Федеральное государственное бюджетное образовательное учреждение
высшего образования «Сибирский государственный университет науки
и технологий имени академика М.Ф. Решетнева»

На правах рукописи

Полякова Анастасия Сергеевна

КОЛЛЕКТИВНЫЕ МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ

05.13.01 – Системный анализ, управление и обработка информации
(космические и информационные технологии)

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель
доктор технических наук, профессор
Семенкин Е.С.

Красноярск – 2019

Оглавление

Введение

Актуальность работы. Основным этапом процесса извлечения знаний из определенного набора данных и формализации таких знаний является построение и последующее использование соответствующей модели. В современной науке существует значительное количество методов и инструментов интеллектуального анализа данных (ИАД). Общей классификацией, унаследованной от области машинного обучения, является разбиение задач ИАД на три группы: обучение с учителем, обучение без учителя и обучение с подкреплением. В обучении с учителем можно выделить две основные группы методов: методы решения задач классификации и регрессии.

Достаточно очевидным обобщением является использование нескольких алгоритмов одновременно, т.е. применение подходов коллективного принятия решений. Это позволяет повысить качество решения задачи ИАД и значительно упрощает поиск компромисса между точностью, простотой и интерпретируемостью каждой отдельной модели.

С точки зрения теории искусственного интеллекта (ИИ) коллектив можно представить в виде набора автономных агентов, работающих над какой-то общей задачей, например, в мультиагентной системе. Можно сказать, что коллектив является интеллектуальным, если он может эффективно использовать интеллект своих членов в ходе решения задачи.

На практике задачи ИАД, имеющие различные характерные особенности (неполнота и неточность исходных данных, высокая вычислительная сложность получения результатов и их формализации, и т.п.), довольно трудно решить с помощью отдельной технологии. К примеру, когда невозможно получить формализованное математическое описание задач интеллектуального анализа данных, применяются быстро развивающиеся интеллектуальные информационные технологии (ИИТ) в виде искусственных нейронных сетей (ИНС). Когда необходимо принимать

решение на основе опыта экспертов, обладающих знаниями в предметной области решаемой задачи, предпочтительнее использовать системы на основе нечеткой логики (НЛС), позволяющие извлекать опыт экспертов, применять и использовать их для решения задачи.

Коллектив из технологий интеллектуального анализа данных представляет собой набор моделей, каждая из которых способна решить поставленную задачу, а их комбинация позволяет повысить эффективность коллектива в целом.

Простейшими методами формирования коллектива являются методы простого и взвешенного голосования. Простое голосование иногда используется для задач классификации, в этом случае решение принимается по принципу большинства. Взвешенное голосование также учитывает степень доверия каждого агента ансамбля при решении задачи. Методы простого и взвешенного голосования часто применяются, когда базовые алгоритмы существенно отличаются друг от друга.

Более сложными методами являются бэггинг [1], бустинг [2, 3, 4], случайные леса [5] и стэкинг [6].

Известными разновидностями бустинга являются AdaBoost [7] и градиентный бустинг [8]. Похожим на бэггинг методом является метод случайных подпространств [9].

Важным моментом при формировании коллектива является не только выбор его членов и их обучение, но также и способ объединения в коллектив. Бустинг и бэггинг являются методами, которые обучают членов коллектива независимо друг от друга, в этом смысле они не производят искусственную специализацию членов коллектива. Альтернативой является, например, перераспределение обучающих примеров между обучающими методами для членов коллектива, приводящие к специализации их на определенной части задачи. Кроме того, возможен вариант, при котором члены коллектива и способ их комбинирования настраиваются одновременно.

Основная идея стэкинга заключается в использовании базовых классификаторов для получения предсказаний (мета-признаков) и использовании их как признаков для некоторого «обобщающего» алгоритма (мета-алгоритма). Иными словами, основной идеей стэкинга является преобразование исходного пространства признаков задачи в новое пространство, точками которого являются предсказания базовых алгоритмов.

Одной из модификаций стэкинга является блендинг [10]. Суть блендинга заключается в выполнении всего одной пары разбиений обучающей выборки (hold-out). Преимуществами такого подхода являются меньшая вычислительная сложность решения и отсутствие риска переобучения, недостатком - неэффективное использование обучающей выборки и необходимость подбора параметров разбиения.

Еще одной модификацией стэкинга является схема, которую можно рассматривать как частный случай предложенного Д. Волпертом [6] общего подхода к формированию обучающей выборки с помощью базовых алгоритмов. Модернизация заключается в том, чтобы получить метапризнак усреднением предсказаний, полученных с помощью различных разбиений на K непересекающихся блоков (K-fold).

В то же время системы коллективного принятия решения на основе НЛС имеют ряд преимуществ:

1. Все процедуры коллективного принятия решений подбираются под конкретную задачу и решение одной задачи никак не поможет построению коллектива алгоритмов для другой. НЛС этого недостатка лишена за счет специальной подсистемы (базы правил), позволяющей накапливать опыт решения других задач, дообучаться и применять их на других задачах.
2. Модели на базе НЛС формализуются на языке, близком к экспертному, и в тех случаях, когда точность решения критична (неприемлемы потери), процедуры поддержки принятия решения,

близкие к естественные языку, за счет их интерпретируемости являются более предпочтительными с точки зрения сертификации и анализа специалистами предметной области.

3. Большинство коллективных форм, особенно простые, чувствительны к выбору агентов в коллектив. А системы с применением нечеткой логики могут в автоматическом режиме отбрасывать неудачные модели или не учитывать влияние слабых агентов при коллективном выводе.
4. Стандартные формы коллективов испытывают затруднения в случае, когда один агент очень сильный, а все остальные – слабые, и объединение в коллектив дает не улучшенное решение, а ослабление решения сильного агента, в то время как процедура на нечеткой логике может формировать решение не хуже лучшего агента.

Отличительной чертой НЛС является то, что модель строится по принципу «белого ящика». НЛС позволяют координировать и объединять опыт экспертов предметной области, а также способны моделировать нелинейные функциональные зависимости произвольной сложности. Все эти свойства дают возможность рассматривать использование НЛС в качестве метода коллективного принятия решений, что позволило бы существенно повысить качество принимаемых решений, а также их интерпретируемость.

Использование коллективов ИИТ позволяет повысить надежность и эффективность конечного решения, если при их формировании уделять особое внимание разнородности входящих в него моделей. Поэтому необходимо ставить задачу эффективного выбора членов коллектива.

За счет автоматизации процессов проектирования ИИТ отпадает необходимость привлечения экспертов и снижаются вычислительные затраты в ходе тестирования для определения наиболее эффективного метода. В то же время, возникающие при этом задачи выбора эффективных

вариантов коллективов требуют применения мощных и универсальных оптимизационных процедур адаптивного типа. Для этого целесообразным представляется использование адаптивных стохастических алгоритмов решения задач глобальной оптимизации алгоритмически заданных функций смешанных переменных, в частности – эволюционных алгоритмов (ЭА). ЭА позволяют в автоматическом режиме выбирать конфигурацию и настраивать параметры коллективных моделей принятия решений на основе нечеткой логики.

Таким образом, разработка и исследование методов автоматизированного формирования коллективных моделей принятия решений на основе нечеткой логики с использованием эволюционных алгоритмов является **актуальной научно-технической задачей**.

Целью диссертационного исследования является повышение эффективности интеллектуальных технологий анализа данных путем автоматизированного формирования коллективов алгоритмов с помощью специальных систем на нечеткой логике.

Для достижения поставленной цели необходимо решить комплекс **задач**.

1. Провести обзор современных методов анализа данных и форм их коллективного взаимодействия.
2. Разработать и реализовать алгоритм коллективного вывода на основе нечеткой логики для решения задач классификации и регрессии.
3. Разработать и реализовать процедуру выбора алгоритмов классификации или регрессии для включения в состав коллектива.
4. Разработать и реализовать процедуру автоматизированного выбора показательных примеров в опорное множество для формирования коллективного вывода.
5. Реализовать в виде программной системы процедуры коллективного принятия решения на основе нечеткой логики.

6. Исследовать работоспособность предложенного алгоритма на тестовых и практических задачах.

Область исследования. Работа выполнена в соответствии со следующими пунктами паспорта специальности 05.13.01:

- разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации;
- методы и алгоритмы интеллектуальной поддержки при принятии управленческих решений в технических, экономических, биологических, медицинских и социальных системах.

Методы исследования.

При выполнении работы использовались методы и подходы теории вероятностей, методы статистической обработки данных, эволюционных вычислений, оптимизации, нечеткой логики, системного анализа данных, выявления закономерностей в исходных данных.

Научная новизна работы.

1. Разработана новая схема формирования коллективного вывода на основе нечеткой логики, отличающаяся иерархической процедурой интеграции правил коллективного вывода.
2. Разработана новая эволюционная процедура выбора агентов для формирования эффективных коллективов, отличающаяся от известных использованием нескольких критериев эффективности.
3. Разработана новая эволюционная процедура автоматизированного формирования базы правил, отличающаяся от известных применением двух уровней эволюции и способом представления решения в бинарном пространстве поиска.
4. Разработана новая система на основе нечеткой логики для формирования коллективов моделей и алгоритмов анализа данных для решения задач классификации и регрессии, отличающаяся от

известных адаптированной процедурой формирования коллективного решения.

5. Разработана комплексная процедура автоматизированного формирования системы коллективного вывода на основе нечеткой логики, отличающаяся возможностью эффективного перераспределения вычислительных ресурсов.

Значение для теории состоит в разработке комплексного подхода к решению задачи ИАД с помощью нового коллективного метода принятия решения на основе нечеткой логики, который является эффективным обобщением отдельных методов интеллектуального анализа данных. Сформированное итоговое решение, получаемое на основе коллектива моделей, эффективнее, так как коллектив всегда работает не хуже самой лучшей модели. Результаты, полученные при выполнении диссертационной работы, создают теоретическую основу для разработки новых технологий распределения ресурсов и распараллеливания процессов в ходе решения сложных трудно формализуемых задач анализа данных.

Практическая ценность.

Разработанные алгоритмические схемы, которые реализованы в виде программной системы на языке программирования Python, являются полноценной библиотекой. Программная система позволяет формировать коллективный вывод при решении задач интеллектуального анализа данных и проектирование коллективов моделей на основе нечеткой логики для задач классификации и регрессии. Программная система протестирована на задачах распознавания лиц по изображению и прогнозирования аффективного (эмоционального) поведения человека по голосу, а также на задаче моделирования технологического процесса металлургического производства.

Реализация результатов работы. В диссертационной работе была разработана программная система, которая прошла регистрацию в Роспатенте.

Диссертационная работа выполнена в рамках проектов:

1. Проект №14.574.21.0037 «Распределенные самоконфигурируемые многоагентные технологии проектирования и управления интеллектуальными информационными сетями» в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014 - 2020 годы».
2. Проект 2.1680.2017/ПЧ «Разработка теоретических основ автоматизации комплексного моделирования сложных систем методами вычислительного интеллекта», в рамках Государственного задания СибГУ на 2017-2019 гг.
3. Российско-словенский проект "Application of artificial intelligence methods on small field robot" ARRS-MS-BI-RU-JR-Prijava/2018/65 на 2019-2020 год.
4. Проект «Разработка системы автоматического распознавания и классификации дефектов элементов панелей солнечных батарей», Фонд содействия развитию малых форм предприятий в научно-технической сфере по программе «У.М.Н.И.К» 2015-2017гг.
5. Российско-германский проект «Advanced feature selection techniques for multimodal emotion recognition» в рамках конкурса германской службы академических обменов «Программа Эйлера» 2015-2016 гг.

Основные защищаемые положения:

1. Предложенная схема иерархической процедуры коллективного вывода на основе системы нечеткой логики позволяет повысить эффективность коллективного вывода в сравнении одноуровневым принятием решения.

2. Разработанная процедура эволюционного выбора агентов для формирования эффективных коллективов моделей позволяет повысить точность работы коллектива.
3. Разработанная процедура автоматизированного формирования базы правил с применением двух уровней эволюции и предложенным способом представления решения в бинарном пространстве поиска позволяет формировать эффективные базы правил с минимальным количеством правил и высоким уровнем обобщения самих правил без потери точности.
4. Разработанная интегрированная процедура автоматизированного формирования системы коллективного вывода на основе нечеткой логики позволяет в автоматизированном режиме управлять вычислительными ресурсами при обучении коллектива и получать коллективы с высокой точностью решения задач классификации и регрессии.

Апробация работы. Основные положения и результаты работы прошли всестороннюю апробацию на Всероссийских и Международных конференциях: «8th International Congress on Advanced Applied Informatics - "7th International Conference on Smart Computing and Artificial Intelligence" SCAI 2019» (2019 г., Toyama, Japan), «The International Workshop "Advanced Technologies in Material Science, Mechanical and Automation Engineering"» (2019 г., г. Красноярск), «The International Workshop on Mathematical Models and their Applications (IWMMA)» (2014, 2016 г., г. Красноярск), «Международная научно-практическая конференция "Решетневские чтения"» (2012-2016 гг., г. Красноярск), «Международная научно-практическая конференция "Актуальные проблемы авиации и космонавтики"» (2011-2016 гг., г. Красноярск), «Всероссийская научно-практическая конференция "Информационно-телекоммуникационные системы и технологии" ИТСиТ-2014» (2014 г., г. Кемерово), «Всероссийская научно-техническая

конференция "Приоритетные направления развития науки и технологий"», (2013 г., г. Тула).

Публикации. По теме диссертации опубликовано 19 печатных работ, из них три статьи в журналах перечня ВАК РФ и три в изданиях, индексируемых в международных базах цитирования Web of Science и/или Scopus. Получено одно свидетельство о регистрации программной системы в Роспатенте.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка литературы из 138 наименований.

1 Коллективный метод принятия решения на основе нечеткой логики

1.1 Коллективы и коллективные формы принятия решений

Пусть имеется обучающая выборка L, и выбрано множество алгоритмов (классификации или регрессии) $A_1, \dots, A_k$. Эти алгоритмы называются базовыми алгоритмами. Для объединения решений используется специальный алгоритм, который называется мета-алгоритмом. Входными данными мета-алгоритма являются решения базовых алгоритмов. Например, в случае мета-классификатора, его входными данными являются решения базовых классификаторов, т.е. его входом является множество меток классов, к которым базовые классификаторы отнесли описание входного объекта. Множество меток на входе мета-алгоритма интерпретируется как множество признаков нового признакового пространства.

Коллективные методы (*Ensemble Methods*) принятия решений представляют собой группу методов, позволяющих рассмотреть гораздо больше возможных альтернатив в групповом решении чем в индивидуальном. Это - мета-алгоритмы, которые объединяют несколько методов машинного обучения в одну прогностическую модель, с целью повышения точности и улучшения результатов.

Коллективы - это наборы обучающих машин, которые каким-то образом объединяют свои решения, или алгоритмы обучения, или разные представления данных, или другие специфические характеристики, чтобы получить более надежные и более точные прогнозы в задачах обучения с учителем и без [11, 12].

В литературе для обозначения коллективов, которые работают вместе для решения задачи машинного обучения, использовалось множество терминов: ансамбль, слияние, комбинация, агрегация, комитет [13, 14, 15, 16,

17, 18], но в данной работе используется термин «коллектив» в его самом широком значении, чтобы охватить весь спектр комбинированных методов.

