками [96].

Особенностью интегрированных (встроенных) методов является принцип отбора признаков, являющийся частью общего механизма построения и обучения модели на конкретных данных. Примером применения подобных методов является отбор признаков во время формирования дерева решений [97]. Однако не каждый алгоритм классификации позволяет встроить процесс отбора в процесс построения модели.

Кроме перечисленных подходов, распространены гибридные методы отбора признаков, которые осуществляют предварительный отбор признаков с помощью фильтров для отсеивания самых неинформативных признаков, а на полученном подмножестве признаков осуществляют построение классификатора и дальнейший отбор в режиме обертки [98]. Этот подход актуален при наличии такого количества признаков или экземпляров в исходном наборе данных, которое потребует значительных затрат ресурсов для многократного перестроения модели при реализации схемы обертки.

Существуют публикации, посвященные разработке алгоритмов отбора признаков для несбалансированных данных. В работах [26, 99] при проведении отбора по схеме обертки используются фитнес-функции, основанные на средней геометрической точности. Отмечается, что качество распознавание наименьшего класса улучшается по сравнению с традиционной фитнес-функцией, основанной на общей точности. Авторы [100] применяют меру симметричной неопределенности, основанной на энтропии, для взвешивания признаков в зависимости от метки класса, чтобы выявить наиболее важные признаки для наиболее редких классов. С целью найти наилучшую комбинацию среди признаков с наибольшими весами используется гармонический

алгоритм. Несмотря на присутствие этапа взвешивания признаков, выходом алгоритма является только бинарный вектор, где ноль означает отсутствие признака в обучении классификатора, единица – за присутствие. Однако есть небольшое количество исследований, в том числе для нечеткого классификатора, где признаки не только сокращаются, но и взвешиваются. В рамках нечетких систем такой подход применяется только для самообучающихся нечетких классификаторов [101] и не затрагивает проблему дисбаланса данных.

Все перечисленные варианты оптимизации нечетких классификаторов могут быть осуществлены с помощью метаэвристических алгоритмов. Однако для работы с несбалансированными данными необходимы метаэвристики, способные эффективно осуществлять как глобальный поиск, так и локальный.

1.5 Метаэвристические алгоритмы

Метаэвристические алгоритмы являются видом эвристик, основанных на имитации физических процессов, природных явлений, поведения животных, социальных механизмов. Они применяются в случае, когда для выполнения задачи достаточно отыскать качественное по какому-либо критерию решение без доказательства его оптимальности. В отличие от классических алгоритмов оптимизации, основанных на производных, метаэвристики имеют меньшую вероятность попадания в локальные экстремумы благодаря использованию случайных переменных и обмену информацией между элементами популяции в случае популяционных алгоритмов.

Метаэвристики можно разделять по принадлежности метафоры, на которой они основаны, к одной из следующих групп [102]:

– эволюционные алгоритмы;

– алгоритмы, имитирующие поведение животных;

– алгоритмы, имитирующие процессы в человеческом обществе;

– алгоритмы, имитирующие природные процессы и явления.

Представителем эволюционных метаэвристик является генетический алгоритм, который представляет решения в виде популяции хромосом. Хромосома состоит из последовательности генов. Ген кодирует один параметр задачи или координату в пространстве поиска. В течение итераций применяются генетические операторы, в которых отражены важные эволюционные принципы наследования: выживание самых приспособленных хромосом и случайные изменения генов. Путем регулирования процедур применения генетических операторов алгоритм добивается того, что приспособленность хромосом в среднем возрастает от поколения к поколению [103]. Другими представителями эволюционных алгоритмов являются

дифференциальная эволюция [104], эволюционная стратегия, эволюционное программирование [105].

Существует множество метаэвристик, имитирующих поведение животных. Одним из первых и основных алгоритмов этой группы является алгоритм роящихся частиц, моделирующий поведение группы животных, например, косяка рыб, в процессе поиска пищи [106]. Движение каждой частицы задается ее лучшей позицией, текущей скоростью, ускорением, заданным предыдущей позицией, и ускорением, заданным лучшей частицей в рое. Алгоритм роящихся частиц имеет большое количество модификаций; например, в работе [107] вместо классического ньютоновского движения используется квантовая модель поведения частиц, а в [108] для улучшения поиска применяется хаотическое преобразование координат. На основе алгоритма роящихся частиц построено множество других метаэвристик. Птичий алгоритм основан на наблюдениях за поведением стай птиц, занимающихся сбором пищи и перемещениями внутри стаи [109]. Крилевый алгоритм построен на поведении группы антарктических рачков. Скопление крилей распространяется в поисках пропитания, постепенно сокращаясь из-за хищничества внутри группы [110].

Метаэвристики третьей группы основаны на воспроизведении поведения человека или общества. Культурные алгоритмы имитируют эволюцию культурного развития, рассматривая череду процессов на микро- и макроуровне. На микроуровне поведенческие черты популяции индивидов меняются посредством применения социальных операторов. Макроуровень отвечает за накопление и осмысление опыта индивидов, который используется для развития популяции [111]. Меметические алгоритмы основаны на нео-дарвиновском принципе эволюции и концепции мема – единицы передачи культурной информации, наследующейся от одного поколения человечества к другому посредством обучения, повторения и т.п. [112]. Алгоритм креативного обучения моделирует человеческий мыслительный процесс при решении абстрактной прикладной задачи. Входной вектор представляет собой некоторую идею, характеризующуюся своей новизной. Далее из нескольких идей формируются новые, пока не появится идея, являющаяся лучшим решением [113]. Метаэвристика «мозговой штурм» также представляет решения в виде идей, но разбивает их на группы – кластеры, имитируя групповое мышление. Лучшие идеи групп становятся центрами своих кластеров. С помощью последовательности операторов, имитирующих процесс естественного отбора, идеи перемешиваются, изменяются на некоторый шаг и фильтруются на основе заданной фитнес-функции [114].

Примерами метаэвристик, построенных на имитации природных процессов, являются гармонический поиск, использующий принцип создания музыкальных фраз [115], алгоритм «минный взрыв», который среди взрывающихся по цепочке осколочных мин ищет самую

взрывоопасную [116], «всемирный потоп», построенный на имитации поиска максимально высокого участка поверхности при процессе затоплении земли [117], и алгоритм гравитационного поиска, оперирующий законами тяготения Ньютона [118].

Общий порядок действий популяционных метаэвристических алгоритмов следующий. Алгоритм осуществляет инициализацию популяции оптимизируемых векторов или принимает на вход уже готовую популяцию. Далее происходит итерационное обновление векторов с помощью различных преобразований. В работе [119] процесс обновления агента сформулирован в виде следующего уравнения:

$$\mathbf{x}(t + 1) = \mathbf{x}(t) + \mathbf{\Delta}(t + 1), \quad (1.9)$$

где $\mathbf{x}(t)$ – текущий агент, $\mathbf{x}(t + 1)$ – обновленный агент, $\mathbf{\Delta}(t + 1)$ – модифицирующий вектор, t – текущая итерация алгоритма. Вектор $\mathbf{\Delta}(t + 1)$ может быть получен следующими способами [119]:

– расчетом градиента;

– комбинацией частей нескольких исходных векторов;

– вычислением произведения разности нескольких решений и случайного числа;

– наложением распределения Гаусса, Коши, Леви.

Величина значений модифицирующего вектора определяет направленность поиска в алгоритме. Если обновленный агент $\mathbf{x}(t + 1)$ мало отличается от исходного вектора $\mathbf{x}(t)$, то поиск направлен на интенсификацию, то есть исследование локальной области решения задачи. В противном случае осуществляется диверсификация – поиск в глобальной области.

Задача алгоритма при осуществлении преобразований состоит в перемещении элементов популяции как можно ближе к искомому экстремуму оптимизируемой функции. Для оценки качества агентов рассчитывается фитнес-функция. Чаще всего фитнес-функция оптимизации совпадает с целевой функцией всей модели [120], но может и отличаться от неё.

После выполнения каждой итерации происходит проверка достижения условия остановки. В качестве условия остановки могут выступать следующие критерии: пройденное количество итераций, достижение заданного значения фитнес-функции, равенство всех элементов популяции, отсутствие существенных изменений в популяции в течение заданного числа итераций и другие. Если критерий остановки достигнут, то решение с лучшей фитнес-функцией подается на выход алгоритма, в противном случае итерации продолжаются.

Метаэвристические алгоритмы не гарантируют нахождения глобального экстремума оптимизируемой функции и обладают, как правило, большим количеством параметров, которые нужно подбирать для решения каждой конкретной задачи [121]. Однако для экономии времени исследователи стараются найти такие параметры алгоритмов, которые являлись бы оптимальными для наибольшего числа задач.

Согласно «теореме о бесплатных завтраках», универсального алгоритма оптимизации, который бы демонстрировал превосходные результаты на всех задачах, не существует [122]. Одни и те же метаэвристики могут демонстрировать различную эффективность не только для разных задач, но и на различных исходных данных. Поэтому разработчикам информационных систем необходим широкий арсенал алгоритмов, знание о их сильных и слабых сторонах, а также понимание взаимосвязи между компонентами алгоритма и спецификой задачи [119]. Далее приведено краткое описание двух метаэвристик, которые будут использованы в работе.

1.5.1 Алгоритм гравитационного поиска. Гравитационный алгоритм, впервые описанный иранской ученой Э. Рашеди (E. Rashedi) [118] в 2009 году, основан на фундаментальных законах тяготения Исаака Ньютона. Популяция входных переменных представляет собой систему частиц, между которыми действуют силы тяготения. Масса частицы зависит от соответствующего ей значения фитнес-функции; частицы с лучшей фитнес-функцией обладают наибольшей массой. Силы тяготения заставляют мелкие частицы двигаться по направлению к крупным, тем самым осуществляя глобальный поиск. Так как на крупные частицы тоже действуют силы тяготения со стороны всей остальной системы, они тоже перемещаются на небольшие расстояния. Таким образом исследуется локальная область вокруг временно лучших решений.

Гравитационный поиск получил широкое распространение среди метаэвристических алгоритмов [123]. В работе [124] применяют модифицированный гравитационный алгоритм с хаотической мутацией для решения задачи уменьшения использования топлива и количества выбросов загрязняющих веществ при работе электроэнергетических систем. Авторы сравнивают результаты исследуемого инструмента с результатами алгоритма роящихся частиц, эволюционного алгоритма и дифференциальной эволюции, и приходят к заключению, что «гравитационный поиск» является более эффективным. Также эта метаэвристика была успешно апробирована при сегментации изображений методом *k*-средних для определения наилучшего положения центров кластеров [125]. Алгоритм продемонстрировал лучшие результаты по сравнению со стандартными средствами цветового квантования, но показал низкую скорость работы.

Бинарный гравитационный алгоритм описан в [126]. Алгоритм практически идентичен непрерывному, изменяется только процесс изменения частиц – вместо прибавления перемещения используется функция трансформации, с помощью которой числовые характеристики частиц конвертируются в битовые значения.

Алгоритм гравитационного поиска хорошо зарекомендовал себя в задачах настройки параметров нечеткого классификатора [99] и отбора признаков в режиме обертки [127], в том числе для несбалансированных данных [128]. Однако он продемонстрировал ряд недостатков,

которые также отмечены в работах [123, 129]: медленную скорость сходимости, застревание в локальных оптимумах, сложность вычислений и трудность подбора параметров.

1.5.2 Алгоритм прыгающих лягушек. Метаэвристический алгоритм, идея которого заключается в имитации процесса поиска пищи популяцией лягушек, был впервые описан в 2003 году [130]. В 2006 году создатели алгоритма сравнили его с генетическим алгоритмом при решении задачи проектирования систем распределения воды и признали свой инструмент более эффективным [131]. С момента своего создания метаэвристика получила несколько модификаций, наибольшим распространением отличается версия, изложенная в [132]. В ней исходная популяция сортируется и последовательно разбивается на независимые группы – мемплексы, в каждой из которых есть свой локальный лидер – лягушка, сумевшая отыскать наиболее удачное место с пищей. Остальные лягушки групп перемещаются в направлении к своему лидеру. Параллельно происходит обмен информацией между популяцией путем перетасовки групп, выделение среди лидеров самого успешного, и постепенное движение всех лягушек к глобальному лидеру.

Метаэвристика «прыгающие лягушки» была использована авторами [133] для определения расположения системы, управляющей мощностями генераторов энергии в распределительной сети для уменьшения потерь мощности. В [134] описано приложение этого алгоритма для планирования маршрута подводных аппаратов. Применение «прыгающих лягушек» для подбора параметров алгоритма регрессии, осуществляющего прогнозирование качества воды, описано в исследовании [135]. Бинарные версии алгоритма прыгающих лягушек применяются для отбора признаков в нечетких классификаторах [136, 137].

В большинстве статей отмечается высокая скорость работы алгоритма и легкость подбора параметров. Однако «прыгающие лягушки» формируют модифицирующий вектор на основе простейшего оператора, состоящего из произведения разности двух векторов и случайного числа, поэтому алгоритм не способен вносить большие изменения в обновляемые вектора, что критично при обширной размерности поиска.

1.5.3 Гибридизация метаэвристик. Для достижения лучших решений метаэвристика должна обеспечивать баланс между интенсификацией и диверсификацией, то есть между интенсивностью и широтой поиска [120]. Другими словами, она должна и предусматривать максимальный охват области поиска, и уметь исследовать близлежащие точки вокруг временно лучших решений. Для этого в алгоритмах различным образом комбинируют локальный и глобальный поиск. В локальном поиске происходит концентрация операций поиска в локальной области, например, передвижение элементов популяции в пределах некоторых групп. Задача локального поиска заключается в проверке, не найдутся ли более качественные решения при внесении мелких изменений в уже существующие. Глобальный поиск направлен на исследование

пространства поиска в широком масштабе, предполагает генерацию большого количества разнообразных решений, перемещение популяции к имеющемуся лучшему элементу популяции и внесение крупных изменений в популяцию. Так как создатели метаэвристик прежде всего стараются подчинить алгоритм оригинальной метафоре, некоторые элементы алгоритма могут оказаться не полностью проработаны. Такая ситуация наблюдается и у двух описанных выше метаэвристик: «гравитационный поиск» осуществляет большие, но практически неуправляемые изменения в оптимизируемых векторах, а «прыгающие лягушки», напротив, инициируют слишком слабые преобразования. Поскольку недостатки алгоритмов противоположны, объединение в один гибрид позволит получить сбалансированный инструмент оптимизации.

1.6 Постановка задачи

Цель классификации заключается в определении для входного объекта **x** из множества объектов $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|X|}\}$ наиболее подходящего класса из множества классов $C = \{c_1, c_2, ..., c_m\}$. Каждый объект описывается вектором значений признаков $\mathbf{x}_p = (x_p^1, x_p^2, ..., x_p^n)$, где x_p^i – значение i-го признака объекта $\mathbf{x}_p$ ($i = \overline{1,n}$), $p = \overline{1,|X|}$.

Во время обучения решающий алгоритм на имеющихся прецедентах (экземплярах данных) восстанавливает зависимость между признаками и классами. В нечетком классификаторе вывод о принадлежности объекта классу строится на основе степени принадлежности этого объекта к отдельным правилам или группам правил с одним и тем же выходным параметром. На рисунке 1.10 в виде черного ящика представлена базовая модель нечеткого классификатора.

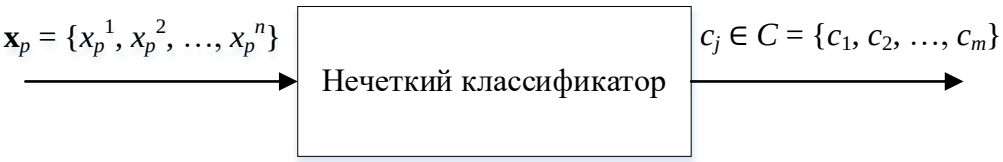

Рисунок 1.10 – Модель базового нечеткого классификатора

База правил нечеткого классификатора состоит из перечня утверждений следующего образца [138]:

$$\text{ЕСЛИ } x^1 = T_{1j}, \text{ И } x^2 = T_{2j}, \text{ И } \ldots, \text{ И } x^n = T_{ni}, \text{ ТО класс} = c_k, \qquad (1.10)$$

где T_{ij} – нечеткий терм, характеризующий признак x^i в j-ом правиле ($j = \overline{1,R}$), R – количество правил в базе, c_k – метка k-го класса ($k = \overline{1,m}$).

Нечеткие термы описывают входные переменные и могут представлять собой различные функции принадлежности: треугольного, трапециевидного, гауссова типа. Последовательность

параметров термов всех признаков составляет вектор антецедентов **θ**. Если количество термов для каждого признака совпадает с числом правил, то размерность этого вектора определяется следующим выражением:

$$|\boldsymbol{\theta}| = n \times R \times p\,, \tag{1.11}$$

где p – количество параметров, которыми задается терм.

Для термов гауссова типа вектор **θ** составляется путем последовательного перечисления двух параметров – координаты вершины по оси абсцисс a_{ij} и стандартного отклонения терма b_{ij}: $\boldsymbol{\theta} = (a_{11}, b_{11}, \ldots, a_{1R}, b_{1R}, a_{21}, b_{21}, \ldots, a_{nR}, b_{nR})$. На рисунке 1.11 приведен пример нечеткого разбиения двух признаков тремя функциями принадлежности гауссова типа, которые и будут использоваться в экспериментальной части исследования.

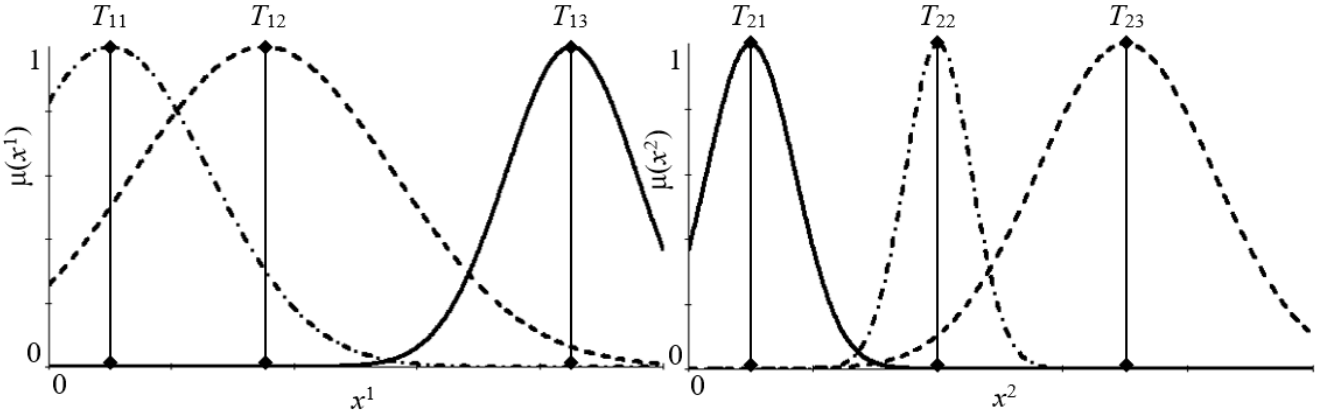

Рисунок 1.11 – Нечеткое разбиение признаков x^1 и x^2 тремя термами треугольного типа

Степень принадлежности некоторой точки x терму гауссова типа T_{ij} вычисляется следующим образом:

$$\mu_{T_{ij}}(x) = \exp\left(-\left(\frac{x - a_{ij}}{b_{ij}}\right)^2\right). \tag{1.12}$$

Количество и расположение термов, а также число и содержание правил определяет алгоритм генерации структуры. После завершения этапа генерации структуры осуществляется процедура вывода и оценка качества построенного классификатора.

Для поступившего на вход классификатора объекта $\mathbf{x}_p$ вычисляется степень его принадлежности каждому классу:

$$\beta_k(\mathbf{x}_p) = \sum_{r_{jk}} \prod_{i=1}^{n} \mu_{T_{ij}}(x_p^i)\,, \tag{1.13}$$

где r_{jk} – правила с выходной меткой класса k, $\mu_{T_{ij}}(x_p^i)$ – значение функции принадлежности терма T_{ij} в точке x_p^i. Выходом является класс с наибольшей степенью принадлежности:

$$class = c_{k^*}, \quad k^* = \arg\max_{1 \le k \le m} \beta_k(\mathbf{x}_p)\,. \tag{1.14}$$

Как было замечено ранее, при построении классификаторов, обрабатывающих несбалансированные данные, важно выбрать адекватный критерий оценки полученной модели. В данной работе будут рассмотрены наборы данных, насчитывающие только два класса. Поэтому наиболее объективными показателями качества можно считать проценты правильной классификации на каждом из этих классов. В качестве объединенной метрики будет использована средняя геометрическая точность *GM*:

$$GM = \sqrt[m]{\prod_{k=1}^{m} (inst_k^* / inst_k)}\,, \tag{1.15}$$

где $inst_k^*$ – количество правильно определенных экземпляров k-го класса, $inst_k$ – количество всех экземпляров k-го класса, $k \in [1; m]$. Чем меньше экземпляров некоторого класса представлено в данных, тем больший вклад в среднюю геометрическую точность будет вносить правильно определенный экземпляр этого класса. Другие достоинства этой метрики заключаются в быстром вычислении, отсутствии коэффициентов, широком распространении среди исследователей алгоритмов машинного обучения.

Цель обучения классификатора заключается в поиске максимума выбранной целевой функции.

1.7 Выводы

1. Проблема дисбаланса данных может быть решена на уровне данных или на уровне алгоритмов. Инструменты генерации искусственных данных могут приводить к увеличению ошибок в данных и усилению перекрытия классов. Объединение решающих алгоритмов в ансамбли усложняет систему, требует больших временных затрат и ухудшает интерпретируемость итоговых моделей. Актуальной задачей является разработка алгоритмов построения нечетких систем для анализа несбалансированных данных, способных демонстрировать высокую эффективность без наличия этапа предобработки данных.

2. Нечеткие системы, основанные на правилах, могут применяться для построения моделей классификации при использовании правил вида Мамдани. Основным достоинством нечетких генетических классификаторов является интерпретируемость – понятность логики вывода модели конечному пользователю. Создание точного, но легко интерпретируемого классификатора является трудной задачей. Построение нечеткого классификатора можно разделить на два этапа: создание базовой структуры и её оптимизация.

3. От качества структуры нечеткого классификатора зависит как точность и скорость вывода модели, так и эффективность алгоритмов оптимизации, применяющихся на следующем шаге построения классификатора. Создание структуры включает в себя генерацию функций

принадлежности (термов) и формирование базы правил. Немногие инструменты формирования структуры способны одновременно учитывать дисбаланс классов, чтобы качественно справляться с распознаванием наименьшего класса, и формировать компактные классификаторы с легко воспринимаемым пользователем числом правил, поэтому этап оптимизации структуры является необходимым.

4. Основные варианты оптимизации нечеткого классификатора – это уточнение структуры, настройка параметров функций принадлежности и отбор признаков. Алгоритмы уточнения структуры направлены на генерацию дополнительных правил или исключение лишних, а также на взвешивание правил. Настройка параметров функций принадлежности необходима для поиска положения и формы термов, максимально точным образом описывающих предметную область. Отбор признаков полезен для снижения сложности модели и снижения риска переобучения. Разработка алгоритмов оптимизации нечеткого классификатора, позволяющих улучшить качество классификации на несбалансированных данных, но не приводящих к значительному увеличению сложности модели и ухудшению интерпретируемости, является актуальной задачей.

5. Метаэвристические алгоритмы являются эффективным инструментом оптимизации, позволяющим реализовать все перечисленные выше варианты улучшения нечеткого классификатора. Для достижения высоких результатов, метаэвристика должна соблюдать баланс между диверсификацией (глобальным поиском) и интенсификацией (поиском в локальной области), однако следование метафоре метаэвристики не всегда позволяет достичь такого баланса. Совмещение в один гибрид двух метаэвристик, склонных к противоположным направлениям поиска, способствует достижению высоких результатов оптимизации.

Глава 2. Алгоритмы построения нечетких классификаторов несбалансированных данных

2.1 Алгоритм формирования структуры нечеткого классификатора на основе метаэвристики «прыгающие лягушки»

Алгоритм предназначен для итерационного процесса генерации и настройки новых правил с целью расширения базы правил нечеткого классификатора [139, 140]. Первичная база правил может быть сгенерирована любым способом, например, алгоритмом кластеризации данных или алгоритмом экстремальных значений признаков классов. Новые правила создаются для тех классов, которые имеют наименьшую долю правильной классификации. Зачастую ими оказываются положительные классы, поэтому алгоритм является полезным инструментом для улучшения качества работы классификатора с несбалансированными данными.

Далее приведено пошаговое описание алгоритма формирования структуры нечеткого классификатора на основе алгоритма прыгающих лягушек. Входными данными является текущая база правил *Base*, количество добавляемых правил *NR* и параметры метаэвристики: *GIt* – количество глобальных итераций, *LIt* – количество локальных итераций, *NM* – число мемплексов, *NA* – число агентов в мемплексе (подмножестве векторов), *const* – константа для генерации новых параметров термов, γ – коэффициент приоритета метрики в фитнес-функции.

Одна итерация создания и настройки правила состоит в следующей последовательности действий. На основе текущей базы правил *Base* осуществляется расчет доли правильной классификации каждого класса в отдельности и выбор класса с наихудшим показателем. Формируется популяция векторов (агентов) $\{\mathbf{R}_1^*,\mathbf{R}_2^*,\ldots,\mathbf{R}_N^*\}$, где N – размер популяции, равный произведению *NA* и *NM*. Каждый вектор **R*** представляет собой вариацию нового правила и состоит из перечня параметров термов и консеквента выбранного класса. Термы генерируются на основе экстремальных значений признаков с некоторым отклонением. Параметры термов гауссова типа в новом правиле **R*** определяются следующим образом:

$$a_{iR^*} = \min(\mathbf{x}^i) + (\max(\mathbf{x}^i) - \min(\mathbf{x}^i)) \times \mathrm{rand}, \tag{2.1}$$

$$b_{iR^*} = (\max(\mathbf{x}^i) - \min(\mathbf{x}^i)) \times \mathrm{rand}/2, \tag{2.2}$$

где a_{iR^*} – координата вершины функции принадлежности для i-го признака, b_{iR^*} – разброс функции, $\min(\mathbf{x}^i)$ и $\max(\mathbf{x}^i)$ – минимальное и максимальное значение i-го признака, rand – равномерно распределенное случайное число из промежутка [0;1]. Наличие случайной компоненты обеспечивает разнообразие в популяции.

Далее запускается глобальный поиск, в котором популяция сортируется по убыванию значения фитнес-функции, после чего счетчик глобальных итераций увеличивается на единицу.

Фитнес-функция отражает улучшение классификации при добавлении нового правила **R*** к текущей базе правил по сравнению с качеством классификации, полученным только на текущей базе:

$$fit(Base \cup \mathbf{R}^*) = score(Base \cup \mathbf{R}^*) - score\left(Base\right), \tag{2.3}$$

где *score*(•) – комбинация средней геометрической точности *GM* и общей точности *Acc*:

$$score = \gamma \times GM + (1-\gamma) \times Acc, \tag{2.4}$$

Коэффициент γ, принадлежащий промежутку [0;1], управляет приоритетом между двумя метриками. Общая точность рассчитывается как отношение суммы правильно определенных классификатором экземпляров к общему количеству экземпляров:

$$Acc = \sum_{k=1}^{m} inst_k^* \Big/ \sum_{k=1}^{m} inst_k \,. \tag{2.5}$$

Общая точность в фитнес-функции полезна для данных с большим дисбалансом, так как в таких случаях при увеличении числа правильно распознанных экземпляров отрицательного класса средняя геометрическая точность практически не изменяется.

После сортировки агенты популяции последовательно разбиваются на подгруппы – мемплексы, внутри которых независимо проводится локальный поиск заданное количество локальных итераций. В каждом мемплексе выбираются **best** и **worst** – векторы с лучшей и худшей фитнес-функцией. На основе выбранных векторов генерируется новое правило:

$$\mathbf{new} = \mathrm{rand} \times const \times (\mathbf{best} - \mathbf{worst}) + \mathbf{worst}\,. \tag{2.6}$$

Если фитнес-функция созданного вектора оказывается лучше, чем у **worst**, то **worst** заменяется на **new**. В противном случае генерация происходит повторно, но на этот раз в **best** помещается глобально лучший агент (первый в популяции). Если и в этом случае не удалось улучшить вектор **worst**, то на его место записывается вектор, сгенерированный путем наложения случайного отклонения на глобально лучший агент.

Когда локальные итерации истекают, алгоритм возвращается к глобальному поиску. После завершения глобальных итераций весь процесс повторяется заново, пока не будут добавлены все *NR* правил. Выходом алгоритма является дополненная база правил нечеткого классификатора.

Далее приведен псевдокод предложенного алгоритма.

Вход: *Base*, *NR*, *GIt*, *LIt*, *NM*, *NA*, *const*, γ.

Выход: *Base*.

цикл пока (*NR* > 0):

Расчет $score(Base) = \gamma \times GM(Base) + (1-\gamma) \times Acc(Base)$.

Определение худшего класса: $c_{\text{worst}} = c_{k^*}$, $k^* = \arg\min_{1 \le k \le m} (inst_k^* / inst_k)$.

Генерация популяции $\{\mathbf{R}_1^*, \mathbf{R}_2^*, \ldots, \mathbf{R}_{NM \times NA}^*\}$ для c_{worst}.

Вычисление фитнес-функции для каждого $\mathbf{R}^*$:

$fit(Base \cup \mathbf{R}^*) = score(Base \cup \mathbf{R}^*) - score(Base)$;

цикл пока ($GIt > 0$):

Сортировка популяции по убыванию фитнес-функции.

Последовательное разбиение популяции на NM групп.

цикл по NM:

цикл пока ($LIt > 0$):

Локальный поиск.

$LIt := LIt - 1$.

конец цикла.

конец цикла.

$GIt := GIt - 1$.

конец цикла.

Добавление $\mathbf{R}^*$ с максимальной фитнес-функцией в *Base*.

$NR := NR - 1$.

конец цикла.

вывод *Base*.

конец алгоритма.

Ниже представлен псевдокод функции локального поиска алгоритма прыгающих лягушек.

Функция Локальный поиск

Определение худшего агента группы $\mathbf{worst} = R_{bf}^*, bf = \arg\min fit(Base \cup R^*)$.

Определение лучшего агента группы $\mathbf{best} = R_{bf}^*, bf = \arg\max fit(Base \cup R^*)$.

Генерация нового вектора $\mathbf{new} = \text{rand} \times const \times (\mathbf{best} - \mathbf{worst}) + \mathbf{worst}$.

если $fit(Base \cup \mathbf{new}) > fit(Base \cup \mathbf{worst})$:

worst := **new**

иначе:

Генерация нового вектора $\mathbf{new} = \text{rand} \times const \times (R_0^* - \mathbf{worst}) + \mathbf{worst}$.

если $fit(Base \cup \mathbf{new}) > fit(Base \cup \mathbf{worst})$:

worst := **new**.

иначе:

Случайная генерация вектора **new** на основе R_0^*.

worst := **new**.

Обновление *fit*(**worst**).

конец функции.

Число добавляемых алгоритмом правил определяется разработчиком системы классификации. При определении этого параметра необходимо исходить из текущей точности классификатора и желаемой сложности. Кроме того, разработчик должен понимать, какое количество правил будут способны воспринять и интерпретировать конечные пользователи.

2.2 Гибридный алгоритм настройки параметров нечеткого классификатора несбалансированных данных

Оптимизация параметров термов является одним из наиболее эффективных способов повышения качества классификатора. Метаэвристические алгоритмы позволяют с небольшой вычислительной сложностью и высокой скоростью найти эвристические решения задачи нахождения экстремумов функции. Для достижения высоких результатов инструмент оптимизации должен соблюдать баланс между интенсивностью и широтой поиска, однако немногие метаэвристики обладают таким свойством.

Гравитационный алгоритм и алгоритм прыгающих лягушек хорошо зарекомендовали себя в задачах оптимизации параметров нечеткого классификатора [89, 99]. Но, так как «гравитационный поиск» представляет собой в большей степени глобальный поиск, а в «прыгающих лягушках» существеннее проработан поиск локальный, их комбинация позволит повысить эффективность оптимизации [141, 142]. Подробно предлагаемый гибрид описан далее.

Входными параметрами являются: вектор параметров антецедентов $\boldsymbol{\theta}_0$, количество итераций глобального и локального поиска *GIt* и *LIt* соответственно, количество агентов в мемплексе *NA*, количество мемплексов *NM*, начальное значение гравитационной постоянной G_0, коэффициент уменьшения α, константы ε и *const*, количество лучших агентов K_{best}.

На основе исходного агента $\boldsymbol{\theta}_0$, представляющего собой последовательный перечень параметров всех термов классификатора, создается популяция $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N-1}\}$, где $N = NA \times NM$. Каждый вектор популяции генерируется путем наложения на $\boldsymbol{\theta}_0$ случайного отклонения. Далее рассчитывается фитнес-функция агентов, производится сортировка по убыванию значения фитнес-функции, и начинается глобальный поиск, взятый из метаэвристики «гравитационный поиск».

Глобальный поиск можно представить в виде следующей последовательности шагов. На первом шаге оцениваются массы векторов:

$$mass_i(t) = \frac{fit(\boldsymbol{\theta}_i(t)) - fit(\boldsymbol{\theta}_{worst}(t))}{fit(\boldsymbol{\theta}_{best}(t)) - fit(\boldsymbol{\theta}_{worst}(t))}, \tag{2.7}$$

$$M_i(t) = mass_i(t) \Big/ \sum_{j=0}^{N-1} mass_j(t), \tag{2.8}$$

где $M_i(t)$ – масса i-го агента на текущей итерации t, $i \in [0, N\text{-}1]$, $t \in [1, GIt]$, $fit(\boldsymbol{\theta}_i(t))$ – значение фитнес-функции i-го агента, $fit(\boldsymbol{\theta}_{worst}(t))$ и $fit(\boldsymbol{\theta}_{best}(t))$ – значение фитнес-функции худшего и лучшего вектора.