В настоящее время коллективные методы представляют собой одно из основных современных направлений исследований в области машинного обучения.

В процессе формирования коллективов методов необходимо пройти следующие этапы:

- предварительная обработка данных (отбор информативных признаков, отбор экземпляров выборки, нормирование данных, восстановление пропусков, удаление выбросов и т.д.);

- выбор структуры использования отдельных алгоритмов (агентов) (параллельная, последовательная или смешанная);

- выбор агентов, в зависимости от постановки задачи (линейная регрессия, метод опорных векторов, искусственная нейронная сеть, метод k – ближайших соседей, деревья решений, системы на нечеткой логике, правила индукции и т.д.);

- выбор алгоритма формирования коллектива агентов (бэггинг, бустинг, случайный лес, блендинг, стэкинг и т.д.);

- выбор способа агрегирования результатов отдельных моделей (простое или взвешенное голосование, усреднение (взвешенное или невзвешенное), ранжирование и т.д.);

- выбор критериев оценки качества полученного результата (*F* – мера, *ROS* – кривые, критерий *CCC*, критерий *MSE* и т.д.).

Комбинирование (агрегирование) моделей дает синергетический эффект, при котором недостатки агентов по отдельности компенсируются достоинствами других, что в свою очередь объясняется возможностью использования более обширного пространства гипотез относительно структуры данных для получения наиболее точной гипотезы.

Одним из дополнительных подходов к гибридизации моделей является разбиение всего множества исходных данных на отдельные кластеры, имеющие однородные статистические характеристики, и построение для них отдельных агентов. Проблема состоит в правильности разбиения на кластеры.

При этом, каждый агент имеет свою специфическую область применения. К примеру, одни агенты лучше справляются с задачами, в которых объекты каждого класса описаны «шарообразными» областями многомерного пространства; другие же предназначены для поиска «ленточных» классов и т.д. Также на различных объектах выборки один агент может ошибаться, в то время как другие дают верный ответ. В случае, когда данные имеют разнородную природу (рисунок 1.1), для выделения групп объектов также целесообразно применять не один агент, а набор различных агентов. Комбинация отдельных технологий в коллективе может компенсировать недостатки обучающих алгоритмов отдельных агентов, а также позволяет получить более эффективные решения в условиях «зашумленных» данных, при наличии в них «пропусков».

Соответственно, коллектив может дать больший эффект, чем применение отдельного агента, увеличивая эффективность и надежность системы в целом.

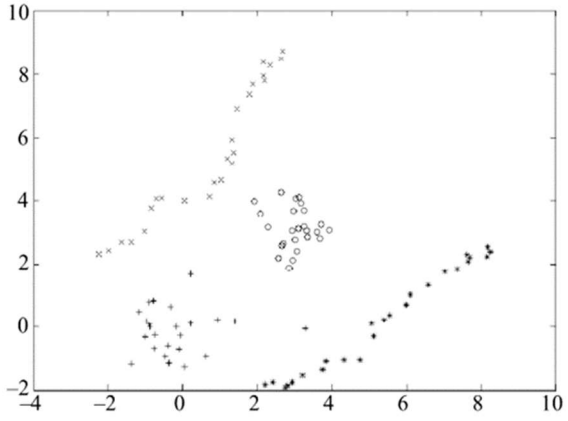

Рисунок 1.1. Пример расположения данных

Под обучением коллектива понимается обучение конечного набора базовых агентов с последующим объединением результатов их прогнозирования в единый прогноз сформированного коллектива. Коллектив дает более точный результат, если:

1. Каждый из базовых агентов сам по себе обладает неплохой точностью.

2. Агенты приводят к разным результатам (ошибаются на разных множествах).

Способы агрегирования результатов агентов. Пусть имеется группа лиц, имеющих право принимать участие в коллективном принятии решений. Предположим, что эта группа рассматривает некоторый набор альтернатив, и каждый член группы осуществляет свой выбор. Ставится задача о выработке решения, которое определенным образом согласует индивидуальные выборы и в каком-то смысле выражает «общее мнение» группы, т.е. принимается за групповой выбор.

Естественно, различным принципам согласования индивидуальных решений будут соответствовать различные групповые решения.

Правила согласования индивидуальных решений при групповом выборе называются правилами голосования.

Для формирования общего решения коллектива алгоритмов наиболее часто используются следующие подходы:

1. Равноправное или взвешенное голосование для задач классификации [19].

2. Простое или взвешенное усреднение для задач регрессии [20, 21].

Рассмотрим задачу классификации на D классов.

$$Y = \{1, 2, \dots, D\} \tag{1.1}$$

Пусть имеется G классификаторов («экспертов») $f_1, f_2, \dots, f_g$:

$$f_g : X \to Y, \quad f_g \in F, \quad g = 1, 2, \dots, G \tag{1.2}$$

В случае простого голосования классификатор строится следующим образом:

$$f(x) = \underset{d=1,\dots,D}{\arg\max} \sum_{g=1}^{G} f_g(x) \quad (1.3)$$

В случае взвешенного голосования (выпуклая комбинация классификаторов, или «смесь экспертов»):

$$f(x) = \underset{d=1,\dots,D}{\arg\max} \sum_{g=1}^{G} \alpha_g \cdot f_g(x), \;\; \alpha_g \geq 0, \quad \sum_{g=1}^{G} \alpha_g = 1 \quad (1.4)$$

или

$$f(x) = \underset{d=1,\dots,D}{\arg\max} \sum_{g=1}^{G} \alpha_g(x) \cdot f_g(x), \;\; \alpha_g(x) \geq 0, \quad \sum_{g=1}^{G} \alpha_g(x) = 1 \quad (1.5)$$

При решении задачи регрессии простое голосование:

$$f(x) = \frac{1}{G} \sum_{g=1}^{G} f_g(x) \quad (1.6)$$

Взвешенное голосование («смесь экспертов»):

$$f(x) = \sum_{g=1}^{G} \alpha_g(x) \cdot f_g(x) \quad (1.7)$$

К примеру, наиболее распространенным является «правило большинства», при котором за групповое решение принимается альтернатива, получившая наибольшее число голосов.

Необходимо понимать, что такое решение отражает лишь распространенность различных точек зрения в группе, а не действительно оптимальный вариант, за который вообще никто может и не проголосовать.

Кроме того, существуют так называемые «парадоксы голосования», наиболее известный из которых парадокс Эрроу [22]. Дж. Эрроу сформулировал требования, выражающие понимание рационального коллективного выбора:

1. Предпочтения каждого индивида должны быть учтены. Не должно быть такого индивида, чье мнение считается главнее для всех, хотя у других индивидов противоположное мнение.

2. Если в результате группового выбора предпочтение было отдано какой-то альтернативе, то это решение не должно меняться, если кто-нибудь из ранее отвергавших ее изменил свое мнение в ее пользу.

3. Порядок альтернатив, решения относительно которых не изменялись в новом групповом упорядочении, не должен меняться.

4. Для любой пары альтернатив возможны такие два множества индивидуальных предпочтений, при которых порядок этих альтернатив противоположен.

Отмеченные требования являются важной предпосылкой рациональности индивидуального выбора. Но универсального правила рационального коллективного выбора, которое бы отвечало всем требованиям, нет. Анализ показал, что возможна ситуация зацикливания (то есть при определенной структуре индивидуальных преимуществ голосование может продолжаться бесконечно, не приводя к принятию однозначного решения) при последовательном осуществлении выбора тремя лицами, потому что при увеличении числа критериев упорядочивания растет вероятность того, что результаты окажутся зацикленными.

Эти парадоксы могут привести, и иногда действительно приводят, к очень неприятным особенностям процедуры голосования: например, бывают случаи, когда группа вообще не может принять единственного решения (нет кворума или каждый голосует за свой уникальный вариант и т.д.), а иногда (при многоступенчатом голосовании) меньшинство может навязать свою волю большинству.

Методы и алгоритмы формирования коллективов. Коллективные методы определяются двумя основными характеристиками: 1) алгоритмы, с помощью которых объединяются разные базовые модели (агенты); 2) методы, с помощью которых генерируются разные и разнообразные базовые агенты. В [23] предложена таксономия, различающая негенеративные коллективные методы (*Non-generative Ensembles Methods*), которые полагаются в основном на первую особенность коллективных методов, и генеративные коллективы (*Generative Ensembles Methods*), которые в основном фокусируются на второй особенности. Стоит отметить, что методы

комбинирования результатов и генерации базовых агентов (*"combination" and "generation"*) так или иначе присутствуют во всех коллективных методах. Различие между этими двумя большими классами заключается в преобладании компоненты комбинирования результатов или генерации агентов в коллективных методах.

Большинство ошибок, возникающих при обучении модели, связаны с тремя основными факторами: дисперсией, шумом и смещением. Используя коллективные методы, можно повысить стабильность окончательной модели и уменьшить ошибку. Комбинируя многие модели, можно уменьшить дисперсию, даже если они по отдельности невелики, поскольку уйдет зависимость от случайных ошибок из одного источника.

Основной принцип моделирования коллектива – объединение слабых моделей для формирования сильной метамодели. Слабый агент - алгоритм обучения, позволяющий работать с вероятностью ошибки меньшей, чем простое угадывание (0.5 для бинарной классификации). Сильный агент - алгоритм обучения, позволяющий добиться произвольно малой ошибки обучения.

Для поддержания разнородности моделей были предложены различные алгоритмы. Например, обучение на различных обучающих множествах или по различным признакам одного множества. Другим примером может служить обучение модели на том же самом обучающем и признаковом множестве путем генерирования различных структур (как в случае нейронных сетей).

Если коллектив состоит из базовых моделей одного типа, чаще всего используются алгоритмы бэггинг (Bagging, Bootstrap aggregation) и бустинг (Boosting) [24]. В случае использования различных базовых алгоритмов, обучаемых на одинаковых данных, результаты прогноза каждой модели используются для обучения мета-алгоритма, который определяет на каких фрагментах данных следует доверять той или иной базовой модели. Такой

подход к созданию коллектива моделей называется стэкинг (Stacking) [24]. Бустинг позволяет уменьшить дисперсию модели, бэггинг приводит к уменьшению смещения модели, а стэкинг позволяет увеличить прогнозирующую силу.

Основной идеей алгоритма бустинг [2, 8, 25, 26] является пошаговое наращивание коллектива алгоритмов. Алгоритм, который присоединяется к коллективу на шаге k обучается по выборке, которая формируется из объектов исходной обучающей выборки, то есть каждая последующая модель коллектива применяется к результатам на выходе предыдущей.

Бустинг основывается на идее о том, что различные алгоритмы обучения ошибаются на различных примерах. Тогда первый алгоритм в композиции будет обучаться на всём наборе, второй - на тех примерах, на которых первый допустил ошибку, третий - на тех примерах, где ошиблись первый и второй алгоритмы, и т.д.

Таким образом, метод бустинга реализует последовательную композицию алгоритмов обучения, в которой каждый алгоритм должен компенсировать ошибки, допущенные композицией предыдущих алгоритмов.

В настоящее время разработано достаточно большое количество методов бустинга, среди которых наиболее популярными являются:

1. AdaBoost.
2. Gradient Boosting Machine (GBM).
3. XGBoost.
4. Light GBM.
5. LPBoost.
6. TotalBoost.
7. BrownBoost.
8. MadaBoost.
9. LogitBoost.

В отличие от бустинга, бэггинг использует параллельное обучение базовых агентов (говоря языком математической логики, бэггинг – улучшающее объединение, а бустинг – улучшающее пересечение). Идея бэггинга состоит в том, что базовый алгоритм многократно обучается на случайных подвыборках с повторениями из обучающей выборки [1, 27, 28]. Такой метод генерации подвыборок принято называть бутстрэп (Bootstrap). В ходе бэггинга происходит следующее:

1. Из множества исходных данных случайным образом отбирается несколько подмножеств, содержащих количество примеров, соответствующее количеству примеров исходного множества.
2. Поскольку отбор осуществляется случайным образом, то набор примеров всегда будет разным: некоторые примеры попадут в несколько подмножеств, а некоторые не попадут ни в одно.
3. На основе каждой выборки строится модель.
4. Выводы моделей агрегируются (путем голосования или усреднения).

Стэкинг имеет два основных отличия от бэггинга и бустинга [6]. Во-первых, стэкинг обычно применяется к моделям, построенным с помощью различных алгоритмов, обучаемых на одинаковых данных, тогда как бэггинг и бустинг учитывают в основном однородные слабые модели. Во-вторых, стэкинг учит объединять базовые модели с использованием концепции метаобучения (т. е. пытается обучить каждого агента, используя алгоритм метаобучения, который позволяет обнаружить лучшую комбинацию выходов базовых моделей), тогда как бэггинг и бустинг объединяют слабые модели с помощью детерминистических алгоритмов [29, 30, 31, 32].

Идея стэкинга такова:

1. Разбить обучающую выборку на два непересекающихся подмножества.
2. Обучить несколько базовых агентов на первом подмножестве.

3. Тестировать базовые агенты на втором подмножестве.
4. Используя предсказания из предыдущего пункта как входные данные, а истинные классы объектов как выход, обучить мета-алгоритм обучения.

Использование коллективных моделей позволяет получить более точные результаты прогнозирования или классификации в сравнении с отдельными моделями в большинстве случаев. Совокупный результат нескольких моделей всегда менее шумный, чем отдельные модели. Это приводит к стабильности и надежности модели. Коллективные модели могут использоваться для захвата как линейных, так и нелинейных связей в данных. Это может быть достигнуто путем использования двух разных моделей и формирования их коллектива.

Недостатками коллективных методов являются:

1. Использование коллективных методов снижает способность интерпретации модели из-за повышенной сложности и делает очень трудным получение каких-либо важных бизнес-идей в конце.
2. Время вычислений и проектирования велико, что очень сильно влияет на разработку приложений реального времени.
3. Подбор моделей для создания коллектива моделей–отдельный трудоемкий процесс.

1.2 Формирование коллективов на основе нечеткой логики

Если отдельная модель слабо обучена, то ее можно усовершенствовать, изменив структуру алгоритма и параметры его обучения. Но эффективность принятого решения определяется не только эффективностью отдельных агентов, входящих в состав коллектива, но и типом внутриколлективных связей (коллективный вывод, отбор агентов в коллектив, распределения ресурсов между агентами).

Как правило, все коллективные методы принятия решения (КМПР) либо очень сильно зависят от выбора агентов, либо завязаны на специфику задачи, т.е. предназначены только для решения конкретного типа задач и не применимы (или «тиражируются») на других задачах. В силу таких особенностей проектирования коллективов моделей применение стандартных подходов зачастую не обеспечивает желаемой эффективности. Предлагаемый в диссертационной работе альтернативный коллективный подход основан на системе на нечеткой логике (Fuzzy Rule-Based System, FRBS).

В 1965 г. профессором Калифорнийского университета Лотфи А.Заде была опубликована работа «Fuzzy Sets», которая явилась начальным толчком к развитию новой математической теории. Л. Заде расширил классическое понятие множества, допустив, что функция принадлежности элемента множеству может принимать любые значения в интервале [0;1], а не только значения 0 либо 1. Такие множества стали называть нечеткими множествами (Fuzzy Sets) [33].

Рассмотрим основные определения, используемые в системах на нечеткой логике (Fuzzy Logic System, FLS).

- В [34] *нечетким множеством* A в некотором (непустом и четком) пространстве X, что обозначается как $A \subseteq X$, называется множество пар

$$A = \{(x, \mu_A(x)); x \in X\}, \text{ где} \mu_A: X \to [0,1] \qquad (1.8)$$

- *Функция принадлежности* (ФП) нечеткого множества A (рисунок 1.2).

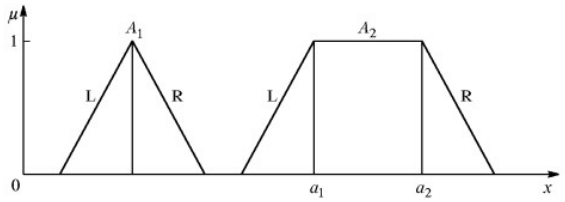

Рисунок 1.2 - Примеры графиков функций принадлежности

- *Нечеткие числа* – нечеткие переменные, определенные на числовой оси, т.е. нечеткое число определяется как нечеткое множество A на множестве действительных чисел R с функцией принадлежности $\mu_A(x) \in [0,1]$, где x - действительное число, т.е. $x \in R$ [34].
- *Лингвистической переменной* (ЛП) называется переменная, значениями которой могут быть слова или словосочетания естественного языка.
- *Терм – множеством* называется множество всех возможных значений ЛП.
- *Терм* – любой элемент терм – множества, который задается нечетким множеством посредством функции принадлежности.
- *Лингвистические переменные* характеризуются пятеркой (x, T(x), U, G, M), в которой x – имя переменной, T(x) – множество термов x, определенное на U, G – синтаксические правила для выработки имен величин x, M – семантические правила для связывания каждой величины с ее смыслом.

Набор ЛП для всех информативных признаков образует *нечеткое правило*.

Формирование нечеткой системы заключается в задании множества нечетких чисел, определяющих семантику используемых ЛП, и в формировании эффективного множества (базы) нечетких правил.

В базовой схеме системы на нечеткой логике (рисунок 1.3):

- в процессе фаззификации происходит преобразование численного значения в символьное нечеткое значение;

- на основе базы правил (БП, Rule Base, RB) осуществляется процедура нечеткого логического вывода с использованием объединений и пересечений;

- и четкое принятие решения – блок дефаззификация – преобразование нечеткого символьного значения в число.

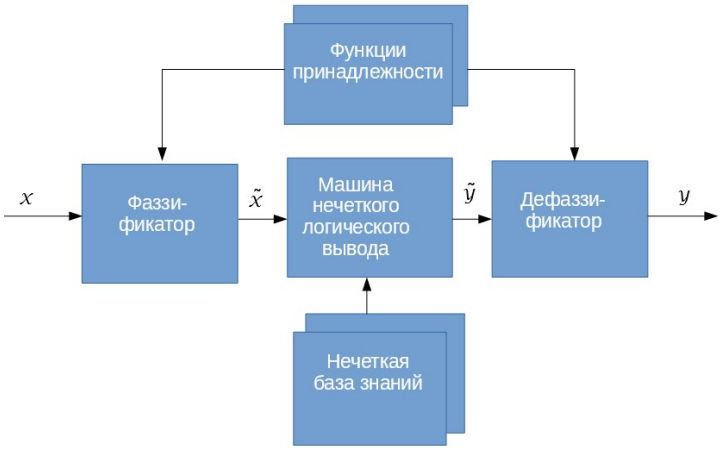

Рисунок 1.3. Базовая схема системы на нечеткой логике

Фаззификация:

1. Преобразует фиксированный вектор влияющих факторов (X) в вектор нечетких множеств $\tilde{X}$, необходимых для нечеткого вывода.

2. Выполняется построение функции принадлежности.

Примеры типовых форм кривых для задания функций принадлежности: треугольная, трапецеидальная и гауссова функции принадлежности.

Треугольная функция принадлежности определяется тройкой чисел (a, b, c), и ее значение в точке x вычисляется согласно выражению:

$$MF(x) = \begin{cases} 1 - \frac{b-x}{b-a}, a \le x \le b \\ 1 - \frac{x-b}{c-b}, b \le x \le c \\ 0, \text{в остальных случаях.} \end{cases} \quad (1.9)$$

Аналогично для задания трапецеидальной функции принадлежности необходима четверка чисел (a, b, c, d):

$$MF(x) = \begin{cases} 1 - \frac{b-x}{b-a}, a \le x \le b \\ 1, b \le x \le c \\ 1 - \frac{x-c}{d-c}, c \le x \le d \\ 0, \text{в остальных случаях.} \end{cases} \quad (1.10)$$

Гауссова функция принадлежности:

$$\mu(u) = \exp(-\frac{(u-b)^2}{2c^2}), \quad (1.11)$$

где b – координата максимума, с – дисперсия.

Нечеткий логический вывод включает в себя 2 части: базу нечетких правил и машину нечеткого вывода.

Основой для проведения операции нечеткого логического вывода являются БП, содержащая нечеткие высказывания в форме "*Если - то*" и функции принадлежности для соответствующих лингвистических термов. Для создания БП извлекаются знания из эксперта, либо она генерируется случайно, либо, получается в результате извлечения нечетких знаний из экспериментальных данных.

Результатом нечеткого вывода является четкое значение переменной y^*на основе заданных четких значений $x_k,\ k = 1..n$.

Рассмотрим подробнее нечеткий вывод на примере механизма Мамдани (Mamdani) [35]. В нем используется минимаксная композиция нечетких множеств. Данный механизм включает в себя следующую последовательность действий.

1. Процедура фаззификации. Для БП с m правилами обозначим степени истинности как

$$A_{ik}(x_k), i = 1..m, k = 1..n \quad (1.12)$$

2. Нечеткий вывод. Сначала определяются уровни "отсечения" для левой части каждого из правил:

$$alfa_i = \min_i(A_k(x_k)), i = 1..m, k = 1..n \quad (1.13)$$

Далее находятся "усеченные" функции принадлежности:

$$B_i^*(y) = \min_i(alfa_i, B_i(y)), i = 1..m \quad (1.14)$$

3. Композиция, или объединение полученных усеченных функций, для чего используется максимальная композиция нечетких множеств:

$$MF(y) = \max_i(\ B_i{}^*(y))\ \ i = 1..m \quad (1.15)$$

где MF(y) – функция принадлежности итогового нечеткого множества.

4. Дефаззификация, или приведение к четкости. Существует несколько методов дефаззификации. Например, центроидный метод.

Предлагаемый в данной главе альтернативный коллективный подход основан на системе на нечеткой логике для решения задач классификации и регрессии. Базовая схема на примере задачи классификации представлена на рисунке 1.4 [36].

Пусть имеется выборка $(\bar{x}_i, {y_i}^j), i = 1,2, \dots, n$, где n – объем выборки; $\bar{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})^T$ – набор значений независимых переменных, m – число переменных задачи; ${y_i}^j = f(\bar{x}_i)$ – соответствующая $\bar{x}_i$ зависимая переменная, $j = 1, v$ - конечное число номеров (имён, меток) классов. Требуется построить нечеткий контроллер, способный эффективно формировать алгоритмы (агентов) в коллектив.

Нечеткий контроллер (НК) принимает решение о выборе классификатора или алгоритма регрессии в зависимости от близости тестового объекта к объектам обучающей выборки и успешностью классификатора на ближайшем объекте.

Задача НК состоит в обеспечении минимальной погрешности решения задачи классификации или регрессии путем эффективного распределения вычислительных ресурсов между алгоритмами.

Рисунок 1.4. Базовая схема коллективного подхода на основе нечеткой логики для решения задачи классификации

При проверке работы алгоритма выборка $(\bar{x}_i, y_i)$ должна быть разбита на 2 части: обучающую и тестовую. Задача заключается в том, чтобы на основании имеющейся обучающей выборки синтезировать решающее правило, которое с минимальной ошибкой различало бы заданные классы, т.е. распознавало бы новые, ранее (при обучении) не встречавшиеся объекты. Тестовая выборка необходима для оценки качества решающего правила.

Для НК формируются три ЛП для входа и одна для выхода:

1. Лингвистическая переменная №1 (рисунок 1.5): *Distance* -близость объекта тестовой выборки к ближайшей точке из обучающей выборки.

Терм – множество = {близко, средне, далеко}.

Метрика ЛП: расстояние по евклидовой метрике.

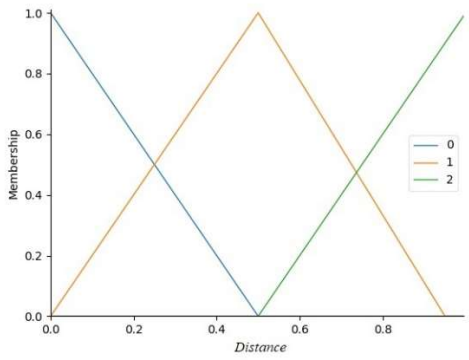

Рисунок 1.5. График базовой переменной для лингвистической переменной №1

2. Лингвистическая переменная №2 (рисунок 1.6): *Error* - разница между выходом модели (агента) на тестовой выборке и в ближайшей точке (точках) обучающего множества (ошибка агента на объекте выборки).

Терм – множество = {низкая, средняя, высокая}.

Метрика ЛП: Для задачи регрессии - расстояние по евклидовой метрике, а в случае задачи классификации – ошибка принимает значение 0, если агентом класс определяется верно, и 1 – если неверно.

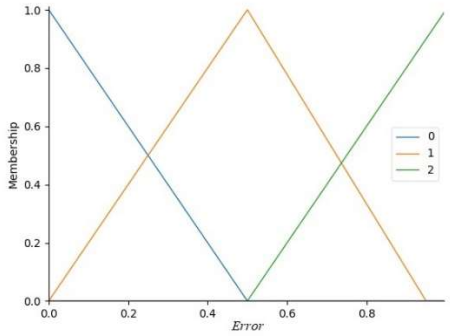

Рисунок 1.6. График базовой переменной для лингвистической переменной №2 (в случае задачи регрессии)

3. Лингвистическая переменная №3 (рисунок 1.7): *Weight_agent* – вес агента, который вычисляется пропорционально ошибке на обучающем множестве

Терм – множество = {низкий, высокий}.

Метрика ЛП: безразмерная величина в интервале от 0 до 1, где 0 – полное отсутствие доверия к алгоритму, а 1 – абсолютное доверие. Сумма весов по всем агентам равна 1.

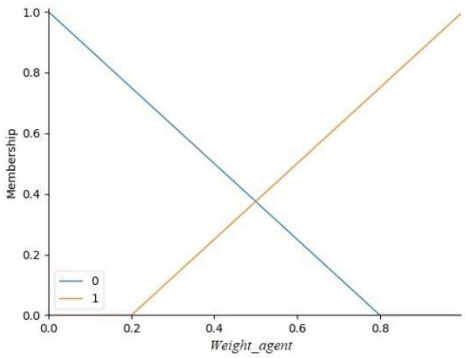

Рисунок 1.7. График базовой переменной для лингвистической переменной №3

При решении задачи регрессии точность каждого агента $Acc_1, Acc_2, \dots, Acc_k$ на обучающей выборке вычисляется на основе критерия - коэффициент корреляции согласованности (*Concordance Correlation Coefficient*) ρ_c:

$$\rho_C = \frac{2\rho \cdot \sigma_x \cdot \sigma_y}{\sigma_x{}^2 + \sigma_y{}^2 + (\mu_x - \mu_y)^2}, \quad (1.16)$$

где μ_x и μ_y средние значения двух переменных, а $\sigma_x{}^2$ и $\sigma_y{}^2$ – соответствующие дисперсии. ρ – коэффициент корреляции между двумя переменными.

В случае задачи классификации в качестве критерия эффективности используется *F – мера*.

Для задачи распознавания с неуравновешенными классами используются такие метрики как точность и полнота. Для описания метрик введём некоторые обозначения:

1. *Ttrue positive*: количество истинно положительных результатов.
2. *True negative*: количество истинно отрицательных результатов.
3. *False positive*: количество ложно положительных результатов.
4. *False negative*: количество ложно отрицательных результатов.

Точность (*Precision*) определяется как отношение числа корректно классифицированных элементов к общему числу элементов:

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \quad (1.17)$$

Полнота (*Recall*) это отношение числа найденных элементов целевого класса, к общему числу элементов целевого класса:

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (1.18)$$

F-мера ($F - measure,\ F1 - score$) позволяет объединить точность и полноту в одной усреднённой величине. F-мера представляет собой гармоническое среднее между точностью и полнотой, которая стремится к нулю, если точность или полнота стремится к нулю:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision \quad recall} \tag{1.19}$$

Пусть имеется Nагентов, $i = 1, N$.

В свою очередь вес *i* агента вычисляется следующим образом:

$$Weight\_agent_i = \frac{Acc_i}{\sum_{j=1}^{k} Acc_j} \tag{1.20}$$

4. Лингвистическая переменная №4 (рисунок 1.8): *Confidence* - степень доверия к агенту, которая вычисляется с помощью нечеткой процедуры вывода, учитывая 3 ЛП входа.

Терм – множество = {низкая, средняя, высокая}.

Базовая переменная: безразмерная величина в интервале от 0 до 1, где 0 – полное отсутствие доверия к алгоритму, а 1 – абсолютное доверие.

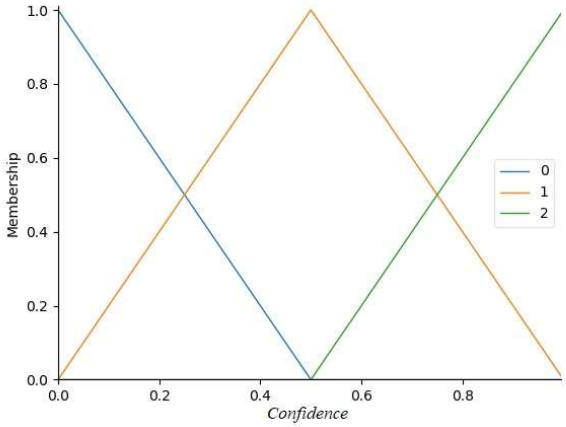

Рисунок 1.8. График базовой переменной для лингвистической переменной №4

Выходом работы НК для каждого объекта выборки из тестового множества является степень доверия к агенту. Нечеткий вывод степени доверия осуществляется по каждому агенту. Выбирается тот агент, степень доверия к которому выше.

БП является достаточно простой и интерпретируемой, и выглядит следующим образом:

IF *error* = высокая THEN *confidence* = низкая;

IF *error* = средняя AND *distance* = близко AND *weight_agent* = высокий THEN *confidence* = высокая;

IF *error* = средняя AND *distance* = средне THEN *confidence* = средняя;

IF *error* = низкая AND *distance* = близко AND *weight_agent* = высокий THEN *confidence* = высокая;

IF *error* = низкая AND *distance* = близко AND *weight_agent* = низкий THEN *confidence* = средняя;

IF *error* = низкая AND *distance* = средне AND *weight_agent* = высокий THEN *confidence* = высокая;

IF *distance* = далеко THEN *confidence* = низкая.

Обучение алгоритмов производится на всем обучающем множестве $X_{\text{обуч}}$.