Равнодействующая сил тяготения между агентом и K_{best} лучшими агентами сообщает агенту ускорение. Благодаря сортировке, K_{best} лучших частиц – это вектора с индексами от 0 до K_{best}-1. Ускорение каждого d-го элемента i-го агента рассчитывается на втором шаге глобального поиска:

$$a_i^d(t) = G(t) \times \sum_{j=0,\, j\neq i,\, j\in K_{\text{best}}}^{N} \text{rand}(0,1) \times \frac{M_j \times (\theta_j^d(t) - \theta_i^d(t))}{\left\| \boldsymbol{\theta}_j(t) - \boldsymbol{\theta}_i(t) \right\| + \varepsilon}, \tag{2.9}$$

где $G(t)$ – значение гравитационной постоянной, обновляющееся на каждой итерации:

$$G(t) = G_0 \times \exp(-\alpha \times t / GIt). \tag{2.10}$$

На третьем шаге определяется скорость как сумма набранного ускорения и случайной компоненты текущей скорости, а также происходит обновление элементов векторов:

$$V_i^d(t+1) = \text{rand}(0,1) \times V_i^d(t) + a_i^d(t), \tag{2.11}$$

$$\theta_i^d(t+1) = \theta_i^d(t) + V_i^d(t+1), \tag{2.12}$$

где $V_i^d(t)$ - текущая скорость, равная на первой итерации нулю.

Далее происходит перерасчет фитнес-функции, сортировка агентов и выполнение заданное количество итераций локального поиска из «прыгающих лягушек».

Локальный поиск заключается в замене худших векторов в мемплексе на новые, при этом фактическое разбиение на подгруппы не проводится, принадлежность к тому или иному мемплексу определяется по индексу агента. Чтобы не заменять постоянно один и тот же вектор, вводится счетчик замены f. Счетчик увеличивается на единицу каждый раз, когда происходит замена на новый вектор, исключая замену случайным образом, и уменьшается до единицы каждый раз, когда достигает значения, равного NA-1. Для создания нового вектора **new** на каждой локальной итерации t_l ($t_l \in [1, LIt]$) выбираются два агента из одного и того же мемплекса mem ($mem \in [0, NM]$): вектор с индексом mem записывается в **best**(t_l), вектор с индексом wr ($wr = N - f \times NM + mem$) записывается в **worst**(t_l). Новый агент **new** генерируется на основе следующего оператора:

$$\mathbf{new} = \text{rand}(0,1) \times const \times \left(\mathbf{best}(t_l) - \mathbf{worst}(t_l)\right) + \mathbf{worst}(t_l). \quad (2.13)$$

Для **new** оценивается значение фитнес-функции; если оно больше, чем фитнес-функция вектора $\boldsymbol{\theta}_{wr}(t_l)$, то агент **new** заменяет $\boldsymbol{\theta}_{wr}(t_l)$, а вектор скорости V_{wr} обнуляется. В противном случае **new** создается заново, но в **best**(t_l) записывается глобально лучший вектор $\boldsymbol{\theta}_0$. Если фитнес-функция **new** по-прежнему не превышает $fit(\boldsymbol{\theta}_{wr}(t_l))$, то на месте $\boldsymbol{\theta}_{wr}(t_l)$ генерируется новый вектор на основе $\boldsymbol{\theta}_0$ с некоторым отклонением. После истечения локальных итераций осуществляется перерасчет фитнесс-функции, сортировка и возврат к глобальному поиску. Выходом гибридного алгоритма является агент с максимальным значением фитнес-функции.

Псевдокод алгоритма, направленного на достижения максимума фитнес-функции, представлен ниже.

Вход: $\boldsymbol{\theta}_0$, *GIt*, *LIt*, *NM*, *NA*, G_0, α, ε, *const*.

Выход: $\boldsymbol{\theta}_0$.

Генерация популяции $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{NM \times NA-1}\}$.

Вычисление фитнес-функции $fit(\boldsymbol{\theta})$, сортировка популяции по убыванию фитнес-функции.

Инициализация счетчика глобальных итераций $t := 0$.

цикл пока ($t < GIt$):

Обновление гравитационной постоянной $G := G_0 \times \exp(-\alpha \times t / GIt)$.

Расчет вектора масс **M**:

$$mass_i := \frac{fit(\boldsymbol{\theta}_i) - fit(\boldsymbol{\theta}_{worst})}{fit(\boldsymbol{\theta}_{best}) - fit(\boldsymbol{\theta}_{worst})}, \; M_i := mass_i \Big/ \sum_{j=0}^{N-1} mass_j .$$

Вычисление вектора ускорения $\mathbf{a}_i$ для каждого агента:

$$\mathbf{a}_i := G \times \sum_{j=0, j \neq i, j \in K_{\text{best}}}^{N} \text{rand}(0,1) \times \frac{M_j \times (\boldsymbol{\theta}_j - \boldsymbol{\theta}_i)}{\left\| \boldsymbol{\theta}_j - \boldsymbol{\theta}_i \right\| + \varepsilon}.$$

Расчет скорости $\mathbf{V}_i := \text{rand}(0,1) \times \mathbf{V}_i + \mathbf{a}_i$.

Обновление векторов популяции $\boldsymbol{\theta}_i := \boldsymbol{\theta}_i + \mathbf{V}_i$.

Расчет фитнес-функции, сортировка популяции по убыванию фитнес-функции.

Локальный поиск.

$t := t + 1$.

Сортировка популяции по убыванию фитнес-функции.

конец цикла.

вывод $\boldsymbol{\theta}_0$.

конец алгоритма.

Далее представлено описание функции локального поиска, взятого из метаэвристики «прыгающие лягушки».

Функция Локальный поиск

mem := 0;

цикл пока (*mem* < *NM*)

f := 0.

цикл пока (*LIt* > 0):

Определение индекса худшего агента группы *wr* := $N - f \times NM + mem$.

Определение худшего агента группы **worst** := $\boldsymbol{\theta}_{wr}$.

Определение лучшего агента группы **best** := $\boldsymbol{\theta}_{mem}$.

Генерация нового вектора

$$\mathbf{new} = \text{rand} \times const \times (\mathbf{best} - \mathbf{worst}) + \mathbf{worst}\ .$$

если *fit*(**new**) > *fit*(**worst**):

worst := **new**.

f := *f* + 1.

иначе:

Генерация нового вектора $\mathbf{new} = \text{rand} \times const \times (\boldsymbol{\theta}_0 - \mathbf{worst}) + \mathbf{worst}$

если *fit*(**new**) > *fit*(**worst**):

worst := **new**.

f := *f* + 1.

иначе:

Случайная генерация вектора **new** на основе $\boldsymbol{\theta}_0$.

worst := **new**.

Обновление *fit*(**worst**).

если *f* > *NA* - 1:

f := 0.

LIt := *LIt* – 1.

конец цикла.

конец цикла.

конец функции.

При вычислении расстояния между векторами (для расчета ускорения) осуществляется проверка на преждевременную сходимость – если все расстояния равны нулю, алгоритм досрочно завершает работу и выдает текущее лучшее решение.

2.3 Алгоритм настройки весовых коэффициентов признаков

Отбор признаков является широко распространенным инструментом для преодоления проблемы переобучения инструментов классификации, поскольку помогает уменьшить сложность модели путем избавления от избыточных и шумовых признаков [143]. Но существует риск, что признаки, обладающие низкой информативностью для обучающих данных, окажутся важными при обработке новых данных, и их исключение приведет к потере полезной информации. Особенно это актуально для несбалансированных наборов данных, обладающих недостаточным для качественного обучения количеством экземпляров класса меньшинства. Для таких данных более подходящей альтернативой может служить введение весовых коэффициентов, отражающих степень важности признаков при расчете вывода модели. Малоинформативный с точки зрения обучения признак получит небольшой вес, однако не будет полностью удален из классификации, благодаря чему будет уменьшена вероятность потери полезной информации.

Использование такого подхода ведет к необходимости разработки эффективных алгоритмов настройки весовых коэффициентов. Так как настройка весов является задачей оптимизации, для её решения можно воспользоваться метаэвристиками, способными функционировать в непрерывном пространстве поиска. Многие метаэвристики создаются именно для работы с непрерывными входными величинами, поэтому еще одно достоинство взвешивания признаков вместо отбора – отсутствие необходимости разрабатывать бинарную версию алгоритма.

Далее предложен алгоритм настройки весов признаков, основанный на гибридном алгоритме из метаэвристик «гравитационный поиск» и «прыгающие лягушки». Идея алгоритма заключается во введении весовых коэффициентов признаков, которые позволят варьировать степень важности признака при формировании выхода классификатора [144].

Степень принадлежности экземпляра объекта k-му классу в таком случае будем определять по формуле:

$$\beta_k(\mathbf{x}_p) = \sum_{r_{jk}} \prod_{i=1}^{n} \left(w_i \times \mu_{T_{ij}}(x_p^i) \right), \tag{2.14}$$

где w_i – вес i-го признака, $w_i \in \mathbf{w} = (w_1, w_2, \ldots, w_n)$. Вес признака принадлежит промежутку от нуля до единицы включительно. При его равенстве нулю признак исключается из классификации, то есть не участвует в расчете степени принадлежности.

Популяция векторов весовых коэффициентов может быть сгенерирована различными способами. Одним из самых простых и быстрых способов является случайная генерация вещественных чисел в промежутке от нуля до единицы, никак не связанная с исследуемыми

данными. Другим подходом является создание популяции на основе количественной оценки взаимосвязи между признаком и выходной переменной. В роли оценочной меры можно использовать взаимную информацию. В таком случае вектор значений взаимной информации нормируется в пределах от нуля до единицы, затем для генерации остальных агентов популяции на имеющийся вектор накладывается случайное отклонение. Такой способ позволит получить отправную точку для алгоритма оптимизации, связанную с особенностями данных.

Созданная популяция подается на вход алгоритму оптимизации, задачей которого является поиск вектора весов признаков, позволяющего достигнуть оптимальное значение фитнес-функции. В роли инструмента для оптимизации используется гибридный алгоритм из метаэвристик «гравитационный поиск» и «прыгающие лягушки», описанный в предыдущем параграфе. При любом изменении вектора весов классификатор должен быть перестроен, а фитнес-функция должна быть рассчитана заново.

Так как предлагаемый алгоритм оптимизации является непрерывным, в ходе вычислений могут быть получены значения, очень близкие, но не равные нулю. Поэтому целесообразно ввести пороговое значение ω, ниже которого элементы вектора весов будут принудительно приравнены к нулю:

$$\text{Если } w_i < \omega, \text{ то } w_i = 0. \tag{2.15}$$

Кроме того, с точки зрения последующей интерпретации нечеткого классификатора важно, чтобы хотя бы один из признаков имел вес, равный единице. Поэтому после каждого обновления вектора стоит проводить нормализацию вектора. В данной работе была использована «минимаксная» нормализация, при которой нормализованное значение веса признака w_i' рассчитывается следующим образом:

$$w_i' = \frac{w_i - w_{\min}}{w_{\max} - w_{\min}}, \tag{2.16}$$

где w_i – значение веса до нормализации, $w_{\min}$ – минимальный вес признака в векторе, $w_{\max}$ – максимальный вес признака в векторе. Таким образом, как минимум один признак будет обладать весом, равным единице. Однако верно и обратное, по крайней мере один признак будет иметь нулевой вес. Впрочем, случаи, когда в наборе данных не присутствует хотя бы один лишний признак, очень редки.

Еще одним изменением в гибридный метаэвристический алгоритм, описанный в параграфе 2.2, является введение динамического обновления коэффициента *const*, применяющегося для генерации нового вектора в локальном поиске. Перед началом работы алгоритма в *const* присваивается начальное значение, установленное пользователем – $const_0$. В конце каждой глобальной итерации *const* уменьшается на 0,1. Когда *const* окажется равным нулю,

в него заново записывается значение $const_0$. Такое динамическое обновление позволит внести большее разнообразие в локальный поиск.

Алгоритм настройки весовых коэффициентов признаков в виде псевдокода приведен далее. Оптимизация направлена на достижение максимума фитнес-функции. Операция проверки веса на близость к нулю и нормализация значений элементов векторов включены в функцию «Нормализация векторов популяции».

Вход: *GIt*, *LIt*, *NM*, *NA*, G_0, α, ε, $const_0$, ω.

Выход: $\mathbf{w}_0$.

Генерация популяции $\mathbf{W} = \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{NM \times NA-1}\}$.

Вычисление фитнес-функции $fit(\mathbf{w})$.

Присваивание $const := const_0$.

Инициализация счетчика глобальных итераций $t := 0$.

цикл пока ($t < GIt$):

Обновление гравитационной постоянной $G := G_0 \times \exp(-\alpha \times t / GIt)$.

Расчет вектора масс **M**:

$$mass_i := \frac{fit(\mathbf{w}_i) - fit(\mathbf{w}_{worst})}{fit(\mathbf{w}_{best}) - fit(\mathbf{w}_{worst})},\ M_i := mass_i \Big/ \sum_{j=0}^{N-1} mass_j .$$

Вычисление вектора ускорения $\mathbf{a}_i$ для каждого агента:

$$\mathbf{a}_i := G \times \sum_{j=0, j \neq i}^{N} \text{rand}(0,1) \times \frac{M_j \times (\mathbf{w}_j - \mathbf{w}_i)}{\left\| \mathbf{w}_j - \mathbf{w}_i \right\| + \varepsilon}.$$

Расчет скорости $\mathbf{V}_i := \text{rand}(0,1) \times \mathbf{V}_i + \mathbf{a}_i$.

Обновление векторов популяции $\mathbf{w}_i := \mathbf{w}_i + \mathbf{V}_i$.

Нормализация векторов популяции.

Расчет фитнес-функции, сортировка популяции по убыванию фитнес-функции.

Локальный поиск.

$t := t + 1$.

Сортировка популяции по убыванию фитнес-функции.

Обновление коэффициента *const*.

конец цикла.

вывод $\mathbf{w}_0$.

Далее представлено описание функции локального поиска.

Функция Локальный поиск

$mem := 0$;

цикл пока ($mem < NM$)

$f := 0$.

цикл пока ($LIt > 0$):

Определение индекса худшего агента группы $wr := N - f \times NM + mem$.

Определение худшего агента группы **worst** := $\mathbf{w}_{wr}$.

Определение лучшего агента группы **best** := $\mathbf{w}_{mem}$.

Генерация нового вектора

$$\mathbf{new} = \text{rand} \times const \times (\mathbf{best} - \mathbf{worst}) + \mathbf{worst}.$$

Нормализация вектора **new**.

если fit(**new**) > fit(**worst**):

worst := **new**.

$f := f + 1$.

иначе:

Генерация нового вектора $\mathbf{new} = \text{rand} \times const \times (\mathbf{w}_0 - \mathbf{worst}) + \mathbf{worst}$.

Нормализация вектора **new**.

если fit(**new**) > fit(**worst**):

worst := **new**.

$f := f + 1$.

иначе:

Случайная генерация вектора **new** на основе $\mathbf{w}_0$.

Нормализация вектора **new**.

worst := **new**.

Обновление fit(**worst**).

если $f > NA - 1$:

$f := 0$.

$LIt := LIt - 1$.

конец цикла.

конец цикла.

Конец функции.

Еще одним достоинством применения процедуры взвешивания признаков является получение новой информации о приоритете признаков при формировании выхода системы. Важность той или иной переменной не всегда очевидна для владельцев данных. Веса позволят не просто сделать выводы в категориях «признак важен» и «признак не важен», как это возможно сделать при классическом отборе, но и выяснить, какие переменные более и менее важны.

2.4 Выводы

1. Разработан алгоритм формирования структуры нечеткого классификатора на основе метаэвристики «прыгающие лягушки», осуществляющий итеративное расширение базы правил путем генерации нового правила для класса с наименьшей долей правильной классификации и дальнейшей настройкой параметров термов этого правила. В роли генератора первичной структуры может выступать любой инструмент. Для настройки параметров термов предложена метаэвристика «прыгающие лягушки». В качестве фитнес-функции предложена функция, составленная из двух метрик: общей точности и средней геометрической точности для улучшения распознавания редких классов. Метрики связаны коэффициентом, который задается пользователем в зависимости от требований к классификатору.

2. Разработан гибридный алгоритм настройки параметров нечеткого классификатора, объединяющий сильные стороны двух метаэвристик: глобальный поиск для осуществления крупных изменений в оптимизируемом векторе из метаэвристики «гравитационный поиск» и локальный поиск для небольших изменений векторов из «прыгающих лягушек». Предложенный алгоритм при использовании фитнес-функции, адекватной для оценки моделей при наличии несбалансированности в данных, позволит улучшать качество распознавания наименьших классов.

3. Разработан алгоритм настройки весовых коэффициентов признаков и способ формирования нечеткого вывода при наличии таких весов. Веса представляют собой вещественные числа, распределенные в пределах от нуля до единицы, включая границы. Для поиска оптимального вектора весов признаков в нечетком классификаторе применяется гибридный алгоритм, состоящий из метаэвристик «гравитационный поиск» и «прыгающие лягушки». Предложена формула для расчета нечеткого вывода при наличии весов и два способа создания популяции для алгоритма оптимизации: случайная генерация и генерация на основе взаимной информации между признаками и классами. При применении в качестве фитнес-функции метрики, способной учитывать дисбаланс классов, алгоритм позволит улучшить качество классификации положительных классов. Значения весов признаков позволят получить новые знания о предметной области.

Глава 3. Исследование эффективности разработанных алгоритмов

Глава посвящена подтверждению эффективности разработанных алгоритмов и включает параграфы следующего содержания.

В первом параграфе описаны наборы данных, используемые в экспериментах.

Второй параграф посвящен задаче выбора оптимальной фитнес-функции при оптимизации нечетких классификаторов несбалансированных данных. Показано, что выбор метрики определяет направление обучения. Так, средняя геометрическая точность позволяет сфокусироваться в первую очередь на наименьшем классе, в то время как общая точность нацелена на улучшение распознавания экземпляров наибольшего класса.

В третьем параграфе приведены результаты создания нечетких классификаторов несбалансированных данных при формировании структуры комбинацией двух алгоритмов: алгоритма экстремальных значений признаков классов и описанного в разделе 2.1 алгоритма добавления правил метаэвристикой «прыгающие лягушки». Показано, что после расширения базы правил качество классификации улучшается на большинстве исследованных наборов данных. Проводится сравнение результатов до и после дополнения баз правил по трем критериям: средней геометрической точности, точности положительного класса и точности отрицательного класса. Полученные на лучших базах правил значения средней геометрической точности сравниваются с результатами аналогичных алгоритмов построения нечетких классификаторов.

Четвертый параграф содержит итоги оптимизации параметров термов нечеткого классификатора гибридным алгоритмом из метаэвристик «гравитационный поиск» и «прыгающие лягушки». Полученные значения метрик качества сравниваются с результатами исходных метаэвристик, функционирующих по отдельности, а также с алгоритмами построения нечетких классификаторов, не применяющих оптимизацию параметров функций принадлежности.

Пятый параграф посвящен исследованию эффективности алгоритма настройки весовых коэффициентов признаков. Так как ранее настройка весов не применялась в нечетких классификаторах рассматриваемого типа, требовалось провести не только проверку работоспособности предложенного алгоритма, но и изучить, как наличие весов влияет на другой инструмент оптимизации – настройку параметров термов.

Пятый параграф включает следующие пункты:

- результаты двух различных схем постановки эксперимента;
- сравнение результатов экспериментов с аналогичными алгоритмами построения нечетких классификаторов;

– сравнение результатов этапов настройки весовых коэффициентов признаков и оптимизации параметров термов, а также их комбинации;
– сравнение результатов настройки весов признаков в зависимости от способа генерации популяции весов;
– подтверждение эффективности применения гибридного алгоритма из метаэвристик «прыгающие лягушки» и «гравитационный поиск» в задаче настройки весов при сравнении со случайным поиском.

В шестом параграфе представлены выводы по проведенным экспериментам и сравнениям.

3.1 Описание экспериментальных данных

Для экспериментов были использованы наборы данных из открытого испанского репозитория «Knowledge Extraction based on Evolutionary Learning» [145]. Так как цель экспериментов заключалась в подтверждении эффективности разработанных алгоритмов, были использованы данные из различных предметных областей: медицины (wisconsin, pima, haberman, newthyroid, ecoli), биологии (abalone), криминалистики (glass), распознавания изображений (vehicle, page-blocks, segment) и других. Выбор наборов ограничивался следующими условиями: в экземплярах отсутствуют пропуски и номинальные признаки, нечеткий классификатор не демонстрирует на этих данных среднюю геометрическую точность, равную единице, сразу после генерации структуры алгоритмом экстремальных значений признаков классов, данные обладают дисбалансом, но имеют не менее пяти экземпляров каждого класса. Большинство наборов, использованных в эксперименте, сформировано путем разбиения набора исходных данных владельцами репозитория на вариации, в которых один класс противопоставляется всем остальным классам (glass0, glass1, vehicle1, vehicle2 и т.д.). Полное описание наборов данных хранится в репозитории [145], основные характеристики наборов приведены в таблице 3.1. Здесь $inst_{all}$ – общее количество экземпляров в наборе, *inst+* – количество экземпляров положительного класса, *inst-* – число экземпляров отрицательного класса. Главной характеристикой набора является коэффициент дисбаланса *IR* – отношение числа экземпляров наибольшего класса к числу экземпляров наименьшего: *IR* = *inst-*/*inst+*. Все наборы содержат только два класса.

Таблица 3.1 – Описание наборов данных для тестирования нечеткого классификатора

№	Наборы данных	Аббревиатура	Признаки	$inst_{all}$	*inst+*	*inst-*	*IR*
1	glass1	gl1	9	214	76	138	1,82
2	ecoli0vs1	ecl0/1	7	220	77	143	1,86
3	wisconsin	wis	9	683	239	444	1,86
4	pima	pm	8	768	268	500	1,87
5	glass0	gl0	9	214	70	144	2,06
6	yeast1	yst1	8	1484	429	1055	2,46

Продолжение таблицы 3.1

№	Наборы данных	Аббревиатура	Признаки	$inst_{all}$	*inst+*	*inst-*	*IR*
7	haberman	hbr	3	306	81	225	2,78
8	vehicle2	vhc2	18	846	218	628	2,88
9	vehicle1	vhc1	18	846	217	629	2,90
10	vehicle3	vhc3	18	846	212	634	2,99
11	glass0123vs456	gl0123/456	9	214	51	163	3,20
12	vehicle0	vhc0	18	846	199	647	3,25
13	ecoli1	ecl1	7	336	77	259	3,36
14	newthyroid2	nwth2	5	215	35	180	5,14
15	newthyroid1	nwth1	5	215	35	180	5,14
16	ecoli2	ecl2	7	336	52	284	5,46
17	segment0	sgm0	19	2308	329	1979	6,02
18	glass6	gl6	9	214	29	185	6,38
19	yeast3	yst3	8	1484	163	1321	8,10
20	ecoli3	ecl3	7	336	35	301	8,60
21	page-blocks0	pb0	10	5472	559	4913	8,79
22	yeast2vs4	yst2/4	8	514	51	463	9,08
23	yeast05679vs4	yst05679/4	8	528	51	477	9,35
24	vowel0	vwl0	13	988	90	898	9,98
25	glass2	gl2	9	214	17	197	11,59
26	glass4	gl4	9	214	13	201	15,46
27	ecoli4	ecl4	7	336	20	316	15,80
28	page-blocks1-3vs4	pb1-3/4	10	472	28	444	15,86
29	abalone9-18	ab9/18	7/8	731	42	689	16,40
30	yeast1458vs7	yst1458/7	8	693	30	663	22,10
31	yeast2vs8	yst2/8	8	482	20	462	23,10
32	yeast4	yst4	8	1484	51	1433	28,10
33	yeast1289vs7	yst1289/7	8	947	30	917	30,57
34	yeast5	yst5	8	1484	44	1440	32,73
35	ecoli0137vs26	ecl0137/26	7	281	7	274	39,14
36	yeast6	yst6	8	1484	35	1449	41,40

В наборе данных abalone9-18 удален признак «Пол», являющийся номинальным.

Все выборки данных перед началом работы алгоритмов проходили через «минимаксную» нормализацию в автоматическом режиме.

3.2 Анализ метрик качества классификации при наличии дисбаланса в данных

Алгоритмы оптимизации проверяют качество найденных решений путем вычисления фитнес-функции. Выбор адекватной меры оценки качества классификации при дисбалансе данных имеет первостепенную важность. Для определения адекватной фитнес-функции были исследованы восемь метрик, для расчета которых достаточно иметь два вектора – действительные метки классов из таблицы наблюдения и метки классов, определенные классификатором. Описание исследуемых метрик приведено далее.

Общая точность (*Acc*.) – это классическая мера оценки качества моделей классификации, рассчитывающаяся как отношение количества правильно распознанных классификатором

экземпляров всех классов по отношению к общему числу экземпляров (формула 2.5). Общая точность не учитывает дисбаланс в классах и не признается адекватной мерой оценки при наличии существенной разницы в количестве экземпляров разных классов [1].

Средняя геометрическая точность (*GM*) для *m* классов определяется как корень *m*-ой степени из произведения доли правильной классификации каждого класса:

$$GM = \sqrt[m]{\prod_{k=1}^{m} class_k} \,. \tag{3.2}$$

где $class_k = inst_k^* / inst_k$.

Недостатком средней геометрической точности является то, что при большом дисбалансе метрика практически не учитывает экземпляры отрицательного класса, что затрудняет получение сбалансированной точности. Например, в случае двух классов, один правильно распознанный экземпляр положительного класса будет более важен с точки зрения этой метрики, чем несколько десятков образцов отрицательного класса.

Целесообразно объединить две предыдущие метрики через коэффициент, который будет регулировать преимущество одной метрики над другой:

$$score = \gamma \times GM + (1-\gamma) \times Acc \,, \tag{3.3}$$

где γ – коэффициент приоритета, принадлежащий промежутку [0;1]. В эксперименте использованы три вариации этой меры оценки с коэффициентом γ, равным одному из трех значений: {0,75; 0,5; 0,25} (обозначены далее как 0,75*GM*0,25*A*, 0,5*GM*0,5*A* и 0,25*GM*0,75*A*, соответственно).

Еще одной распространенной оценкой качества моделей при классификации несбалансированных данных является *F*1-мера (*F*1):

$$F1 = tp / (tp + 0{,}5(fp + fn)) \,, \tag{3.4}$$

где *tp* – число верно определенных экземпляров положительного класса, *fp* – количество экземпляров, неверно отнесенных классификатором к положительному классу, *fn* – число неверно отнесенных к отрицательному классу экземпляров.

Мера качества, названная сбалансированной точностью (*balanced*), вычисляется как среднее арифметическое по точности каждого класса. Для данных с двумя классами сбалансированная точность может быть представлена следующим образом:

$$balanced = (class_1 + class_2) / 2 \,, \tag{3.5}$$

Последняя рассмотренная метрика – коэффициент Жаккара – является коэффициентом сходства и обычно применяется в задачах кластеризации. Для классификации коэффициент Жаккара может быть рассчитан по формуле:

$$Jaccard = tp/(tp + fp + fn), \tag{3.6}$$

Все метрики принадлежат промежутку от нуля, что соответствует худшему значению, до единицы – лучшего значения.

В таблице 3.2 приведен процент правильной классификации положительного класса, получаемый после оптимизации параметров нечетких классификаторов исходным алгоритмом прыгающих лягушек, описанным в работе [89]. Метаэвристика запускалась десять раз на пятикратной кросс-валидации, далее значения точности были усреднены. В эксперименте использованы наборы данных (НД) с дисбалансом, превышающим девять (с 22 по 36 номер из таблицы 3.1), чтобы избежать возможного смещения результатов из-за влияния данных с малым дисбалансом. Были установлены следующие параметры алгоритма прыгающих лягушек: 50 агентов, 20 глобальных итераций, 25 локальных итераций, 5 мемплексов, коэффициент обновления *const* = 1,2.

Таблица 3.2 – Процент правильной классификации положительного класса после оптимизации параметров «прыгающими лягушками» с различными фитнесс-функциями

НД	*Acc*.	*GM*	0,75*GM*0,25*A*	0,5*GM0,5A*	0,25*GM0,75A*	*F*1	*balanced*	*Jaccard*
22	63,4 ± 3,9	**78,1** ± 2,9	75,4 ± 2,4	71,5 ± 3,0	69,4 ± 2,3	64,8 ± 2,7	77,2 ± 2,9	62,2 ± 2,5
23	33,2 ± 3,4	**68,4** ± 3,1	65,4 ± 2,2	60,3 ± 4,3	55,2 ± 3,9	31,1 ± 3,0	67,8 ± 3,5	30,9 ± 5,7
24	63,7 ± 3,0	**94,4** ± 2,2	91,8 ± 1,6	86,6 ± 3,2	73,8 ± 3,9	64,3 ± 4,3	93,3 ± 1,1	64,0 ± 3,7
25	1,8 ± 2,6	63,0 ± 6,1	58,5 ± 5,5	48,2 ± 12,5	26,8 ± 9,9	3,5 ± 2,8	**64,8** ± 6,4	0,5 ± 0,9
26	79,0 ± 4,4	84,0 ± 2,4	83,5 ± 2,8	82,5 ± 3,5	78,5 ± 4,1	77,0 ± 3,0	**85,5** ± 1,8	73,5 ± 4,1
27	37,7 ± 15,2	71,0 ± 6,1	74,0 ± 8,5	71,3 ± 6,9	70,7 ± 6,3	45,7 ± 12,3	**72,3** ± 6,5	39,7 ± 9,0
28	88,2 ± 3,8	88,9 ± 3,2	**92,0** ± 3,5	87,6 ± 2,9	90,6 ± 1,8	82,9 ± 6,5	89,7 ± 2,5	83,2 ± 2,7
29	27,4 ± 2,3	**67,1** ± 3,6	60,0 ± 4,6	55,5 ± 3,2	46,1 ± 3,9	25,8 ± 3,2	65,1 ± 3,3	23,1 ± 3,4
30	1,0 ± 1,4	49,7 ± 3,7	38,3 ± 5,7	29,0 ± 5,7	15,0 ± 4,0	0,7 ± 1,1	**50,7** ± 7,5	2,3 ± 2,3
31	50,5 ± 1,8	**58,0** ± 4,0	54,5 ± 3,8	54,0 ± 5,0	53,5 ± 3,5	49,0 ± 2,4	55,0 ± 3,0	51,5 ± 2,8
32	5,9 ± 2,0	71,6 ± 2,4	70,6 ± 3,5	66,6 ± 2,1	57,0 ± 4,5	8,9 ± 2,5	**73,1** ± 2,4	8,2 ± 3,8
33	1,3 ± 1,6	55,7 ± 6,1	49,0 ± 5,0	29,7 ± 4,4	16,3 ± 3,0	3,7 ± 3,2	**57,7** ± 4,8	3,0 ± 2,4
34	57,5 ± 4,0	88,5 ± 1,1	88,7 ± 2,3	87,1 ± 2,5	82,5 ± 2,4	54,5 ± 5,3	**89,8** ± 1,4	53,8 ± 7,9
35	59,0 ± 9,2	**69,0** ± 7,2	67,0 ± 12,2	63,0 ± 9,8	55,0 ± 14,0	61,0 ± 10,8	62,0 ± 10,4	39,0 ± 17,0
36	36,6 ± 5,8	**80,3** ± 1,5	79,7 ± 1,5	75,1 ± 2,2	64,3 ± 2,9	32,9 ± 6,3	78,6 ± 4,0	29,7 ± 6,5
Ср.	40,4 ± 4,3	**72,5** ± 3,7	69,9 ± 4,3	64,5 ± 4,7	57,0 ± 4,7	40,4 ± 4,6	72,2 ± 4,1	37,6 ± 5,0

В таблице 3.3 продемонстрирована точность отрицательного класса, полученная на тех же классификаторах.