Разрабатываемый алгоритм формирования коллектива с помощью нечеткой логики представлен ниже:

Алгоритм принятия решения с помощью нечеткого контроллера:

1. *На вход коллектива поступает объект тестовой выборки* $\bar{x}_{i\ \text{тест}}$, *значение выходов агентов для i-го объекта тестовой выборки* $y_{i\ \text{тест},k}$, $k = 1, N$, *где* N – *количество агентов в коллективе; обучающая выборка* $X_{\text{обуч}}$, *значение выходов объекта на обучающей выборке* $Y_{\text{обуч}}$, *значения выходов агентов* $Y_{k,\text{обуч}}, k = 1, N$.
2. *Вычисляются значения лингвистических переменных:*
   a. *Вычислить* $\rho(\bar{x}_{i\ \text{тест}}, x_j), \forall x_j \in X_{\text{обуч}}$. *Объект* x_j, *для которого* $\min\limits_{x_j \in X_{\text{обуч}}} \rho(\bar{x}_{i\ \text{тест}}, x_j)$ *назовем* x^* *и соответствующее ему значение выхода* y^*. *Вычислить значение функции принадлежности* $\mu_1\left(\min\limits_{x_j \in X_{\text{обуч}}} \rho(\bar{x}_{i\ \text{тест}}, x_j)\right)$ *по лингвистической переменной «Близость объекта обучающей выборки к тестовой выборке».*
   b. *Вычислить* $E(y^*, y_{i\ \text{обуч},k})$, $k = 1, N$. E – *функция показывающая близость реального выхода объекта* x^* *и*

соответствующего значения агента. E зависит от вида решаемой задачи (регрессия или классификация) и выбранной метрики. Вычислить значение функции принадлежности, $\mu_2{}^k\left(E(y^*,\ y_{i\ \text{обуч},k})\right),\ k = 1, N$. *по лингвистической переменной «Успешность классификатора (алгоритма регрессии) на объекте выборки».*

c. Вычислить $E(Y_{\text{обуч}}, Y_{\text{обуч},k})$, *где E – функция оценивающая точность k-го агента на обучающей выборке,* $k = 1, N$ *Вычислить значение функции принадлежности* $\mu_3{}^k\left(E(Y_{\text{обуч}}, Y_{\text{обуч},k}),\right)$ *по лингвистической переменной «вес агента».*

3. *На основе значений лингвистических переменных* $\mu_1{}^k, \mu_2{}^k,\ \mu_3{}^k$ *(где* $k = 1, N$ *– количество алгоритмов в коллективе) рассчитывается степень доверия(степень принадлежности) к каждому алгоритму методом центра тяжести* $\mu_k = F(\mu_1{}^k, \mu_2{}^k, \mu_3{}^k)$.
4. *На основе полученных значений принадлежности определяется k алгоритм в правиле с максимальной степенью доверия:*$k^* = arg\left(\max_k(\mu_k)\right)$.

Введем модификацию в описанный выше подход к формированию коллективной системы принятия решения на основе системы на нечеткой логике с помощью гибридизации FRBS и формирования итогового решения с помощью среднего (*mean*) и взвешенного среднего (*Wmean*), позволяющая получить решение точнее, чем решение лучшего агента из коллектива. Предложенный подход называется “*FRBS+Wmean*” или “*FRBS+mean*” (или “*FLS+Wmean*” или “*FLS+mean*”) в зависимости от способа формирования итогового решения (рисунок 1.9) [37].

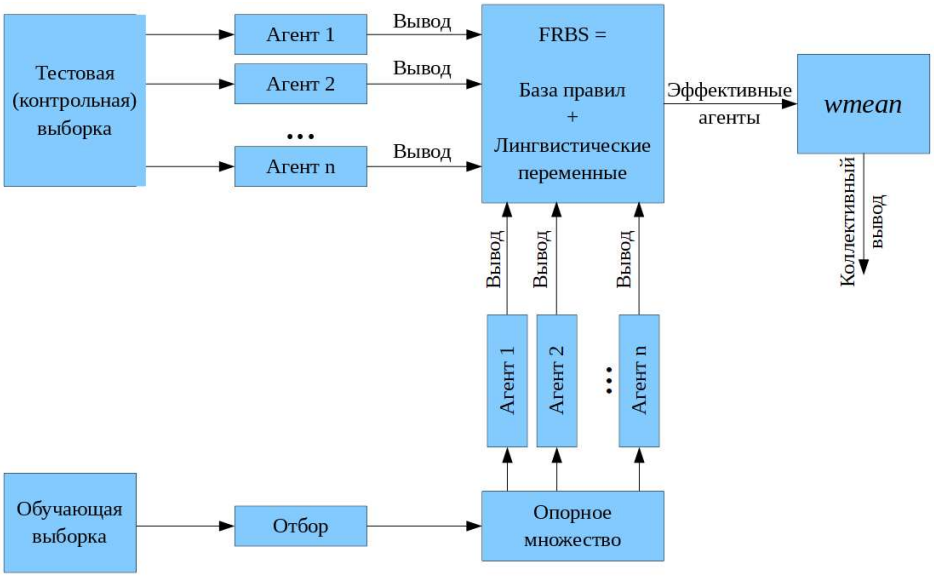

Рисунок 1.9. Общая схема коллективного вывода на основе подхода *“FRBS+Wmean”*

Варьируемыми параметрами алгоритма *“FRBS+Wmean”* являются два показателя: количество ближайших точек из обучающего множества к объекту из тестовой выборки (*nPoints*), а также количество агентов, композиция которых будет применяться для получения решения для объекта из тестовой выборки (*nAgent*).

Особенностью данной системы является то, что FRBS на основе опорного множества, принимает решение о выборе множества эффективных агентов *nAgent.*

При построении решающего правила для каждой тестовой точки, решение будет приниматься тем агентом, степень доверия к которому выше. Эта схема работает в том случае, если принятие решения происходит по одному агенту, то есть *nAgent* = 1.

В случае если *nAgent* >1, то для текущего решения с помощью системы на нечеткой логике отбирается *nAgent* лучших агентов (*nAgent* – параметр алгоритма), а из них итоговое решение принимается средним или взвешенным средним. При этом веса агентов определяются пропорционально их уверенности на данном примере.

Таким образом, система на нечеткой логике, позволяющая получить степень доверия агента, зависит от пяти параметров, переданных в НК:

$$FRBS(Distance, Error, Weigh\_agent, nAgent, nPoints) = \mu \quad (1.21)$$

Задача FRBS состоит в максимизации точности решения задачи регрессии путем эффективного распределения вычислительных ресурсов между агентами. На рисунке 1.9 представлена общая схема описанного выше подхода, для которой исходная выборка разбивается на 3 части: обучающая, тестовая и контрольная.

Из обучающей выборки производится отбор точек в опорное множество. Опорное множество позволяет оценить эффективность агентов в различных точках тестового пространства. То, как использовать информацию об эффективности агентов на точках опорного множества, и то как эту информацию проецировать на точки тестового пространства определяется с помощью FRBS. Формирование опорного множества может выполняться различными методами отбора. На основе обучающей выборки происходит обучение агентов в отдельности. На тестовой выборке происходит оценка эффективности обучения агентов и обучение базы нечетких правил в FRBS. На контрольной выборке происходит итоговая оценка эффективности работы системы в целом.

В качестве модификации алгоритма принятия решения с помощью НК можно принять различные изменения. С помощью кластеризации объектов обучающей выборки, к примеру, можно сократить число вычислений метрики, так как текущий объект выборки будет соотноситься уже не с каждым объектом обучающей выборки, а с центром кластера. В таком случае предпочтение отдается классификатору, показавшему наибольшую эффективность на данном кластере.

Традиционные методы классификации работают с объектами, параметры которых заданы исключительно в четком виде, что затрудняет их практическое использование при работе с объектами нечеткой природы. В

настоящее время для классификации подобных объектов активно развиваются методы, основанные на нечеткой логике. Данные методы формируют классы, границы которых размыты, а объект может одновременно относиться к нескольким из них с различными степенями принадлежности.

Когда объект принадлежит к разным кластерам в равной степени или не принадлежит ни одному, то алгоритм может ошибаться. Эту проблему можно успешно решить с помощью нечеткой классификации. В этом случае нечеткий вывод будет выполняться с помощью композиции классификаторов или алгоритмов регрессии. Вместо однозначного ответа на вопрос “к какому классу относится объект?”, он определяет уровень принадлежности того, что объект принадлежит к тому или иному классу. Таким образом, утверждение «объект А принадлежит к классу 1 с вероятностью 90%, к классу 2 — 10%» верно и более удобно.

1.3 Исследование эффективности коллективных систем на нечеткой логике на тестовых задачах

Исследование эффективности предложенной коллективной системы на основе нечеткой логики FRBS проводилось на основе известных наборов тестовых задач регрессии и классификации, взятых с репозиториев UCI [38] и KEEL [39] (описание характеристик представлено в таблице 1.1).

Набор данных «Elevators» относится к задаче управления самолетом F16. Целевая переменная связана с действием, выполняемым на элеваторах самолета.

Набор данных «MV» представляет собой искусственный набор данных с зависимостями между значениями атрибута. Целевые значения переменной выхода формируется на основе логических правил типа «IF-THEN».

Набор данных «House-16H» представляет собой задачу прогнозирования средней цены на дом в регионе на основе демографических показателей и состояния рынка жилья. Число в имени означает количество атрибутов наборов данных. Следующая буква обозначает некоторое приближение к сложности задачи.

Наборы данных «Pageblocks» - представляет собой задачу классификации областей распознанного текста, «Phoneme» - задачу классификации назальных и оральных звуков, а «Satimage» - задачу классификации пикселей на снимках со спутника.

Таблица 1.1. Задачи классификации и регрессии [38, 39]

Тип задачи	**№п/п**	**Наборы данных**	**Описание набора данных**
Задача регрессии	1	Elevators	16599 экземпляров выборки, 18 признаков
	2	MV	40768 экземпляров выборки, 10 признаков
	3	House-16H	22784 экземпляров выборки, 16 признаков
Задача классификации	1	Pageblocks	5472 экземпляров выборки, 10 признаков, 5 классов
	2	Phoneme	5404 экземпляров выборки, 5 признаков, 2 класса
	3	Satimage	6435 измерений, 36 переменных, 6 классов

Всего тестировалось 14 комбинаций алгоритма FRBS – подходы *“FRBS+Wmean”* и *“FRBS+mean”* со значением параметра *nAgent* варьирующимся от 1 до 7. Также был проведен сравнительный анализ эффективности с лучшим, худшим и средним агентом, а также с методами коллективного принятия решения *mean* и *Wmean*. Методы работали в одинаковых условиях с параметром *nPoints*=1.

Разбиение выборки производилось на обучающую (*learning*) – 60%, тестовую (*testing*) – 25% и контрольную (*validation*) – 15% от общего числа точек.

В таблице 1.2 представлены результаты работы *“FRBS+Wmean”* и *“FRBS+mean”* при решении задач классификации с указанием эффективности на основе критерия F-меры на тестовой (**Ptest**) и на контрольной выборке (**Pvalid**).

Для формирования коллектива в состав были включены агенты, построенные на основе библиотеки «Scikit-learn» (Python):

1. Коллектив деревьев решений методом градиентного бустинга (GBR).

2. Алгоритм k – ближайших соседей для задачи классификации (KNC).

3. Линейная модель метода опорных векторов для задачи классификации (LinearSVC).

4. Линейная регрессия для задачи классификации (LogisticR).

5-6. Искусственная нейронная сеть (многослойный персептрон) (MLP). Структура сети: 200x100x50x20, 200x200 нейронов на соответствующих слоях, сигмоидальная активационная функция.

7. Коллектив деревьев принятия решений методом «случайного леса» (RFR). Количество деревьев в коллективе: 10, 50. Глубина дерева: 18. Количество признаков, используемых одним деревом: 100.

8. Метод опорных векторов для задачи классификации (SVC).

По приведенным результатам можно судить о том, что на наборе данных «Pageblocks» предложенный подход показывает результат лучше лучшего из агентов при количестве агентов *nAgent*=1 и *nAgent*=2 и итогового вывода взвешенным средним *Wmean*. На наборах данных «Phoneme» и «Satimage» на тестовой выборке получен результат равный по значению лучшему агенту, а на контрольной результат несколько хуже лучшего агента, но сопоставимый с ним. Однако полученный результат значительно лучше среднего агента и лучше процедур *mean, Wmean*.

Таблица 1.2. Точность решения задач классификации с помощью коллективной системы на основе нечеткой логики FRBS

№ п/п	Коллективный метод	Задача классификации					
		«Pageblocks»		«Phoneme»		«Satimage»	
		Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
1	Лучший агент	0,958	0,957	0,897	**0,912**	0,911	**0,895**
2	Худший агент	0,884	0,892	0,725	0,729	0,725	0,760
3	Средний агент	0,924	0,927	0,779	0,784	0,803	0,809
4	*mean*	0,922	0,925	0,775	0,769	0,833	0,817
5	*Wmean*	0,938	0,930	0,794	0,771	0,850	0,836
6	*FRBS+M (nAgent=1)*	0,950	**0,961**	0,897	0,889	0,911	0,890
7	*FRBS+W (nAgent=1)*	0,950	**0,961**	0,897	0,889	0,911	0,890
8	*FRBS+M (nAgent=2)*	0,949	0,950	0,874	0,835	0,895	0,885
9	*FRBS+W (nAgent=2)*	0,950	**0,961**	0,897	0,889	0,911	0,890
10	*FRBS+M (nAgent=3)*	0,957	0,955	0,839	0,826	0,875	0,866
11	*FRBS+W (nAgent=3)*	0,959	0,954	0,839	0,826	0,880	0,881
12	*FRBS+M (nAgent=4)*	0,951	0,945	0,812	0,812	0,844	0,865
13	*FRBS+W (nAgent=4)*	0,957	0,957	0,839	0,829	0,880	0,869
14	*FRBS+M (nAgent=5)*	0,948	0,942	0,808	0,809	0,845	0,836
15	*FRBS+W (nAgent=5)*	0,952	0,946	0,808	0,809	0,845	0,836
16	*FRBS+M (nAgent=6)*	0,939	0,930	0,808	0,779	0,842	0,825
17	*FRBS+W (nAgent=6)*	0,953	0,946	0,808	0,809	0,845	0,836
18	*FRBS+M (nAgent=7)*	0,930	0,928	0,794	0,767	0,833	0,824
19	*FRBS+W (nAgent=7)*	0,940	0,930	0,794	0,767	0,842	0,833

В таблице 1.3 представлены результаты работы *“FRBS+Wmean”* и *“FRBS+mean”* при решении задач регрессии с указанием эффективности на основе критерия ρ_c

Таблица 1.3. Точность решения задач регрессии с помощью коллективной системы на основе нечеткой логики FRBS

№ п/п	Коллективный метод	Задача регрессии					
		«House-16H»		«Elevators»		«MV»	
		Ptest	Pvalid	Ptest	Pvalid	Ptest	Pvalid
1	Лучший агент	0,752	0,722	0,902	0,894	0,973	**0,966**
2	Худший агент	0,310	0,290	0,625	0,612	0,630	0,631
3	Средний агент	0,531	0,514	0,837	0,829	0,891	0,878
4	*Mean*	0,605	0,582	0,883	0,874	0,942	0,933
5	*Wmean*	0,617	0,588	0,887	0,877	0,947	0,938
6	*FRBS+M (nAgent=1)*	0,742	0,716	0,872	0,860	0,967	0,961
7	*FRBS+W (nAgent=1)*	0,742	0,716	0,872	0,860	0,967	0,961
8	*FRBS+M (nAgent=2)*	0,749	**0,723**	0,883	0,872	0,971	0,963
9	*FRBS+W (nAgent=2)*	0,749	**0,723**	0,883	0,872	0,971	0,963
10	*FRBS+M (nAgent=3)*	0,747	0,721	0,883	0,881	0,971	0,963
11	*FRBS+W (nAgent=3)*	0,747	0,721	0,883	0,881	0,971	0,963
12	*FRBS+M (nAgent=4)*	0,734	0,712	0,903	**0,899**	0,970	0,963
13	*FRBS+W (nAgent=4)*	0,734	0,712	0,903	**0,898**	0,970	0,963
14	*FRBS+M (nAgent=5)*	0,703	0,677	0,892	0,892	0,965	0,956
15	*FRBS+W (nAgent=5)*	0,704	0,678	0,892	0,892	0,965	0,956
16	*FRBS+M (nAgent=6)*	0,674	0,647	0,895	0,893	0,957	0,948
17	*FRBS+W (nAgent=6)*	0,675	0,648	0,894	0,893	0,958	0,949
18	*FRBS+M (nAgent=7)*	0,649	0,624	0,896	0,889	0,951	0,941
19	*FRBS+W (nAgent=7)*	0,652	0,627	0,896	0,889	0,952	0,942

. В состав коллектива также были включены агенты, построенные на основе библиотеки «Scikit-learn» (Python):

1. GBR.

2. Алгоритм k – ближайших соседей для задачи регрессии (KNR).

3. Линейная регрессия, которая строится на основе метрики L1 (LLasso).

4. Линейная регрессия, которая строится методом наименьших квадратов (LR).

5. Гребневая линейная регрессия, которая строится на основе метрики L2 (LRidge).

6. MLP. Структура сети: 200x100x50x20 нейронов на соответствующих слоях, сигмоидальная активационная функция.

7. RFR. Количество деревьев в коллективе: 10, 50. Глубина дерева: 18. Количество признаков, используемых одним деревом: 100.

8. Метод опорных векторов для регрессии (Support Vector Regression, SVR)

На наборе данных «MV» предложенный алгоритм коллективного принятия решения на основе нечеткой логики FRBS показывает сопоставимую эффективность с эффективностью лучшего агента, как на контрольной, так и на тестовой выборке. Но в случае задач «Elevators» (*nAgent*=4) и «House-16H» (*nAgent*=2) эффективность FRBS в комбинации с *mean* и *Wmean* выше, чем у лучшего агента.

Как в случае решения задач регрессии, так и классификации, значительное увеличение параметра алгоритма *nAgent*>5 не повышает эффективность решения задач.

ВЫВОДЫ

В данной главе были рассмотрены такие методы коллективного принятия решения как простое или взвешенное голосование, усреднение (взвешенное или невзвешенное), ранжирование, бэггинг, бустинг, случайный лес, блендинг, стэкинг и т.д. в задачах классификации и регрессии.

Предложен новый метод формирования коллективного решения на основе нечеткой логической системы. Основу логической системы составляет база нечетких правил, состоящая из семи правил, и три ЛП (*Error, Distance, Weight_agent*), описывающие успешность классификатора. Эффективность предложенного подхода была исследована на трех тестовых задачах классификации и трех тестовых задачах регрессии. Показано, что предложенный метод коллективного принятия решения решает задачу либо на уровне, сопоставимом с уровнем лучшего агента, либо лучше лучшего агента в коллективе. На всех задачах предложенный подход превзошел известные процедуры формирования коллективного принятия решения: простое и взвешенное голосование.

Эффективность формирования нечеткой системы FRBS для коллективного вывода зависит от состава коллектива и примеров, на основе которой происходит обучение каждого из агентов. Соответственно возникает задача обоснованного выбора агентов в состав коллектива и задача формирования опорного множества примеров для повышения эффективности решения задач классификации и регрессии.

2 Автоматизированное формирование состава коллектива и опорного множества

2.1 Эволюционная процедура автоматизированного формирования опорного множества

При формировании коллектива и обучении агента может использоваться вся обучающая выборка. Однако использовать всю обучающую выборку для определения степени уверенности агента на тестовой точке – является вычислительно затратным.

Высокая временная и вычислительная сложность коллективного подхода на основе нечеткой логики для решения задачи классификации или регрессии является следствием того, что для вычисления лингвистической переменной *Error* («близость объекта обучающей выборки к тестовой выборке») приходится использовать весь набор данных, то есть необходимо выполнять сравнение между каждой парой экземпляров в наборе данных для поиска ближайшего объекта. Формирование опорной выборки позволит сократить эти затраты.

В предложенную схему коллективного вывода на основе подхода *“FRBS+Wmean”* (рисунок 1.9.), описанную в первой главе, вводится понятие “опорное множество” (опорная выборка), под которым подразумевается подмножество обучающего множества [40]. Опорное множество будет использоваться в коллективном выводе с помощью системы на нечеткой логике, в соответствии с чем для точки из тестового множества определяется одна (в случае $nPoints=1$) или несколько ближайших точек (в случае $nPoints>1$) не из обучающего множества, а из опорного. В зависимости от того, насколько эта точка близка к объекту из тестового множества и насколько успешно справляется на ней алгоритм, определяется уверенность агента на данной тестовой точке. С помощью НК итоговое решение

принимает либо агент с наибольшей уверенностью, либо *nAgent* –лучших агентов методом усреднения.

Выбор экземпляров (*Instance Selection*) является предметом исследований, представляющим большой интерес в настоящее время, но проблема не была изучена в достаточной степени в отношении регрессионных наборов данных, в том числе из-за сложности этого типа набора данных [41]. В то время как в классификации число классов или значений, которые должны быть определены, обычно очень мало, выходная переменная в регрессии является непрерывной, так что число возможных значений для предсказания не ограничено [42].

Методы формирования выборок [43, 44] разделяют на методы выбора прототипа (*Prototype selection methods)* и методы построения прототипа (*Prototype construction methods)*. Здесь под прототипом подразумевается выделяемая подвыборка относительно исходной выборки. Методы выбора прототипа не модифицируют, а только отбирают наиболее важные экземпляры из исходной выборки. Данные методы в зависимости от стратегии формирования решений разделяют на методы с добавлением (*Incremental methods*) [45] и удалением (*Decremental methods*) [46]. Также отдельно выделяют методы фильтрации шума (*Noise filtering methods*) [47], методы конденсации (*Condensation methods*) [48, 49] и методы на основе стохастического поиска [50].

Методы построения прототипа на основе исходной выборки создают искусственные экземпляры, позволяющие описывать исходную выборку. Среди данных методов выделяют методы на основе кластер-анализа (*Cluster analisys based methods*) [51, 52], методы преобразования сжатием (*Data squashing methods*) [53] и нейросетевые методы (*Neural network based methods*) [54].

Общими недостатками данных методов является высокая итеративность и значительное время работы, а также неопределенность в

выборе параметров метода. Методы на основе кластер-анализа характеризуются такими недостатками, как неопределенность выбора числа кластеров, метрики и начальных параметров метода кластеризации и обучения. Методы преобразования сжатием формируют прототипы, которые сложно интерпретировать.

Коллективный метод принятия решения на основе нечеткой логики показал свою эффективность при решении задач регрессии и классификации [36, 37]. Однако данный метод при нечетком выводе использует опорное множество, позволяющее оценить уверенность агентов в каждой точке тестового пространства. Если в качестве опорного множества используется вся обучающая выборка это приводит к высоким вычислительным затратам.

Сокращение набора данных имеет две основных цели: уменьшение требований к вычислительным ресурсам, а также времени на обработку задачи обучения. Таким образом, выбор правильного подмножества примеров позволяет уменьшить размер выборки и повысить эффективность обучения.

В данной главе способы уменьшения объемов опорного множества при сохранении точности результатов. В качестве таких методов рассматриваются: генетический алгоритм, алгоритм кластеризации k-средних и случайный отбор экземпляров.

Формирование опорной выборки случайным образом (*Random Instance Selection, RIS*) для коллективного вывода. Одной из проблем, возникающих при анализе реальных наборов данных, является большое количество содержащихся в них примеров, что делает процесс обучения вычислительно дорогостоящим и предотвращает использование определенных алгоритмов в определенных наборах данных. Эта проблема еще более значима для алгоритмов, основанных на критериях соседства. Поиск окрестностей каждого экземпляра существенно возрастает вместе с размером набора данных. Очевидная альтернатива, которая поможет

противостоять этой проблеме, связанной с размером, и ускорит вычисление ближайшего соседа, состоит в сокращении количества экземпляров в обучающем наборе, при этом не допуская увеличения ошибки прогнозирования.

Таким образом, стратегия выбора экземпляра должна решать вопрос компромисса между объемом итогового множества (набора экземпляров) и качеством решения задачи регрессии или классификации.

Вероятностные методы [55, 56, 57] предполагают случайное извлечение набора экземпляров из исходной выборки, причем каждый экземпляр исходной выборки имеет ненулевую вероятность, быть включенным в формируемую выборку.

К вероятностным методам извлечения выборок относят:

– простой случайный отбор (*Simple random sampling*): из исходной выборки случайным образом отбирается заданное число экземпляров;

– систематический отбор (*Systematic sampling*): исходная выборка упорядочивается определенным образом и разбивается на последовательные группы экземпляров, в каждой из которых выбирается для включения в формируемую выборку объект с заданным порядковым номером в группе;

– стратифицированный отбор (*Stratification sampling*): исходная выборка разделяется на непересекающиеся однородные подмножества – страты, представляющие все виды экземпляров, в каждом из которых применяется случайный или систематический отбор;

– вероятностный пропорциональный к объему отбор (*Probability proportional to size sampling*): используется, когда имеется «вспомогательная переменная» или «метрика объема», которая предполагается связанной с интересующей переменной для каждого экземпляра, вероятность выбора для каждого элемента исходной выборки будет пропорциональна его метрике объема.

Формирование опорной выборки с помощью алгоритма кластеризации k-средних (k-means) для коллективного вывода. Детерминированные методы формирования выборок [56, 57, 58] предполагают извлечение экземпляров на основе предположений об их полезности (информативности), при этом некоторые экземпляры могут не быть выбраны или вероятность их выбора не может быть точно определена; они, как правило, основаны на кластерном анализе и стремятся обеспечить топологическое подобие исходной выборке. К детерминированным методам формирования выборок относят методы:

– удобного отбора (*Convenience sampling*): формирует нерепрезентативную выборку из наиболее легко доступных для исследования объектов;

– квотного отбора (*Quota sampling*): исходная выборка разделяется на отличающиеся свойствами подгруппы, после чего из каждой подгруппы выбираются объекты на основе заданной пропорции);

– целевого отбора (*Judgmental (purposive) sampling*): объекты извлекаются из исходной выборки исследователем в соответствии с его мнением относительно их пригодности для исследования.

Достоинствами данных методов являются их относительная простота и возможность оценки ошибки выборки, а недостатками – то, что они не гарантируют, что сформированная выборка малого объема будет хорошо отображать свойства исходной выборки, а также не будет избыточной и не будет искусственно упрощать задачу.

Формирование опорной выборки с помощью генетического алгоритма (ГА) для коллективного вывода. ГА являются эволюционными методами поиска, моделирующим процессы природной эволюции, которые используются для решения различных задач оптимизации, как простых, так и сложных систем [34]. Основой возникновения ГА послужила модель биологической эволюции и методы случайного поиска.

Формализованная постановка задачи безусловной однокритериальной оптимизации имеет следующий вид [59]:

Необходимо найти

$$x_{max} = arg \max_{x \epsilon X} f(\bar{x}), \quad (2.1)$$

где X – множество всех возможных решений; $f(\bar{x})$ – функционал, определенный на данном множестве, возвращающий действительное число из интервала $(-\infty; +\infty)$; $\bar{x} \in X$ имеет вид $\bar{x} = (x_1; \dots; x_i; \dots; x_n)^T$.

Главным преимуществом эволюционных алгоритмов (ЭА) является возможность решения задач за счет комбинирования элементов случайности и направленности аналогично тому, что происходит в природе. Еще несколько важных элементов, используемых в ЭА, это - моделирование процессов селекции, мутации и наследования [60].

Основные понятия ГА [61] описываются в таблице 2.1.

Таблица 2.1. Основные понятия ГА

Вещественная строка	Вектор-столбец x конечной длины n, компоненты которого могут принимать значения из множества $[a_i; b_i], (i = \overline{1,n})$.
Индивид (Фенотип)	Некоторое решение, выраженное в терминах поставленной задачи.
Хромосома (Генотип)	Решение задачи (вектор-столбец x конечной длины n, компоненты которого могут принимать значения из множества$[0; 1]$) . Состоит из генов.
Ген	Компонент x_i бинарного вектора $x \epsilon X$.
Поколение	Одна итерация работы ГА.
Популяция	Множество индивидов, обрабатываемых на текущем поколении.
Функция пригодности	Преобразованная целевая функция для использования в генетическом алгоритме на бинарных строках.
Пригодность	Значение функции пригодности

Основной алгоритм ГА состоит из следующих шагов [34]:

1. Инициализация исходной популяции хромосом.
2. Оценка приспособленности хромосом в популяции.

3. Формирование новой популяции из полученных популяций потомков и родителей.
4. Проверка условия остановки алгоритма. Если выполняется, алгоритм останавливается, переход к 6.
5. Применение генетических операторов:

    5.1) Селекция – отбор родителей из текущей популяции.

    5.2) Скрещивание – получение популяции потомков из текущей популяции.

    5.3) Мутация – полученные потомки подвергаются мутации.
6. Выбор «наилучшей хромосомы».

Переформулируем задачу оптимизации как задачу нахождения максимума некоторой функции $f(x_1, x_2, \dots, x_n)$, называемой функцией приспособленности (*Fitness function*). Она должна принимать неотрицательные значения на ограниченной области определения. Для работы ГА выбирают множество параметров оптимизации и кодируют их в последовательность конечной длины.

Особью (хромосомой) будет называться строка, являющаяся конкатенацией строк упорядоченного набора параметров (рисунок 2.1.):

1010 10110 101 … 10101

| x_1 | x_2 | x_3 | … | x_n |

Рисунок 2.1. Кодирование хромосомы в ГА

Совокупность "особей" - популяция, каждая из которых представляет возможное решение данной проблемы. Каждая особь оценивается мерой ее "приспособленности" согласно тому, насколько "хорошо" соответствующее ей решение задачи.

Алгоритм работает, пока не пройдет заданное количество итераций алгоритма, или пока не найдено решение соответствующей точности.

Рассмотрим основные операторы ГА.

Селекция - это оператор, посредством которого индивиды (т.е. хромосомы), имеющие более высокое значение целевой функции, получают большую возможность для спаривания и порождения потомков.

Пропорциональная селекция. Вероятность индивида быть отобранным пропорциональна его пригодности. Пусть f_i - пригодность индивида i, N - размер популяции, тогда $\bar{f}$ - средняя пригодность популяции.

$$\bar{f} = \frac{1}{N} \cdot \sum_i f_i \ , i = \overline{1,N} \tag{2.2}$$

В пропорциональной селекции индивид i выбирается для репродукции с следующей вероятностью:

$$p_i = \frac{f_i}{\sum_j f_j} = \frac{f_i}{\bar{f} \cdot N}, \ i = \overline{1,N}, \text{j}= \overline{1,N} \tag{2.3}$$

*Ранговая селекция***.** При применении ранговой селекции в задаче минимизации индивиды популяции ранжируются в соответствии с их пригодностью: $R_i < R_j$ если $f(x^i) \leq f(x^j)$.