Таблица 3.3 – Процент правильной классификации отрицательного класса после оптимизации параметров «прыгающими лягушками» с различными фитнесс-функциями

НД	*Acc*.	*GM*	0,75*GM*0,25*A*	0,5*GM0,5A*	0,25*GM0,75A*	*F*1	*balanced*	*Jaccard*
22	98,1 ± 0,2	92,9 ± 0,8	94,4 ± 1,4	95,9 ± 1,0	97,2 ± 0,3	98,1 ± 0,2	91,7 ± 1,4	**98,3** ± 0,5
23	**98,0** ± 0,4	84,7 ± 1,1	88,0 ± 0,6	91,1 ± 1,0	93,9 ± 1,0	**98,0** ± 0,7	86,9 ± 1,5	97,9 ± 0,7
24	98,6 ± 0,4	91,5 ± 1,0	92,1 ± 0,7	94,2 ± 0,7	97,1 ± 0,5	**98,8** ± 0,4	91,1 ± 0,9	**98,8** ± 0,4
25	98,0 ± 1,0	73,9 ± 3,2	79,6 ± 2,7	84,1 ± 2,7	92,0 ± 2,2	98,4 ± 0,7	73,2 ± 2,6	**99,1** ± 0,7
26	**99,3** ± 0,3	97,6 ± 0,6	97,9 ± 0,3	97,8 ± 0,3	98,0 ± 0,4	99,2 ± 0,2	97,6 ± 0,5	99,0 ± 0,3

Продолжение таблицы 3.3

НД	*Acc*.	*GM*	0,75*GM*0,25*A*	0,5*GM0,5A*	0,25*GM0,75A*	*F*1	*balanced*	*Jaccard*
27	97,8 ± 0,9	94,9 ± 0,9	96,2 ± 1,2	97,0 ± 0,7	96,4 ± 1,1	**98,3** ± 0,5	95,3 ± 1,7	98,0 ± 0,9
28	**99,0** ± 0,4	97,8 ± 0,5	98,3 ± 0,6	98,1 ± 0,5	98,6 ± 0,3	98,7 ± 0,5	97,7 ± 0,6	98,9 ± 0,3
29	99,5 ± 0,3	85,8 ± 1,5	88,6 ± 1,2	94,2 ± 0,6	96,8 ± 0,5	99,5 ± 0,2	86,6 ± 1,4	**99,7** ± 0,1
30	99,6 ± 0,3	72,9 ± 1,9	81,5 ± 1,6	88,3 ± 1,3	95,9 ± 1,2	99,7 ± 0,3	71,7 ± 2,1	**99,8** ± 0,2
31	**99,7** ± 0,1	91,5 ± 2,2	97,2 ± 1,3	98,8 ± 0,8	99,3 ± 0,3	99,6 ± 0,2	93,1 ± 1,6	**99,7** ± 0,1
32	99,5 ± 0,1	87,3 ± 0,9	89,5 ± 1,1	91,3 ± 0,3	94,6 ± 0,5	**99,6** ± 0,1	87,3 ± 1,1	**99,6** ± 0,3
33	**99,6** ± 0,1	76,6 ± 1,6	81,4 ± 2,0	91,0 ± 1,8	96,9 ± 0,5	**99,6** ± 0,2	74,0 ± 3,4	**99,6** ± 0,2
34	98,9 ± 0,1	96,8 ± 0,2	96,8 ± 0,3	97,0 ± 0,3	97,4 ± 0,1	**99,0** ± 0,2	96,7 ± 0,2	98,9 ± 0,1
35	99,1 ± 0,2	90,4 ± 1,7	89,7 ± 4,8	93,3 ± 2,4	97,8 ± 1,0	99,1 ± 0,2	82,8 ± 4,1	**99,2** ± 0,2
36	99,4 ± 0,1	93,0 ± 0,8	94,0 ± 0,9	95,5 ± 0,6	97,3 ± 0,2	**99,6** ± 0,1	93,3 ± 0,7	99,5 ± 0,1
Ср.	98,9 ± 0,3	88,5 ± 1,3	91,0 ± 1,4	93,9 ± 1,0	96,6 ± 0,7	99,0 ± 0,3	87,9 ± 1,6	**99,1** ± 0,3

Статистическое сравнение критерием Фридмана показало, что в обоих случаях существует значимое различие в полученных результатах (асимптотическая значимость меньше 0,001). Средние ранги, вычисленные в процессе сравнения, собраны в таблице 3.4 ($class_1$ – положительный класс, $class_2$ – отрицательный класс). Чем больше ранг, тем лучше фитнесс-функция справляется с определением класса. Для определения лучшей метрики необходимо рассчитать расстояние от метрики до идеальной точки, оба ранга которой равняются максимальному значению (восьми). Расстояние также приведено в таблице 3.4.

Таблица 3.4 – Средние ранги по критерию Фридмана и расстояние до идеальной точки

Метрика	Ранг для $class_1$	Ранг для $class_2$	Расстояние до идеальной точки
Acc.	2,33	6,6	5,84
GM	7,13	1,67	6,39
0,75*GM*0,25*A*	6,47	3,07	5,16
0,5*GM0,5A*	5	3,93	5,06
0,25*GM0,75A*	4	4,93	5,04
*F*1	2,33	7,03	5,75
balanced	7,13	1,4	6,66
Jaccard	1,6	7,37	6,43

Ниже представлен рисунок 3.1, демонстрирующий положение метрик на фронте Парето согласно вычисленным средним рангам по критерию Фридмана. Идеальная точка находится в правом верхнем углу (зеленый круг). Точки, попавшие во фронт Парето, отмечены красным.

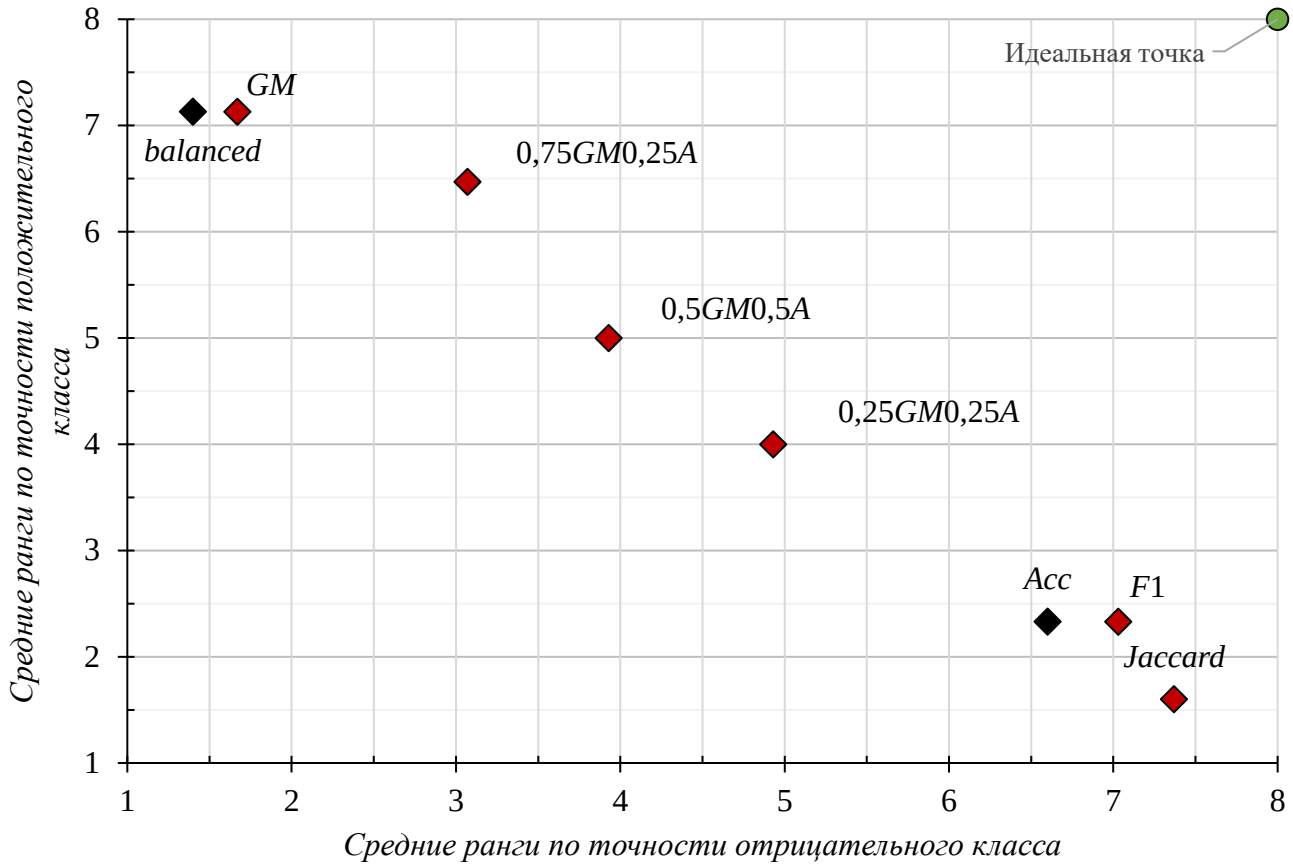

Рисунок 3.1 – Фронт Парето по средним рангам критерия Фридмана при сравнении точности положительного и отрицательного классов, полученной различными метриками

Значения рангов показывают, что фитнесс-функция, соответствующая средней геометрической точности, позволила получить после настройки параметров термов «прыгающими лягушками» наибольшую точность положительного класса по сравнению с другими метриками. Лучшую точность отрицательного класса продемонстрировал классификатор, настроенный с помощью критерия Жаккара.

Определить метрику, показавшую максимальную эффективность сразу по двум критериям, можно путем вычисления расстояния между идеальной точкой и положением метрики на рисунке 3.1. Наименьшее расстояние между идеальной точкой продемонстрировала авторская метрика 0,25*GM0,75A*; другие версии этой метрики также оказались поблизости. Это значит, что при необходимости получить сбалансированную точность лучше всего использовать метрики, комбинирующие среднюю геометрическую и общую точности. Выбирать коэффициент γ необходимо, исходя из требований к модели и величины коэффициента дисбаланса обрабатываемых данных.

3.3 Проверка эффективности алгоритма формирования структуры нечеткого классификатора несбалансированных данных на основе итерационного добавления правил метаэвристикой «прыгающие лягушки»

3.3.1 Описание эксперимента. Целью эксперимента является подтверждение следующих утверждений: предложенный алгоритм итерационного добавления правил метаэвристикой «прыгающие лягушки», описанный в параграфе 2.1, способен улучшать среднюю геометрическую точность первичных баз правил на несбалансированных данных с двумя классами; предложенный инструмент в комбинации с алгоритмом генерации структуры на основе экстремальных значений признаков классов способен демонстрировать среднюю геометрическую точность на несбалансированных данных с двумя классами, сопоставимую с аналогичными методами построения нечетких классификаторов.

Алгоритм экстремальных значений признаков классов выбран в качестве инструмента создания первичной базы правил нечеткого классификатора благодаря следующим достоинствам: гарантированному созданию правил для всех имеющихся в обучении классов и получению базы правил минимального объема, равного количеству классов [67]. Чем меньше объем первичной базы правил, тем меньше времени будут занимать вычисления алгоритма, формирующего новые правила.

Эксперимент проводился по схеме пятикратной кросс-валидации. На всех выборках комбинация алгоритма на основе экстремальных значений классов и алгоритма прыгающих лягушек запускалась по пять раз для каждого количества добавляемых правил. Были зафиксированы результаты построения нечетких классификаторов при добавлении одного, двух, пяти и семи правил. В качестве термов использованы функции Гаусса. Параметры рассматриваемого алгоритма (параграф 2.1) были следующими: 15 глобальных итераций, 25 локальных итераций, 5 мемплексов, 5 векторов в мемплексе. Константа для генерации новых параметров термов *const* для расчета формулы 2.6 равнялась 1,2. Все параметры были подобраны эмпирически как наиболее универсальные для рассматриваемых наборов данных. Коэффициент γ в фитнес-функции равнялся 0,5 для равного приоритета между общей и средней геометрической точностью (формула 2.4).

3.3.2 Результаты эксперимента. В таблице 3.5 представлены результаты эксперимента на тестовых выборках. Столбец «АЭПК» демонстрирует среднюю геометрическую точность, полученную на классификаторах, построенных до добавления правил алгоритмом экстремумов признаков классов. В этом случае количество правил равнялось двум. Далее приведены результаты классификации после добавления одного, двух, пяти и семи правил «прыгающими лягушками» (столбец «АЭПК+ПЛ»). Предпоследняя строка демонстрирует усредненные

значения по всем наборам данных, последняя – разность между исходной точностью и точностью после расширения базы правил.

Таблица 3.5 – Средняя геометрическая точность нечеткого классификатора до и после добавления правил алгоритмом прыгающих лягушек

Данные		АЭПК	АЭПК+ПЛ			
		2 правила	3 правила	4 правила	7 правил	9 правил
1	glass1	40,5	61,0 ± 3,8	62,8 ± 6,0	**68,9** ± 5,1	68,5 ± 5,2
2	ecoli0vs1	88,8	95,1 ± 1,8	96,4 ± 1,2	**95,8** ± 1,9	95,4 ± 2,4
3	wisconsin	73,4	93, 5 ± 1,0	**95,1** ± 1,1	95,0 ± 1,6	94,5 ± 1,0
4	pima	55,6	68,1 ± 1,9	70,6 ± 1,6	71,6 ± 1,8	**72,2** ± 2,3
5	glass0	60,1	74,4 ± 3,6	73,9 ± 3,4	75,9 ± 4,4	**78,4** ± 3,8
6	yeast1	39,6	65,2 ± 2,6	66,2 ± 1,3	69,0 ± 1,2	**69,9** ± 1,3
7	haberman	44,3	46,1 ± 3,4	50,5 ± 3,8	55,0 ± 4,4	**56,5** ± 4,8
8	vehicle2	40,0	59,2 ± 2,7	67,1 ± 4,3	80,2 ± 3,0	**82,8** ± 4,0
9	vehicle1	41,9	50,1 ± 2,7	61,8 ± 3,4	66,1 ± 2,9	**68,4** ± 1,9
10	vehicle3	39,1	43,3 ± 2,4	52,9 ± 3,1	64,6 ± 2,4	**66,4** ± 2,6
11	glass0123vs456	87,6	**93,2** ± 1,5	93,1 ± 2,1	90,8 ± 3,1	89,7 ± 2,7
12	vehicle0	55,5	71,8 ± 4,2	75,5 ± 3,3	81,6 ± 1,8	**86,4** ± 2,2
13	ecoli1	80,8	86,8 ± 1,8	**88,4** ± 1,6	88,1 ± 2,0	88,3 ± 2,2
14	newthyroid2	**99,2**	97,9 ± 0,5	98,2 ± 0,1	96,8 ± 2,5	96,8 ± 2,1
15	newthyroid1	**99,2**	97,2 ± 1,6	96,1 ± 2,0	94,9 ± 3,6	94,1 ± 3,9
16	ecoli2	34,2	81,2 ± 4,6	84,9 ± 4,1	89,3 ± 2,7	**90,5** ± 2,8
17	segment0	88,1	92,6 ± 1,1	94,0 ± 1,0	89,0 ± 1,0	**97,6** ± 0,8
18	glass6	22,8	83,7 ± 5,1	86,5 ± 4,9	86,5 ± 4,4	**88,3** ± 4,7
19	yeast3	85,5	88,5 ± 1,4	90,1 ± 0,8	**90,4** ± 1,8	90,0 ± 1,3
20	ecoli3	50,8	84,0 ± 3,1	**87,1** ± 3,0	85,5 ± 3,3	81,7 ± 4,7
21	page-blocks0	63,6	74,1 ± 1,8	81,9 ± 1,6	79,8 ± 2,3	**88,2** ± 0,9
22	yeast2vs4	67,3	82,9 ± 3,9	84,0 ± 3,2	**86,7** ± 4,9	85,5 ± 3,8
23	yeast05679vs4	61,9	69,2 ± 3,2	74,2 ± 3,6	**74,4** ± 3,6	73,7 ± 3,9
24	vowel0	83,9	84,4 ± 1,0	86,6 ± 1,2	88,9 ± 3,1	**92,1** ± 3,4
25	glass2	10,8	57,8 ± 10,0	**61,7** ± 10,6	61,1 ± 8,7	53,4 ± 14,2
26	glass4	23,1	76,1 ± 8,8	74,8 ± 14,4	**79,4** ± 10,5	75,9 ± 12,5
27	ecoli4	68,7	88,0 ± 4,7	89,4 ± 3,1	**89,9** ± 3,1	84,3 ± 3,0
28	page-bl.1-3vs4	75,1	85,7 ± 3,7	89,2 ± 3,9	**90,1** ± 5,1	86,8 ± 5,1
29	abalone9-18	58,7	67,4 ± 3,3	71,4 ± 3,7	**75,0** ± 2,3	72,2 ± 7,3
30	yeast1458vs7	45,8	**59,6** ± 3,2	56,6 ± 6,2	54,7 ± 6,7	52,7 ± 9,9
31	yeast2vs8	68,8	68,5 ± 3,3	69,7 ± 4,0	**70,2** ± 4,5	66,2 ± 8,0
32	yeast4	65,7	70,2 ± 2,5	75,7 ± 2,7	**79,9** ± 2,1	77,3 ± 3,9
33	yeast1289vs7	54,1	62,4 ± 3,1	63,1 ± 4,3	61,3 ± 6,0	**63,3** ± 7,2
34	yeast5	72,9	92,5 ± 1,6	91,8 ± 2,0	**93,5** ± 2,1	92,5 ± 2,9
35	ecoli0137vs26	0,0	**71,3** ± 3,8	68,5 ± 8,8	61,2 ± 16,3	65,8 ± 9,5
36	yeast6	51,2	82,3 ± 3,5	**84,0** ± 2,5	83,4 ± 2,1	83,0 ± 4,0
Среднее		58,3	75,7 ± 3,1	78,2 ± 3,6	79,6 ± 3,8	**79,7** ± 4,3
Процент улучшения		-	17,4	19,9	21,3	**21,4**

Полужирным начертанием выделены наибольшие значения средней геометрической точности по каждому набору данных. В среднем лучший результат был получен при добавлении семи правил, однако и разброс в результатах выше именно в этом случае.

Для многих задач, связанных с классификацией несбалансированных данных, наибольшей важностью обладает процент правильной классификации наименьшего класса. В таблице 3.6

представлена точность положительного класса на исходной структуре нечеткого классификатора (столбец АЭПК) и после добавления правил метаэвристикой «прыгающие лягушки» (АЭПК+ПЛ).

Таблица 3.6 – Процент правильной классификации положительного класса до и после добавления правил алгоритмом прыгающих лягушек

Данные		АЭПК	АЭПК+ПЛ			
		2 правила	3 правила	4 правила	7 правил	9 правил
1	glass1	31,8	**69,1** ± 12,6	62,2 ± 11,2	68,6 ± 9,4	65,9 ± 8,8
2	ecoli0vs1	**97,3**	92,1 ± 2,5	94,2 ± 1,7	94,2 ± 3,6	93,7 ± 4,0
3	wisconsin	94,1	91,1 ± 1,6	**94,7** ± 1,7	93,6 ± 2,8	93,0 ± 2,1
4	pima	57,5	68,5 ± 5,9	70,1 ± 5,2	**70,5** ± 3,4	70,4 ± 3,6
5	glass0	45,7	**78,9** ± 6,3	78,0 ± 5,9	74,3 ± 8,2	74,9 ± 6,9
6	yeast1	**98,1**	66,6 ± 4,6	63,5 ± 3,3	68,3 ± 2,0	68,1 ± 3,1
7	haberman	**54,3**	34,2 ± 3,8	47,5 ± 7,9	48,3 ± 7,7	50,6 ± 7,3
8	vehicle2	22,0	61,6 ± 7,1	63,6 ± 7,9	79,8 ± 5,4	**82,1** ± 5,5
9	vehicle1	57,2	39,7 ± 3,9	**73,2** ± 6,0	64,9 ± 5,0	67,3 ± 4,6
10	vehicle3	48,6	36,1 ± 4,4	65,9 ± 8,1	63,7 ± 4,2	**67,5** ± 4,2
11	glass0123vs456	80,4	**95,6** ± 2,5	93,6 ± 4,8	87,7 ± 5,5	85,0 ± 5,1
12	vehicle0	34,7	69,4 ± 8,8	80,0 ± 9,8	84,4 ± 5,0	**87,8** ± 3,1
13	ecoli1	75,5	89,6 ± 2,9	**90,5** ± 1,7	86,8 ± 3,8	86,6 ± 3,4
14	newthyroid2	**100,0**	96,6 ± 0,9	97,1 ± 0,0	94,9 ± 5,0	94,9 ± 3,7
15	newthyroid1	98,3	99,1 ± 0,4	99,3 ± 0,4	98,9 ± 0,4	**99,6** ± 0,2
16	ecoli2	15,1	84,2 ± 7,9	84,0 ± 6,1	85,6 ± 5,3	**87,5** ± 5,6
17	segment0	84,2	95,1 ± 1,7	95,0 ± 1,4	87,4 ± 2,9	**97,0** ± 1,6
18	glass6	14,0	78,5 ± 8,9	**80,9** ± 9,3	78,9 ± 6,9	80,4 ± 8,3
19	yeast3	**94,5**	87,6 ± 2,0	88,3 ± 2,4	88,7 ± 3,2	87,1 ± 2,3
20	ecoli3	37,1	**89,1** ± 5,0	88,6 ± 4,1	81,1 ± 5,5	74,9 ± 7,8
21	page-blocks0	42,8	62,4 ± 4,4	80,8 ± 4,0	76,8 ± 5,4	**87,7** ± 1,5
22	yeast2vs4	77,5	76,7 ± 7,7	80,3 ± 6,6	**82,6** ± 8,3	78,4 ± 6,8
23	yeast05679vs4	**80,4**	58,5 ± 5,3	69,5 ± 7,7	68,6 ± 6,1	66,0 ± 6,8
24	vowel0	84,4	83,6 ± 1,4	87,1 ± 1,8	87,8 ± 5,6	**89,6** ± 6,0
25	glass2	6,7	**63,7** ± 17,9	60,7 ± 16,8	52,7 ± 13,6	42,0 ± 13,9
26	glass4	13,3	**78,7** ± 12,3	70,0 ± 20,3	71,3 ± 12,0	65,3 ± 14,9
27	ecoli4	50,0	85,0 ± 10,4	**86,0** ± 4,8	83,0 ± 5,6	73,0 ± 5,2
28	page-bl.1-3vs4	58,0	86,8 ± 7,7	**88,8** ± 6,1	85,2 ± 10,1	78,3 ± 8,5
29	abalone9-18	50,6	66,1 ± 6,6	68,9 ± 5,8	**70,6** ± 3,8	64,7 ± 12,3
30	yeast1458vs7	36,7	**72,0** ± 7,2	54,7 ± 10,1	46,7 ± 11,5	46,0 ± 10,7
31	yeast2vs8	50,0	53,0 ± 4,0	57,0 ± 2,4	**58,0** ± 7,2	51,0 ± 11,2
32	yeast4	74,4	64,7 ± 5,4	75,4 ± 6,6	**77,6** ± 3,9	70,7 ± 6,6
33	yeast1289vs7	46,7	**72,0** ± 6,7	61,3 ± 7,5	53,3 ± 9,9	55,3 ± 12,5
34	yeast5	54,2	94,8 ± 1,1	**95,7** ± 0,7	90,2 ± 4,0	88,6 ± 5,5
35	ecoli0137vs26	0,0	**72,0** ± 3,2	68,0 ± 9,6	58,0 ± 16,0	60,0 ± 11,2
36	yeast6	31,4	**82,3** ± 5,5	79,4 ± 3,9	77,7 ± 3,4	76,0 ± 5,9
Среднее		55,5	74,9 ± 5,6	**77,6** ± 5,9	76,1 ± 6,2	75,2 ± 6,4
Процент улучшения		-	19,4	**22,1**	20,6	19,7

На тридцати наборах данных процент правильной классификации положительного класса увеличился после добавления правил. Лучший средний результат достигнут на четырех правилах.

В таблице 3.7 продемонстрирована точность наибольшего класса после увеличения объема базы правил.

Таблица 3.7 – Процент правильной классификации отрицательного класса до и после добавления правил алгоритмом прыгающих лягушек

Данные		АЭПК	АЭПК+ПЛ			
		2 правила	3 правила	4 правила	7 правил	9 правил
1	glass1	**74,1**	55,9 ± 7,1	65,1 ± 7,4	70,7 ± 7,1	72,2 ± 6,9
2	ecoli0vs1	81,2	98,2 ± 2,2	**98,9** ± 1,5	97,6 ± 1,6	97,2 ± 1,6
3	wisconsin	58,5	95,9 ± 1,4	95,5 ± 1,4	**96,3** ± 0,8	96,2 ± 1,0
4	pima	65,2	68,5 ± 3,3	71,7 ± 3,7	73,1 ± 2,7	**74,2** ± 2,7
5	glass0	**84,7**	71,0 ± 5,6	70,7 ± 5,2	78,6 ± 4,6	83,1 ± 4,5
6	yeast1	16,3	64,5 ± 3,8	69,5 ± 3,1	69,9 ± 2,5	**71,9** ± 2,2
7	haberman	37,3	64,4 ± 5,3	55,7 ± 7,8	**64,4** ± 4,8	64,1 ± 4,7
8	vehicle2	73,9	58,4 ± 6,6	72,3 ± 7,9	81,0 ± 4,8	**83,7** ± 4,2
9	vehicle1	30,9	64,3 ± 5,4	52,7 ± 5,3	67,8 ± 2,9	**70,1** ± 4,7
10	vehicle3	31,7	53,2 ± 4,7	43,6 ± 5,2	**66,4** ± 3,9	65,6 ± 3,9
11	glass0123vs456	**95,7**	91,0 ± 1,7	92,9 ± 1,7	94,2 ± 2,0	95,0 ± 1,6
12	vehicle0	**92,7**	75,6 ± 7,4	72,6 ± 4,0	79,3 ± 3,6	85,1 ± 3,2
13	ecoli1	87,6	84,6 ± 1,6	86,7 ± 2,0	89,8 ± 1,9	**90,6** ± 2,4
14	newthyroid2	98,3	**99,4** ± 0,0	99,3 ± 0,2	98,9 ± 0,9	99,0 ± 0,6
15	newthyroid1	**100,0**	95,4 ± 2,7	93,1 ± 3,7	91,4 ± 6,4	89,1 ± 7,3
16	ecoli2	**99,7**	79,1 ± 6,5	86,4 ± 5,1	93,5 ± 2,1	94,0 ± 1,9
17	segment0	92,2	90,3 ± 1,6	93,5 ± 2,2	90,7 ± 1,7	**98,2** ± 0,5
18	glass6	**98,9**	90,4 ± 5,2	93,1 ± 3,1	95,7 ± 2,4	97,5 ± 2,2
19	yeast3	77,5	89,5 ± 1,3	92,0 ± 1,5	92,1 ± 1,1	**93,1** ± 0,8
20	ecoli3	**98,3**	80,4 ± 4,8	86,6 ± 2,0	91,7 ± 1,9	91,2 ± 1,5
21	page-blocks0	**94,8**	88,5 ± 3,6	83,2 ± 3,6	83,4 ± 3,3	88,8 ± 0,7
22	yeast2vs4	64,9	90,2 ± 3,2	88,4 ± 2,4	91,7 ± 2,3	**93,7** ± 1,8
23	yeast05679vs4	48,6	**84,4** ± 4,8	80,8 ± 3,7	82,5 ± 3,2	83,8 ± 2,4
24	vowel0	83,5	85,6 ± 1,2	86,4 ± 1,3	90,3 ± 2,2	**95,0** ± 1,5
25	glass2	**96,4**	58,0 ± 7,5	68,9 ± 5,2	75,1 ± 4,2	80,3 ± 5,7
26	glass4	**99,5**	80,1 ± 7,2	86,7 ± 4,3	94,8 ± 2,7	95,1 ± 2,3
27	ecoli4	**99,7**	92,1 ± 4,2	93,7 ± 2,2	98,2 ± 0,9	98,6 ± 0,7
28	page-bl.1-3vs4	**99,1**	85,5 ± 4,9	90,1 ± 3,7	95,9 ± 2,4	96,9 ± 2,1
29	abalone9-18	73,2	71,3 ± 3,5	74,9 ± 4,0	80,8 ± 2,4	**83,2** ± 2,2
30	yeast1458vs7	59,9	50,0 ± 2,7	63,7 ± 4,7	67,7 ± 3,7	**72,2** ± 4,1
31	yeast2vs8	**97,2**	91,5 ± 3,1	87,2 ± 6,4	88,0 ± 2,0	89,4 ± 3,3
32	yeast4	58,4	76,8 ± 3,5	76,6 ± 4,0	82,4 ± 1,8	**85,0** ± 1,9
33	yeast1289vs7	63,4	55,1 ± 2,9	67,0 ± 3,1	72,7 ± 3,1	**75,9** ± 2,5
34	yeast5	**99,0**	90,6 ± 2,8	88,3 ± 3,9	97,1 ± 0,6	97,2 ± 0,5
35	ecoli0137vs26	**99,6**	91,0 ± 6,8	93,8 ± 3,0	97,4 ± 1,6	98,0 ± 1,3
36	yeast6	**94,7**	84,1 ± 2,2	90,5 ± 1,7	91,2 ± 1,6	92,8 ± 1,3
Среднее		78,5	79,0 ± 4,0	80,9 ± 3,6	85,3 ± 2,7	**87,1** ± 2,6
Процент улучшения		-	0,5	2,4	6,8	**8,6**

После добавления правил алгоритмом прыгающих лягушек только для девятнадцати наборов удалось улучшить точность отрицательного класса. Наибольший процент правильной классификации был получен на девяти правилах.

3.3.3 Сравнение результатов построенных нечетких классификаторов с аналогами. Проведенный эксперимент продемонстрировал, что значение средней геометрической точности

классификатора может быть увеличено после добавления правил метаэвристикой «прыгающие лягушки». Только на двух наборах данных (newthyroid1 и newthyroid2) нечеткий классификатор показал лучшее значение средней геометрической точности до добавления новых правил. Особенностью этих двух наборов является то, что доля правильной классификации положительного класса равняется единице на первичной базе правил, полученной алгоритмом экстремальных значений признаков классов. Добавление правил привело к улучшению распознавания отрицательного класса, но снизило долю правильной классификации положительного класса. В остальных 34 случаях увеличение числа правил позволило увеличить среднюю геометрическую точность от 1 процента (yeast2vs8) до 71 процента (ecoli0137vs26).

Вывод о пользе итерационного добавления правил подтверждает статистическое сравнение с помощью непараметрического критерия Уилкоксона (таблица 3.8). Проведено сравнение по каждой исследуемой метрике – средней геометрической точности *GM* (по данным из таблицы 3.5), точности положительного класса $class_1$ (данные из таблицы 3.6) и точности отрицательного класса $class_2$ (по данным из таблицы 3.7). Нулевая гипотеза гласит, что между результатами нет статистических различий; уровень значимости равен 0,05. Положительное значение стандартизированной статистики критерия (ССК) показывает превосходство результатов, полученных после добавления правил, над точностью классификатора с исходной базой правил.

Таблица 3.8 – Статистическое сравнение средней геометрической точности классификаторов до и после дополнения базы правил

Количество добавленных правил	Метрика	Асимптотическая значимость	ССК	Нулевая гипотеза
1	*GM*	<0,001	5,090	Отклоняется
	$class_1$	0,001	3,236	Отклоняется
	$class_2$	0,626	-0,487	Принимается
2	*GM*	<0,001	5,137	Отклоняется
	$class_1$	<0,001	4,195	Отклоняется
	$class_2$	0,814	0,236	Принимается
5	*GM*	<0,001	5,106	Отклоняется
	$class_1$	<0,001	4,116	Отклоняется
	$class_2$	0,116	1,571	Принимается
7	*GM*	<0,001	5,075	Отклоняется
	$class_1$	<0,001	3,896	Отклоняется
	$class_2$	0,029	2,184	Отклоняется

Во всех случаях для наиболее важных критериев – средней геометрической точности и точности положительного класса – нулевая гипотеза отклоняется, а ССК является положительным. Следовательно, классификаторы с дополненными базами правил показали более высокие результаты, чем классификаторы, структура которых построена только алгоритмом на основе экстремальных значений классов.

Для разных наборов данных количество правил, на котором были достигнуты лучшие показатели, различается. В таблице 3.9 собраны исходные и лучшие результаты для каждого набора, сгруппированные по трем метрикам – средняя геометрическая точность (*GM*), точность положительного класса ($class_1$) и точность отрицательного класса ($class_2$). В скобках приведено количество правил, на котором получен показатель.

Таблица 3.9 – Результаты нечетких классификаторов на лучших базах правил

Данные		*GM*		$class_1$		$class_2$	
		АЭПК	АЭПК+ПЛ	АЭПК	АЭПК+ПЛ	АЭПК	АЭПК+ПЛ
1	glass1	40,5	68,9 ± 5,1 (7)	31,8	69,1 ± 12,6 (3)	74,1	72,2 ± 6,9 (9)
2	ecoli0vs1	88,8	96,4 ± 1,2 (4)	97,3	94,2 ± 1,7 (4)	81,2	98,9 ± 1,5 (4)
3	wisconsin	73,4	95,1 ± 1,1 (4)	94,1	94,7 ± 1,7 (4)	58,5	96,3 ± 0,8 (7)
4	pima	55,6	72,2 ± 2,3 (9)	57,5	70,5 ± 3,4 (7)	65,2	74,2 ± 2,7 (9)
5	glass0	60,1	78,4 ± 3,8 (9)	45,7	78,9 ± 6,3 (3)	84,7	83,1 ± 4,5 (9)
6	yeast1	39,6	69,9 ± 1,2 (9)	98,1	68,3 ± 2,0 (7)	16,3	71,9 ± 2,2 (9)
7	haberman	44,3	56,4 ± 4,8 (9)	54,3	50,6 ± 7,3 (9)	37,3	64,4 ± 4,8 (7)
8	vehicle2	40,0	82,8 ± 4,0 (9)	22,0	82,1 ± 5,5 (9)	73,9	83,7 ± 4,2 (9)
9	vehicle1	41,9	68,4 ± 1,9 (9)	57,2	73,2 ± 6,0 (4)	30,9	70,1 ± 4,7 (9)
10	vehicle3	39,1	66,4 ± 2,6 (9)	48,6	67,5 ± 4,2 (9)	31,7	66,4 ± 3,9 (7)
11	glass0123vs456	87,6	93,2 ± 1,5 (3)	80,4	95,6 ± 2,5 (3)	95,7	95,0 ± 1,6 (9)
12	vehicle0	55,5	86,4 ± 2,2 (9)	34,7	87,8 ± 3,1 (9)	92,7	85,1 ± 3,2 (9)
13	ecoli1	80,8	88,4 ± 1,6 (4)	75,5	90,5 ± 1,7 (4)	87,6	90,6 ± 2,4 (9)
14	newthyroid2	99,2	98,2 ± 0,1 (4)	100,0	97,1 ± 0,0 (4)	98,3	99,4 ± 0,0 (3)
15	newthyroid1	99,2	97,2 ± 1,6 (3)	98,3	99,6 ± 0,2 (9)	100,0	95,4 ± 2,7 (3)
16	ecoli2	34,2	90,5 ± 2,8 (9)	15,1	87,5 ± 5,6 (9)	99,7	94,0 ± 1,9 (9)
17	segment0	88,1	97,6 ± 0,8 (9)	84,2	97,0 ± 1,6 (9)	92,2	98,2 ± 0,5 (9)
18	glass6	22,8	88,2 ± 4,7 (9)	14,0	80,9 ± 9,3 (4)	98,9	97,5 ± 2,2 (9)
19	yeast3	85,5	90,4 ± 1,8 (7)	94,5	88,7 ± 3,2 (7)	77,5	93,1 ± 0,8 (9)
20	ecoli3	50,8	87,1 ± 3,0 (4)	37,1	89,1 ± 5,0 (3)	98,3	91,7 ± 1,9 (7)
21	page-blocks0	63,6	88,2 ± 0,9 (9)	42,8	87,7 ± 1,5 (9)	94,8	88,8 ± 0,7 (9)
22	yeast2vs4	67,3	86,7 ± 4,9 (7)	77,5	82,6 ± 8,3 (7)	64,9	93,7 ± 1,8 (9)
23	yeast05679vs4	61,9	74,3 ± 3,6 (7)	80,4	69,5 ± 7,7 (4)	48,6	84,4 ± 4,8 (3)
24	vowel0	83,9	92,1 ± 3,4 (9)	84,4	89,6 ± 6,0 (9)	83,5	95,0 ± 1,5 (9)
25	glass2	10,8	61,7 ± 10,6 (4)	6,7	63,7 ± 17,9 (3)	96,4	80,3 ± 5,7 (9)
26	glass4	23,1	79,4 ± 10,4 (7)	13,3	78,7 ± 12,3 (3)	99,5	95,1 ± 2,3 (9)
27	ecoli4	68,7	89,9 ± 3,1 (7)	50,0	86,0 ± 4,8 (4)	99,7	98,6 ± 0,7 (9)
28	page-bl.1-3vs4	75,1	90,1 ± 5,1 (7)	58,0	88,8 ± 6,1 (4)	99,1	96,9 ± 2,1 (9)
29	abalone9-18	58,7	75,0 ± 2,3 (7)	50,6	70,6 ± 3,8 (7)	73,2	83,2 ± 2,2 (9)
30	yeast1458vs7	45,8	59,6 ± 3,2 (3)	36,7	72,0 ± 7,2 (3)	59,9	72,2 ± 4,1 (9)
31	yeast2vs8	68,8	70,2 ± 4,5 (7)	50,0	58,0 ± 7,2 (7)	97,2	91,5 ± 3,1 (3)
32	yeast4	65,7	79,9 ± 2,1 (7)	74,4	77,6 ± 3,9 (7)	58,4	85,0 ± 1,9 (9)
33	yeast1289vs7	54,1	63,3 ± 7,2 (7)	46,7	72,0 ± 6,7 (3)	63,4	75,9 ± 2,5 (9)
34	yeast5	72,9	93,5 ± 2,1 (7)	54,2	95,7 ± 0,7 (4)	99,0	97,2 ± 0,5 (9)
35	ecoli0137vs26	0,0	71,3 ± 3,8 (3)	0,0	72,0 ± 3,2 (3)	99,6	98,0 ± 1,3 (9)
36	yeast6	51,2	84,0 ± 2,5 (4)	31,4	82,3 ± 5,5 (3)	94,7	92,8 ± 1,3 (9)
Среднее		58,3	81,4 ± 3,3 (6,7)	55,5	80,8 ± 5,2 (5,6)	78,5	87,5 ± 2,5 (8,0)
Процент улучшения		-	23,1	-	25,3	-	9,0

Таким образом, применение алгоритма итерационного добавления правил на основе «прыгающих лягушек» позволило добиться прироста в средней геометрической точности на 23

процента, в точности наименьшего класса на 25 процентов и в точности наибольшего класса на 9 процентов.

В таблице 3.10 приведено сравнение полученных результатов с аналогичными алгоритмами формирования нечетких классификаторов из статьи [6]. Их краткое описание приведено в параграфе 1.3 данной работы. Алгоритмы Chi-3 и Chi-5 используют по три и пять антецедентов в правиле соответственно. Все алгоритмы, кроме HFRBCS, генерируют по 30 правил для каждого класса. За исключением E-алгоритма, все аналоги применяются в комбинации с алгоритмом предобработки данных SMOTE, поэтому находятся в более привилегированном положении, чем E-алгоритм и предлагаемая комбинация АЭПК+ПЛ.

При сравнении использованы лучшие результаты нечетких классификаторов, построенные комбинацией АЭПК+ПЛ; они представлены в последнем столбце таблицы 3.10. В скобках указано, на каком количестве правил получен показатель.

Таблица 3.10 – Сопоставление средней геометрической точности с аналогичными алгоритмами построения нечетких классификаторов

Данные	Chi-3	Chi-5	Ishibuchi	E-алгоритм	HFRBCS	АЭПК+ПЛ
gl1	64,9 ± 6,9	64,9 ± 6,9	59,3 ± 10,3	0,0 ± 0,0	**73,7** ± 4,7	68,9 ± 5,1 (7)
ecl0/1	92,3 ± 5,9	95,6 ± 5,2	**96,7** ± 2,4	95,3 ± 4,8	93,6 ± 6,5	96,4 ± 1,2 (4)
wis	88,9 ± 2,1	43,6 ± 5,9	95,8 ± 1,4	**96,0** ± 1,6	88,2 ± 1,6	95,1 ± 1,1 (4)
pm	66,8 ± 5,9	66,8 ± 2,3	71,1 ± 4,5	55,0 ± 4,6	68,7 ± 5,3	**72,2** ± 2,3 (9)
gl0	64,1 ± 3,5	63,7 ± 1,8	69,4 ± 7,7	0,0 ± 0,0	76,6 ± 8,1	**78,4** ± 3,8 (9)
yst1	67,7 ± 1,9	69,7 ± 1,5	51,4 ± 12,2	0,0 ± 0,0	**71,7** ± 2,4	69,9 ± 1,2 (9)
hbr	58,9 ± 6,0	60,4 ± 2,4	**62,7** ± 2,8	4,9 ± 11,1	57,1 ± 4,1	56,4 ± 4,8 (9)
vhc2	85,5 ± 3,4	87,2 ± 3,0	67,8 ± 5,0	43,8 ± 13,2	**90,6** ± 2,2	82,8 ± 4,0 (9)
vhc1	70,9 ± 4,3	**71,9** ± 1,3	64,9 ± 4,4	3,1 ± 6,9	71,8 ± 2,6	68,4 ± 1,9 (9)
vhc3	**69,2** ± 4,9	63,1 ± 2,0	63,1 ± 4,1	0,0 ± 0,0	66,8 ± 3,3	66,4 ± 2,6 (9)
gl0123/456	85,8 ± 3,0	85,9 ± 1,7	88,6 ± 5,2	82,1 ± 7,0	88,4 ± 4,0	**93,2** ± 1,5 (3)
vhc0	86,4 ± 3,1	84,9 ± 1,6	75,9 ± 1,4	39,1 ± 16,5	**88,9** ± 2,0	86,4 ± 2,2 (9)
ecl1	85,3 ± 9,8	86,1 ± 8,6	85,7 ± 2,9	77,8 ± 7,9	84,2 ± 12,7	**88,4** ± 1,6 (4)
nwth2	89,8 ± 10,8	96,3 ± 6,7	94,2 ± 4,2	88,6 ± 3,8	**99,7** ± 0,6	98,2 ± 0,1 (4)
nwth1	87,4 ± 8,1	95,4 ± 8,8	89,0 ± 13,5	88,5 ± 8,8	**98,6** ± 2,5	97,2 ± 1,6 (3)
ecl2	88,0 ± 5,5	87,6 ± 5,0	87,0 ± 4,4	70,4 ± 15,4	87,6 ± 8,2	**90,5** ± 2,8 (9)
sgm0	95,0 ± 0,5	95,9 ± 1,2	42,5 ± 2,8	95,3 ± 1,1	97,5 ± 1,1	**97,6** ± 0,8 (9)
gl6	83,9 ± 9,8	78,1 ± 7,8	86,3 ± 8,2	**90,2** ± 3,8	87,0 ± 10,8	88,2 ± 4,7 (9)
yst3	90,1 ± 4,1	89,3 ± 3,3	77,1 ± 17,7	82,0 ± 2,3	**90,4** ± 2,3	**90,4** ± 1,8 (7)
ecl3	87,6 ± 4,1	**91,6** ± 5,0	85,4 ± 3,7	75,5 ± 8,7	90,8 ± 4,4	87,1 ± 3,0 (4)
pb0	79,9 ± 4,3	87,3 ± 1,9	32,2 ± 9,6	64,5 ± 2,8	**91,4** ± 0,7	88,2 ± 0,9 (9)
yst2/4	86,8 ± 5,5	86,4 ± 7,4	70,9 ± 23,5	80,9 ± 9,1	**89,3** ± 4,2	86,7 ± 4,9 (7)
yst05679/4	78,9 ± 6,0	76,0 ± 6,4	**79,5** ± 9,5	60,0 ± 16,4	73,2 ± 7,5	74,3 ± 3,6 (7)
vwl0	98,4 ± 0,6	97,9 ± 1,8	89,0 ± 6,6	89,6 ± 6,1	**98,8** ± 1,6	92,1 ± 3,4 (9)
gl2	47,7 ± 10,2	49,2 ± 8,2	43,6 ± 15,7	9,9 ± 22,1	54,8 ± 20,6	**61,7** ± 10,6 (4)
gl4	**85,0** ± 13,8	81,8 ± 11,2	78,3 ± 17,7	83,4 ± 19,9	70,4 ± 40,5	79,4 ± 10,4 (7)
ecl4	91,3 ± 7,4	92,1 ± 8,4	86,9 ± 8,7	92,4 ± 8,2	**93,0** ± 8,2	89,9 ± 3,1 (7)
pb1-3/4	91,9 ± 4,8	92,9 ± 9,5	94,5 ± 4,9	94,1 ± 10,3	**98,6** ± 0,7	90,1 ± 5,1 (7)

Продолжение таблицы 3.10

Данные	Chi-3	Chi-5	Ishibuchi	E-алгоритм	HFRBCS	АЭПК+ПЛ
ab9/18	63,9 ± 11,0	66,5 ± 10,7	65,8 ± 9,2	32,3 ± 20,6	67,6 ± 14,0	**75,0** ± 2,3 (7)
yst1458/7	62,4 ± 4,6	58,8 ± 8,6	40,8 ± 16,6	0,0 ± 0,0	**62,5** ± 6,3	59,6 ± 3,2 (3)
yst2/8	72,8 ± 15,0	**78,8** ± 8,6	72,8 ± 15,0	72,8 ± 15,0	72,5 ± 15,1	70,2 ± 4,5 (7)
yst4	83,0 ± 3,1	**83,1** ± 2,6	71,4 ± 23,3	32,2 ± 20,6	82,6 ± 2,3	79,9 ± 2,1 (7)
yst1289/7	**76,1** ± 7,2	69,3 ± 4,6	48,6 ± 16,9	50,0 ± 13,6	69,4 ± 4,4	63,3 ± 7,2 (7)
yst5	93,4 ± 5,4	93,6 ± 2,1	**94,9** ± 0,4	88,2 ± 7,0	94,2 ± 2,6	93,5 ± 2,1 (7)
ecl0137/26	71,0 ± 41,4	49,6 ± 46,4	71,3 ± 41,7	**73,7** ± 43,1	71,5 ± 41,8	71,3 ± 3,8 (3)
yst6	87,5 ± 10,6	87,7 ± 9,3	**88,4** ± 6,1	51,7 ± 13,8	84,9 ± 12,9	84,0 ± 2,5 (4)
Среднее	80,0 ± 7,1	78,6 ± 6,4	73,4 ± 9,6	57,3 ± 9,6	**81,8 ±** 7,6	81,4 ± 3,3 (6,7)

Для попарного сравнения результатов был использован критерий Уилкоксона. Полученные значения критерия показаны в таблице 3.11. Положительное значение стандартизированной статистики критерия (ССК) свидетельствует о превосходстве результатов комбинации АЭПК+ПЛ над результатами аналогов.

Таблица 3.11 – Сравнение средней геометрической точности, полученной различными нечеткими классификаторами, непараметрическим критерием Уилкоксона

Аналог	Асимптотическая значимость	ССК	Нулевая гипотеза
Chi-3	0,242	1,17	Принимается
Chi-5	0,315	1,005	Принимается
Ishibuchi	<0,001	3,723	Отклоняется
E-алгоритм	<0,001	4,556	Отклоняется
HFRBCS	0,396	-0,848	Принимается

По значению средней геометрической точности результаты нечетких классификаторов, построенных с помощью алгоритма на основе экстремумов признаков классов и метаэвристики «прыгающие лягушки», статистически неразличимы с результатами иерархических нечетких классификаторов HFRBCS, а также алгоритмов Chi-3 и Chi-5. Средняя геометрическая точность, полученная после применения АЭПК+ПЛ, существенно превосходит результаты Ishibuchi и E-алгоритма, последний из которых позиционируется как способ построения базы правил при наличии дисбаланса в данных.

Разброс результатов, достигнутых с помощью комбинации АЭПК+ПЛ меньше, чем у аналогов, что подтверждается статистическим сравнением (таблица 3.12). Так как нулевая гипотеза отклоняется во всех случаях, можно сделать вывод о лучшей стабильности нечетких классификаторов, структура которых сформирована комбинацией алгоритма экстремумов признаков классов и метаэвристикой «прыгающие лягушки». Чем больше ССК, тем существеннее различается разброс результатов между аналогом и предложенным алгоритмом.

Таблица 3.12 – Сравнение критерием Уилкоксона разброса средней геометрической точности исследуемых нечетких классификаторов

Аналог	Асимптотическая значимость	ССК	Нулевая гипотеза
Chi-3	<0,001	4,572	Отклоняется
Chi-5	0,001	3,224	Отклоняется
Ishibuchi	<0,001	4,823	Отклоняется
E-алгоритм	<0,001	4,336	Отклоняется
HFRBCS	<0,001	3,488	Отклоняется

Так как применение комбинации алгоритма экстремальных значений признаков классов и итеративной процедуры добавления нечетких правил метаэвристикой «прыгающие лягушки» не требует предварительной предобработки данных и позволяет достигнуть сопоставимую или превосходящую аналогичные алгоритмы среднюю геометрическую точность на меньшем количестве правил, её использование с целью получения точной и компактной системы классификации является предпочтительным.

3.4 Исследование гибридного алгоритма оптимизации параметров нечеткого классификатора

3.4.1 Описание эксперимента. Эксперимент направлен на изучение качества работы гибридного алгоритма, основанного на метаэвристиках «гравитационный поиск» и «прыгающие лягушки», при оптимизации параметров функций принадлежности нечетких классификаторов несбалансированных данных.

Для генерации структуры классификатора был использован алгоритм на основе экстремальных значений признаков классов. Параметры термов построенных классификаторов настраивались исследуемыми инструментами: метаэвристикой «гравитационный поиск» (ГП) [112], метаэвристикой «прыгающие лягушки» (ПЛ) [114] и гибридным алгоритмом (ГП+ПЛ), описанном в параграфе 2.2. Каждый алгоритм запускался по десять раз, далее были подсчитаны средние значения метрик качества. Параметры алгоритмов указаны в таблице 3.13; они подобраны эмпирически как наиболее универсальные для всех исследуемых наборов данных.

Таблица 3.13 – Параметры алгоритмов оптимизации

Параметр	ГП	ПЛ	ГП+ПЛ
Число итераций	1000 итераций	20 глобальных итераций, 50 локальных	20 глобальных итераций, 50 локальных
Размер популяции	40 векторов	40 векторов (5 мемплексов по 8 агентов)	40 векторов (5 мемплексов по 8 агентов)
Переменные	$G_0 = 100$, $\alpha = 10$, $\varepsilon = 0{,}01$	*const* = 1,2	$G_0 = 100$, $\alpha = 10$, $\varepsilon = 0{,}01$, *const* = 1,2

3.4.2 Результаты эксперимента. В таблице 3.14 представлены усредненные по десяти прогонам результаты настройки параметров нечетких классификаторов на тестовых выборках. Помимо средней геометрической точности (*GM*), использованной в качестве фитнес-функции всеми метаэвристиками, в таблице также представлено время работы алгоритмов (*Time*).

Таблица 3.14 – Результаты настройки параметров нечеткого классификатора гибридным алгоритмом и составляющими его метаэвристиками в отдельности

Данные	*GM*			*Time*		
	ГП+ПЛ	ПЛ	ГП	ГП+ПЛ	ПЛ	ГП
glass1	64,2 ± 5,3	62,0 ± 5,4	**64,7 ±** 5,2	14,4 ± 0,6	**11,4** ± 0,5	77,4 **±** 7,5
ecoli0vs1	96,0 ± 2,4	96,1 ± 2,0	**96,2 ±** 2,4	14,5 ± 0,2	**10,3** ± 0,9	53,7 **±** 7,2
wisconsin	**96,1 ±** 1,4	94,8 ± 1,6	95,9 ± 1,2	38,0 ± 1,1	**30,8** ± 1,1	65,1 ± 47,1
pima	**72,6 ±** 2,1	72,1 ± 2,4	71,9 ± 1,9	**34,8** ± 1,0	35,4 ± 1,3	208,5 ± 6,5
glass0	**78,4 ±** 3,1	78,2 ± 4,4	73,4 ± 5,7	16,9 ± 0,5	**15,3** ± 1,5	34,0 ± 20,1
yeast1	71,3 ± 2,6	**71,6 ±** 2,7	69,4 ± 2,2	90,0 **±** 2,8	60,1 ± 3,5	**2,9** ± 0,0
haberman	59,2 ± 4,9	52,9 ± 6,0	**62,5 ±** 3,9	8,6 ± 0,1	**8,5 ±** 0,4	51,4 **±** 2,3
vehicle2	**87,0 ±** 3,6	84,9 ± 3,1	77,8 ± 4,9	73,8 ± 2,8	**66,0** ± 2,7	427,4 ± 12,8
vehicle1	**70,9 ±** 2,6	69,6 ± 2,5	66,9 ± 3,0	73,1 ± 2,9	**65,4** ± 3,1	427,1 ± 11,4
vehicle3	68,8 ± 3,3	**69,1 ±** 2,4	65,3 ± 2,9	73,5 **±** 3,0	**66,7** ± 3,7	430,9 ± 13,1
glass0123vs456	90,9 ± 3,5	**91,2 ±** 3,2	90,8 ± 3,1	14,5 **±** 0,6	**10,1** ± 0,4	77,1 ± 3,0
vehicle0	**93,2 ±** 1,3	92,5 ± 1,8	82,9 ± 3,6	**72,7** ± 2,7	74,6 ± 3,5	498,9 ± 40,4
ecoli1	**88,2 ±** 3,2	88,0 ± 3,2	86,4 ± 4,3	20,4 ± 0,3	13,4 ± 0,6	**10,1** ± 14,9
newthyroid2	**98,5 ±** 1,7	97,8 ± 1,9	97,8 ± 2,5	9,2 ± 0,2	**7,7** ± 0,4	60,2 ± 1,0
newthyroid1	96,8 ± 3,2	**98,2 ±** 1,7	97,7 ± 2,6	9,2 **±** 0,1	**8,0** ± 0,5	59,9 ± 1,4
ecoli2	90,1 ± 2,9	**91,0 ±** 2,8	71,4 ± 28,6	20,8 **±** 0,4	13,4 ± 0,6	**0,8** ± 0,0
segment0	**98,7 ±** 0,7	98,4 ± 0,5	94,5 ± 0,9	205,6 ± 5,2	**204,7** ± 25,2	823,7 ± 136,5
glass6	**86,3 ±** 4,6	83,6 ± 5,6	75,5 ± 17,1	16,5 ± 0,9	**13,4** ± 1,7	33,2 ± 24,7
yeast3	90,6 ± 1,8	**91,1 ±** 1,8	90,4 ± 2,2	89,1 **±** 3,2	62,5 ± 6,6	**2,8** ± 0,0
ecoli3	**86,7 ±** 5,6	83,7 ± 6,0	66,5 ± 26,6	20,6 ± 0,3	13,3 ± 0,5	**0,7** ± 0,0
page-blocks0	**85,8 ±** 1,8	81,2 ± 2,2	83,7 ± 3,0	276,9 ± 8,7	**250,8** ± 17,9	1473,0 ± 17,2
yeast2vs4	86,3 ± 3,5	**86,7 ±** 3,0	84,8 ± 3,7	33,4 **±** 0,9	22,1 ± 1,0	**1,1** ± 0,0
yeast05679vs4	**76,3 ±** 7,3	75,7 ± 8,1	71,8 ± 7,0	34,0 ± 0,8	23,6 ± 1,2	**1,2** ± 0,0
vowel0	94,5 ± 1,8	**94,7 ±** 1,8	91,5 ± 2,1	73,2 **±** 2,0	62,5 ± 3,5	**2,9** ± 0,1
glass2	70,3 ± 8,9	**72,2 ±** 6,9	48,0 ± 22,9	17,2 **±** 0,4	18,0 ± 1,6	**13,7** ± 13,5
glass4	82,6 ± 12,0	**84,2 ±** 9,9	72,6 ± 18,8	16,4 **±** 0,6	**13,0** ± 1,1	23,7 ± 20,5
ecoli4	**91,3 ±** 6,0	90,1 ± 5,6	90,9 ± 5,1	20,8 ± 0,3	13,4 ± 0,4	**8,8** ± 12,9
page-bl.1-3vs4	94,2 ± 4,3	**95,4 ±** 3,7	92,4 ± 4,9	30,5 **±** 0,7	**22,9** ± 1,2	159,2 ± 4,3
abalone9-18	**84,7 ±** 3,2	83,0 ± 5,2	71,5 ± 6,0	42,3 ± 1,2	28,3 ± 1,6	**1,3** ± 0,0
yeast1458vs7	**66,0 ±** 5,6	60,9 ± 7,3	52,9 ± 11,3	44,1 ± 1,7	31,3 ± 2,7	**1,4** ± 0,0
yeast2vs8	**71,7 ±** 11,5	69,4 ± 12,0	67,9 ± 10,7	32,0 ± 0,6	21,5 ± 0,9	**1,0** ± 0,0
yeast4	79,4 ± 3,5	**79,8 ±** 3,6	78,9 ± 3,6	89,3 **±** 2,8	63,2 ± 4,3	**2,8** ± 0,0
yeast1289vs7	**68,2 ±** 8,9	67,0 ± 6,5	62,3 ± 5,9	59,6 ± 1,6	42,8 ± 4,4	**1,8** ± 0,0
yeast5	93,1 ± 4,8	92,5 ± 4,8	**94,3 ±** 3,1	90,5 ± 0,9	59,5 **±** 2,6	**2,8 ±** 0,0
ecoli0137vs26	**60,5 ±** 37,3	59,3 ± 38,8	42,4 ± 44,6	14,2 ± 0,3	**12,1** ± 0,8	64,7 ± 25,7
yeast6	85,5 ± 5,8	**85,7 ±** 6,0	84,8 ± 6,5	90,3 **±** 1,6	63,3 ± 4,8	**2,8** ± 0,0
Среднее	**82,6 ±** 5,2	81,8 ± 5,2	77,5 ± 7,9	52,2 ± 1,5	**42,8** ± 3,0	141,9 ± 12,3

В таблице 3.15 содержатся полученные с помощью исследуемых метаэвристик проценты правильной классификации положительного ($class_1$) и отрицательного ($class_2$) классов.

Таблица 3.15 – Точность наименьшего и наибольшего классов после настройки параметров термов нечетких классификаторов

Данные	$class_1$			$class_2$		
	ГП + ПЛ	ПЛ	ГП	ГП + ПЛ	ПЛ	ГП
glass1	64,0 ± 7,9	**68,9 ±** 10,6	63,1 ± 10,2	66,0 ± 9,5	58,1 ± 10,1	**68,9 ±** 9,7
ecoli0vs1	94,3 ± 4,8	**95,1 ±** 4,4	94,8 ± 4,1	**97,9 ±** 2,6	97,3 ± 2,8	97,6 ± 2,4
wisconsin	**96,1 ±** 2,4	93,2 ± 2,7	95,6 ± 2,1	96,3 ± 1,1	**96,4 ±** 1,2	96,2 ± 1,2
pima	**71,8 ±** 4,3	70,7 ± 5,5	70,9 ± 4,7	73,6 ± 3,3	**74,0 ±** 4,3	73,3 ± 3,7
glass0	**82,0 ±** 5,5	80,0 ± 5,9	79,3 ± 11,1	75,5 ± 6,1	**76,7 ±** 5,9	69,7 ± 8,6
yeast1	**70,9 ±** 4,1	70,5 ± 5,2	69,7 ± 4,3	71,8 ± 3,2	**72,9 ±** 3,4	69,5 ± 4,9
haberman	51,3 ± 7,6	43,9 ± 9,8	**56,4 ±** 7,8	69,6 ± 5,8	66,5 ± 8,4	**70,6 ±** 4,9
vehicle2	**87,1 ±** 4,8	86,7 ± 5,9	79,1 ± 10,7	**87,0 ±** 3,3	83,5 ± 4,5	77,9 ± 6,9
vehicle1	**68,2 ±** 6,8	65,9 ± 6,8	65,8 ± 7,2	**74,4 ±** 4,1	74,3 ± 4,6	68,9 ± 5,3
vehicle3	67,3 ± 8,3	**68,5 ±** 5,6	61,9 ± 6,8	**71,1 ±** 4,2	70,3 ± 4,9	69,6 ± 3,9
glass0123vs456	90,3 ± 7,3	**90,9 ±** 6,8	90,2 ± 6,7	91,9 ± 4,4	**92,0 ±** 4,1	91,9 ± 4,3
vehicle0	94,8 ± 2,9	**95,0 ±** 3,3	89,3 ± 5,7	**91,7 ±** 2,6	90,3 ± 3,0	77,8 ± 7,1
ecoli1	**92,1 ±** 7,2	92,0 ± 6,7	90,2 ± 7,5	**85,0 ±** 5,6	84,7 ± 6,3	83,2 ± 5,7
newthyroid2	**98,0 ±** 2,5	97,1 ± 3,4	96,6 ± 4,3	**99,1 ±** 1,5	98,5 ± 1,5	**99,1 ±** 1,3
newthyroid1	94,6 ± 6,6	**97,7 ±** 3,2	96,1 ± 5,0	99,3 ± 1,0	98,8 ± 1,3	**99,4 ±** 0,9
ecoli2	**88,3 ±** 5,1	**88,3 ±** 5,8	68,7 ± 27,5	92,1 ± 2,5	94,0 ± 2,2	**94,2 ±** 3,7
segment0	**98,3 ±** 1,3	98,2 ± 1,1	98,0 ± 1,5	**99,0 ±** 0,6	98,7 ± 0,7	91,2 ± 2,1
glass6	**78,2 ±** 8,9	73,7 ± 9,3	69,0 ± 17,3	**95,9 ±** 3,1	95,6 ± 3,3	92,1 ± 6,4
yeast3	**91,2 ±** 3,0	**91,2 ±** 3,4	89,5 ± 3,6	90,1 ± 1,9	91,1 ± 1,7	**91,4 ±** 2,2
ecoli3	**87,4 ±** 15,1	81,7 ± 15,1	63,4 ± 27,8	87,1 ± 4,5	87,3 ± 4,9	**90,6 ±** 5,5
page-blocks0	**80,3 ±** 3,5	72,7 ± 4,5	79,8 ± 3,7	**91,8 ±** 1,3	90,9 ± 2,3	87,9 ± 2,9
yeast2vs4	80,5 ± 7,1	**81,3 ±** 6,3	79,2 ± 6,9	**92,9 ±** 1,8	92,8 ± 1,6	91,0 ± 2,6
yeast05679vs4	**68,7 ±** 14,3	67,9 ± 15,3	63,9 ± 13,3	**86,6 ±** 3,3	86,1 ± 2,6	82,8 ± 4,5
vowel0	96,8 ± 3,4	**97,3 ±** 3,1	94,4 ± 4,5	**92,3 ±** 1,6	92,2 ± 1,6	88,8 ± 1,8
glass2	70,0 ± 18,1	**73,7 ±** 13,7	53,7 ± 29,4	**73,5 ±** 6,8	72,7 ± 7,4	68,9 ± 17,4
glass4	78,3 ± 19,2	**78,7 ±** 18,8	67,7 ± 27,2	90,6 ± 4,2	**92,6 ±** 4,1	89,3 ± 5,8
ecoli4	85,5 ± 12,0	83,5 ± 11,2	**87,5 ±** 10,8	**98,0 ±** 1,5	97,9 ± 1,7	95,2 ± 2,6
page-bl.1-3vs4	**89,7 ±** 8,1	94,1 ± 7,1	86,6 ± 9,0	**99,3 ±** 0,8	96,9 ± 1,6	99,0 ± 1,1
abalone9-18	**84,6 ±** 6,7	82,6 ± 10,1	60,7 ± 10,6	85,3 ± 4,1	84,5 ± 3,1	**85,5 ±** 3,3
yeast1458vs7	**59,3 ±** 10,4	51,7 ± 12,8	47,7 ± 19,1	**74,8 ±** 4,0	74,5 ± 5,1	65,5 ± 6,6
yeast2vs8	**57,0 ±** 15,6	55,0 ± 16,2	54,5 ± 15,8	**94,9 ±** 4,1	92,5 ± 5,1	88,3 ± 6,8
yeast4	72,4 ± 6,7	72,6 ± 6,7	**74,5 ±** 7,8	87,6 ± 1,9	**88,1 ±** 1,6	84,1 ± 3,1
yeast1289vs7	**60,7 ±** 15,5	57,7 ± 12,4	56,0 ± 12,4	**80,8 ±** 4,4	79,8 ± 4,6	71,9 ± 6,6
yeast5	90,2 ± 9,4	88,9 ± 9,2	**93,4 ±** 6,0	**96,7 ±** 1,0	**96,7 ±** 1,0	95,2 ± 1,2
ecoli0137vs26	**57,0 ±** 37,6	**57,0 ±** 38,4	43,0 ± 45,2	**97,3 ±** 1,8	96,5 ± 2,0	97,2 ± 2,0
yeast6	79,4 ± 11,1	79,7 ± 11,2	**81,1 ±** 12,6	92,7 ± 1,1	**92,9 ±** 1,6	89,8 ± 2,8
Среднее	**79,9 ±** 8,7	79,0 ± 8,8	75,3 ± 11,4	**87,5 ±** 3,2	86,9 ± 3,5	84,8 ± 4,5

И по проценту правильной классификации положительного класса, и по точности отрицательного класса гибридный алгоритм продемонстрировал лучшие показатели при усреднении по всем наборам данным.

3.4.3 Сравнение результатов построенных нечетких классификаторов. Сравнение критерием Уилкоксона средней геометрической точности, полученной путем настройки нечетких классификаторов каждым из трех участвующих в эксперименте инструментов, показывает, что существует значимое статистическое различие между результатами гибрида и

исходными метаэвристиками (таблица 3.16). Положительное значение ССК свидетельствует о превосходстве результатов гибридного алгоритма.

Таблица 3.16 – Попарное сравнение результатов классификации после оптимизации термов предлагаемым гибридом и отдельными метаэвристиками

Метаэвристики	Критерий	Асимптотическая значимость	ССК	Нулевая гипотеза
ПЛ	*GM*	0,017	2,388	Отклоняется
	Time	<0,001	5,027	Отклоняется
	$class_1$	0,090	1,698	Принимается
	$class_2$	0,035	2,106	Отклоняется
ГП	*GM*	<0,001	4,430	Отклоняется
	Time	0,239	-1,178	Принимается
	$class_1$	<0,001	3,582	Отклоняется
	$class_2$	<0,001	3,728	Отклоняется

По времени работы лучший результат показал алгоритм прыгающих лягушек: при сравнении с гибридом асимптотическая значимость меньше 0,001, что свидетельствует о наличии статистического различия в исследуемом показателе. Время работы «гравитационного поиска» и гибридного алгоритма оказывается сопоставимым (асимптотическая значимость 0,239). Но при рассмотрении временных затрат метаэвристики «гравитационный поиск» можно заметить, что есть наборы, на которых алгоритм выдает результат очень быстро (yeast1, ecoli2, yeast3 и другие). Равенство разброса нулю показывает, что эта ситуация повторялась во время каждого из десяти запусков. С другой стороны, есть наборы, на которых «гравитационный поиск» работал намного больше, чем остальные алгоритмы (wisconsin, pima, vehicle2 и другие). Такой нестабильный режим работы показывает, что метаэвристика «гравитационный поиск» не подходит для построения нечетких классификаторов на несбалансированных данных.

Можно заключить, что при необходимости достижения лучшей средней геометрической точности между тремя алгоритмами стоит предпочесть гибридный вариант, но для наиболее быстрого получения результатов нужно использовать «прыгающие лягушки».

Проведено статистическое сравнение средней геометрической точности, полученной с помощью гибридного алгоритма, с результатами аналогичных классификаторов из таблицы 3.10. Нулевая гипотеза (НГ) гласит, что между исследуемыми выборками нет статистических различий. Итоги сравнения приведены в таблице 3.17. Положительное значение ССК показывает о превосходстве результатов предложенной комбинации метаэвристик над аналогами.

Таблица 3.17 – Статистическое сравнение критерием Уилкоксона средней геометрической точности после оптимизации параметров классификатора гибридным алгоритмом с аналогами

Аналог	Асимптотическая значимость	ССК	Нулевая гипотеза
Chi-3	0,029	2,184	Отклоняется
Chi-5	0,018	2,372	Отклоняется
Ishibuchi	<0,001	3,928	Отклоняется
E-алгоритм	<0,001	4,587	Отклоняется
HFRBCS	0,671	0,424	Принимается

Нечеткие классификаторы несбалансированных данных, параметры которых настроены комбинацией ГП+ПЛ, всего лишь на двух нечетких правилах показывают статистически различимые результаты с алгоритмами Chi-3, Chi-5, Ishibuchi и E-алгоритмом, а также сопоставимые результаты с иерархическими нечеткими системами HFRBCS. По сравнению с результатами до оптимизации (таблица 3,5, столбец АЭПК), средняя геометрическая точность и точность положительного класса после оптимизации параметров алгоритмом ГП+ПЛ увеличились в среднем на 24 процента, точность отрицательного класса на 9 процентов.

3.5 Проверка эффективности алгоритма настройки весовых коэффициентов признаков в нечетком классификаторе несбалансированных данных

Для исследования эффективности разработанного алгоритма настройки весовых коэффициентов признаков, описанного в параграфе 2.3, был проведен эксперимент на основе двух схем построения нечеткого классификатора несбалансированных данных. Использование двух схем обусловлено необходимостью формирования методики построения нечеткого классификатора при наличии двух этапов оптимизации: этапа настройки весовых коэффициентов признаков и этапа поиска оптимальных параметров термов.