Турнирная селекция. Пользователем задается размерность турнира N_{tur}. Случайным образом из популяции индивидов выбираются N_{tur} индивидов. Из выбранных N_{tur} индивидов для дальнейшей обработки отбирается индивид с наилучшей функцией пригодности.

Скрещивание – оператор, посредством которого индивиды меняются частями хромосом, таким образом, создавая хромосомы потомков.

Одноточечное скрещивание. Хромосомы родителей «разрезаются» в выбранной случайным образом общей точке и производится обмен правыми частями (рисунок 2.2.).

$P1$:	11	111
$P2$:	00	000
$P1'$:	11	000
$P2'$:	00	111

Рисунок 2.2. Процедура одноточечного скрещивания

Двухточечное скрещивание. Хромосома рассекается на две части и полученные куски обмениваются (рисунок 2.3.).

$P1$:	111	01	00
$P2$:	000	11	10
$P1'$:	111	11	00
$P2'$:	000	01	10

Рисунок 2.3. Процедура одноточечного скрещивания

Равномерное скрещивание. Каждый ген потомка выбирается случайным образом из соответствующих генов родителей.

Замечание: в этом случае родителей может быть больше двух, в том числе возможно участие всей популяции родителей в целом (Gene pool recombination)

Мутация – оператор внесения небольших изменений в значениях одного или нескольких генов в хромосоме. В двоичных хромосомах мутация состоит в инвертировании случайным образом выбранного бита генотипа (рисунок 2.4.). В генетических алгоритмах мутация применяется к генам с очень низкой вероятностью $p_m \epsilon [0.001, 0.01]$. Хорошим эмпирическим правилом считается выбор вероятности мутации из соотношения $p_m = \frac{1}{M}$, где M – число бит в хромосоме.

$P1$:	$011\mathbf{1}111$
$P1'$:	010111

Рисунок 2.4. Процедура мутации

В работе использовалось три типа мутации. Тогда, при высокой мутации вероятность мутации равна $\frac{3}{M}$, при средней – $\frac{1}{M}$, и при слабой – $\frac{1}{3M}$.

Элитизм - введение в популяцию лучшего индивида (группы лучших индивидов) предыдущей итерации [61].

В работе [62] представлено использование ГА для обнаружения выбросов. Этот подход не зависит от свойств и цели набора данных, поэтому его можно использовать как для задач классификации, так и для задач регрессии. Однако результаты получены только для двух очень маленьких наборов данных (35 экземпляров с 2 признаками и 21 экземпляр с 3 признаками).

В другой же работе [63] был представлен генетический подход с использованием довольно сложных эволюционных алгоритмов многокритериальной оптимизации до 300 000 оценок функции пригодности.

Совсем недавно, также следуя генетическому подходу, обратились к проблеме в рамках многокритериальных нечетких систем, основанных на правилах (FRBS [64]). Наконец, некоторые другие авторы сосредоточили свои усилия на адаптации методов выбора экземпляров к регрессии, которые изначально были предназначены для классификации. В [42] сверточные нейронные сети (Convolutional Neural Networks, CNN) и метод, в котором измерение исключается, если его класс отличается от класса ближайших соседей (Edited Nearest Neighbor, ENN), были адаптированы для работы с проблемами регрессии. Недавно коллективы отбора экземпляров выявили свои преимущества в классификации [65]. Адаптация коллективов отбора экземпляров к регрессии была сделана в работе [66], в которой авторы объединили свои собственные алгоритмы, чем смогли достигнуть лучшей производительности и более высокого сжатия.

В качестве эволюционной процедуры для автоматизированного отбора экземпляров в опорное множество из обучающего при решении задач классификации или регрессии предлагается использовать ГА безусловной однокритериальной оптимизации со специальной схемой кодирования:

- генотип хромосомы представляет собой двоичную последовательность;

- фенотип - целочисленные значения, которые определяют номер точки из обучающего множества для ее включения в опорное множество.

Количество точек, кодируемых в бинарную строку определяется пользователем.

2.2 Анализ эффективности выбора опорного множества

Для исследования эффективности способов уменьшения объемов опорного множества было рассмотрено несколько методов.

Одним из предлагаемых методов выбора экземпляров является *RIS*, который отбирает *nPoints* экземпляров из обучающего набора без учета значения их выхода. В опорное множество производится отбор экземпляров случайным образом в соотношении, заданном пользователем. Исследование производится с помощью 30 запусков, с последующим усреднением полученных результатов. При этом сформированное опорное множество является подмножеством исходного обучающего, элементы которого не должны повторяться.

В качестве детерминированных методов для отбора экземпляров в данной главе предлагается использовать *алгоритм кластерного анализа k-means*. Исходная выборка разделяется на кластеры, из группы экземпляров каждого кластера выбираются точки ближайшие к центру кластера для формируемой выборки. Параметр k – обозначает итоговое количество точек, отобранных в опорное множество. Из каждого кластера выбирается по одной точке из ближайших к центру. Для решения задач регрессии в этой работе параметр k варьируется от 10 до 1000 с шагом 50 и до 10000 с шагом 1000.

В свою очередь, *ГА* для автоматизированного формирования опорной выборки запускался со следующими параметрами: двухточечное скрещивание, турнирная селекция, средняя мутация. Количество индивидов - 30, количество прогонов – 10.

Исследование эффективности предложенного способа кодирования хромосомы ГА в сравнении с методами RIS и k-means проводилось на основе трех известных тестовых задачах регрессии: «House-16H», «MV» и «Elevators» [40]. В качестве критерия эффективности используется коэффициент ρ_c . Количество точек *nPoints* варьировалось от 10 до полного размера выборки, *nAgent* = 1.

Результаты, полученные для набора данных «House-16H» представлены на рисунках 2.5. и 2.6.

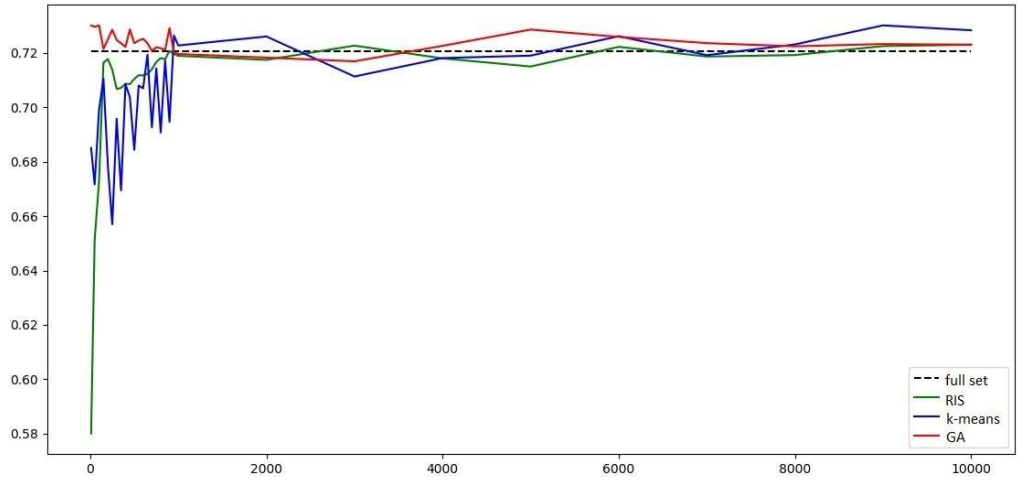

Рисунок 2.5. График зависимости точности работы коллектива от объема опорной выборки при решении задачи регрессии на наборе данных "House-16H"

по оси абсцисс представлено число точек, входящих в опорное множество, а по оси ординат – точность решения задачи регрессии на тестовой выборке

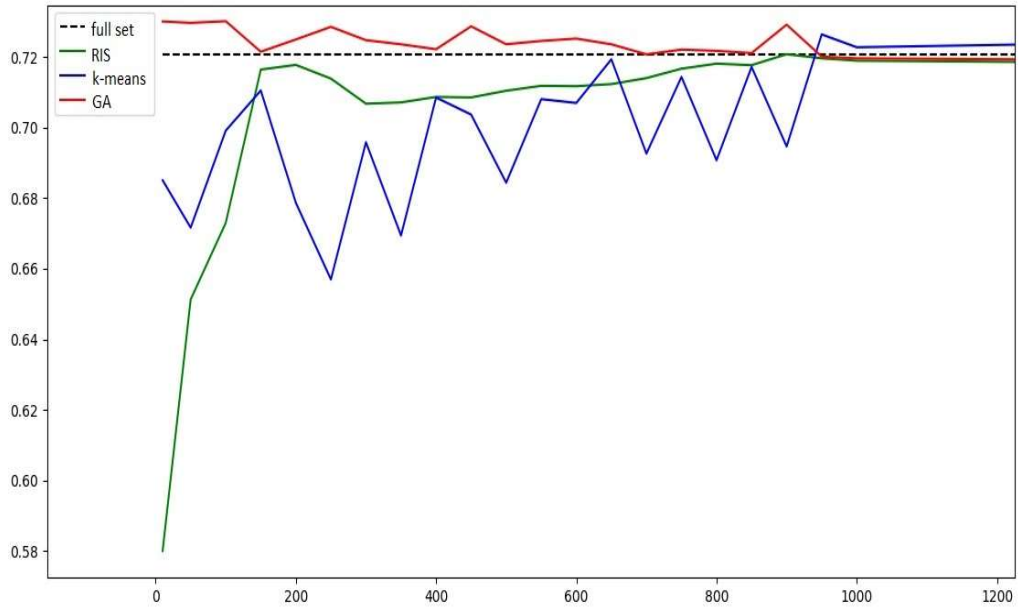

Рисунок 2.6. График зависимости точности работы коллектива от объема опорной выборки при решении задачи регрессии на наборе данных "House-16H" в интервале до 1200 точек
по оси абсцисс представлено число точек, входящих в опорное множество, а по оси ординат – точность решения задачи регрессии на тестовой выборке

Анализ результатов для набора данных " House-16H " показывает, что с помощью ГА можно сформировать опорное множество из точек объема от 10 до 100 точек, которое позволит формировать коллективный вывод эффективнее, чем при использовании всей обучающей выборки в качестве опорного множества. При этом опорное множество от 1000 и более можно формировать любым из предложенных методов. Однако RIS более прост в реализации и требует меньше вычислительных затрат.

В таблице 2.2 представлены результаты решения задачи регрессии на наборе данных "House-16H", полученные с применением алгоритмов отбора экземпляров в коллективном выводе на основе нечеткой логики на контрольной выборке (**Pvalid**).

Таблица 2.2. Сравнительный анализ эффективности используемых алгоритмов отбора экземпляров при коллективном выводе в задаче "House-16H"

Число точек опорного множества	**$Pvalid_{RIS}$**	**$Pvalid_{k\text{-}means}$**	**$Pvalid_{ГА}$**
10	0,580079	0,685055	**0,730037**
50	0,651289	0,671642	0,729628
100	0,672964	0,699105	**0,730108**
150	0,716403	0,710479	0,721461
200	0,717732	0,678685	0,724927
250	0,713852	0,656970	0,728510
300	0,706741	0,695821	0,724725
350	0,707101	0,669411	0,723560
400	0,708658	0,708478	0,722160
450	0,708514	0,703658	0,728651
500	0,710410	0,684368	0,723576
550	0,711803	0,708032	0,724505
600	0,711705	0,706948	0,725178
650	0,712293	0,719311	0,723561
700	0,713991	0,692613	0,720718
750	0,716657	0,714330	0,722064
800	0,718080	0,690725	0,721712
850	0,717652	0,717149	0,721078
900	**0,720749**	0,694630	0,729124
950	0,719593	**0,726363**	0,720083
1000	0,718916	**0,722722**	0,719590
2000	0,717425	**0,726031**	0,718261
3000	**0,722669**	0,711310	0,716895
4000	0,717970	0,718099	0,722583
5000	0,714987	0,718992	0,728567
6000	**0,722212**	**0,726126**	0,725872
7000	0,718660	0,719154	0,723582
8000	0,719195	**0,723101**	0,722413
9000	**0,722498**	**0,730115**	0,723250
10000	**0,723006**	**0,728305**	0,723018
15948	**0,720667**	**0,720667**	0,720667

Результаты, полученные для набора данных "MV", показаны на рисунках 2.7. и 2.8.

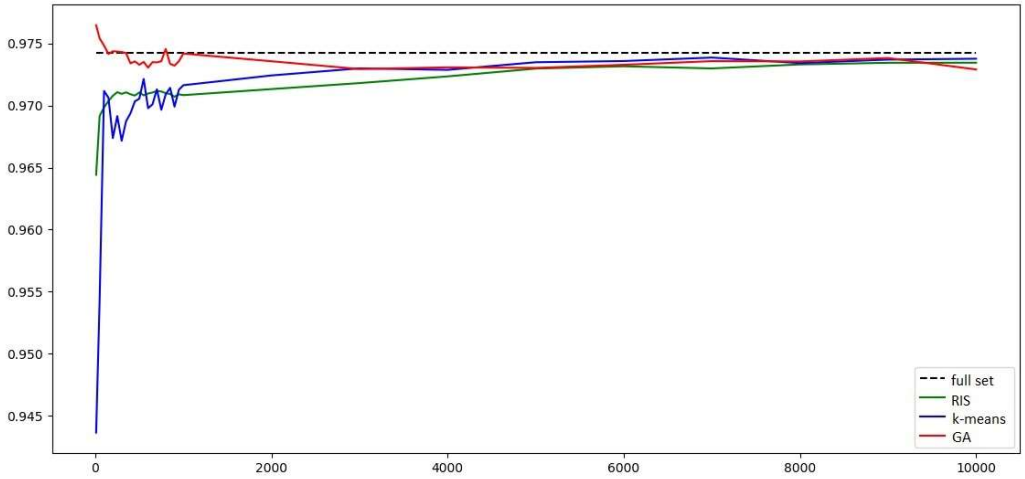

Рисунок 2.7. График зависимости точности работы коллектива от объема опорной выборки при решении задачи регрессии на наборе данных "MV" по оси абсцисс представлено число точек, входящих в опорное множество, а по оси ординат – точность решения задачи регрессии на тестовой выборке

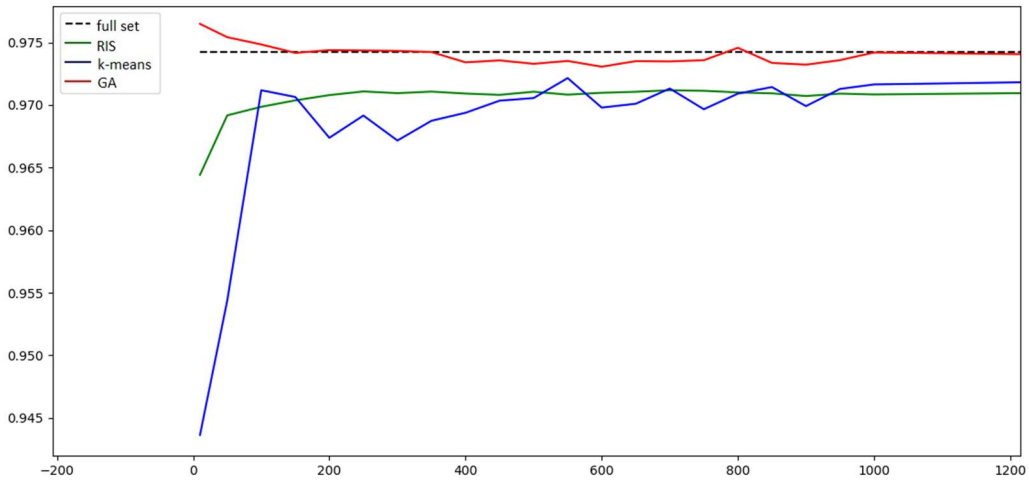

Рисунок 2.8. График зависимости точности работы коллектива от объема опорной выборки при решении задачи регрессии на наборе данных "MV" в интервале до 1200 точек

по оси абсцисс представлено число точек, входящих в опорное множество, а по оси ординат – точность решения задачи регрессии на тестовой выборке

В таблице 2.3 представлены результаты решения задачи регрессии на наборе данных "MV" на контрольной выборке (**Pvalid**).