3.5.1 Первая схема эксперимента. Первая схема включала три этапа: формирование структуры классификатора алгоритмом экстремальных значений признаков классов, настройку весов признаков гибридным алгоритмом, сочетающем метаэвристики «гравитационный поиск» и «прыгающие лягушки», и оптимизацию параметров термов тем же гибридом. Популяция весов для алгоритма настройки весовых коэффициентов генерировалась двумя способами: первый включал оценку взаимной информации (ГП+ПЛ+ВИ), второй – случайную генерацию (ГП+ПЛ+СГ).

И при оптимизации термов, и при настройке весовых коэффициентов выполнялось по 20 глобальных и 25 локальных итераций; популяция состояла из 40 векторов (5 мемплексов по 8 агентов). При поиске оптимальных функций принадлежности гибридным алгоритмом из параграфа 2.2 были использованы следующие параметры: $G_0 = 100$, $\alpha = 10$, $\varepsilon = 0{,}01$, $const = 1{,}2$. Для настройки весов для алгоритма, изложенного в параграфе 2.3, применялся другой набор переменных: $G_0 = 10$, $\alpha = 2$, $\varepsilon = 0{,}01$, $const_0 = 1$. При получении меньшего веса признака, чем значение порога ω, равного 0,2, вес принудительно приравнивался к нулю. В качестве фитнес-функции на всех этапах оптимизации применялась средняя геометрическая точность.

Значения средней геометрической точности, снятые после второго (веса) и третьего этапа (веса + термы) этой схемы, приведены в таблице 3.18. В последней строке таблицы приведена средняя разность между точностью, полученной после этапа генерации структуры (таблица 3.5,

столбец АЭПК), и представленным в столбце этапом оптимизации. Результаты усреднены по 15 запускам.

Таблица 3.18 – Средняя геометрическая точность нечетких классификаторов после настройки весов и после комбинации настройки весов и термов

Данные	АЭПК	ГП+ПЛ+ВИ		ГП+ПЛ+СГ	
		Веса	Веса + термы	Веса	Веса + термы
glass1	40,5	**61,0** ± 0,8	59,0 ± 2,7	60,4 ± 0,8	60,4 ± 2,2
ecoli0vs1	88,8	96,8 ± 0,4	**97,0** ± 0,7	96,7 ± 0,2	96,4 ± 1,5
wisconsin	73,4	92,0 ± 0,1	**95,1** ± 0,8	92,0 ± 0,0	**95,1** ± 0,6
pima	55,6	64,8 ± 0,5	**69,5** ± 1,2	64,5 ± 0,4	69,3 ± 0,6
glass0	60,1	**77,6** ± 1,0	74,9 ± 2,6	77,5 ± 0,7	74,1 ± 2,5
yeast1	39,6	59,8 ± 0,5	61,5 ± 1,0	60,4 ± 0,5	**62,7** ± 1,3
haberman	44,3	43,6 ± 0,0	49,8 ± 1,8	43,6 ± 0,0	**51,6** ± 1,3
vehicle2	40	70,0 ± 1,3	67,6 ± 2,0	**70,3** ± 1,5	68,2 ± 1,9
vehicle1	41,9	61,9 ± 1,2	62,8 ± 1,3	61,3 ± 1,3	**63,2** ± 1,2
vehicle3	39,1	57,9 ± 1,3	61,2 ± 0,9	58,6 ± 1,1	**62,0** ± 1,0
glass0123vs456	87,6	89,7 ± 1,5	90,6 ± 0,9	89,6 ± 1,1	**91,0** ± 1,7
vehicle0	55,5	69,9 ± 0,9	**76,1** ± 1,8	70,0 ± 0,7	75,9 ± 2,1
ecoli1	80,8	89,2 ± 0,2	89,5 ± 0,5	89,3 ± 0,1	**89,9** ± 0,5
newthyroid2	**99,2**	98,6 ± 0,4	95,0 ± 1,4	98,9 ± 0,2	95,2 ± 1,1
newthyroid1	**99,2**	98,8 ± 0,1	95,8 ± 1,6	98,8 ± 0,1	95,1 ± 1,8
ecoli2	34,2	86,2 ± 0,4	86,1 ± 1,2	86,4 ± 0,2	**86,8** ± 1,7
segment0	88,1	**95,8** ± 0,4	90,6 ± 1,6	95,7 ± 0,6	89,1 ± 3,2
glass6	22,8	73,2 ± 4,3	**85,5** ± 2,5	78,9 ± 5,6	**85,5** ± 1,7
yeast3	85,5	**89,8** ± 0,4	88,7 ± 1,7	89,5 ± 0,4	88,7 ± 1,5
ecoli3	50,8	**86,3** ± 0,4	85,6 ± 1,8	86,2 ± 0,4	85,6 ± 2,2
page-blocks0	63,6	76,6 ± 2,7	**77,7** ± 2,1	75,4 ± 3,4	76,3 ± 1,2
yeast2vs4	67,3	86,7 ± 0,7	84,6 ± 1,2	**86,8** ± 0,7	86,2 ± 1,7
yeast05679vs4	61,9	73,4 ± 1,6	**78,1** ± 0,8	72,5 ± 1,1	78,0 ± 1,0
vowel0	83,9	**89,7** ± 0,2	83,2 ± 3,4	**89,7** ± 0,3	83,0 ± 2,6
glass2	10,8	35,7 ± 3,8	48,4 ± 2,5	38,3 ± 4,3	**50,3** ± 7,5
glass4	23,1	23,9 ± 6,4	72,9 ± 8,1	28,3 ± 8,4	**76,1** ± 8,2
ecoli4	68,7	90,6 ± 0,3	88,8 ± 3,0	**91,3** ± 1,1	90,5 ± 3,2
page-blocks1-3vs4	75,1	**89,4** ± 2,1	85,3 ± 4,6	**89,4** ± 2,1	85,4 ± 3,6
abalone9-18	58,7	75,7 ± 1,5	73,2 ± 3,5	**76,1** ± 1,3	73,0 ± 2,0
yeast1458vs7	45,8	50,9 ± 1,6	**57,8** ± 3,0	50,6 ± 1,3	55,3 ± 4,9
yeast2vs8	68,8	74,2 ± 1,6	73,8 ± 2,4	**74,8** ± 1,1	72,7 ± 2,1
yeast4	65,7	68,9 ± 0,8	**80,4** ± 1,6	68,5 ± 0,4	79,5 ± 1,4
yeast1289vs7	54,1	61,2 ± 0,3	**63,1** ± 2,6	61,2 ± 0,3	59,6 ± 2,0
yeast5	72,9	93,9 ± 0,6	93,6 ± 1,4	93,6 ± 0,9	**94,0** ± 0,7
ecoli0137vs26	0	52,3 ± 8,6	**74,0** ± 6,3	58,9 ± 9,2	72,0 ± 8,8
yeast6	51,2	77,3 ± 0,7	84,6 ± 2,0	77,8 ± 0,9	**85,2** ± 1,9
Среднее	58,3	74,5 ± 1,4	77,8 ± 2,2	75,0 ± 1,5	**77,9** ± 2,3
Улучшение	-	16,2	19,5	16,7	19,6

Настройка весов гибридным алгоритмом позволила улучшить среднюю геометрическую точность исходных нечетких классификаторов в среднем на 16 процентов. При последующей оптимизации термов качество возрастает еще на 3 процента.

Попарное сравнение критерием Уилкоксона результатов первой схемы с аналогами из таблицы 3.10 приведено в таблице 3.19. Если значение стандартизированной статистики

критерия (ССК) отрицательное, то результаты лучше у аналогов, в противном случае – у предлагаемых алгоритмов оптимизации нечетких классификаторов.

Таблица 3.19 – Попарное сравнение средней геометрической точности нечетких классификаторов с настройкой весов и комбинацией настройки весов и термов с аналогами

Алгоритм	Этап	Аналог	АЗ	ССК	НГ
ГП+ПЛ+ВИ	Настройка весов	Chi-3	0,017	-2,388	Отклоняется
		Chi-5	0,017	-2,388	Отклоняется
		Ishibuchi	0,649	0,456	Принимается
		E-алгоритм	<0,001	3,833	Отклоняется
		HFRBCS	0,001	-3,362	Отклоняется
	Настройка весов + оптимизация термов	Chi-3	0,162	-1,398	Принимается
		Chi-5	0,124	-1,540	Принимается
		Ishibuchi	0,087	1,712	Принимается
		E-алгоритм	<0,001	4,242	Отклоняется
		HFRBCS	0,005	-2,797	Отклоняется
ГП+ПЛ+СГ	Настройка весов	Chi-3	0,018	-2,372	Отклоняется
		Chi-5	0,034	-2,121	Отклоняется
		Ishibuchi	0,540	0,613	Принимается
		E-алгоритм	<0,001	3,896	Отклоняется
		HFRBCS	0,001	-3,323	Отклоняется
	Настройка весов + оптимизация термов	Chi-3	0,153	-1,430	Принимается
		Chi-5	0,140	-1,477	Принимается
		Ishibuchi	0,066	1,838	Принимается
		E-алгоритм	<0,001	4,226	Отклоняется
		HFRBCS	0,004	2,844	Отклоняется

Точность классификаторов, получаемая сразу после настройки весовых коэффициентов, уступает большинству рассматриваемых аналогичных классификаторов. После проведения оптимизации параметров результаты улучшаются: предлагаемый инструмент опережает E-алгоритм, показывает сопоставимую точность с Ishibuchi, Chi-3 и Chi-5, и уступает только иерархическим нечетким системам HFRBCS.

Таким образом, можно заключить, что при отсутствии этапа предобработки данных и использовании базы правил минимального объема, настройка весовых коэффициентов признаков должна сопровождаться оптимизацией параметров термов для достижения сопоставимой точности с большинством аналогичных алгоритмов.

3.5.2 Вторая схема эксперимента. Во второй схеме эксперимента изменено количество этапов и их порядок: первый этап так же заключался в формировании структуры алгоритмом экстремальных значений признаков классов, на втором осуществлялась оптимизация термов (Термы) алгоритмом из параграфа 2.2, на третьем проводилась настройка весовых коэффициентов признаков алгоритмом, описанном в 2.3 (термы + веса), на четвертом повторно оптимизировались параметры термов (термы + веса + термы). Параметры алгоритмов совпадали с приведенными в первой схеме эксперимента. Усредненные по 15 запускам результаты классификации представлены в таблице 3.20.

Таблица 3.20 – Средняя геометрическая точность нечетких классификаторов после проведения трех этапов оптимизации

Данные	ГП+ПЛ+ВИ			ГП+ПЛ+СГ		
	Термы	Термы + веса	Термы + веса + термы	Термы	Термы + веса	Термы + веса + термы
gl1	59,8 ± 3,1	61,0 ± 3,3	**62,8** ± 2,2	59,8 ± 2,5	60,6 ± 2,5	61,7 ± 3,2
ecl0/1	96,4 ± 1,5	96,5 ± 1,2	**96,6** ± 1,0	96,4 ± 1,1	**96,6** ± 1,2	95,7 ± 0,9
wis	94,8 ± 1,2	**95,2** ± 0,8	95,0 ± 0,8	94,6 ± 1,1	94,8 ± 1,0	**95,2** ± 0,7
pm	68,0 ± 1,7	69,4 ± 1,6	**70,1** ± 1,7	69,1 ± 1,8	69,6 ± 1,7	68,7 ± 1,5
gl0	71,0 ± 2,4	72,6 ± 1,9	72,8 ± 2,9	70,3 ± 1,8	72,2 ± 2,3	**73,4** ± 2,2
yst1	64,5 ± 1,7	65,3 ± 1,5	65,5 ± 1,8	65,4 ± 1,2	**66,0** ± 1,3	64,8 ± 1,6
hbr	61,2 ± 2,6	61,0 ± 2,2	62,0 ± 2,8	61,9 ± 2,7	**62,2** ± 2,6	61,7 ± 1,9
vhc2	67,8 ± 2,3	70,2 ± 2,5	68,7 ± 3,0	69,3 ± 3,1	**71,1** ± 3,4	68,4 ± 3,2
vhc1	63,4 ± 1,6	65,0 ± 1,4	64,6 ± 1,0	64,3 ± 1,3	**65,2** ± 1,1	64,4 ± 1,4
vhc3	65,2 ± 1,2	65,2 ± 0,9	**65,8** ± 0,8	64,4 ± 1,2	65,0 ± 0,9	65,2 ± 1,1
gl0123/456	88,1 ± 2,3	90,3 ± 1,7	**91,5** ± 1,5	87,6 ± 1,7	88,6 ± 1,6	89,8 ± 1,8
vhc0	76,2 ± 2,7	**79,7** ± 2,9	77,5 ± 3,0	74,1 ± 2,3	76,6 ± 1,8	78,1 ± 2,6
ecl1	86,9 ± 1,3	87,2 ± 1,0	**88,4** ± 0,8	87,8 ± 1,5	88,2 ± 1,2	87,1 ± 1,0
nwth2	95,2 ± 1,8	94,4 ± 1,6	94,7 ± 2,0	95,5 ± 1,5	**95,6** ± 1,5	94,9 ± 1,1
nwth1	**95,2** ± 1,5	94,6 ± 1,5	94,5 ± 2,0	94,7 ± 2,2	94,7 ± 2,2	94,8 ± 2,1
ecl2	87,3 ± 2,0	87,1 ± 1,9	86,2 ± 1,9	86,7 ± 1,4	87,0 ± 1,3	**87,5** ± 1,7
sgm0	82,9 ± 2,3	86,2 ± 2,1	86,7 ± 3,6	83,7 ± 2,1	**87,0** ± 1,7	83,1 ± 3,8
gl6	89,2 ± 1,6	89,7 ± 2,3	89,9 ± 2,9	89,8 ± 1,9	**90,3** ± 2,1	89,1 ± 3,0
yst3	82,9 ± 3,9	85,2 ± 2,9	**87,4** ± 2,3	83,5 ± 2,8	86,2 ± 2,8	84,5 ± 3,6
ecl3	85,7 ± 2,0	85,6 ± 1,3	**85,9** ± 1,4	85,1 ± 1,6	85,5 ± 1,9	85,1 ± 2,3
pb0	75,4 ± 3,2	78,3 ± 2,1	**79,5** ± 1,4	76,7 ± 2,4	78,4 ± 1,7	76,3 ± 3,3
yst2/4	83,4 ± 1,9	84,2 ± 1,9	**84,7** ± 1,7	83,8 ± 2,4	84,2 ± 2,2	84,5 ± 2,3
yst05679/4	74,5 ± 2,4	75,1 ± 2,1	**77,9** ± 1,2	76,7 ± 2,7	76,5 ± 2,7	77,1 ± 1,6
vwl0	79,1 ± 3,8	**83,6** ± 3,2	82,5 ± 3,2	77,1 ± 4,0	80,6 ± 3,6	82,3 ± 3,5
gl2	56,7 ± 4,4	58,8 ± 6,2	**59,2** ± 4,1	56,4 ± 5,9	58,8 ± 4,8	54,2 ± 6,6
gl4	82,0 ± 3,9	**83,7** ± 3,4	79,4 ± 6,1	79,4 ± 6,4	82,0 ± 4,5	77,6 ± 7,1
ecl4	88,3 ± 2,9	89,0 ± 2,6	**89,7** ± 2,5	87,8 ± 3,5	87,8 ± 3,4	88,8 ± 2,7
pb1-3/4	80,1 ± 5,2	83,3 ± 4,9	86,9 ± 4,0	80,8 ± 3,0	83,5 ± 2,6	**87,4** ± 3,4
ab9/18	69,3 ± 4,2	69,9 ± 3,6	70,5 ± 3,7	70,7 ± 3,1	**72,5** ± 2,1	72,3 ± 3,3
yst1458/7	54,3 ± 4,9	55,0 ± 5,1	55,1 ± 3,7	52,6 ± 4,9	53,5 ± 5,2	**56,5** ± 4,1
yst2/8	72,7 ± 2,8	73,4 ± 3,2	**73,9** ± 1,9	71,6 ± 3,4	71,8 ± 3,1	71,8 ± 3,3
yst4	79,8 ± 1,6	**80,4** ± 1,4	**80,4** ± 1,2	79,3 ± 1,8	79,5 ± 1,5	79,1 ± 1,0
yst1289/7	61,2 ± 4,9	60,9 ± 5,4	60,9 ± 4,0	61,4 ± 3,1	62,5 ± 3,0	**63,0** ± 4,1
yst5	92,8 ± 1,6	93,7 ± 1,0	**94,0** ± 1,4	93,2 ± 1,4	93,8 ± 1,2	93,0 ± 1,9
ecl0137/26	68,9 ± 6,5	70,4 ± 5,0	**72,1** ± 2,8	67,7 ± 5,5	67,9 ± 5,5	67,9 ± 4,5
yst6	84,5 ± 1,8	**85,4** ± 1,7	84,8 ± 1,6	83,8 ± 2,1	83,8 ± 2,2	82,8 ± 2,1
Среднее	77,3 ± 2,7	78,5 ± 2,5	**78,8** ± 2,3	77,3 ± 2,6	78,3 ± 2,4	78,0 ± 2,7
Улучшение	19,0	20,2	**20,5**	19,0	20,0	19,7

Использование настройки весовых коэффициентов признаков после оптимизации параметров термов позволило улучшить качество классификации на один процент. Повторение оптимизации термов поспособствовало увеличению точности не на всех наборах данных, а для некоторых даже ухудшило точность.

Наибольшей средней геометрической точности удалось достигнуть после применения оптимизации термов и настройки весов алгоритмом, популяция которого была создана с помощью расчета взаимной информации. По сравнению с исходным нечетким классификатором точность выросла на 20 процентов.

Сравнение результатов второй схемы эксперимента с аналогами из таблицы 3.10 критерием Уилкоксона приведено в таблице 3.21. Отрицательное значение стандартизированной статистики критерия (ССК) демонстрирует, что лучший результат демонстрирует аналогичный алгоритм, положительное свидетельствует о превосходстве предлагаемых алгоритмов оптимизации нечетких классификаторов.

Таблица 3.21 – Попарное сравнение средней геометрической точности нечетких классификаторов с комбинацией настройки термов и весов с аналогами

Алгоритм	Этап	Аналог	АЗ	ССК	НГ
ГП+ПЛ+ВИ	Оптимизация термов	Chi-3	0,078	-1,760	Принимается
		Chi-5	0,077	-1,767	Принимается
		Ishibuchi	0,102	1,634	Принимается
		E-алгоритм	<0,001	3,849	Отклоняется
		HFRBCS	0,002	-3,095	Отклоняется
	Оптимизация термов + настройка весов	Chi-3	0,265	-1,115	Принимается
		Chi-5	0,215	-1,241	Принимается
		Ishibuchi	0,008	2,655	Отклоняется
		E-алгоритм	<0,001	4,140	Отклоняется
		HFRBCS	0,010	-2,592	Отклоняется
	Оптимизация термов + настройка весов + оптимизация термов	Chi-3	0,572	-0,566	Принимается
		Chi-5	0,396	-0,848	Принимается
		Ishibuchi	0,005	2,828	Отклоняется
		E-алгоритм	<0,001	4,195	Отклоняется
		HFRBCS	0,025	-2,247	Отклоняется
ГП+ПЛ+СГ	Оптимизация термов	Chi-3	0,087	-1,712	Принимается
		Chi-5	0,106	-1,618	Принимается
		Ishibuchi	0,126	1,532	Принимается
		E-алгоритм	<0,001	3,802	Отклоняется
		HFRBCS	0,002	3,048	Отклоняется
	Оптимизация термов + настройка весов	Chi-3	0,245	-1,163	Принимается
		Chi-5	0,215	-1,241	Принимается
		Ishibuchi	0,019	2,342	Отклоняется
		E-алгоритм	<0,001	4,038	Отклоняется
		HFRBCS	0,007	-2,702	Отклоняется
	Оптимизация термов + настройка весов + оптимизация термов	Chi-3	0,182	-1,335	Принимается
		Chi-5	0,182	-1,335	Принимается
		Ishibuchi	0,028	2,199	Отклоняется
		E-алгоритм	<0,001	3,943	Отклоняется
		HFRBCS	0,004	2,914	Отклоняется

Сравнение результатов второй схемы эксперимента с аналогами также показало, что комбинация этапов оптимизации термов и настройки параметров позволило получить более точный классификатор, чем при использовании только оптимизации параметров термов. Повторная оптимизация термов улучшила среднюю геометрическую точность только при

генерации исходной популяции весов на основе взаимной информации. Возможно, увеличение количества итераций позволило бы поднять качество классификации, так как оптимизация параметров на 500 итерациях показывает результат существенно хуже, чем на 1000 итерациях (подраздел 3.3).

3.5.3 Сравнение этапов оптимизации нечеткого классификатора несбалансированных данных. Для выбора лучшей методики оптимизации нечеткого классификатора проведено попарное сравнение результатов двух схем эксперимента между собой (таблица 3.22). Настройка весовых коэффициентов признаков обозначена как «веса», оптимизация параметров термов как «термы». Положительное значение стандартизированной статистики критерия (ССК) показывает, что результаты первого этапа лучше, чем результаты второго.

Таблица 3.22 – Попарное сравнение средней геометрической точности после проведения различных этапов оптимизации нечеткого классификатора гибридным метаэвристическим алгоритмом

Сравниваемые этапы		ГП+ПЛ+ВИ			ГП+ПЛ+СГ		
№ 1	№ 2	АЗ	ССК	НГ	АЗ	ССК	НГ
Веса	Веса + термы	0,129	-1,516	Прин.	0,116	-2,419	Прин.
Термы	Термы + веса	< 0,001	-4,478	Откл.	< 0,001	-5,020	Откл.
Термы + веса	Термы + веса + термы	0,020	-2,317	Откл.	0,314	1,007	Прин.
Веса	Термы	0,615	0,503	Прин.	0,519	-0,644	Прин.
Веса + термы	Термы + веса	0,752	-0,316	Прин.	0,912	-0,110	Прин.
Веса + термы	Термы + веса + термы	0,239	-1,178	Прин.	0,894	0,134	Прин.

По значениям ССК можно сделать вывод, что комбинация этапов позволяет получить большую среднюю геометрическую точность, чем применение только одного из этапов.

При генерации популяции весов признаков на основе взаимной информации лучшие результаты были продемонстрированы классификаторами при трехэтапной схеме – оптимизации термов, настройке весов и повторной оптимизации параметров термов. При случайной генерации повторная настройка функций принадлежности ухудшила результаты. Эти утверждения также подтверждаются сравнением этапов критерием Фридмана (таблица 3.23).

Таблица 3.23 – Сравнение различных этапов оптимизации нечеткого классификатора между собой критерием Фридмана

Этапы	ГП+ПЛ+ВИ	ГП+ПЛ+СГ
Веса	2,88	2,79
Веса + термы	3,08	3,35
Термы	2,14	2,35
Термы + веса	3,14	3,53
Термы + веса + термы	3,76	2,99
Асимптотическая значимость	0,001	0,014
Нулевая гипотеза	Отклоняется	Отклоняется

3.5.4 Сравнение результатов настройки весов в зависимости от выбранного способа генерации первичной популяции. Для определения, какой из двух используемых способов генерации популяции весов является более эффективным, было проведено попарное сравнение результатов совпадающих этапов критерием Уилкоксона. Так как этап оптимизации параметров, проводимый первым во второй схеме эксперимента, не связан с генерацией популяции весов, он не был включен в сравнение. Результаты анализа приведены в таблице 3.24. Положительное значение стандартизированной статистики критерия показывает, что на данном этапе результаты, полученные после генерации популяции весов на основе взаимной информации, были лучше, чем после случайной генерации.

Таблица 3.24 – Сравнение результатов оптимизации нечеткого классификатора при использовании двух способов генерации популяции весовых коэффициентов признаков

Этапы	Асимптотическая значимость	Стандартизированная статистика критерия	Нулевая гипотеза
Веса	0,381	-0,876	Принимается
Веса + термы	0,712	-0,369	Принимается
Термы + веса	0,671	0,424	Принимается
Термы + веса + термы	0,003	2,930	Отклоняется

Существенное различие в результатах было выявлено только при трехэтапной схеме оптимизации (термы + веса + термы). В таблице 3.25 приведен результат ранжирования всех достигнутых результатов с помощью критерия Фридмана

Таблица 3.25 – Упорядоченные по возрастанию средние ранги построенных нечетких классификаторов

Этапы и способ генерации популяции весов	Средний ранг
Термы (любая генерация)	4,11
Веса, случайная генерация	5,15
Термы + веса + термы, случайная генерация	5,18
Веса, взаимная информация	5,28
Веса + термы, случайная генерация	5,88
Веса + термы, взаимная информация	6,06
Термы + веса, случайная генерация	6,08
Термы + веса, взаимная информация	6,14
Термы + веса + термы, взаимная информация	7,01

При таком способе сравнения видно, что генерацию популяции на основе взаимной информации можно считать более предпочтительной, так как одинаковые этапы с этим способом формирования популяции получают больший ранг, чем при случайной генерации.

3.5.5 Анализ дополнительных метрик качества нечетких классификаторов при наличии этапа настройки весов. Помимо средней геометрической точности, которую можно рассматривать как обобщающий критерий качества, важно проанализировать результаты по прочим метрикам – точности отдельных классов и количеству признаков. В приложении А приведены проценты правильной классификации положительного и отрицательного классов,

полученные после оптимизации. В таблице 3.26 приведены усредненные результаты по всем наборам данных и средний ранг Фридмана. Результаты упорядочены по возрастанию ранга.

Таблица 3.26 – Средний процент правильной классификации наименьшего и наибольшего классов после проведения оптимизации и средний ранг по критерию Фридмана

Точность положительного класса			Точность отрицательного класса		
Этап и способ генерации	Среднее	Ранг	Этап и способ генерации	Среднее	Ранг
Веса, случайная генерация	73,5 ± 2,2	4,64	Термы, взаимная информация	80,0 ± 3,1	4,03
Термы, случайная генерация	77,3 ± 4,3	4,72	Термы, случайная генерация	80,0 ± 2,9	4,11
Веса, взаимная информация	73,6 ± 1,9	4,86	Термы + веса, случайная генерация	81,0 ± 2,7	5,60
Термы, взаимная информация	77,4 ± 4,2	5,15	Веса + термы, случайная генерация	80,7 ± 2,3	5,35
Термы + веса + термы, случайная генерация	77,9 ± 4,1	5,72	Термы + веса + термы, случайная генерация	80,8 ± 2,8	5,35
Термы + веса + термы, взаимная информация	78,0 ± 3,7	5,76	Веса + термы, взаимная информация	80,9 ± 2,1	5,71
Веса + термы, случайная генерация	77,4 ± 3,6	5,88	Веса, случайная генерация	81,7 ± 1,2	5,83
Веса + термы, взаимная информация	77,5 ± 3,4	5,92	Термы + веса, взаимная информация	81,3 ± 2,6	5,86
Термы + веса, взаимная информация	78,1 ± 3,9	6,13	Веса, взаимная информация	81,4 ± 1,0	6,00
Термы + веса, случайная генерация	78,1 ± 3,9	6,22	Термы + веса + термы, взаимная информация	81,9 ± 2,5	7,17

При сравнении критерием Фридмана достигнутых значений точности положительного класса асимптотическая значимость оказывается равной 0,175, что позволяет принять нулевую гипотезу о том, что распределения результатов одинаковы. В случае точности отрицательного класса нулевая гипотеза отклоняется (асимптотическая значимость равна 0,001). Алгоритм настройки весовых коэффициентов классов сфокусировался на отрицательном классе; комбинация с оптимизацией параметров термов поспособствовала смещению фокуса на положительный класс, однако в некоторых случаях это негативно отразилось на качестве распознавания отрицательного класса. В целом, результаты сравнения согласуются с итогами сравнения по средней геометрической точности (таблица 3.24): лучшие значения одновременно по двум классам показывают комбинации «термы + веса» и «термы + веса + термы» с генерацией популяции весов на основе взаимной информации.

Самостоятельно алгоритм настройки признаков повышает точность наименьшего класса примерно на 18 процентов, точность наибольшего – примерно на 3 процента.

В процессе эксперимента фиксировались суммы весовых коэффициентов признаков для каждого набора данных. Если бы алгоритм посчитал все признаки равнозначными, то после проведения процедуры нормализации сумма весов равнялась бы исходному количеству признаков. По таблице 3.27 видно, что это не так – суммы значительно меньше, чем число

признаков в данных (*F*). Можно заключить, что происходит и настройка весов, и исключение некоторых признаков после получения низкого веса (порог равнялся 0,2 до нормализации).

Таблица 3.27 – Исходное количество признаков в наборах данных и сумма весовых коэффициентов после их настройки гибридным алгоритмом из метаэвристик «гравитационный поиск» и «прыгающие лягушки»

Данные	*F*	Веса	Веса	Термы + веса	Термы + веса
		ГП+ПЛ+ВИ	ГП+ПЛ+СГ	ГП+ПЛ+ВИ	ГП+ПЛ+СГ
glass1	9	2,2 ± 0,2	2,4 ± 0,2	3,9 ± 0,4	4,2 ± 0,3
ecoli0vs1	7	2,7 ± 0,2	3,2 ± 0,3	2,8 ± 0,3	3,5 ± 0,3
wisconsin	9	4,3 ± 0,3	3,7 ± 0,2	4,7 ± 0,3	4,2 ± 0,4
pima	8	2,1 ± 0,2	2,1 ± 0,1	3,1 ± 0,4	3,5 ± 0,3
glass0	9	2,5 ± 0,1	2,9 ± 0,2	3,8 ± 0,4	4,2 ± 0,5
yeast1	8	2,6 ± 0,2	2,8 ± 0,2	3,5 ± 0,3	3,7 ± 0,4
haberman	3	1,9 ± 1,2	1,4 ± 0,1	1,4 ± 0,1	1,5 ± 0,2
vehicle2	18	1,9 ± 0,2	2,5 ± 0,3	5,5 ± 0,7	6,1 ± 0,8
vehicle1	18	2,3 ± 0,2	2,3 ± 0,2	6,3 ± 0,8	6,4 ± 0,7
vehicle3	18	1,5 ± 0,1	1,6 ± 0,3	6,2 ± 0,9	6,6 ± 0,8
glass0123/456	9	2,9 ± 0,2	3,2 ± 0,3	3,8 ± 0,4	4,1 ± 0,5
vehicle0	18	3,2 ± 0,5	3,5 ± 0,5	5,5 ± 0,8	6,1 ± 1,0
ecoli1	7	1,4 ± 0,1	1,7 ± 0,1	2,9 ± 0,3	3,1 ± 0,3
newthyroid2	5	2,7 ± 0,1	2,4 ± 0,2	2,7 ± 0,1	2,5 ± 0,2
newthyroid1	5	2,3 ± 0,1	2,3 ± 0,1	2,7 ± 0,2	2,5 ± 0,2
ecoli2	7	3,0 ± 0,2	3,4 ± 0,2	3,2 ± 0,1	3,4 ± 0,3
segment0	19	3,5 ± 0,2	3,7 ± 0,3	5,7 ± 0,8	5,6 ± 0,8
glass6	9	2,0 ± 0,1	1,9 ± 0,2	3,8 ± 0,3	4,0 ± 0,3
yeast3	8	2,0 ± 0,3	2,8 ± 0,4	3,1 ± 0,4	3,3 ± 0,2
ecoli3	7	2,8 ± 0,1	3,1 ± 0,1	3,2 ± 0,2	3,4 ± 0,3
page-blocks0	10	1,4 ± 0,2	1,3 ± 0,2	3,8 ± 0,4	3,8 ± 0,4
yeast2vs4	8	2,4 ± 0,1	2,8 ± 0,1	3,1 ± 0,4	3,7 ± 0,3
yeast05679vs4	8	1,3 ± 0,1	1,5 ± 0,1	3,2 ± 0,3	3,8 ± 0,4
vowel0	13	4,1 ± 0,3	5,1 ± 0,4	4,4 ± 0,5	4,8 ± 0,5
glass2	9	2,0 ± 0,2	2,1 ± 0,2	3,8 ± 0,2	4,0 ± 0,4
glass4	9	2,2 ± 0,3	2,2 ± 0,3	3,7 ± 0,3	4,0 ± 0,3
ecoli4	7	2,4 ± 0,1	2,9 ± 0,2	3,1 ± 0,2	3,3 ± 0,3
page-bl.13vs2	10	3,3 ± 0,3	3,4 ± 0,3	4,4 ± 0,5	4,2 ± 0,4
abalone9-18	7	2,3 ± 0,1	2,4 ± 0,2	3,3 ± 0,3	3,6 ± 0,4
yeast1458vs7	8	3,4 ± 0,2	3,4 ± 0,3	3,2 ± 0,3	3,6 ± 0,3
yeast2vs8	8	1,8 ± 0,1	2,2 ± 0,2	3,2 ± 0,4	4,0 ± 0,3
yeast4	8	2,6 ± 0,2	2,8 ± 0,2	3,4 ± 0,5	3,7 ± 0,5
yeast1289vs7	8	2,2 ± 0,1	2,4 ± 0,2	3,4 ± 0,2	3,8 ± 0,3
yeast5	8	3,7 ± 0,2	3,9 ± 0,2	3,3 ± 0,4	3,8 ± 0,3
ecoli0137vs26	7	2,4 ± 0,2	2,4 ± 0,2	3,0 ± 0,1	3,3 ± 0,4
yeast6	8	3,1 ± 0,3	3,6 ± 0,2	3,3 ± 0,4	3,7 ± 0,4
Среднее	9,4	2,5 ± 0,2	2,7 ± 0,2	3,7 ± 0,4	4,0 ± 0,4

После сочетания оптимизации термов и настройки весов значения сумм весов оказываются большими, чем при использовании только настройки весовых коэффициентов. Учитывая, что и лучшие точности были получены при комбинации «термы + веса», можно заключить, что при применении в качестве способа генерации структуры алгоритма экстремальных значений признаков классов важно сначала найти оптимальное положение

термов, а затем настраивать признаки. После чего стоит провести повторную оптимизацию параметров, на случай если в каких-либо признаках были шумы.

3.5.6 Подтверждение эффективности гибридного метаэвристического алгоритма при настройке весовых коэффициентов признаков нечеткого классификатора. Последняя часть эксперимента по проверке эффективности инструмента настройки весовых коэффициентов признаков была посвящена сравнению результатов при применении в качестве инструмента оптимизации гибридного алгоритма на основе метаэвристик «гравитационный поиск» и «прыгающие лягушки» и простейшего алгоритма оптимизации – случайного поиска. В течение 500 итераций случайный поиск генерировал популяцию из 40 векторов весовых коэффициентов признаков; лучший полученный результат подавался на выход алгоритма. Средняя геометрическая точность после настройки весов случайным поиском приведена в таблице 3.28.

Таблица 3.28 – Результаты построения нечетких классификаторов несбалансированных данных после применения настройки весов признаков случайным поиском

Данные	*GM*	Данные	*GM*	Данные	*GM*
glass1	56,8 ± 3,8	ecoli1	88,1 ± 1,3	glass2	17,6 ± 4,0
ecoli0vs1	94,4 ± 2,6	newthyroid2	95,5 ± 1,7	glass4	23,2 ± 5,4
wisconsin	84,4 ± 1,8	newthyroid1	95,6 ± 2,6	ecoli4	82,8 ± 5,0
pima	63,9 ± 1,5	ecoli2	81,1 ± 3,9	page-bl.1-3vs4	75,7 ± 7,8
glass0	75,6 ± 3,4	segment0	70,9 ± 3,2	abalone9-18	67,8 ± 4,2
yeast1	57,0 ± 3,4	glass6	68,6 ± 3,3	yeast1458vs7	31,4 ± 14,7
haberman	40,8 ± 1,0	yeast3	87,8 ± 1,5	yeast2vs8	43,6 ± 16,0
vehicle2	35,4 ± 5,3	ecoli3	83,4 ± 4,4	yeast4	72,6 ± 5,0
vehicle1	44,1 ± 2,5	page-blocks0	44,9 ± 3,4	yeast1289vs7	52,9 ± 6,1
vehicle3	39,8 ± 2,6	yeast2vs4	80,4 ± 4,6	yeast5	91,0 ± 2,1
gl.0123vs456	84,6 ± 3,9	yeast05679vs4	71,9 ± 5,3	ecoli0137vs26	60,6 ± 23,3
vehicle0	66,0 ± 2,0	vowel0	85,1 ± 2,5	yeast6	70,4 ± 7,0
Среднее			66,3 ± 4,8		

Чтобы подтвердить эффективность использования для настройки весов гибридного алгоритма из метаэвристик «гравитационный поиск» и «прыгающие лягушки», достаточно сравнить результаты нечетких классификаторов после проведения настройки весовых коэффициентов признаков из первой схемы эксперимента (таблица 3.17) до оптимизации параметров термов и результаты случайного поиска. Сравнение непараметрическим критерием Уилкоксона продемонстрировало, что есть существенные различия в точности между алгоритмами (таблица 3.29). Положительное значение стандартизированной статистики критерия свидетельствует о превосходстве предложенного гибридного алгоритма над случайным поиском (СП).

Таблица 3.29 – Сравнение результатов гибридного алгоритма и случайного поиска критерием Уилкоксона при настройке весовых коэффициентов признаков нечетких классификаторов несбалансированных данных

Алгоритмы	Асимптотическая значимость	Стандартизированная статистика критерия	Нулевая гипотеза
ГП+ПЛ+ВИ и СП	< 0,001	4,101	Отклоняется
ГП+ПЛ+СГ и СП	< 0,001	4,368	Отклоняется

Сравнение подтвердило наличие превосходства результатов гибридного алгоритма над случайным поиском при настройке весов признаков нечеткого классификатора.

3.6 Выводы

Для подтверждения эффективности разработанных алгоритмов были проведены эксперименты на 36 несбалансированных наборах данных. Результаты экспериментов позволяют сделать следующие выводы.

1. Алгоритм формирования структуры нечеткого классификатора несбалансированных данных на основе итерационного добавления правил метаэвристикой «прыгающие лягушки» позволил улучшить среднюю геометрическую точность на 23 процента, точность положительного класса на 25 процентов и точность отрицательного класса на 9 процентов по сравнению с результатами до добавления правил (результаты по лучшим базам правил, таблица 3.9). Разработанный алгоритм показал лучшее среднее геометрическое значение точности, чем общеизвестные алгоритмы Ishibuchi и E-алгоритм, хотя первый из них использует генерацию дополнительных экземпляров положительного класса, а второй позиционируется как специальный алгоритм для несбалансированных данных. При сравнении с результатами алгоритмов Chi-3, Chi-5 и HFRBCS, применяющих этап исправления данных, получена сопоставимая точность.

Далее представлен перечень достоинств алгоритма формирования структуры классификатора на основе «прыгающих лягушек»:

– алгоритм позволяет нечеткому классификатору показать превосходящую или сопоставимую точность с общеизвестными аналогами, которые используют предобработку данных;

– инструмент не требует применения методов исправления данных;

– алгоритм может работать в комбинации с любым алгоритмом генерации первичной структуры нечеткого классификатора;

– в комбинации с алгоритмом экстремальных значений признаков классов предложенный алгоритм добавления правил позволяет достичь качественных результатов даже при добавлении

небольшого количества правил, которое позволяет классификатору оставаться интерпретируемым. В выполненном эксперименте объем базы не превышал девяти правил, в то время как приведенные аналоги используют как минимум 60 правил.

Количество добавляемых правил и фитнес-функция должны определяться пользователем алгоритма в зависимости от требований к конечной модели.

2. Алгоритм настройки параметров термов нечеткого классификатора, представляющий собой гибрид между метаэвристиками «гравитационный поиск» и «прыгающие лягушки», показал наличие статистически значимой разницы в средней геометрической точности при сравнении с исходными метаэвристиками. По сравнению с точностью до оптимизации параметров термов предложенному алгоритму удалось увеличить среднюю геометрическую точность и точность положительного класса на 24 процента, точность отрицательного класса на 9 процентов.

Классификаторы, построенные алгоритмом экстремальных значений признаков классов, после настройки параметров функций принадлежности продемонстрировали результаты, превосходящие Chi-3, Chi-5, Ishibuchi и E-алгоритм, а также сопоставимую среднюю геометрическую точность по сравнению с HFRBCS, хотя аналоги используют большее количество правил и предобработку данных (за исключением E-алгоритма).

3. Алгоритм настройки весовых коэффициентов признаков нечеткого классификатора, основанный на гибриде метаэвристик «гравитационный поиск» и «прыгающие лягушки», по сравнению с точностью до введения весов позволил улучшить среднюю геометрическую точность на 16 процентов, точность положительного и отрицательного классов на 18 и 3 процента соответственно. Проверены два способа генерации популяции весовых коэффициентов признаков; генерация на основе взаимной информации в целом по эксперименту продемонстрировала результаты лучше, чем случайная генерация. Продемонстрировано, что предложенный инструмент сокращает число признаков в наборе данных. В сравнении с аналогами алгоритм показывает лучшую точность, чем E-алгоритм, сопоставимую точность с Ishibuchi, и уступает алгоритмам Chi-3, Chi-5 и HFRBCS.

При последовательном выполнении этапов оптимизации параметров термов, настройки весов признаков и повторной оптимизации термов, увеличение средней геометрической точности составило 20 процентов. Предложенная комбинация этапов показала большую среднюю геометрическую точность по сравнению с Ishibuchi и E-алгоритмом, сопоставимую точность с Chi-3 и Chi-5, но уступила алгоритму HFRBCS, создающего нечеткие классификаторы с более сложной структурой. Следовательно, после оптимизации параметров термов и весовых коэффициентов признаков гибридом на основе метаэвристик «гравитационный поиск» и «прыгающие лягушки», нечеткие классификаторы способны демонстрировать сопоставимое или

превосходящее значение средней геометрической точности всего на двух правилах по сравнению с большинством рассматриваемых аналогов, использующих большее число правил и алгоритм добавления экземпляров положительного класса SMOTE для исправления дисбаланса.

4. Проведено исследование метрик качества классификации на данных с большим коэффициентом дисбаланса (коэффициент дисбаланса от 9 до 41). Предложенная метрика, состоящая из совмещения средней геометрической точности и общей точности, показала лучшие результаты по соотношению точности положительного и отрицательного классов при коэффициенте приоритета γ, равного 0,25.

Глава 4. Практическое применение результатов диссертационного исследования

Результаты диссертационной работы были использованы при разработке программного обеспечения совместно с сотрудниками ОГАУЗ «Родильный дом №1», расположенного в городе Томск. До настоящего времени наиболее тяжелым осложнением беременности и родов во всем мире являются акушерские кровотечения. Летальность во время беременности и родов от массивной кровопотери и геморрагического шока составляет от 16 до 36% случаев материнской смертности в разных регионах мира [146] и 18-23% в России [147, 148]. Более 70% всех кровотечений в акушерстве относятся к послеродовым кровотечениям [146]. Среди факторов риска развития послеродовых кровотечений выделяют нарушения системы гемостаза – существовавшие ранее заболевания (болезнь Виллебранда, гемофилия), приобретенные коагулопатии (тромбоцитопении), ДВС-синдром, применение антикоагулянтов [149]. Некоторые из этих нарушений существуют и до беременности, но с течением беременности усугубляются, другие развиваются во время беременности. В любом случае, выявление женщин с нарушениями свертывания крови должно проводиться на ранних сроках беременности, и в дальнейшем такие пациентки требуют постоянного контроля со стороны акушеров и гематологов.

Порядок оказания медицинской помощи по профилю «акушерство и гинекология» (утвержден приказом Министерства здравоохранения Российской Федерации от 20 октября 2020 года № 1130н [150]) предусматривает, помимо клоттинговых тестов, проведение тромбоэластографии (электрокоагулографии) для контроля гемостаза. Но понимание значения этого метода исследования системы гемостаза, по мнению ряда авторов [151], есть далеко не всегда. Возможно, это связано отчасти с проблемой интерпретации результатов исследования клиницистами. Одним из решений указанной проблемы является программный классификатор анализа данных исследования системы гемостаза, призванный оказать помощь практическим врачам в оценке результатов и, соответственно, ускорить процесс диагностики и лечения.

Целью разработки нечетких классификаторов для оценки системы свертывания крови у беременных женщин являлось упрощение процесса работы врача клинико-диагностической лаборатории.

4.1 Описание данных для классификации

Для сбора данных были проведены исследования на анализаторе реологических свойств крови АРП-01 «МЕДНОРД». Материалом для исследования служила цельная венозная кровь беременных женщин на различных строках гестации с предварительно выявленными нарушениями свертывающей системы крови на скрининговых этапах обследования. В процессе

регистрации определялись следующие реологические показатели (признаки) [152]: A_f – начальный показатель агрегатного состояния крови; A_r – интенсивность спонтанной агрегации тромбоцитов; r – период реакции; k – время образования сгустка; AM – фибрин-тромбоцитарная константа крови; T – время формирования фибрин-тромбоцитарной структуры сгустка; F – суммарный показатель ретракции и спонтанного лизиса сгустка. Описание типа данных и области определение признаков приведено в таблице 4.1.

Таблица 4.1 – Описание входных признаков

Признак	Обозначение	Тип	Область определения
Возраст	V	Целочисленный	[15; 41]
Срок беременности	S	Вещественный	[6; 40]
Начальный показатель агрегатного состояния крови	A_f	Целочисленный	[47; 91]
Интенсивность спонтанной агрегации тромбоцитов	A_r	Целочисленный	[-19; -2]
Период реакции	r	Вещественный	[0,6; 16]
Время образования сгустка	k	Вещественный	[0,9; 10,5]
Фибрин-тромбоцитарная константа крови	AM	Целочисленный	[368; 995]
Время формирования фибрин-тромбоцитарной структуры сгустка	T	Вещественный	[18,3; 94,9]
Суммарный показатель ретракции и спонтанного лизиса сгустка	F	Вещественный	[0; 88,5]

Полученные значения показателей сравнивались с таковыми для здоровой группы лиц, и по характеру их изменения производилась оценка состояния системы гемостаза по трем составляющим: структурные характеристики, хронометрические характеристики, общее состояние системы свертывания крови [153, 154].

Для всех трех характеристик было сформировано по одному набору данных. Каждый набор состоит из 393 записей пациенток, состоящих на наблюдении в ОГАУЗ «Родильный дом №1» г. Томска. Врачом клинико-диагностической лаборатории определены классы для каждой записи. В качестве класса выделялись три основных состояния свертывающей системы: нормальное (класс 1), гиперкоагуляционный сдвиг (класс 2), который характеризуется укорочением хронометрических показателей (r, k, T) и увеличением амплитудных характеристик, и гипокоагуляционный сдвиг (класс 0), выявляемый при удлинении хронометрических характеристик и снижении амплитудных параметров.

В таблице 4.2 приведено распределение экземпляров по классам в данных по каждой характеристике системы свертывания крови и рассчитан коэффициент дисбаланса IR (отношение числа образцов наибольшего класса к числу образцов наименьшего).

Таблица 4.2 – Количество экземпляров и коэффициент дисбаланса в наборах данных

Набор данных	Гипокоагуляция (класс 0)	Нормальное состояние (класс 1)	Гиперкоагуляция (класс 1)	*IR*
Общее состояние системы свертывания	41	208	144	5,1
Хронометрические характеристики	54	229	110	4,2
Структурные характеристики	23	243	127	10,6

Все состояния требуют определенную реакцию со стороны врача-клинициста для коррекции сдвигов и предотвращения потенциальных кровотечений или тромбозов. Однако определение типа изменения зачастую представляет определенную сложность для врача. Для облегчения работы был предложен способ оценки системы гемостаза на основе реологических свойств крови в виде нечеткого классификатора [155].

4.2 Построение нечеткого классификатора для оценки системы свертывания крови

Построение нечеткого классификатора было осуществлено следующими инструментами:

– алгоритмом генерации структуры на основе экстремальных значений признаков классов (АЭПК);

– алгоритмом оптимизации параметров термов комбинацией метаэвристик «гравитационный поиск» и «прыгающие лягушки» (параграф 2.2);

– алгоритмом формирования структуры на основе итерационного добавления правил метаэвристикой «прыгающие лягушки» (параграф 2.1);

– алгоритмом настройки весовых коэффициентов комбинацией метаэвристик «гравитационный поиск» и «прыгающие лягушки» параграф (2.3).

В нечетких правилах были использованы функции принадлежности трапециевидной формы, показавшие лучшие результаты на этапе генерации структуры с помощью АЭПК. В таблице 4.3 приведены параметры инструментов оптимизации.

Таблица 4.3 – Параметры алгоритмов

Алгоритм	Фитнес-функция	Количество итераций	Количество агентов	Прочие параметры
Алгоритм оптимизации параметров термов	$0{,}75GM+0{,}25Acc$	300 глобальных, 100 локальных	50 агентов: 5 мемплексов по 10 агентов	*const* = 1,2; $G_0 = 100$, $\alpha = 10$, $\varepsilon = 0{,}01$
Алгоритм формирования структуры	$0{,}75GM+0{,}25Acc$	20 глобальных, 50 локальных	50 агентов: 5 мемплексов по 10 агентов	*const* = 1,4
Алгоритм настройки весовых коэффициентов	GM	40 глобальных, 25 локальных	50 агентов: 5 мемплексов по 10 агентов	*const* = 1; $G_0 = 100$, $\alpha = 10$, $\varepsilon = 0{,}01$

Если вес признака оказывался меньше значения 0,1, то он исключался из набора. Совместно с заказчиком было решено, что число правил в базе не должно превышать шести, чтобы избежать излишнего усложнения системы.

После построения моделей для каждого используемого признака были подкорректированы термы, находящиеся ближе к каждой из двух границ области определения признака. Коррекция включала передвижение крайних параметров термов таким образом, чтобы при получении входных значений, выходящих за пределы области определения обучающих данных, значение признака было корректно обработано.

4.3 Результаты построения нечетких классификаторов

На основе обучающих данных были построены три классификатора для определения различных характеристик системы свертывания крови – оценки структурных и хронометрических признаков, а также общего её состояния. Каждый классификатор оперировал с тремя классами, соответствующими состоянию системы гемостаза – гиперкоагуляция, нормокоагуляция и гипокоагуляция.

Обучающие наборы данных были разбиты на непосредственно обучающую и валидационную часть в пропорции 75 на 25 процентов. Позднее были получены дополнительные образцы данных 125 пациенток для проверки эффективности построенных классификаторов, далее они будут обозначены как тестовые данные.

4.3.1 Хронометрические признаки системы свертывания крови. Хронометрические признаки характеризуют процесс образования тромба (таблица 4.4).

Таблица 4.4 – Результаты построения классификатора для оценки хронометрических признаков свертывающей системы пациенток

Параметр	Значение
Количество правил	3
Количество признаков	2
Признаки	*r*; *k*
Веса признаков	1; 1
Общая точность на обучении	96,6
Средняя геометрическая точность на обучении	97,0
Общая точность на валидации	98,0
Средняя геометрическая точность на валидации	98,8
Общая точность на тестировании	95,2
Средняя геометрическая точность на тестировании	97,2

В таблице 4.5 приведены матрицы ошибок для обучающей, тестовой и валидационной выборки, а также итоговый процент правильной классификации каждого класса.

Таблица 4.5 – Количество правильно и ошибочно классифицированных экземпляров каждого класса при оценке хронометрических признаков системы свертывания крови

Обучающие данные				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	40	0	0	100
Норма	3	167	2	97,1
Гиперкоагуляция	0	5	78	94,0
Данные валидации				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	14	0	0	100
Норма	0	55	2	96,5
Гиперкоагуляция	0	0	27	100
Тестовые данные				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	23	0	0	100
Норма	0	30	0	100
Гиперкоагуляция	0	6	66	91,7

База правил нечеткого классификатора, определяющего хронометрические признаки свертывающей системы крови, содержит по одному правилу на каждый класс.

4.3.2 Структурные признаки свертывающей системы крови. Структурные признаки отражают реологические свойства образовавшегося сгустка крови (таблица 4.6).

Таблица 4.6 – Результаты классификатора для оценки структурных признаков

Параметр	Значение
Количество правил	6
Количество признаков	3
Признаки	*r*; *k*; *AM*
Веса признаков	0,2; 0,25; 1
Общая точность на обучении	98,3
Средняя геометрическая точность на обучении	98,8
Общая точность на валидации	100
Средняя геометрическая точность на валидации	100
Общая точность на тестировании	96,0
Средняя геометрическая точность на тестировании	97,4

В таблице 4.7 приведены матрицы ошибок и точность каждого класса.

Таблица 4.7 – Количество правильно и ошибочно классифицированных экземпляров каждого класса при оценке структурных характеристик системы коагуляции

Обучающие данные				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	17	0	0	100
Норма	0	173	3	98,3
Гиперкоагуляция	0	2	94	97,9
Данные валидации				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	6	0	0	100
Норма	0	61	0	100
Гиперкоагуляция	0	0	31	100

Продолжение таблицы 4.7

Тестовые данные				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	5	0	0	100
Норма	0	68	4	94,4
Гиперкоагуляция	0	1	47	97,9

База правил нечеткого классификатора, определяющего структурные признаки свертывающей системы крови, насчитывает одно правило для класса «гипокоагуляция», два правила для класса «норма» и три правила для класса «гиперкоагуляция».

4.3.3 Оценка общего состояния системы свертывания крови. Итоговым результатом служит комплексная оценка системы гемостаза (таблица 4.8).

Таблица 4.8 – Результаты тестирования классификатора для оценки общего состояния свертывающей системы

Параметр	Значение
Количество правил	4
Количество признаков	5
Признаки	*V*; *S*; *r*; *k*; *AM*
Веса признаков	0,2; 0,3; 0,7; 1; 1
Общая точность на обучении	97,3
Средняя геометрическая точность на обучении	97,4
Общая точность на валидации	94,9
Средняя геометрическая точность на валидации	96,7
Общая точность на тестировании	99,2
Средняя геометрическая точность на тестировании	99,0

В таблице 4.9 приведены матрицы ошибок для обучающей, валидационной и тестовой выборки, а также итоговый процент правильной классификации каждого класса.

Таблица 4.9 – Количество правильно и ошибочно классифицированных экземпляров каждого класса при оценке общего состояния системы свертывания крови у беременных

Обучающие данные				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	30	1	0	96,8
Норма	7	149	0	95,5
Гиперкоагуляция	0	0	108	100
Данные валидации				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	10	0	0	100
Норма	5	47	0	90,4
Гиперкоагуляция	0	0	36	100
Тестовые данные				
Классы	Гипокоагуляция	Норма	Гиперкоагуляция	Точность класса
Гипокоагуляция	11	0	0	100
Норма	0	33	1	97,1
Гиперкоагуляция	0	0	79	100

База правил нечеткого классификатора, оценивающего общее состояние свертывающей системы крови, включает два правила для класса «гипокоагуляция» и по одному правилу на классы «норма» и «гиперкоагуляция».

4.4 Описание разработанного программного обеспечения

Результаты исследовательской работы легли в основу программного обеспечения для оценки состояния свертывающей системы крови у беременных женщин, применяемого в клинико-диагностической лаборатории ОГАУЗ «Родильный дом №1». Программа создана совместно с сотрудниками родильного дома. Для программирования использован языке C#. На рисунке 4.1 изображена главная форма разработанной программы.

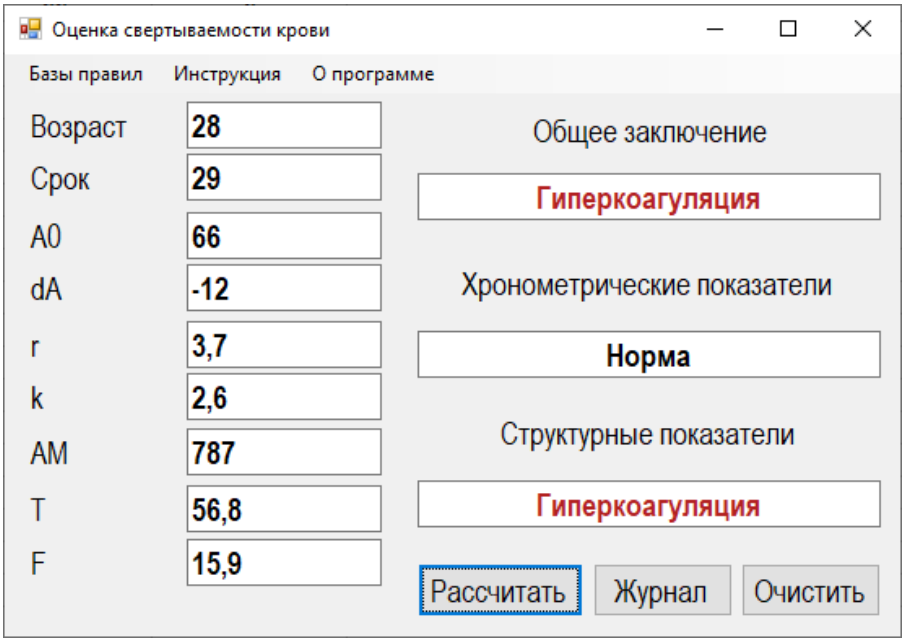

Рисунок 4.1 – Главная форма разработанной программы для оценки состояния системы свертывания крови у беременных женщин

На главной форме располагаются поля для ввода данных, поля для вывода данных, кнопка для расчета вывода классификатора, кнопка для очистки данных во входных полях, кнопка для записи данных в журнал.

На форме «Базы правил» приведены термы и базы правил для каждой из задач классификации (рисунок 4.2).

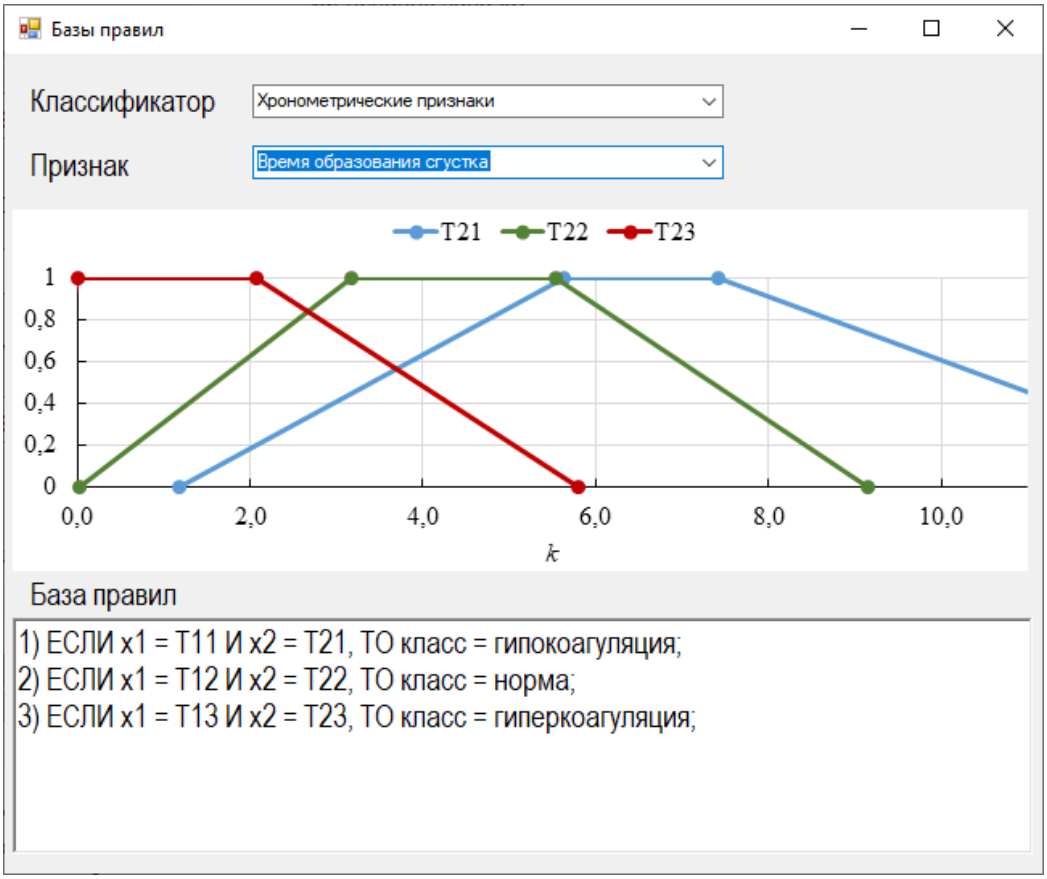

Рисунок 4.2 – Форма для вывода термов и баз правил встроенных классификаторов

Форма «Инструкция» содержит инструкцию для пользователей приложения.

После использования кнопки «Рассчитать» осуществляется считывание данных из входных полей и проверка полученных значений на соответствие типу данных и областям определения соответствующих признаков. Если входное значение не принадлежит требуемому типу данных (вещественному или целому), то программа выдаст ошибку с информацией о некорректности ввода. Если считанное значение какого-либо признака не входит в его область определения, программа выдаст предупреждение, но осуществит расчет.

Акт внедрения результатов диссертационного исследования в рабочий процесс клинико-диагностической лаборатории ОГАУЗ «Родильный дом №1» г. Томска приведен в приложении Б.

4.5 Выводы

1. Для оценки системы свертывания крови у беременных женщин построены три нечетких классификатора. Алгоритм настройки весовых коэффициентов признаков позволил не только корректно расставить веса признаков, но и существенно сократить количество признаков. Наиболее информативными признаками оказались следующие: r – период реакции; k – время образования сгустка; AM – фибрин-тромбоцитарная константа крови. Несмотря на небольшое количество правил, достигнута высокая общая и средняя геометрическая точность

классификации: при оценке общего состояния системы свертывания крови на тестовых данных было получено 99 процентов по обеим метрикам, 95 и 97 процентов соответственно на хронометрических характеристиках, а также 96 и 97 процентов при анализе структурных характеристик системы коагуляции.

2. Совместно с сотрудниками родильного дома подготовлено программное обеспечение, которое может применяться клинико-диагностическими лабораториями для оценки свертывающей системы крови у беременных женщин путем анализа результатов реологических свойств крови коагулометром АРП-01 «МЕДНОРД».

Заключение

В процессе выполнения диссертационной работы разработаны алгоритмы построения и оптимизации нечетких классификаторов, способствующие повышению средней геометрической точности на несбалансированных данных при генерации изначальной структуры алгоритмом экстремальных значений признаков классов. Их применение позволяет создавать точные, компактные и интерпретируемые модели. Полученные в процессе эксперимента нечеткие классификаторы продемонстрировали сопоставимую среднюю геометрическую точность с аналогами, применяющими дополнительный этап предобработки данных и имеющих более сложную и менее интерпретируемую структуру.

В рамках исследования выполнены следующие задачи.

1. Проведен обзор существующих методов обработки несбалансированных данных при построении систем классификации. Выдвинуто предположение, что изменение способа оценки качества нечетких классификаторов при наличии эффективных алгоритмов их построения позволит работать с несбалансированными данными без применения этапа предобработки. При исследовании процесса построения нечетких классификаторов выделены два основных этапа – формирование структуры и её дальнейшая оптимизация. Разобраны три основных направления оптимизации: уточнение структуры, настройка термов и отбор признаков. Выяснено, что все три способа оптимизации могут быть эффективно решены путем применения метаэвристических алгоритмов. При поставке задачи обучение классификатора сформулировано как поиск максимума целевой функции.

2. Разработан алгоритм формирования структуры нечеткого классификатора несбалансированных данных на основе метаэвристики «прыгающие лягушки, предназначенной для итеративного создания и настройки правил для классов с наименьшей долей правильной классификации. В роли фитнес-функции используется разность метрики, объединяющей среднюю геометрическую и общую точность, между дополненной и исходной базами правил. Предложенный алгоритм в комбинации с алгоритмом экстремальных значений признаков классов позволил улучшить среднюю геометрическую точность на 23 процента, точность положительного класса на 25 процентов и точность отрицательного класса на 9 процентов по сравнению с результатами до добавления правил. Алгоритм позволяет создавать классификаторы, демонстрирующие при меньшем числе правил сопоставимую или большую среднюю геометрическую точность по сравнению с общеизвестными алгоритмами построения нечетких классификаторов Chi, Ishibuchi и HFRBCS в комбинации с техникой добавления экземпляров наименьшего класса SMOTE, а также E-алгоритмом.

3. Разработан гибридный алгоритм настройки параметров нечеткого классификатора, основанный на комбинации метаэвристики «гравитационный поиск» с локальным поиском из метаэвристики «прыгающие лягушки». По сравнению с результатами до оптимизации алгоритм позволил увеличить среднюю геометрическую точность классификации и точность положительного класса в среднем на 24 процента при возрастании точности отрицательного класса на 9 процентов на исследованных несбалансированных наборах данных. При существенно меньшем количестве используемых правил алгоритм продемонстрировал сопоставимую точность с комбинацией HFRBCS+SMOTE и большую точность по сравнению с Chi+SMOTE, Ishibuchi+SMOTE и E-алгоритмом.

4. Разработан алгоритм настройки весовых коэффициентов признаков, отражающих важность признака при формировании вывода нечеткого классификатора. Предложена формула для расчета вывода нечеткого классификатора при наличии весов. Для создания первичной популяции весов при проведении настройки предложены два подхода: оценка взаимной информации между признаками и выходной переменной, а также случайная генерация. Настройка вектора весов осуществляется гибридным алгоритмом из метаэвристик «гравитационный поиск» и «прыгающие лягушки». Алгоритм позволил увеличить среднюю геометрическую точность классификации в среднем на 16 процентов относительно точности до введения весов; точность положительного и отрицательного классов была улучшена на 18 и 3 процента соответственно. В сравнении с аналогами алгоритм показывает лучшие результаты, чем E-алгоритм, сопоставимую точность с Ishibuchi, и уступает алгоритмам Chi-3, Chi-5 и HFRBCS со значительно более сложной структурой и наличием этапа предобработки данных алгоритмом SMOTE.

При последовательном выполнении этапов оптимизации параметров термов, настройки весов признаков и повторной оптимизации термов, увеличение средней геометрической точности составило 20 процентов. Предложенная комбинация этапов показала большую среднюю геометрическую точность по сравнению с Ishibuchi и E-алгоритмом, сопоставимую точность с Chi-3 и Chi-5, и уступила только алгоритму HFRBCS. Полученные результаты показывают, что для проведения настройки весов необходима качественная структура нечеткого классификатора, которой можно добиться путем оптимизации параметров термов.

5. Разработанные алгоритмы применены при создании нечетких классификаторов для программного обеспечения, предназначенного для оценки текущего состояния свертывающей системы крови у пациенток ОГАУЗ «Родильный дом №1». Построенные классификаторы продемонстрировали высокое качество работы: при оценке общего состояния системы свертывания крови на тестовых данных средняя геометрическая точность составила 99 процентов, при анализе хронометрических и структурных характеристик значение этой метрики

равнялось 97 процентам. Созданное совместно с сотрудниками родильного дома программное обеспечение используется в клинико-диагностической лаборатории. Согласно полученному акту о внедрении (приложение Б), использование данного программного обеспечения позволяет сократить время выдачи заключения в среднем на 60 минут (при исходной средней длительности выдачи анализа в 120 минут).

Разработанные алгоритмы можно применять при построении нечетких классификаторов для решения практических задач и в научно-исследовательских целях при анализе данных.

Литература

1 Gil, M.A. Editorial of the special issue “Statistics with imperfect data”/ M.A. Gil, G. González-Rodríguez, R. Kruse // Information Sciences. – 2013. – 245. – P. 1–3.

2 Nguyen, G.H. Learning pattern classification tasks with imbalanced data sets / G.H. Nguyen, A. Bouzerdoum, S.L. Phung // Pattern Recognition / ed. Peng-Yeng Yin. – London: IntechOpen, 2009. – P. 193-208. – ISBN 978-953-307-014-8.

3 *Ходашинский, И. А.* Построение нечеткого классификатора алгоритмом гравитационного поиска / И. А. Ходашинский, М. Б. Бардамова, В. С. Ковалев // Доклады ТУСУР. – 2017. – Т. 20, № 2. – С. 84–87.