Таблица 2.3. Сравнительный анализ эффективности используемых алгоритмов отбора экземпляров при коллективном выводе в задаче "MV"

Число точек опорного множества	**$Pvalid_{RIS}$**	**$Pvalid_{k\text{-}means}$**	**$Pvalid_{ГА}$**
10	0,964418	0,943627	**0,976483**
50	969171	0,954363	**0,975417**
100	0,969856	0,971180	0,974839
150	0,970370	0,970639	0,974165
200	0,970791	0,967377	0,974382
250	0,971084	0,969157	0,974356
300	0,970947	0,967161	0,974324
350	0,971069	0,968741	0,974232
400	0,970915	0,969378	0,973408
450	0,970810	0,970345	0,973565
500	0,971061	0,970553	0,973292
550	0,970827	0,972151	0,973521
600	0,970977	0,969793	0,973059
650	0,971055	0,970099	0,973505
700	0,971181	0,971315	0,973483
750	0,971136	0,969661	0,973578
800	0,971005	0,970909	0,974569
850	0,970931	0,971434	0,973365
900	0,970717	0,969912	0,973221
950	0,970909	0,971284	0,973583
1000	0,970843	0,971644	0,974197
2000	0,971330	0,972431	0,973576
3000	0,971815	**0,973004**	0,972946
4000	0,972355	0,972882	0,973085
5000	0,972985	**0,973496**	0,973040
6000	**0,973169**	**0,973592**	0,973290
7000	0,972998	**0,973873**	0,973585
8000	**0,973319**	**0,973427**	0,973560
9000	**0,973452**	**0,973708**	0,973836
10000	**0,973459**	**0,973780**	0,972916
40768	**0,974235**	**0,974235**	0,974235

При анализе набора данных "MV" ГА также показал большую эффективность в сравнении с другими методами на малом размере опорного множества. С ростом размера опорного множества эффективность ГА снижается. Это в первую очередь связано с существенным расширением пространства поиска, что в свою очередь требует больше вычислительных ресурсов.

Результаты, полученные для набора данных "Elevators", показаны на рисунках 2.9. и 2.10.

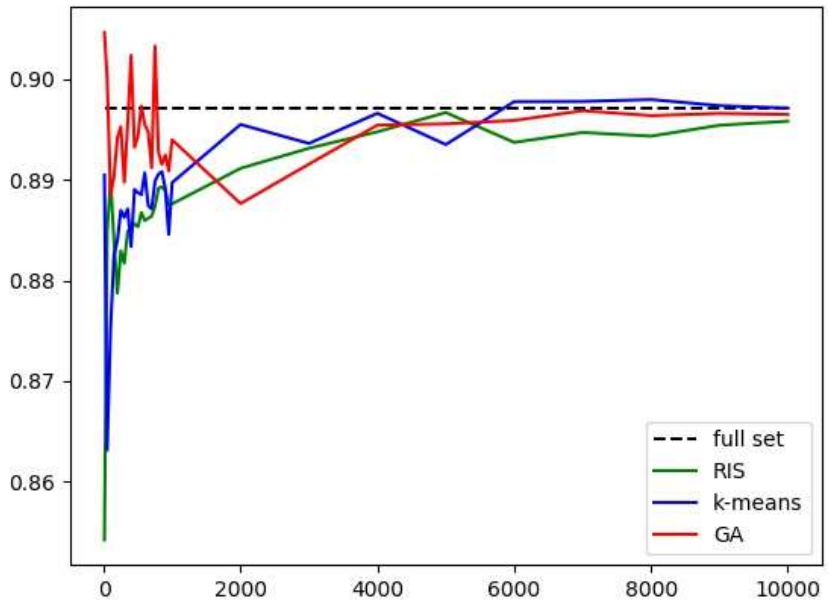

Рисунок 2.9. График зависимости точности работы коллектива от объема опорной выборки при решении задачи регрессии на наборе данных "Elevators"

по оси абсцисс представлено число точек, входящих в опорное множество, а по оси ординат – точность решения задачи регрессии на тестовой выборке

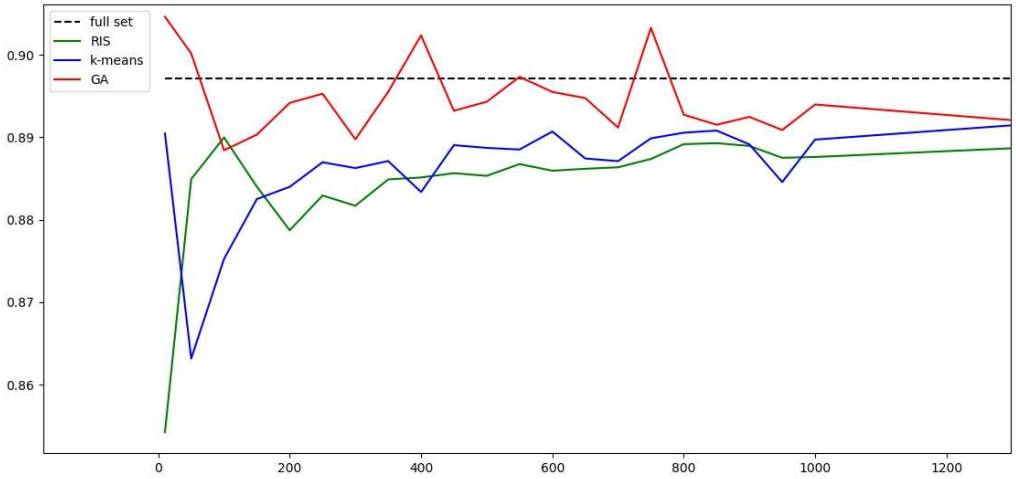

Рисунок 2.10. График зависимости точности работы коллектива от объема опорной выборки при решении задачи регрессии на наборе данных "Elevators"в интервале до 1200 точек
по оси абсцисс представлено число точек, входящих в опорное множество, а по оси ординат – точность решения задачи регрессии на тестовой выборке

В таблице 2.4 также представлены результаты решения задачи регрессии на наборе данных "Elevators" на контрольной выборке (**Pvalid**).

На наборе данных "Elevators" все методы показывают большой разброс в точности при формировании опорного множества малого объема. Однако при достаточном количестве вычислительных ресурсов ГА способен сформировать эффективное опорное множество малого объема. Для формирования опорного множества большего объема (6000 примеров и более) лучшим оказывается k-means, однако разница с остальными методами является не значительной, а вычислительные ресурсы в сравнении с RIS методом очень значительными.

Таблица 2.4. Сравнительный анализ эффективности используемых алгоритмов отбора экземпляров при коллективном выводе в задаче "Elevators"

Число точек опорного множества	**$Pvalid_{RIS}$**	**$Pvalid_{k\text{-}means}$**	**$Pvalid_{ГА}$**
10	0,854236	**0,890460**	**0,904618**
50	0,884926	0,863159	**0,900163**
100	0,889978	0,875251	0,888424
150	0,884003	0,882508	0,890321
200	0,878722	0,883989	0,894171
250	0,882937	0,886968	0,895278
300	0,881694	0,886263	0,889753
350	0,884893	0,887114	0,895513
400	0,885126	0,883358	**0,902354**
450	0,885643	0,889044	0,893218
500	0,885324	0,888713	0,894306
550	0,886750	0,888514	0,897321
600	0,885940	0,890698	0,895493
650	0,886181	0,887425	0,894752
700	0,886356	0,887105	0,891178
750	0,887367	0,889870	**0,903263**
800	0,889163	**0,890554**	0,892736
850	0,889285	**0,890813**	0,891515
900	0,888949	0,889135	0,892472
950	0,887511	0,884568	0,890878
1000	0,887609	0,889716	0,893972
2000	**0,891134**	**0,895491**	0,887634
3000	**0,893129**	**0,893609**	0,891552
4000	**0,894761**	**0,896601**	0,895424
5000	**0,896676**	**0,893497**	0,895551
6000	**0,893713**	**0,897755**	0,895890
7000	**0,894694**	**0,897778**	0,896856
8000	**0,894334**	**0,897966**	0,896367
9000	**0,895411**	**0,897358**	0,896587
10000	**0,895814**	**0,897115**	0,896485
16599	**0,897156**	**0,897156**	0,897156

В результате проведенных исследований на всех трех тестовых задачах регрессии можно сделать выводы о том, что сокращение опорной выборки не приводит к снижению точности работы коллектива, а в некоторых случаях

повышает ее. Это можно объяснить тем, что не всегда ближайшие точки к оцениваемому примеру являются показательными, например, из-за переобучения отдельных агентов или особенностей зависимости между входными параметрами и выходом.

2.3 Эволюционная процедура автоматизированного формирования состава коллектива

За счет правильного формирования опорного множества “плохие” агенты будут исключены. С другой стороны, за счет правильного выбора агентов даже в непоказательных точках опорного множества будет принято верное решение. Соответственно оба этапа (формирование опорного множества и отбор агентов) являются взаимосвязанными.

С содержательной точки зрения в КМПР каждый агент должен улучшать или не ухудшать значение своей функции полезности, или система в целом должна улучшать качество решения общей задачи.

Выбор агентов может производиться как случайным образом, так и детерменированными способами.

Случайный выбор агентов производится из моделей, обученных на одном и том же множестве, а градиентный бустинг в свою очередь позволяет наращивать состав коллектива таким образом, чтобы ошибка предыдущих агентов уменьшалась.

При формировании коллектива на основе нечеткой логики *“FRBS+Wmean”* обучение алгоритмов может производиться двумя способами. Алгоритмы обучения предлагаемого подхода для решения задачи классификации и регрессии представлены ниже:

Схема формирования коллектива (Алгоритм 1):

1. *Все члены коллектива (алгоритмы) обучаются на множестве* $X_{\text{обуч}}$.
2. *Формируется упорядоченный список алгоритмов следующим образом:*
   - 2.1. *Определить t=1, и* $Ensemble = \emptyset$, $\varepsilon > 0$ *- точность (порог), N – максимальное число алгоритмов в коллективе,* $X = X_{\text{обуч}}$.
   - 2.2. *Среди всех алгоритмов находится тот "*$Algorithm_t$*" для классификации у которого доля верно классифицированных объектов на* X *, больше. Для регрессии выбирается тот алгоритм, у которого ошибка на* X *меньше.* $Ensemble = Ensemble \cup Algorithm_t$.
   - 2.3. *Формируется новое множество* $X = X \setminus \{x_i : Algorithm_t(x_i) = y_i\}$. *Определить t=t+1.*
   - 2.4. *Вычисляется ошибка работы коллектива* $Error\ (Ensemble) = \frac{|X|}{|X_{\text{обуч}}|} * 100\%$.

Для задачи классификации $|X|$ *– множество объектов, неверно классифицированных коллективом алгоритмов из сформированного множества* $Ensemble$. *Для задачи регрессии* $|X|$ *– множество объектов, для которых значение ошибки меньше порогового, где порог задается пользователем классифицированных коллективом алгоритмов из сформированного множества* $Ensemble$.

   - 2.5. *Повторять 2.2. - 2.4. до тех пор, пока* $X \neq \emptyset$, *либо* $Error > \varepsilon$, *либо* $|Ensemble| < N$.

Схема формирования коллектива (Алгоритм 2):

1. *Определяется упорядоченный список классификаторов (алгоритмов регрессии)* $Algorithm$.
2. *Обучение происходит следующим образом:*

2.1. *Определить t=1, и* $Ensemble = \emptyset$, $\varepsilon > 0$ - *точность, N – максимальное число классификаторов (алгоритмов регрессии) в коллективе,* $X = X_{\text{обуч}}$.

2.2. $Algorithm_t$ *обучается на X.*

2.3. $Ensemble = Ensemble \cup Algorithm_t$.

2.4. *Формируется новое множество* $X = X \setminus \{x_i : Algorithm_t(x_i) = y_i\}$. *Определить t=t+1.*

2.5. *Вычисляется ошибка работы коллектива* $Error\ (Ensemble) = \frac{|X|}{|X_{\text{обуч}}|} * 100\%$. *Для задачи классификации* $|X|$ *– множество объектов, неверно классифицированных коллективом алгоритмов из сформированного множества* $Ensemble$. *Для задачи регрессии* $|X|$ *– множество объектов, для которых значение ошибки меньше порогового, где порог задается пользователем классифицированных коллективом алгоритмов из сформированного множества* $Ensemble$.

2.6. *Повторять 2.2. - 2.5. до тех пор, пока* $X \neq \emptyset$, *либо* $Error > \varepsilon$, *либо* $|Ensemble| < N$.

В схеме 1 обучение алгоритмов производится на всем обучающем множестве $X_{\text{обуч}}$. Коллектив строится из методов, которые показали наибольшую эффективность при решении задачи классификации (регрессии).

В схеме 2 обучение алгоритмов производится последовательно. На всем множестве $X_{\text{обуч}}$ обучается только первый алгоритм. Все оставшиеся объекты последовательно до обучаются следующим выбранным алгоритмом.

В обоих схемах алгоритм принятия решения производится на основе нечеткого контролера.

Так как успех коллективов алгоритмов зависит от набора соответствующих агентов и от их разнообразия, возникает необходимость разработки процедуры автоматизированного отбора агентов в состав коллектива. Избыток агентов в составе коллектива приводит к возникновению эффекта переобучения. Недостаток агентов может не привести к повышению точности решения задачи. В связи с этим возникает необходимость учета нескольких критериев.

Формирование состава коллектива с помощью эволюционных алгоритмов оптимизации. Эволюционные алгоритмы позволяют проектировать и отбирать эффективных агентов, исходя из степени их приспособленности к задаче и к достижению цели. Во время взаимодействия агентов осуществляется отбор наиболее успешных, которые затем используются для генерации нового поколения агентов, среди которых опять применяются те же самые процедуры оценки и отбора. В итоге наиболее успешное поколение решает задачу и достигает целей наиболее эффективным образом.

Многие проблемы проектирования и поддержки принятия решений можно сформулировать как проблемы оптимизации. И в большинстве реальных задач необходимо принимать решение, основываясь не на одном критерии, или показателе качества, а на их совокупности. При решении сложных задач классификации и регрессии коллективными методами необходимо учитывать совокупность двух критериев: точность решения задачи и ее вычислительная сложность.