4 Метаэвристические методы отбора информативных классифицирующих признаков / И. А. Ходашинский, А. Е. Анфилофьев, М. Б. Бардамова, К. С. Сарин // Информационные и математические технологии в науке и управлении. – 2017. – № 2 (6). – С. 11–18.

5 Метаэвристические методы оптимизации параметров нечетких классификаторов / И. А. Ходашинский, А. Е. Анфилофьев, М. Б. Бардамова [и др.] // Информационные и математические технологии в науке и управлении. – 2016. – № 1. – С. 73-80.

6 Fernández, A. Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets / A. Fernández, M. J. del Jesus, F. Herrera // International Journal of Approximate Reasoning. – 2009. – Vol. 50, N 3. – P. 561–577.

7 Special issue on recent advances in theory, methodology and applications of imbalanced learning / ed. H. He // IEEE Transactions on Neural Networks and Learning Systems. – 2018. – Vol. 29, N 3. – P. 763.

8 Iris Data Set [Электронный ресурс]. – URL: http://archive.ics.uci.edu/ml/datasets/Iris (дата обращения: 05.07.2021).

9 Standard classification data sets. Knowledge extraction based on evolutionary learning [Электронный ресурс]. – URL: https://sci2s.ugr.es/keel/category.php?cat=clas (дата обращения: 10.10.2020).

10 Customer churn prediction using improved balanced random forests / Y. Xie, X. Li, E.W.T. Ngai, W. Ying // Expert Systems with Applications. – 2009. – Vol. 36, N 3, P. 1. – P. 5445-5449.

11 Бардамова, М. Б. Нечеткий классификатор несбалансированных медицинских данных с применением алгоритма прыгающих лягушек / М. Б. Бардамова // Сборник избранных статей научной сессии ТУСУР. – Томск: В-Спектр, 2019. – Т. 1, № 1-2. – С. 41–44.

12 A comparative study of machine learning algorithms in predicting severe complications after bariatric surgery / Y. Cao, X. Fang, J. Ottosson [et al.] // Journal of Clinical Medicine. – 2019. – Vol. 8, N 5. – P. 668.

13 Fathy, Y. Learning with imbalanced data in smart manufacturing: a comparative analysis / Y. Fathy, M. Jaber, A. Brintrup // IEEE Access. – 2021. – Vol. 9. – P. 2734-2757.

14 KDD cup 1999 Data. UCI machine learning archive [Электронный ресурс]. – URL: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (дата обращения: 25.10.2020)

15 Zuech, R. Detecting web attacks using random undersampling and ensemble learners / R. Zuech, J. Hancock, T.M. Khoshgoftaar // Journal of Big Data. – 2021. – Vol. 8. – P. 75.

16 Bagui, S. Resampling imbalanced data for network intrusion detection datasets / S. Bagui, K. Li // Journal of Big Data. – 2021. – Vol. 8. – P. 6.

17 Khor, K.C. The effectiveness of sampling methods for the imbalanced network intrusion detection data set / K.C. Khor, C.Y. Ting, S. Phon-Amnuaisuk // Recent Advances on Soft Computing and Data Mining. Advances in Intelligent Systems and Computing / eds.: T. Herawan, R. Ghazali, M. Deris. – Cham: Springer, 2014. – Vol 287. – P. 613-622. – ISBN 978-3-319-07692-8.

18 Learning from class-imbalanced data: review of methods and application / G. Haixiang, L. Yijing, J. Shang [et al.] // Expert Systems with Applications. – 2017. – Vol. 73. – P. 220-239.

19 Vuttipittayamongkol, P. On the class overlap problem in imbalanced data classification / P. Vuttipittayamongkol, E. Elyan, A. Petrovski // Knowledge-Based Systems. – 2021. – Vol. 212. – P. 106631.

20 Imbalanced learning: foundations, algorithms, and applications / Eds.: H. He, Y. Ma. – New Jersey: John Wiley & Sons, Inc., 2013. – 216 p. – ISBN 9781118646106.

21 He, H. Learning from Imbalanced Data / H. He, E.A. Garcia // IEEE Transactions on Knowledge and Data Engineering. – 2009. – Vol. 21. – P. 1263-1284.

22 Hand, D. Measuring classifier performance: A coherent alternative to the area under the ROC curve / D. Hand // Machine Learning. – 2009. – Vol. 77. – P. 103-123.

23 Ferri, C. An experimental comparison of performance measures for classification / C. Ferri, J. Haernandez-Orallo, R. Modroiu // Pattern Recognition Letters. – 2009. – Vol. 30. – P. 27-38.

24 A new approach for imbalanced data classification based on data gravitation / L. Peng, H. Zhang, B. Yang, Y. Chen // Information Sciences. – 2014. – Vol. 288. – P. 347-373.

25 Classification of imbalanced data by oversampling in kernel space of support vector machines / J. Mathew, C. K. Pang, M. Luo, W. H. Leong // IEEE Transactions on Neural Networks and Learning Systems. – 2018. – Vol. 29. – P. 4065- 4076.

26 Du, Lm. Feature selection for multi-class imbalanced data sets based on genetic algorithm / Lm. Du, Y. Xu, H. Zhu // Annals of Data Science. – 2015. – Vol. 2. – P. 293-300.

27 D‘Addabbo, A. Parallel selective sampling method for imbalanced and large data classification / A. D‘Addabbo, R. Maglietta // Pattern Recognition Letters. – 2015. – Vol. 62. – P. 61-67.

28 Hart, P. The condensed nearest neighbor rule / P. Hart // IEEE Transactions on Information Theory. – 1968. – Vol. 14, N 3. – P. 515-516.

29 Tomek, I. Two modifications of CNN / I. Tomek // IEEE Transactions on Systems Man and Cybernetics. – 1976. – Vol. 6. – P. 769-772.

30 Smith, M.R. An instance level analysis of data complexity / M.R. Smith, T. Martinez, C. Giraud-Carrier // Machine Learning. – 2014. – Vol. 95, N 2. – P. 225-256.

31 Sobhani, P. Learning from imbalanced data using ensemble methods and cluster-based undersampling / P. Sobhani, H. Viktor, S. Matwin // New Frontiers in Mining Complex Patterns. NFMCP 2014: New Frontiers in Mining Complex Patterns / Eds.: Appice A., Ceci M., Loglisci C. [et al.]. – Cham: Springer, 2014. – Vol. 8983. – P. 69-83.

32 Fast-CBUS: a fast clustering-based undersampling method for addressing the class imbalance problem / N. Ofek, L. Rokach, R. Stern, A. Shabtai // Neurocomputing. – 2017. – Vol. 243. – P. 88-102.

33 Electroencephalogram emotion recognition based on dispersion entropy feature extraction using random over-sampling imbalanced data processing / X.-W. Ding, Z.-T. Liu, D.-Y. Li [et al] // IEEE Transactions on Cognitive and Developmental Systems. – 2021. – P. 1-1.

34 SMOTE: synthetic minority over-sampling technique / N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer // Journal of Artificial Intelligence Research. – 2002. – Vol. 16. – 321-357.

35 Koziarski, M. Radial-based undersampling for imbalanced data classification / N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer // Pattern Recognition. – 2020. – Vol. 102. – P. 107262.

36 Han, H. Borderline-SMOTE: a new over-sampling method in imbalanced data aets learning / H. Han, W.Y. Wang, B.H. Mao // Lecture Notes in Computer Science. – 2005. – Vol. 3644. – P. 878–887.

37 Bunkhumpornpat, C. Safe-Level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem / C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap // Advances in Knowledge Discovery and Data Mining (PAKDD 2009). Lecture Notes in Computer Science / Eds.: T. Theeramunkong, B. Kijsirikul, N. Cercone, T.B. Ho – Berlin, Heidelberg: Springer, 2009. – Vol 5476. – P. 475-482.

38 Jo, T. Class imbalances versus small disjuncts / T. Jo, N. Japkowicz // ACM SIGKDD Explorations Newsletter. – 2004. – Vol. 6, N 1. – P. 40-49.

39 ADASYN: adaptive synthetic sampling approach for imbalanced learning / H. He, Y. Bai, E. A. Garcia, S. Li // Proceedings of the 5th IEEE International Joint Conference on Neural Networks. – IEEE, 2008. – P. 1322-1328.

40 Generative adversarial nets / I. Goodfellow, J. Pouget-Abadie, M. Mirza [et al.] // Advances in neural information processing systems / Eds.: Z. Ghahramani, M. Welling, C. Cortes [et al.]. – NY: Curran Associates, Inc., 2014. – Vol. 27. – P. 2672-2680.

41 SMOTE–IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering / J. A. Saez, J. Luengo, C. García-Osorioa, L. I. Kuncheva // Information Sciences. – 2015. – Vol. 291. – P. 184-203.

42 A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data / J. A. Sanz, D. Bernardo, F. Herrera [et al.] // IEEE Transactions on Fuzzy Systems. – 2015. – Vol. 23, N 4. – P. 973-990.

43 Kim, S. Ordinal classification of imbalanced data with application in emergency and disaster information services / S. Kim, H. Kim, Y. Namkoong // IEEE Intelligent Systems. – 2016. – Vol. 31, N 5. – P. 50-56.

44 Ling, C. X. Cost-sensitive learning and the class imbalance problem / C. X. Ling, V. S. Sheng // Encyclopedia of Machine Learning / Eds: C. Sammut, G.I. Webb. – Boston: Springer US, 2011. – P. 231-235. – ISBN 978-0-387-34558-1.

45 Ali, A. Classification with class imbalance problem: A review / A. Ali, S.M. Shamsuddin, A. Ralescu // International Journal of Advanced Soft Computing Applications. – 2013. – Vol. 5, N 3. – P. 176-204.

46 Lemnaru, C. Imbalanced classification problems: systematic study, issues and best practices / C. Lemnaru, R. Potolea // Enterprise Information Systems. ICEIS 2011. Lecture Notes in Business Information Processing / Eds.: R. Zhang, J. Zhang, Z. Zhang [et al.]. – Berlin, Heidelberg: Springer, 2012. – Vol 102. – P. 35-50.

47 Wang, Y. An ensemble learning imbalanced data classification method based on sample combination optimization / Y. Wang // Journal of Physics: Conference Series. – 2019. – Vol. 1284. – P. 012035.

48 Hybrid classifier ensemble for imbalanced data / K. Yang, Z. Yu, X. Wen [et al] // IEEE Transactions on Neural Networks and Learning Systems. – 2020. – Vol. 31, N 4. – P. 1387-1400.

49 Zadeh, L.A. Fuzzy sets / L.A. Zadeh // Information and Control. – 1965. – Vol. 8, N 3. – P. 338-353.

50 Прикладные нечеткие системы: пер. с япон. / К. Асаи, Д. Ватада, С. Иваи [и др.]; перевод с яп. Ю. Н. Чернышова; под ред. Т. Тэрано, К. Асаи, М. Сугэно. – М.: Мир, 1993. – 368 с.

51 Zadeh, L.A. Fuzzy Sets as a Basis for Theory of Possibility / L.A. Zadeh // Fuzzy Sets and Systems. – 1999. – Vol. 100, sup. 1. – P. 9-34.

52 Mamdani, E. H. Application of fuzzy algorithms for control of simple dynamic plant / E. H. Mamdani // Proceedings of the Institution of Electrical Engineers. – 1974. – Vol. 121, N 12. – P. 1585–1588.

53 Takagi, T. Fuzzy identification of systems and its applications to modeling and control / T. Takagi, M. Sugeno // Readings in Fuzzy Sets for Intelligent Systems / Eds.: D. Dubois, H. Prade, R. R. Yager. – Waltham: Morgan Kaufmann, 1993. – P. 387-403. – ISBN 978-1-4832-1450-4.

54 Ojha, V. Heuristic design of fuzzy inference systems: a review of three decades of research / V. Ojha, A. Abraham, V. Snasel // Engineering Applications of Artificial Intelligence. – 2019. – Vol. 85. – P. 845-864.

55 Ten years of genetic fuzzy systems: current framework and new trends / O. Cordon, F. Gomide, F. Herrera [et al] // Fuzzy Sets and Systems. – 2004. – Vol. 141, N 1. – 5-31.

56 Sahin, S. Hybrid expert systems: a survey of current approaches and applications / S. Sahin, M. R. Tolun, R. Hassanpour // Expert Systems with Applications. – 2012. – Vol. 39, N 4. – P. 4609-4617.

57 Pelusi, D. On designing optimal control systems through genetic and neuro-fuzzy techniques / D. Pelusi // 2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). – IEEE, 2011. – P. 134-139.

58 Jang, J.-S.R. ANFIS: adaptive-network-based fuzzy inference system / J.-S.R. Jang // IEEE Transactions on Systems, Man, and Cybernetics. – 1993. – Vol. 23, N 3. – P. 665-685.

59 Петрова, И.Ю. Прогнозирование электропотребления с помощью нейро-нечеткой системы ANFIS / И.Ю. Петрова, А.А. Глебов // Машиностроение и компьютерные технологии. – 2006. – N 7. – С. 3.

60 Angelov P. Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density / P. Angelov, R. Yager // IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS). – IEEE, 2011. – P. 62–69.

61 Анфилофьев, А.Е. Отбор признаков для классификатора на основе системы Ангелова-Ягера / А. Е. Анфилофьев // Сборник избранных статей научной сессии ТУСУР. – 2018. – N 1-3. – С. 106-109.

62 Gravitational search for designing a fuzzy rule-based classifiers for handwritten signature verification / M. B. Bardamova, A. Konev, I. Hodashinsky, A. Shelupanov // Journal of Communications Software and Systems. – 2019. – Vol. 15, N 3. – P. 254-261.

63 Горбунов, И. В. Методы построения трехкритериальных Парето-оптимальных нечетких классификаторов / И. В. Горбунов, И. А. Ходашинский // Искусственный интеллект и принятие решений. – 2015. – N 2. – С. 75-87.

64 Chi, Z. Fuzzy algorithms with applications to image processing and pattern recognition / Z. Chi, H. Yan, T. Pham // Advances in Fuzzy Systems – Applications and Theory, Vol 10. – Singapore: World Scientific Pub Co Inc, 1996. – 240 p. – eBook ISBN 978-981-4498-85-2.

65 Ishibuchi, H. Rule weight specification in fuzzy rule-based classification systems / H. Ishibuchi, T. Yamamoto // IEEE Transactions on Fuzzy Systems. – 2005. – Vol. 13, N 4. – P. 428-435.

66 Xu, L. Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification E-algorithm / L. Xu, M.Y. Chow, L.S. Taylor // IEEE Transactions on Power Systems. – 2007. – Vol. 22, N 1. – P. 164-171.

67 Алгоритмы структурной идентификации компактных и точных нечетких систем / И. А. Ходашинский, И. В. Горбунов, К. С. Сарин, С. Р. Субханкулова // Информационные и математические технологии в науке и управлении. – 2016. – N 1. – С. 82-93.

68 Ходашинский, И. А. Идентификация нечетких систем: методы и алгоритмы / И. А. Ходашинский // Проблемы управления. – 2009. – N 4. – С. 15-23.

69 Корышев, Н. П. Алгоритм формирования базы правил нечёткого классификатора на основе алгоритма кластеризации K-средних и метаэвристического алгоритма «китов» / Н. П. Корышев, И. А. Ходашинский // Доклады ТУСУР. – 2021. – Т. 24, N 1. – С. 42-47.

70 The fuzzy inference system with rule bases generated by using the fuzzy C-means to predict regional minimum wage in Indonesia / S. Handoyo, M. Marji, I. N. Purwanto, F. Jie // International Journal of Operations and Quantitative Management. – 2019. – Vol. 24, N 4. – P. 272-296.

71 Freitas, A. A. Data mining and knowledge discovery with evolutionary algorithms / A. A. Freitas. – Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2002. – 265 p. – (Natural Computing Series). – eBook ISBN 978-3-662-04923-5.

72 De Jong, K.A. Using genetic algorithms for concept learning / K.A. De Jong, W.M. Spears, D.F. Gordon // Machine Learning. – 1993. – Vol. 13. – P. 161-188.

73 Fernández, A. Analysing the hierarchical fuzzy rule based classification systems with genetic rule selection / A. Fernández, M. J. del Jesus, F. Herrera // 2010 4th International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS). – IEEE, 2010. – P. 69-74.

74 Alcala-Fdez, J. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning / J. Alcala-Fdez, R. Alcala, F. Herrera // IEEE Transactions on Fuzzy Systems. – 2011. – Vol. 19, N 5. – P. 857-872.

75 González-Muñoz, A. Multi-stage genetic fuzzy systems based on the iterative rule learning approach / A. González-Muñoz, F. Herrera // Mathware & soft computing. – 1997. – Vol. 4, N 3. – P. 233-249.

76 Del Jesus, M. J. MOGUL: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach / M. J. Del Jesus, F. Herrera, M. Lozano // International Journal of Intelligent Systems. – 1999. – Vol. 14, N 11. – P. 1123-1153.

77 Fuzzy rule weight modification with particle swarm optimization / T. Chen, Q. Shen, P. Su, C. Shang // Soft Computing. – 2016. – Vol. 20. – P. 2923-2937.

78 Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data / V. López, S. Del Río, J.M. Benítez, F. Herrera // Fuzzy Sets and Systems. – 2015. – Vol. 258. – P. 5-38.

79 Zolghadri Jahromi, M. A proposed method for learning rule weights in fuzzy rule-based classification systems / M. Zolghadri Jahromi, M. Taheri // Fuzzy Sets and Systems. – 2008. – Vol. 159, N 4. – P. 449-459.

80 Alcala, R. A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection / R. Alcala, J. Alcala-Fdez, F. Herrera // IEEE Transactions on Fuzzy Systems. – 2007. – Vol. 15, N 4. – P. 616-635.

81 Kumar, P.G. Fuzzy classifier design using modified genetic algorithm / P.G. Kumar, D. Devaraj // International Journal of Computational Intelligence Systems. – 2010. – Vol. 3. – P. 334-342.

82 Aydogan, E. K. HGA: hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems / E. K. Aydogan, I. Karaoglan, P. M. Pardalos // Applied Soft Computing. – 2012. – Vol. 12, N 2. – P. 800-806.

83 Fazzolari, M. A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm / M. Fazzolari, R. Alcala, F. Herrera // Applied Soft Computing. – 2014. – Vol. 24. – P. 470-481.

84 Бардамова, М. Б. Применение нечеткого классификатора для прогнозирования риска возникновения и развития сердечно-сосудистых заболеваний / М. Б. Бардамова, В. С. Ковалев, И. В. Горбунов // Материалы докладов международной научно-практической конференции «Электронные средства и системы управления». – Томск: В-Спектр, 2015. – N 1. – С. 248-252.

85 Novaković, J. Toward optimal feature selection using ranking methods and classification algorithms / J. Novaković, P. Strbac, D. Bulatović // Yugoslav Journal of Operations Research. – 2011. – Vol. 1. – P. 119-135.

86 Бардамова, М. Б. Бинаризация непрерывных метаэвристик в задачах отбора признаков для нечетких классификаторов / М. Б. Бардамова, И. А. Ходашинский // Труды VII всероссийской научной-практической конференции «Нечеткие системы, мягкие вычисления и интеллектуальные технологии». – СПб.: Политехника-сервис, 2017. – Т. 2. – С. 18–25.

87 Ходашинский, И. А. Применение ранжирования и схем кроссвалидации при отборе признаков для нечеткого классификатора / И. А. Ходашинский, Ф. Е. Анфилофьев, М. Б. Бардамова [и др.] // Информационные и математические технологии в науке и управлении. – 2018. – № 2 (10). – С. 31–41.

88 Ходашинский, И.А. Построение нечеткого классификатора на основе методов гармонического поиска / И. А. Ходашинский, М. А. Мех // Программирование. – 2017. – N 1. – С. 54-56.

89 Hodashinsky, I.A. Using shuffled frog-leaping algorithm for feature selection and fuzzy classifier design / I. A. Hodashinsky, M. B. Bardamova, V. S. Kovalev // Scientific and Technical Information Processing. – 2019. – Vol. 46. – P. 381-387.

90 Feature selection based on swallow swarm optimization for fuzzy classification / I. Hodashinsky, K. Sarin, A. Shelupanov, A. Slezkin // Symmetry. – 2019. – Vol. 11. – P. 1423.

91 Аутентификация пользователя по динамике подписи на основе нечеткого классификатора / И.А. Ходашинский, Е.Ю. Костюченко, К.С. Сарин [и др.] // Компьютерная оптика. – 2018. – Т. 42, N 4. – С. 657-666.

92 Ходашинский, И. А. Модификации алгоритма прыгающих лягушек для отбора признаков в нечетком классификаторе при аутентификации пользователя по рукописной подписи / И. А. Ходашинский, М. Б. Бардамова // Информационные и математические технологии в науке и управлении. – 2020. – 4(20). – С 75–83.

93 Fuzzy classifier design for network intrusion detection using the gravitational search algorithm / M. B. Bardamova, A. A. Konev, I. A. Hodashinsky, A. A. Shelupanov // Journal of Physics: Conference Series. – 2019. – Vol. 1145. – P. 012008.

94 Liu, H. Toward integrating feature selection algorithms for classification and clustering / H. Liu, L. Yu // IEEE Transactions on Knowledge and Data Engineering. – 2005. – Vol. 17, N 4. – P. 491-502.

95 Shawky, D. M. A feature selection method using misclassified patterns / D. M. Shawky, A. F. Ali // International Journal of Computer Theory and Engineering. – 2011. – Vol. 3, N 5. – P. 643-651.

96 Bolon-Canedo, V. Feature selection for high-dimensional data / V. Bolon-Canedo, N. Sanchez-Marono, A. Alonso-Betanzos // Progress in Artificial Intelligence. – 2016. – Vol. 5. – P. 65-75.

97 Witten, I. H. Data mining: practical machine learning tools and techniques / I. H. Witten, E. Frank, M. Hall. – 3nd ed. – Waltham: Morgan Kaufmann, 2011. – 664 p. – eBook ISBN: 9780080890364.

98 Glowworm swarm based informative attribute selection using support vector machines for simultaneous feature selection and classification / A. Gurav, V. Nair, U. Gupta, J. Valadi // Swarm, Evolutionary, and Memetic Computing. SEMCCO 2014. Lecture Notes in Computer Science / Eds.: B. Panigrahi, P. Suganthan, S. Das. – Cham: Springer, 2015. – Vol. 8947. – P. 27-37.

99 Application of the gravitational search algorithm for constructing fuzzy classifiers of imbalanced data / M. Bardamova, I. Hodashinsky, A. Konev, A. Shelupanov // Symmetry. – 2019. – 11. – P. 1458.

100 Feature selection for high dimensional imbalanced class data using harmony search / A. Moayedikia, K.-L. Ong, Y. L. Boo [et al] // Engineering Applications of Artificial Intelligence. – 2017. – Vol. 57. – P. 38-49.

101 Lughofer, E. On-line incremental feature weighting in evolving fuzzy classifiers / E. Lughofer // Fuzzy Sets and Systems. – 2011. – Vol. 163, N 1. – P. 1-23.

102 Brownlee, J. Clever algorithms: nature–inspired programming recipes / J. Brownlee. – Raleigh: Lulu.com, 2011. – 436 p. – ISBN: 9781446785065.

103 Курейчик, В. М. Генетические алгоритмы / В. М. Курейчик // Известия ЮФУ. Технические науки. – 1998. – N 2. – С. 4-7.

104 Storn, R. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces / R. Storn, K. Price // Journal of Global Optimization. – 1997. – Vol. 11. – P. 341-359.

105 Карпенко, А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие / А. П. Карпенко. – М.: Издательство МГТУ им. Н. Э. Баумана, 2014. – 446 с. – ISBN 978-5-7038-3949-2.

106 Kennedy, J. Particle swarm optimization / J. Kennedy, R. Ebenhart // Proceedings of the 1995 IEEE International Conference on Neural Networks. – Perth: IEEE Service Center, 1995. – P. 1942-1948.

107 Ходашинский, И.А. Применение гибридного квантового алгоритма роящихся частиц для идентификации параметров нечетких аппроксиматоров / И.А. Ходашинский, Д.С. Синьков // Информатика и системы управления. – 2013. – N 2 (36). – С. 56-63.

108 Hodashinsky, I. A. Tuning fuzzy systems parameters with chaotic particle swarm optimization / I. A. Hodashinsky, M. B. Bardamova // Journal of Physics Conference Series. – 2017. – Vol. 803. – P. 012053.

109 A new bio-inspired optimization algorithm: bird swarm algorithm / X. Gao, L. Lu, Y. Liu, H. Zhang // Journal of Experimental & Theoretical Artificial Intelligence. – 2016. – Vol. 208, N 4. – P. 673-687.

110 Сравнительный анализ эффективности метаэвристических алгоритмов при построении нечетких классификаторов / М. Б. Бардамова, А. Е. Анфилофьев, В. С. Ковалев, И. В. Филимоненко // Сборник научных трудов IV Международной летней школы-семинара по искусственному интеллекту «Интеллектуальные системы и технологии: современное состояние и перспективы». – СПб.: Политехника-сервис, 2017. – С. 22–31.

111 Chung, C.-J. A testbed for solving optimization problems using cultural algorithms / C.-J. Chung, R. G. Reynolds // Proceedings of Conference on Evolutionary Programming. – MIT Press, Cambridge, 1996. – P. 225-236.

112 Сахаров, М. К. Меметические алгоритмы для решения задачи глобальной нелинейной оптимизации. Обзор / М. К. Сахаров, А. П. Карпенко // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. – 2015. – N 12. – С. 119-142.

113 Feng, X. A novel optimization algorithm inspired by the creative thinking process / X. Feng, R. Zou, H. Yu // Soft Computing. – 2015. – С. 2955-2972.

114 An improved brain storm optimization with differential evolution strategy for applications of ANNs / Z. Cao, X. Hei, L. Wang [et al] // Mathematical Problems in Engineering. – 2015. – Vol. 2015. – P. 1-18.

115 Geem, Z.W. A new heuristic optimization algorithm: harmony search / Z.W. Geem, J.H. Kim, G.V. Loganathan // Simulation. – 2001. – Vol. 76, N 2. – P. 60-68.

116 Mine blast algorithm: A new population-based algorithm for solving constrained engineering optimization problems / A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi // Applied Soft Computing. – 2013. – Vol. 13, N 5. – P. 2592-2612.

117 Ravi, V. Modified Great Deluge Algorithm versus Other Metaheuristics in Reliability Optimization / V. Ravi // Computational Intelligence in Reliability Engineering. Studies in Computational Intelligence / Ed.: G. Levitin. – Berlin, Heidelberg: Springer. – 2007. – Vol. 40. – P. 21–36.

118 Rashedi, E. GSA: A gravitational search algorithm / E. Rashedi, H. Nezamabadi-pour, S. Saryazdi // Information Sciences. – 2009. – Vol. 179. – P. 2232-2248.

119 Ходашинский, И. А. Методы повышения эффективности роевых алгоритмов оптимизации / И. А. Ходашинский // Автоматика и телемеханика. – 2021. – N 6. – С. 3-45.

120 Карпенко, А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие / А. П. Карпенко. — М.: Издательство МГТУ им. Н. Э. Баумана, 2014. — 446 с.

121 Основанные на производных и метаэвристические методы идентификации параметров нечетких моделей / И.А. Ходашинский, В.Ю. Гнездилова, П.А. Дудин, А.В. Лавыгина // Труды VIII международной конференции «Идентификация систем и задачи управления» SICPRO 2008. – М: Институт проблем управления им. В.А. Трапезникова РАН, 2009. – С. 501–529.

122 Wolpert, D. No free lunch theorems for optimization / D. Wolpert, W. Macready // IEEE Transactions on Evolutionary Computation. – 1997. – Vol. 1. – P. 67-82.

123 Sabri, N. M. An overview of gravitational search algorithm utilization in optimization problems / N. M. Sabri, M. Puteh, M. R. Mahmood // 2013 IEEE 3rd International Conference on System Engineering and Technology. – IEEE, 2013. – P. 61-66.

124 An improved gravitational search algorithm for solving short-term economic/environmental hydrothermal scheduling / H. Tian, X. Yuan, Y. Huang, X. Wu // Soft Computing. – 2015. – Vol. 19. – P. 2783–2797.

125 Лисин, А.В. Применение метаэвристических алгоритмов к решению задач кластеризации методом k-средних / А.В. Лисин, Р.Т. Файзуллин // Компьютерная оптика. – 2015. – Т. 39, N3. – С. 406–412.

126 Rashedi, E. GSA: binary gravitational search algorithm / E. Rashedi, H. Nezamabadi-pour, S. Saryazdi // Natural Computing. – 2010. – Vol. 9. – P. 727-745.

127 A Fuzzy Classifier with Feature Selection Based on the Gravitational Search Algorithm / M. Bardamova, A. Konev, I. Hodashinsky, A. Shelupanov // Symmetry. – 2018. – Vol. 10. – P. 609.

128 Bardamova, M. B. Designing fuzzy classifiers with feature selection by the binary gravitational search algorithm for imbalanced data / M. B. Bardamova // Материалы докладов XIV Международной научно-практической конференции «Электронные средства и системы управления». – Томск: В-Спектр, 2018. – Ч.2 – С. 266–269.

129 Improved chaotic gravitational search algorithms for global optimization / D. Shen, T. Jiang, W. Chen [et al] // 2015 IEEE Congress on Evolutionary Computation (CEC). – IEEE, 2015. – P. 1220-1226.

130 Eusuff, M. M. Optimization of water distribution network design using the shuffled frog leaping algorithm / M. M. Eusuff, K. E. Lansey // Journal of Water Resources Planning and Management. – 2003. – Vol. 129, N 3. – PP. 210-225.

131 Eusuff, M. M. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization / M. M. Eusuff, K.E. Lansey, F. Pasha // Engineering Optimization. – 2006. – Vol. 38, N 2. – P. 129-154.

132 Elbeltagi, E. A modified shuffled frog-leaping optimization algorithm: applications to project management / E. Elbeltagi, T. Hegazy, D. Grierson // Structure and Infrastructure Engineering. – 2007. – Vol. 3, N 1. – P. 53-60.

133 Afzalan, E. Optimal placement and sizing of DG in radial distribution networks using SFLA/ E. Afzalan, M.A. Taghikhani, M. Sedighizadeh // International Journal of Energy Engineering. – 2012. – Vol.2, N 3. – P. 73-77.

134 Application of shuffled frog leaping algorithm to long term generation expansion planning / M. Jadidoleslam, E. Bijami, N. Amiri [et al] // International Journal of Computer and Electrical Engineering. – 2012. – Vol. 4, N 2. – P. 115-120.

135 Mahmoudi, N. Integration of shuffled frog leaping algorithm and support vector regression for prediction of water quality parameters / N. Mahmoudi, H. Orouji, E. Fallah-Mehdipour // Water Resources Management. – 2016. – Vol. 30. – P. 2195–2211.

136 Bardamova, M. B. Binarization of the Shuffled frog leaping algorithm for feature selection in fuzzy classifiers / M. B. Bardamova // Электронные средства и системы управления: материалы докладов XVI Международной научно-практической конференции. – Томск: В-Спектр, 2020. – Ч. 2. – С. 232–235.

137 Бардамова, М. Б. Способы адаптации алгоритма прыгающих лягушек к бинарному пространству поиска при решении задачи отбора признаков / М. Б. Бардамова, А. Г. Буймов, В. Ф. Тарасенко // Доклады ТУСУР. – 2020. – Т. 23, № 4. – С. 57–62.

138 Hodashinsky, I.A. Identification of the parameters of fuzzy approximators and classifiers based on the cuckoo search algorithm / I. A. Hodashinsky, D. Y. Minina, K. S. Sarin // Optoelectronics, Instrumentation and Data Processing. – 2015. – Vol. 51. – P. 234–240.

139 Бардамова, М. Б. Формирование структуры нечеткого классификатора алгоритмом на основе экстремумов классов, дополненного алгоритмом прыгающих лягушек / М. Б. Бардамова // Сборник избранных статей по материалам международной научно-технической конференции «Научная сессия ТУСУР». – Томск: В-Спектр, 2020. – Ч. 2. – С. 49–51.

140 Бардамова, М.Б. Формирование структуры нечеткого классификатора комбинацией алгоритма экстремумов классов и алгоритма «прыгающих лягушек» для несбалансированных данных с двумя классами / М.Б. Бардамова, И. А. Ходашинский // Автометрия. – 2021. – Т. 57, №4. – С. 54-64.

141 Бардамова, М. Б. Оптимизация параметров нечеткого классификатора комбинацией алгоритмов гравитационного поиска и прыгающих лягушек / М. Б. Бардамова //

Сборник трудов XVII Международной конференции «Перспективы развития фундаментальных наук». – Томск: Изд-во Томск. гос. ун-та систем упр. и радиоэлектроники, 2020. – Т. 7. – С. 23–25.

142 Bardamova, M. B. Optimization of fuzzy classifier parameters with a combination of gravitational search algorithm and shuffled frog leaping algorithm / M. B. Bardamova, I.A. Hodashinsky // Journal of Physics: Conference Series. – 2020. – Vol. 1611, No. 1. – P. 012068.

143 Ходашинский, И. А. Исследование эффективности бинарного гравитационного алгоритма при построении нечетких классификаторов с отбором признаков / И. А. Ходашинский, М. Б. Бардамова // Материалы IV Всероссийской Поспеловской конференции с международным участием «Гибридные и синергетические интеллектуальные системы». – Калининград: Изд-во БФУ им. Иммануила Канта, 2018. – С. 448–455.

144 Bardamova, M. Hybrid Algorithm for Tuning Feature Weights in a Fuzzy Classifier / M. Bardamova, I. Hodashinsky // 2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). – IEEE, 2021. – P. 0354-0357.

145 Imbalanced data sets for classification. Knowledge extraction based on evolutionary learning. – URL: https://sci2s.ugr.es/keel/imbalanced.php (дата обращения: 07.09.2020).

146 Global causes of maternal death: a WHO systematic analysis / L. Say, D. Chou, A. Gemmill [et al] // The Lancet Global Health. – 2014. – Vol. 2, N 6. – P. e323-e333.

147 Куликов, А.В. Протокол неотложной помощи при кровотечении в акушерстве. Методические рекомендации / А.В. Куликов, С.В. Мартиросян, Т.А. Обоскалова. – Екатеринбург: ГОУ ВПО «Уральская государственная медицинская академия Росздрава», 2010. – 38с.

148 Айламазян, Э.К. Еще один взгляд на проблему акушерских кровотечений / Э.К. Айламазян, М.А. Репина, Т.У. Кузьминых // Журнал акушерства и женских болезней. – 2008. – Т. LVII, N 3. – С. 3-11.

149 Шифман, Е.М. Интенсивная терапия и анестезия при кровопотере в акушерстве. Клинические рекомендации / Е.М. Шифман, А.В. Куликов, С.Р. Беломестнов // Анестезиология и реаниматология. – 2014. – N 1. – С. 76-78.

150 Об утверждении Порядка оказания медицинской помощи по профилю "акушерство и гинекология: приказ М-ва здравоохранения Рос. Федерации от 20.10.2020 № 1130н // Официальный интернет-портал правовой информации: гос. система правовой информации. – URL: http://publication.pravo.gov.ru/Document/View/0001202011130037?index=0&rangeSize=1 (дата обращения: 24.03.2021).

151 Интраоперационный контроль гемостаза в акушерстве / С. Беломестнов, А. Жилин, А. Матковский [и др.] // Медицина целевые проекты. – 2014. – N 18. – С. 24-25.

152 Тютрин, И. И. Методика исследования и интегральной оценки реологических свойств крови (Расшифровка и интерпретация графика изменений агрегатного состояния крови) / И. И. Тютрин, М. Н. Шписман, А. И. Стеценко // Актуальные проблемы клинических исследований крови. – 1997. – С. 8-18.

153 Прогнозирование результатов исследования реологических свойств крови у беременных женщин для оценки свертывающей системы с использованием нечеткого классификатора / И. А. Ходашинский, И. Б. Бардамова, М. Б. Бардамова [и др.] // Материалы Всероссийской научно-практической конференции в рамках I Конгресса «Здравоохранение России. Технологии опережающего развития». – Томск: STT, 2015. – С. 95–98.

154 *Ходашинский, И.А.* Комплексная оценка параметров коагуляции у беременных женщин с помощью нечеткого классификатора / И.А. Ходашинский, И.Б. Бардамова, М.Б. Бардамова // Медицинская техника. – 2017. – № 3(303). – С. 52–55.

155 *Бардамова, М. Б.* Построение нечеткого классификатора для оценки состояния системы гемостаза у беременных женщин / М. Б. Бардамова // Сборник трудов XIV Международной научно-практической конференции «Молодежь и современные информационные технологии». – Томск: ТПУ, 2016. – Т. 2. – С. 294–295.

Приложение А

Точность классов после построения нечеткого классификатора с настройкой весов

В таблице А.1 представлена точность положительного класса после построения структуры алгоритмом экстремумов признаков классов (АЭПК) и оптимизации. Для генерации популяции весов использована взаимная информация. Схемы эксперимента описаны в параграфе 3.5.

Таблица А.1 – Точность наименьшего класса при наличии этапа настройки весов с генерацией популяции на основе взаимной популяции

Данные	АЭПК	Веса	Веса + термы	Термы	Термы + веса	Термы + веса + термы
gl1	31,8	78,1 ± 2,8	68,4 ± 3,5	63,1 ± 6,1	61,0 ± 5,7	60,9 ± 5,4
ecl0/1	97,3	95,9 ± 0,3	95,4 ± 1,1	95,7 ± 1,6	95,4 ± 1,3	94,8 ± 1,3
wis	94,1	90,8 ± 0,0	96,2 ± 1,4	94,8 ± 1,9	95,5 ± 1,3	95,6 ± 0,9
pm	57,5	69,9 ± 1,5	70,4 ± 2,8	70,8 ± 2,6	70,2 ± 2,6	70,6 ± 3,4
gl0	45,7	81,8 ± 0,7	83,6 ± 2,7	85,7 ± 4,0	86,3 ± 4,0	83,2 ± 4,1
yst1	98,1	61,2 ± 1,6	64,5 ± 2,2	65,9 ± 2,7	66,0 ± 2,1	65,8 ± 2,7
hbr	54,3	34,1 ± 1,9	52,3 ± 3,6	58,5 ± 5,6	59,9 ± 4,7	58,8 ± 4,3
vhc2	22	68,5 ± 3,1	67,6 ± 3,4	68,8 ± 4,7	70,5 ± 4,9	69,4 ± 4,1
vhc1	57,2	64,6 ± 3,3	60,3 ± 3,1	64,3 ± 3,3	65,4 ± 2,8	63,5 ± 2,2
vhc3	48,6	52,8 ± 2,4	56,3 ± 1,7	64,7 ± 3,4	64,5 ± 2,8	66,1 ± 2,7
gl0123/456	80,4	87,6 ± 1,3	90,8 ± 2,9	89,4 ± 3,8	91,6 ± 3,4	91,5 ± 2,8
vhc0	34,7	62,3 ± 1,5	86,0 ± 4,2	86,6 ± 4,0	89,4 ± 1,7	87,8 ± 4,0
ecl1	75,5	94,9 ± 0,0	95,8 ± 0,6	91,7 ± 2,0	91,5 ± 2,1	93,4 ± 1,3
nwth2	100	97,1 ± 0,0	94,1 ± 2,2	93,5 ± 2,8	92,8 ± 2,8	93,1 ± 2,9
nwth1	98,3	100,0 ± 0,0	95,0 ± 1,4	94,5 ± 1,8	94,5 ± 1,8	92,4 ± 4,2
ecl2	15,1	85,3 ± 0,0	84,9 ± 2,9	87,2 ± 3,7	87,2 ± 3,5	86,7 ± 2,8
sgm0	84,2	97,6 ± 0,6	94,5 ± 2,2	93,3 ± 2,8	94,3 ± 2,2	94,7 ± 2,1
gl6	14	67,4 ± 4,9	80,2 ± 3,0	85,2 ± 3,2	85,8 ± 4,5	85,6 ± 4,3
yst3	94,5	91,6 ± 1,0	90,5 ± 1,8	84,9 ± 5,5	87,7 ± 3,3	89,5 ± 2,6
ecl3	37,1	88,6 ± 0,0	89,1 ± 4,0	87,8 ± 4,2	87,2 ± 3,4	88,0 ± 2,8
pb0	42,8	66,4 ± 5,4	74,9 ± 4,5	70,1 ± 6,5	72,9 ± 4,9	74,4 ± 2,8
yst2/4	77,5	87,9 ± 0,5	81,2 ± 2,7	78,3 ± 3,0	78,8 ± 3,8	78,2 ± 3,3
yst05679/4	80,4	63,9 ± 1,6	71,6 ± 1,1	69,2 ± 4,0	69,0 ± 3,0	72,8 ± 2,5
vwl0	84,4	90,0 ± 0,0	86,5 ± 4,6	81,6 ± 5,2	85,0 ± 4,8	83,0 ± 5,5
gl2	6,7	28,6 ± 5,1	54,7 ± 10,6	63,0 ± 7,0	65,4 ± 11,1	62,2 ± 6,7
gl4	13,3	25,1 ± 6,1	71,8 ± 9,0	81,3 ± 5,2	82,7 ± 6,5	77,3 ± 8,4
ecl4	50	85,7 ± 1,2	89,3 ± 3,0	85,3 ± 4,4	85,7 ± 4,7	87,0 ± 4,3
pb1-3/4	58	86,4 ± 2,8	88,4 ± 6,1	81,4 ± 7,6	84,4 ± 6,1	89,7 ± 5,3
ab9/18	50,6	69,9 ± 2,1	70,5 ± 3,4	66,6 ± 5,6	66,1 ± 5,9	67,1 ± 4,2
yst1458/7	36,7	62,4 ± 3,1	48,4 ± 7,0	48,2 ± 7,5	47,8 ± 7,1	48,7 ± 6,8
yst2/8	50	64,3 ± 2,3	61,3 ± 4,1	59,3 ± 1,8	59,3 ± 3,0	59,3 ± 3,0
yst4	74,4	62,9 ± 0,0	71,6 ± 2,1	75,6 ± 3,3	75,9 ± 2,8	75,9 ± 2,0
yst1289/7	46,7	59,6 ± 0,8	57,3 ± 4,7	56,0 ± 7,6	55,6 ± 8,1	54,2 ± 8,8
yst5	54,2	92,3 ± 1,2	94,7 ± 1,5	95,0 ± 2,2	95,3 ± 1,8	94,9 ± 2,5
ecl0137/26	0	54,0 ± 6,4	72,0 ± 4,5	68,0 ± 8,3	70,0 ± 5,3	72,0 ± 3,5
yst6	31,4	79,2 ± 1,1	81,1 ± 3,0	81,3 ± 2,8	81,9 ± 2,7	81,3 ± 2,7
Среднее	55,5	73,6 ± 1,9	77,5 ± 3,4	77,4 ± 4,2	78,1 ± 3,9	78,0 ± 3,7

В таблице А.2 содержится точность положительного класса при случайной генерации популяции весов признаков.

Таблица А.1 – Процент правильной классификации наименьшего класса при наличии этапа настройки весов со случайной генерацией популяции

	АЭПК	Веса	Веса + термы	Термы	Термы + веса	Термы + веса + термы
gl1	31,8	78,4 ± 1,7	65,1 ± 3,7	61,4 ± 3,4	61,6 ± 4,2	60,8 ± 4,9
ecl0/1	97,3	96,3 ± 0,4	95,2 ± 1,3	95,2 ± 1,4	95,3 ± 1,4	93,7 ± 1,3
wis	94,1	90,8 ± 0,0	95,3 ± 0,9	95,9 ± 1,5	95,7 ± 1,3	96,2 ± 1,4
pm	57,5	68,7 ± 0,9	69,0 ± 1,6	70,1 ± 4,2	70,9 ± 3,3	70,1 ± 2,5
gl0	45,7	81,1 ± 1,0	85,6 ± 3,4	84,6 ± 3,8	84,7 ± 3,5	85,6 ± 2,4
yst1	98,1	61,4 ± 2,1	65,7 ± 2,9	65,7 ± 3,1	66,0 ± 2,9	66,3 ± 3,3
hbr	54,3	33,4 ± 0,0	47,9 ± 3,2	59,8 ± 5,3	60,4 ± 5,4	61,9 ± 6,3
vhc2	22,0	68,2 ± 3,1	66,5 ± 2,9	68,7 ± 4,6	71,9 ± 4,7	69,9 ± 5,2
vhc1	57,2	63,3 ± 4,2	60,3 ± 1,7	63,7 ± 3,2	64,9 ± 2,7	64,0 ± 2,2
vhc3	48,6	54,3 ± 2,2	56,4 ± 2,1	65,7 ± 3,1	66,3 ± 2,8	65,4 ± 2,5
gl0123/456	80,4	85,3 ± 2,2	91,8 ± 3,2	86,8 ± 3,2	87,9 ± 3,5	89,8 ± 3,4
vhc0	34,7	60,7 ± 1,8	87,2 ± 3,0	85,5 ± 5,0	86,7 ± 3,6	89,5 ± 2,6
ecl1	75,5	94,9 ± 0,0	95,9 ± 0,8	92,3 ± 2,2	92,6 ± 1,8	90,8 ± 1,4
nwth2	100,0	99,4 ± 0,9	94,7 ± 2,0	94,7 ± 2,7	94,7 ± 2,7	93,5 ± 2,3
nwth1	98,3	100,0 ± 0,0	94,5 ± 3,3	93,5 ± 3,0	93,5 ± 3,0	93,3 ± 3,6
ecl2	15,1	85,6 ± 0,9	86,7 ± 3,1	86,8 ± 2,5	86,7 ± 2,6	87,1 ± 2,8
sgm0	84,2	95,8 ± 1,0	92,3 ± 3,5	93,0 ± 2,6	93,6 ± 2,7	92,2 ± 3,7
gl6	14,0	73,5 ± 5,3	80,9 ± 2,2	85,4 ± 3,5	86,1 ± 3,2	84,4 ± 5,2
yst3	94,5	91,8 ± 1,1	90,9 ± 2,0	84,6 ± 4,2	87,3 ± 3,8	85,7 ± 4,1
ecl3	37,1	88,0 ± 0,9	88,4 ± 3,7	88,6 ± 3,0	88,4 ± 3,3	88,6 ± 2,7
pb0	42,8	64,9 ± 5,4	73,7 ± 3,6	69,9 ± 4,4	73,5 ± 3,2	70,3 ± 5,8
yst2/4	77,5	87,9 ± 1,0	80,9 ± 3,3	78,5 ± 3,8	78,6 ± 3,5	79,1 ± 2,8
yst05679/4	80,4	64,6 ± 2,5	72,4 ± 1,9	72,5 ± 3,7	71,9 ± 3,6	72,7 ± 3,1
vwl0	84,4	90,1 ± 0,4	82,7 ± 3,8	78,4 ± 6,5	80,3 ± 4,8	85,4 ± 3,4
gl2	6,7	27,2 ± 4,6	56,8 ± 12,3	66,6 ± 9,5	68,4 ± 6,6	60,9 ± 9,3
gl4	13,3	19,8 ± 6,1	74,2 ± 9,6	80,0 ± 9,8	82,7 ± 8,4	77,3 ± 10,8
ecl4	50,0	87,0 ± 2,4	88,0 ± 6,0	87,3 ± 3,8	86,0 ± 4,7	87,0 ± 4,9
pb1-3/4	58,0	84,5 ± 4,0	89,5 ± 4,8	83,6 ± 6,6	87,1 ± 5,9	91,6 ± 5,6
ab9/18	50,6	70,6 ± 2,2	70,9 ± 3,6	68,6 ± 5,8	70,7 ± 4,8	70,7 ± 6,3
yst1458/7	36,7	62,4 ± 3,7	50,7 ± 6,9	46,7 ± 7,1	47,1 ± 8,0	51,1 ± 6,5
yst2/8	50,0	64,3 ± 1,7	58,3 ± 3,1	56,3 ± 4,6	56,3 ± 4,1	58,0 ± 3,6
yst4	74,4	62,4 ± 0,9	73,7 ± 2,0	74,9 ± 3,6	74,1 ± 2,9	72,7 ± 2,0
yst1289/7	46,7	59,8 ± 0,9	51,3 ± 3,6	55,1 ± 7,0	56,0 ± 5,6	57,3 ± 6,8
yst5	54,2	91,3 ± 1,8	95,2 ± 1,6	94,1 ± 2,3	95,1 ± 1,8	94,5 ± 3,0
ecl0137/26	0,0	58,7 ± 10,8	75,3 ± 10,4	68,7 ± 6,0	68,7 ± 6,0	67,3 ± 5,6
yst6	31,4	78,9 ± 1,5	82,3 ± 4,1	80,0 ± 3,4	79,6 ± 3,5	79,0 ± 4,1
Среднее	55,5	73,5 ± 2,2	77,4 ± 3,6	77,3 ± 4,3	78,1 ± 3,9	77,9 ± 4,1

Таблица А.3 демонстрирует процент правильной классификации отрицательного класса после оптимизации. В качестве способа генерации популяции весовых коэффициентов признаков использована взаимная информация.

Таблица А.3 – Точность набольшего класса при наличии этапа настройки весов с генерацией популяции на основе взаимной популяции

Данные	АЭПК	Веса	Веса + термы	Термы	Термы + веса	Термы + веса + термы
gl1	74,1	47,6 ± 1,2	54,4 ± 3,9	59,8 ± 6,5	63,3 ± 6,1	67,5 ± 3,6
ecl0/1	81,2	97,3 ± 0,2	98,4 ± 0,7	97,3 ± 2,2	97,7 ± 1,6	98,5 ± 1,0
wis	58,5	93,2 ± 0,0	94,8 ± 0,9	94,9 ± 0,5	95,0 ± 0,6	94,6 ± 1,1
pm	65,2	60,6 ± 0,7	68,7 ± 2,1	66,0 ± 1,8	69,2 ± 2,7	70,2 ± 1,4
gl0	84,7	71,8 ± 0,6	67,9 ± 2,9	60,2 ± 4,4	62,3 ± 3,9	64,8 ± 4,7
yst1	16,3	61,6 ± 1,9	60,7 ± 1,5	64,0 ± 3,7	65,4 ± 3,1	65,9 ± 3,5
hbr	37,3	59,4 ± 2,1	57,3 ± 3,8	66,3 ± 3,4	66,0 ± 3,3	68,3 ± 3,6
vhc2	73,9	67,7 ± 4,8	66,6 ± 4,2	68,8 ± 4,8	71,4 ± 3,7	69,9 ± 5,3
vhc1	30,9	61,0 ± 2,7	67,7 ± 2,1	63,8 ± 2,6	65,2 ± 2,0	66,5 ± 2,1
vhc3	31,7	64,1 ± 1,1	68,6 ± 1,8	66,3 ± 2,1	66,7 ± 1,7	66,3 ± 2,0
gl0123/456	95,7	94,4 ± 0,6	91,7 ± 1,2	87,9 ± 2,7	89,7 ± 1,8	92,1 ± 1,3
vhc0	92,7	80,8 ± 1,4	70,2 ± 3,7	68,1 ± 4,7	72,1 ± 4,2	69,7 ± 4,7
ecl1	87,6	84,5 ± 0,1	84,3 ± 0,5	82,8 ± 2,3	83,5 ± 1,5	83,9 ± 1,0
nwth2	98,3	99,9 ± 0,2	96,0 ± 1,1	97,1 ± 1,1	96,3 ± 1,1	96,6 ± 1,5
nwth1	100,0	98,3 ± 0,1	96,1 ± 1,2	96,1 ± 1,3	94,9 ± 1,5	97,0 ± 1,1
ecl2	99,7	88,5 ± 0,2	87,5 ± 1,4	87,6 ± 1,7	87,4 ± 1,9	86,1 ± 1,8
sgm0	92,2	96,6 ± 0,2	88,5 ± 4,2	75,1 ± 4,3	80,0 ± 3,3	80,8 ± 7,1
gl6	98,9	93,7 ± 0,7	91,9 ± 2,0	93,9 ± 2,1	94,3 ± 1,9	95,4 ± 1,5
yst3	77,5	88,0 ± 0,3	88,3 ± 1,3	81,9 ± 4,4	83,2 ± 3,3	85,9 ± 2,6
ecl3	98,3	85,8 ± 0,2	84,2 ± 1,6	84,7 ± 2,4	84,9 ± 2,0	84,7 ± 1,5
pb0	94,8	89,4 ± 1,7	81,5 ± 3,6	82,5 ± 3,4	85,0 ± 3,0	85,5 ± 2,2
yst2/4	64,9	86,1 ± 0,2	92,1 ± 1,1	89,7 ± 1,8	90,8 ± 1,7	92,2 ± 1,3
yst05679/4	48,6	83,3 ± 0,2	85,1 ± 0,8	82,4 ± 3,5	83,7 ± 2,2	84,7 ± 1,4
vwl0	83,5	89,7 ± 0,2	83,8 ± 3,0	78,5 ± 5,2	83,3 ± 4,7	83,4 ± 3,5
gl2	96,4	75,4 ± 2,3	54,8 ± 2,8	58,8 ± 3,0	58,8 ± 4,5	59,9 ± 4,0
gl4	99,5	91,6 ± 3,8	82,6 ± 2,8	86,0 ± 2,9	86,8 ± 2,7	87,5 ± 1,6
ecl4	99,7	96,9 ± 0,3	94,9 ± 1,1	92,6 ± 2,4	93,5 ± 1,6	93,5 ± 2,3
pb1-3/4	99,1	94,5 ± 1,7	85,2 ± 3,5	83,5 ± 6,5	85,8 ± 4,2	86,0 ± 3,2
ab9/18	73,2	82,9 ± 0,6	77,7 ± 2,9	73,8 ± 3,8	76,0 ± 2,8	76,4 ± 3,5
yst1458/7	59,9	45,2 ± 1,5	67,8 ± 3,0	66,5 ± 4,9	68,3 ± 4,1	66,4 ± 2,0
yst2/8	97,2	92,3 ± 2,0	92,2 ± 2,4	93,8 ± 3,9	95,3 ± 2,1	95,0 ± 1,5
yst4	58,4	76,3 ± 0,0	87,0 ± 1,2	84,7 ± 2,0	85,7 ± 1,5	85,9 ± 2,0
yst1289/7	63,4	64,4 ± 0,9	71,0 ± 1,6	71,0 ± 3,4	71,0 ± 3,5	71,9 ± 3,7
yst5	99,0	96,1 ± 0,1	92,8 ± 1,0	91,0 ± 2,1	92,4 ± 0,9	93,2 ± 0,8
ecl0137/26	99,6	93,3 ± 1,8	90,5 ± 2,6	91,9 ± 2,8	91,8 ± 2,8	92,3 ± 2,5
yst6	94,7	78,0 ± 1,0	90,8 ± 1,0	89,3 ± 2,0	90,4 ± 0,9	89,9 ± 1,0
Среднее	78,5	81,4 ± 1,0	80,9 ± 2,1	80,0 ± 3,1	81,3 ± 2,6	81,9 ± 2,5

Таблица А.4 показывает процент правильной классификации отрицательного класса после оптимизации. При создании популяции весов признаков применялась случайная генерация.

Таблица А.4 – Точность набольшего класса при наличии этапа настройки весов со случайной генерацией популяции

Данные	АЭПК	Веса	Веса + термы	Термы	Термы + веса	Термы + веса + термы
gl1	74,1	47,9 ± 1,1	57,7 ± 3,6	60,7 ± 4,8	61,7 ± 4,7	64,9 ± 4,5
ecl0/1	81,2	97,2 ± 0,0	97,7 ± 2,0	97,8 ± 1,4	98,0 ± 1,4	97,9 ± 1,1
wis	58,5	93,2 ± 0,1	94,8 ± 0,9	93,5 ± 1,9	94,0 ± 1,7	94,3 ± 1,2
pm	65,2	61,2 ± 0,7	70,0 ± 1,8	68,9 ± 2,2	69,1 ± 2,2	68,1 ± 2,4
gl0	84,7	74,3 ± 1,8	65,4 ± 3,2	60,0 ± 4,0	62,6 ± 4,3	64,2 ± 4,1
yst1	16,3	61,8 ± 2,5	60,6 ± 2,1	66,2 ± 2,8	66,8 ± 2,3	64,4 ± 2,9
hbr	37,3	60,9 ± 0,0	58,5 ± 3,2	66,7 ± 3,8	66,7 ± 4,0	63,4 ± 4,1
vhc2	73,9	73,6 ± 3,7	71,3 ± 3,4	71,2 ± 3,4	71,3 ± 3,9	68,7 ± 4,8
vhc1	30,9	62,2 ± 5,5	67,0 ± 1,3	65,8 ± 1,9	66,3 ± 2,1	66,0 ± 1,9
vhc3	31,7	64,0 ± 1,9	69,1 ± 2,3	64,2 ± 2,3	64,4 ± 2,5	65,7 ± 2,0
gl0123/456	95,7	94,6 ± 0,6	90,6 ± 1,2	89,3 ± 1,8	90,1 ± 1,6	90,5 ± 2,2
vhc0	92,7	81,6 ± 1,1	66,9 ± 3,3	65,8 ± 4,7	68,7 ± 5,3	69,6 ± 4,7
ecl1	87,6	84,3 ± 0,2	84,5 ± 0,7	84,0 ± 1,2	84,3 ± 1,1	84,0 ± 1,3
nwth2	98,3	98,3 ± 0,4	95,9 ± 1,6	96,5 ± 1,0	96,7 ± 1,0	96,6 ± 0,9
nwth1	100,0	97,7 ± 0,3	96,1 ± 0,6	96,2 ± 2,0	96,2 ± 2,0	96,7 ± 1,2
ecl2	99,7	88,0 ± 0,7	87,2 ± 1,8	87,2 ± 2,7	87,9 ± 2,3	88,3 ± 1,3
sgm0	92,2	95,8 ± 0,6	86,7 ± 3,6	76,6 ± 4,7	81,5 ± 3,1	76,3 ± 7,2
gl6	98,9	93,5 ± 1,2	91,8 ± 1,4	95,1 ± 1,8	95,2 ± 1,7	94,7 ± 2,1
yst3	77,5	87,3 ± 0,6	86,8 ± 1,8	83,7 ± 3,7	85,6 ± 2,5	83,9 ± 4,0
ecl3	98,3	85,8 ± 0,3	84,0 ± 1,5	82,9 ± 2,5	83,7 ± 2,4	83,2 ± 2,6
pb0	94,8	89,9 ± 1,6	80,7 ± 4,7	85,5 ± 1,8	84,6 ± 1,4	84,3 ± 1,9
yst2/4	64,9	86,0 ± 0,4	92,3 ± 1,2	90,0 ± 1,8	90,6 ± 1,6	90,9 ± 2,6
yst05679/4	48,6	82,8 ± 0,8	85,1 ± 0,8	82,9 ± 2,8	83,0 ± 3,0	83,3 ± 1,7
vwl0	83,5	89,6 ± 0,3	84,5 ± 4,1	77,4 ± 5,7	82,1 ± 4,8	80,3 ± 5,3
gl2	96,4	76,3 ± 3,5	54,7 ± 4,3	55,9 ± 4,5	57,3 ± 4,9	57,5 ± 4,5
gl4	99,5	93,5 ± 3,1	84,6 ± 3,3	83,6 ± 3,9	84,8 ± 3,8	84,5 ± 3,3
ecl4	99,7	96,4 ± 0,4	94,1 ± 1,5	89,6 ± 3,5	91,2 ± 2,4	91,9 ± 2,5
pb1-3/4	99,1	95,5 ± 1,4	83,1 ± 4,3	81,7 ± 6,6	82,8 ± 5,8	85,3 ± 3,6
ab9/18	73,2	83,1 ± 0,8	76,7 ± 1,9	74,7 ± 4,3	75,8 ± 2,6	75,8 ± 2,9
yst1458/7	59,9	45,8 ± 1,8	65,5 ± 2,8	63,8 ± 2,5	65,1 ± 3,8	67,4 ± 3,0
yst2/8	97,2	91,9 ± 1,1	93,4 ± 2,2	94,0 ± 2,5	94,3 ± 2,1	93,5 ± 3,3
yst4	58,4	76,6 ± 0,9	86,4 ± 1,3	84,7 ± 2,2	85,8 ± 1,3	86,6 ± 1,4
yst1289/7	63,4	64,2 ± 0,8	71,6 ± 2,4	72,0 ± 3,5	72,5 ± 2,4	72,8 ± 3,4
yst5	99,0	96,0 ± 0,2	93,0 ± 0,6	92,6 ± 1,1	92,8 ± 1,1	91,7 ± 1,6
ecl0137/26	99,6	91,3 ± 3,0	87,6 ± 3,9	91,3 ± 2,9	91,5 ± 3,2	91,9 ± 3,1
yst6	94,7	78,2 ± 0,2	89,6 ± 1,1	89,5 ± 1,4	89,8 ± 1,3	89,4 ± 1,6
Среднее	78,5	81,7 ± 1,2	80,7 ± 2,3	80,0 ± 2,9	81,0 ± 2,7	80,8 ± 2,8

Приложение Б

Акт о внедрении результатов диссертационного исследования в рабочий процесс

Областное государственное автономное учреждение здравоохранения
«Родильный дом №1»

г. Томск, пр. Ленина, 65, тел.факс: (3822) 53-33-96; e-mail: roddom1@rdom1.tomsk..ru сайт: roddom-1.tomsk.ru,

АКТ ВНЕДРЕНИЯ

результатов диссертационной работы
на соискание ученой степени кандидата технических наук
Бардамовой Марины Борисовны

Комиссия в составе:

председатель – главный врач Е.Ю.Агаркова,

члены комиссии

— заместитель главного врача по клинико-экспертной работе Н.А.Мазур,

— заведующий клинико-диагностической лабораторией И.Б.Бардамова

составила настоящий акт о том, что результаты диссертационной работы Бардамовой М.Б. «Алгоритмы построения нечетких классификаторов несбалансированных данных на основе метаэвристик «гравитационный поиск» и «прыгающие лягушки» были внедрены в деятельность ОГАУЗ «Родильный дом №1» и используются в повседневной практике.

Разработанное программное обеспечение применяется для оценки состояния системы свертывания крови пациенток при обработке результатов анализа реологических свойств крови коагулометром АРП-01 «Меднорд». Программа позволяет оценивать хронометрические, структурные и общие показатели состояния свертывающей системы.

Использование предоставленного программного обеспечения позволяет сократить время выдачи заключения в среднем на 60 минут. Программа помогает выявлять сложные случаи для дальнейшего наблюдения, а также используется для обучения специалистов клинической лабораторной диагностики и практикующих врачей.

Главный врач _____ Е. Ю. Агаркова

Члены комиссии:

Заместитель главного врача по клинико-экспертной работе _____ Н. А. Мазур

Заведующий клинико-диагностической лабораторией _____ И. Б. Бардамова

Приложение В

Акт о внедрении результатов диссертационной работы в учебный процесс

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»

УТВЕРЖДАЮ
Проректор по научной работе
и инновациям
канд. тех. наук, доцент
Лощилов А. Г.
«___» _____ 2021

АКТ

о внедрении в учебный процесс результатов диссертационной работы на соискание ученой степени кандидата технических наук Бардамовой Марины Борисовны

Комиссия в составе председателя Давыдовой Е.М., декана факультета безопасности, членов: Конева А.А., доцента кафедры комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС), Костюченко Е.Ю., доцента кафедры КИБЭВС, подтверждают, что результаты диссертационной работы Бардамовой М.Б. «Алгоритмы построения нечетких классификаторов несбалансированных данных на основе метаэвристик «гравитационный поиск» и «прыгающие лягушки» применяются в учебном процессе кафедры КИБЭВС при организации занятий по дисциплинам «Информатика» и «Теория информации» при подготовке специалистов по направлениям и бакалавров по направлению.

Изучение возможностей применения метаэвристических алгоритмов для улучшения качества моделей машинного обучения, описанных Бардамовой М.Б. в лабораторном занятии по введению в искусственный интеллект в рамках дисциплины «Информатика», позволяет студентам ознакомиться с многообразием применения метаэристик для решения задач оптимизации и способствует развитию интереса первокурсников к проведению дальнейших исследований в области искусственного интеллекта.

В соавторстве с Ходашинским И.А. издано методическое пособие для выполнения практических и самостоятельных работ по дисциплине «Теория информации» (Ходашинский И.А., Бардамова М.Б. Теория информации: методические указания для выполнения практических и самостоятельных работ. – Томск, В-Спектр, 2018. – 64 с.). В пособие включены темы, затронутые в диссертационном исследовании и касающиеся количественной оценки взаимосвязи между переменными.

Материалы диссертации используются в научно-исследовательских работах студентов факультета безопасности.

Председатель комиссии Давыдова Е.М.

Члены комиссии Конев А.А.

Костюченко Е.Ю.

Приложение Г

Свидетельства о государственной регистрации программ для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2018614316

Программа настройки параметров нечеткого классификатора на основе алгоритма гравитационного поиска

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (ТУСУР) (RU)***

Авторы: ***Бардамова Марина Борисовна (RU),***
Ходашинский Илья Александрович (RU)

Заявка № **2017662551**
Дата поступления **04 декабря 2017 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***04 апреля 2018 г.***

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2019612574

Программа отбора признаков для нечеткого классификатора на основе бинарного алгоритма гравитационного поиска со статическими функциями трансформации

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (ТУСУР) (RU)***

Авторы: ***Бардамова Марина Борисовна (RU),***
Ходашинский Илья Александрович (RU)

Заявка № **2019610957**
Дата поступления **05 февраля 2019 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***22 февраля 2019 г.***

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

RU2021611060

ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2021611060
Дата регистрации: **21.01.2021**
Номер и дата поступления заявки:
2021610342 19.01.2021
Дата публикации и номер бюллетеня:
21.01.2021 Бюл. № 2
Контактные реквизиты:
нет

Автор(ы):
Бардамова Марина Борисовна (RU),
Ходашинский Илья Александрович (RU)
Правообладатель(и):
Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)

Название программы для ЭВМ:
Программа добавления правил на основе алгоритма прыгающих лягушек для нечеткого классификатора несбалансированных данных

Реферат:
Программа осуществляет итерационный процесс добавления нечетких правил к первичной базе правил нечеткого классификатора. Генерация и оптимизация правил осуществляется для класса с наименьшей относительной точностью. Параметры антецедентов генерируются на основе экстремальных значений признаков и случайной компоненты, далее проводится настройка термов метаэвристическим алгоритмом «Прыгающие лягушки». В качестве фитнесс-функции используется компромисс между общей и средней геометрической точностью. Разработанная программа может быть использована для построения нечетких классификаторов несбалансированных данных. Тип ЭВМ: IBM PC; ОС: Windows 7/8/10.

Язык программирования: C#

Объем программы для ЭВМ: 33 КБ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

RU2021611138

ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2021611138
Дата регистрации: **22.01.2021**
Номер и дата поступления заявки:
2021610354 19.01.2021
Дата публикации и номер бюллетеня:
22.01.2021 Бюл. № 2

Автор(ы):
Бардамова Марина Борисовна (RU),
Ходашинский Илья Александрович (RU)
Правообладатель(и):
Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)

Название программы для ЭВМ:
Программа настройки параметров нечеткого классификатора несбалансированных данных комбинацией гравитационного алгоритма и алгоритма прыгающих лягушек

Реферат:
Программа предназначена для оптимизации параметров термов в нечетком классификаторе с целью улучшения качества классификации. Особенность программы заключается в комбинации двух метаэвристик: гравитационный алгоритм используется в качестве внешнего глобального поиска, вложенный алгоритм прыгающих лягушек применяется для локального поиска. Частота вхождения в локальный поиск регулируется входным параметром. В качестве фитнесс-функции используется компромисс между общей и средней геометрической точностью. Тип ЭВМ: IBM PC-совмест. ПК. ОС: Windows 7/8/10.

Язык программирования: C#

Объем программы для ЭВМ: 463 КБ