Современные средства формирования классификационных и регрессионных моделей позволяют строить модели практически в автоматизированном режиме. И для одной задачи можно получить десятки эффективных агентов. Однако, с ростом количества агентов в коллективе уменьшается способность к обобщению и коллектив склонен к переобучению. Поэтому, при максимизации точности коллектива на

тестовых данных необходимо решать минимизацию количества агентов в коллективе.

В настоящее время генетические алгоритмы являются одним из наиболее эффективных средств решения задач многокритериальной оптимизации.

В качестве эволюционной процедуры для автоматизированного формирования состава коллектива с возможностью учета критериев предлагается использовать многокритериальный эволюционный генетический алгоритм (Multi-Objective Evolutionary Genetic Algorithm, MOEGA) - генетический алгоритм недоминируемой сортировки (Nondominated Sorting Genetic Algorithm, NSGA-II) [67], представленный в работе [37].

NSGA-II позволяет автоматизировать формирование состава коллектива, тем самым экономя вычислительные ресурсы (позволяет минимизировать количество агентов), и решать поставленные задачи достаточно качественно (повышая способность к обобщению результата).

В дополнение, этот алгоритм отличается относительно простой реализацией и настройкой, а также различными принципами поиска решений. NSGA-II основан на идее доминирования Парето. Практика применения NSGA-II и получаемые им высокие результаты, как правило, ограничиваются 2-, 3–критериальными задачами оптимизации. Так как при увеличении количества критериев сходимость имеет тенденцию к ухудшению точности. Причина этого видится в том, что с ростом числа критериев увеличивалась скорость заполнения популяции недоминируемыми решениями.

При формировании коллектива необходимо выбирать, кто из агентов будет входить в состав коллектива при принятии конечного решения. Состав коллектива с использованием MOEGA кодируется в бинарную строку, где один агент закодирован с помощью одного бита. Длина бинарной строки

постоянна, так как кодируется для максимального количества агентов, имеющих возможность включения в коллектив [37].

На рисунке 2.11. представлен образец кодирования бинарной строки MOEGA, длина которой равняется количеству всех имеющихся агентов *F*. Бинарная строка представляет собой набор нулей и единиц, где 0 – означает что этот агент не принимает участие в принятии решения, 1 – участвует.

Рисунок 2.11. Способ кодирования состава коллектива агентов для коллективного принятия решения

В качестве критериев используется коэффициент корреляции согласованности ρ_c для задачи регрессии (F-мера в случае задачи классификации) и количество агентов *F*, формирующих коллектив.

2.4 Анализ эффективности отбора агентов в коллектив

Исследование эффективности предложенного способа кодирования хромосомы для многокритериального алгоритма NSGA-II проводится на основе тестовых задач регрессии и классификации (описание характеристик представлено в таблице 1.1) со следующими параметрами запусков: 10 поколений, 50 индивидов.

В данной главе обучение агентов производится на всем обучающем множестве, а затем состав агентов в системе *“FRBS+Wmean”* и *“FRBS+mean”* формируется с помощью эволюционной процедуры. Параметры запусков: *nPoints*= 1, *nAgent* варьировался от 1 до 5. В качестве критериев эффективности используются коэффициент корреляции

согласованности ρ_c для задач регрессии и F-мера для задач классификации. Разбиение выборки проводилось в следующем соотношении: 60% - обучающая выборка, 25% - тестовая, а 15% - контрольная.

Сравнительный анализ эффективности решения задач регрессии и классификации с помощью FRBS с исходным количеством агентов (описанных в предыдущей главе) и с оптимизированным составом коллектива представлен в таблицах 2.5 - 2.10, где **Ptest** – точность на тестовой выборке, а **Pvalid** – на контрольной.

Таблица 2.5. Результаты решения задачи классификации «Pageblocks» с оптимизацией состава коллектива с помощью NSGA-II

№ п/п	**Коллективный метод**	**Задача классификации «Pageblocks»**			
		Без оптимизации		**NSGA-II**	
		Ptest	**Pvalid**	**Ptest**	**Pvalid**
1	Лучший агент	0,958	0,957	0,958	0,957
2	Худший агент	0,884	0,892	0,884	0,892
3	Средний агент	0,924	0,927	0,924	0,927
4	*mean*	0,922	0,925	0,956	0,955
5	*Wmean*	0,938	0,930	0,938	0,930
6	*FRBS+M (nAgent=1)*	0,950	**0,961**	0,960	**0,963**
7	*FRBS+W (nAgent=1)*	0,950	**0,961**	0,960	**0,962**
8	*FRBS+M (nAgent=2)*	0,949	0,950	0,949	0,948
9	*FRBS+W (nAgent=2)*	0,950	**0,961**	0,960	0,952
10	*FRBS+M (nAgent=3)*	0,957	0,955	0,958	0,955
11	*FRBS+W (nAgent=3)*	0,959	0,954	0,959	0,951
12	*FRBS+M (nAgent=4)*	0,951	0,945	0,952	0,945
13	*FRBS+W (nAgent=4)*	0,957	0,957	0,958	0,955
14	*FRBS+M (nAgent=5)*	0,948	0,942	0,949	0,942
15	*FRBS+W (nAgent=5)*	0,952	0,946	0,954	0,945

Таблица 2.6. Результаты решения задачи классификации «Phoneme» с оптимизацией состава коллектива с помощью NSGA-II

№ п/п	Коллективный метод	Задача классификации «Phoneme»			
		Без оптимизации		NSGA-II	
		Ptest	Pvalid	Ptest	Pvalid
1	Лучший агент	0,897	**0,912**	0,897	0,912
2	Худший агент	0,725	0,729	0,725	0,729
3	Средний агент	0,779	0,784	0,779	0,784
4	*mean*	0,775	0,769	0,875	0,880
5	*Wmean*	0,794	0,771	0,794	0,771
6	*FRBS+M (nAgent=1)*	0,897	0,889	0,912	**0,918**
7	*FRBS+W (nAgent=1)*	0,897	0,889	0,912	**0,920**
8	*FRBS+M (nAgent=2)*	0,874	0,835	0,880	0,867
9	*FRBS+W (nAgent=2)*	0,897	0,889	0,912	0,889
10	*FRBS+M (nAgent=3)*	0,839	0,826	0,855	0,847
11	*FRBS+W (nAgent=3)*	0,839	0,826	0,855	0,847
12	*FRBS+M (nAgent=4)*	0,812	0,812	0,812	0,828
13	*FRBS+W (nAgent=4)*	0,839	0,829	0,855	0,838
14	*FRBS+M (nAgent=5)*	0,808	0,809	0,808	0,805
15	*FRBS+W (nAgent=5)*	0,808	0,809	0,808	0,805

Таблица 2.7. Результаты решения задачи классификации «Satimage» с оптимизацией состава коллектива с помощью NSGA-II

№ п/п	Коллективный метод	Задача классификации «Satimage»			
		Без оптимизации		NSGA-II	
		Ptest	Pvalid	Ptest	Pvalid
1	Лучший агент	0,911	**0,895**	0,911	**0,895**
2	Худший агент	0,725	0,760	0,725	0,760
3	Средний агент	0,803	0,809	0,803	0,809
4	*mean*	0,833	0,817	0,884	0,881
5	*Wmean*	0,850	0,836	0,850	0,836
6	*FRBS+M (nAgent=1)*	0,911	0,890	0,911	0,890
7	*FRBS+W (nAgent=1)*	0,911	0,890	0,911	**0,894**
8	*FRBS+M (nAgent=2)*	0,895	0,885	0,895	0,888
9	*FRBS+W (nAgent=2)*	0,911	0,890	0,911	0,888
10	*FRBS+M (nAgent=3)*	0,875	0,866	0,880	0,876
11	*FRBS+W (nAgent=3)*	0,880	0,881	0,885	0,885
12	*FRBS+M (nAgent=4)*	0,844	0,865	0,871	0,868
13	*FRBS+W (nAgent=4)*	0,880	0,869	0,882	0,874
14	*FRBS+M (nAgent=5)*	0,845	0,836	0,858	0,831
15	*FRBS+W (nAgent=5)*	0,845	0,836	0,864	0,833

По результатам, представленным в таблицах 2.5 – 2.7. можно сделать вывод об эффективности применения процедуры автоматизированного формирования состава коллектива для каждой из трех задач классификации.

Таблица 2.8. Результаты решения задачи регрессии «House-16H» с оптимизацией состава коллектива с помощью NSGA-II

№ п/п	Коллективный метод	Задача регрессии «House-16H»			
		Без оптимизации		NSGA-II	
		Ptest	Pvalid	Ptest	Pvalid
1	Лучший агент	0,752	0,722	0,752	0,722
2	Худший агент	0,310	0,290	0,310	0,290
3	Средний агент	0,531	0,514	0,531	0,514
4	*mean*	0,605	0,582	0,767	0,741
5	*Wmean*	0,617	0,588	0,764	0,743
6	*FRBS+M (nAgent=1)*	0,742	0,716	0,752	0,721
7	*FRBS+W (nAgent=1)*	0,742	0,716	0,752	0,721
8	*FRBS+M (nAgent=2)*	0,749	**0,723**	0,760	**0,724**
9	*FRBS+W (nAgent=2)*	0,749	**0,723**	0,760	**0,724**
10	*FRBS+M (nAgent=3)*	0,747	0,721	0,762	**0,727**
11	*FRBS+W (nAgent=3)*	0,747	0,721	0,762	**0,727**
12	*FRBS+M (nAgent=4)*	0,734	0,712	0,747	0,720
13	*FRBS+W (nAgent=4)*	0,734	0,712	0,747	0,720
14	*FRBS+M (nAgent=5)*	0,703	0,677	0,713	0,683
15	*FRBS+W (nAgent=5)*	0,704	0,678	0,713	0,684

Таблица 2.9. Результаты решения задачи регрессии «Elevators» с оптимизацией состава коллектива с помощью NSGA-II

№ п/п	Коллективный метод	Задача регрессии «Elevators»			
		Без оптимизации		NSGA-II	
		Ptest	Pvalid	Ptest	Pvalid
1	Лучший агент	0,902	0,894	0,902	0,894
2	Худший агент	0,625	0,612	0,625	0,612
3	Средний агент	0,837	0,829	0,837	0,829
4	*mean*	0,883	0,874	0,894	0,897
5	*Wmean*	0,887	0,877	0,895	0,897
6	*FRBS+M (nAgent=1)*	0,872	0,860	0,885	0,894
7	*FRBS+W (nAgent=1)*	0,872	0,860	0,885	0,894
8	*FRBS+M (nAgent=2)*	0,883	0,872	0,893	0,894
9	*FRBS+W (nAgent=2)*	0,883	0,872	0,923	0,904
10	*FRBS+M (nAgent=3)*	0,883	0,881	0,926	0,903
11	*FRBS+W (nAgent=3)*	0,883	0,881	0,926	0,903
12	*FRBS+M (nAgent=4)*	0,903	**0,899**	0,925	**0,910**
13	*FRBS+W (nAgent=4)*	0,903	**0,898**	0,926	**0,910**
14	*FRBS+M (nAgent=5)*	0,892	0,892	0,886	0,900
15	*FRBS+W (nAgent=5)*	0,892	0,892	0,886	0,900

Таблица 2.10. Результаты решения задачи регрессии «MV» с оптимизацией состава коллектива с помощью NSGA-II

№ п/п	Коллективный метод	Задача регрессии «MV»			
		Без оптимизации		NSGA-II	
		Ptest	Pvalid	Ptest	Pvalid
1	Лучший агент	0,973	**0,966**	0,973	0,966
2	Худший агент	0,630	0,631	0,630	0,631
3	Средний агент	0,891	0,878	0,891	0,878
4	*mean*	0,942	0,933	0,972	**0,972**
5	*Wmean*	0,947	0,938	0,972	**0,972**
6	*FRBS+M (nAgent=1)*	0,967	0,961	0,971	0,971
7	*FRBS+W (nAgent=1)*	0,967	0,961	0,972	0,971
8	*FRBS+M (nAgent=2)*	0,971	0,963	0,972	**0,972**
9	*FRBS+W (nAgent=2)*	0,971	0,963	0,972	**0,972**
10	*FRBS+M (nAgent=3)*	0,971	0,963	0,972	**0,972**
11	*FRBS+W (nAgent=3)*	0,971	0,963	0,972	**0,972**
12	*FRBS+M (nAgent=4)*	0,970	0,963	0,971	0,971
13	*FRBS+W (nAgent=4)*	0,970	0,963	0,971	0,971
14	*FRBS+M (nAgent=5)*	0,965	0,956	0,964	0,964
15	*FRBS+W (nAgent=5)*	0,965	0,956	0,965	0,964

Результаты, представленные в таблицах 2.8 – 2.10. также показывают эффективность применения процедуры автоматизированного формирования состава коллектива при решении задач регрессии.

ВЫВОДЫ

В главе 2 предложена процедура формирования эффективного состава коллектива с помощью многокритериального ГА - NSGA-II. В ходе исследований показано, что выбор эффективных агентов с помощью многокритериальной эволюционной процедуры позволяет существенно повысить точность работы и обобщающую способность коллектива.

На примере решения задач регрессии показано, что для формирования малых опорных выборок (количество примеров менее 500) генетический алгоритм оказывается эффективнее других подходов RIS и k-средних по критерию ρ_c. При этом удается получить точность работы коллектива выше, чем на полной выборке. С ростом объема опорной выборки снижается разница в эффективности представленных методов. На достаточно больших объемах (8000 и более) разница между данными методами практически отсутствует.

Вычислительные эксперименты показывают, что при эффективном выборе точек в опорное множество можно существенно снизить затраты вычислительных ресурсов (количество вычислений целевой функции) при сохранении точности результата.

В результате проведенных исследований во второй главе показано, что автоматизированное формирование состава коллектива и опорного множества позволяет повысить эффективность решения задач классификации и регрессии.

В свою очередь, выбор агентов определяется базой нечетких правил, интегрированных в FRBS. Работа БП определяется ее параметрами, например, типом и количеством правил, операторами агрегации правил и операторами вывода. Работа же всей FRBS в совокупности зависит еще и от параметров ЛП. Соответственно возникает задача автоматизированного

выбора БП с соответствующими ЛП для повышения эффективности формирования FRBS.

3 Автоматизированное формирование интеллектуальной системы на основе нечеткой логики для коллективного принятия решения

3.1 Методы и алгоритмы автоматизированного формирования системы на нечеткой логике

Нечеткие системы были успешно применены к задачам в классификации [68], моделировании [69], управлении [70] и в значительном числе других приложений. В большинстве случаев ключом к успеху была способность нечетких систем включать экспертные знания человека.

Системы на нечеткой логике состоят из БП, содержащей нечеткие высказывания в форме "Если-То" и функций принадлежностей для соответствующих лингвистических термов. Также они содержат механизм логического вывода, который включает четыре этапа: введение нечеткости (фаззификация), нечеткий вывод, композиция и приведение к четкости, или дефаззификация.

Алгоритмы нечеткого вывода различаются главным образом видом используемых правил, логических операций и разновидностью метода дефаззификации. В настоящее время разработаны модели нечеткого вывода Мамдани [71], Такаги – Сугено [72], Ларсена и Цукамото [73].

В подходе описанном в первой главе *“FRBS+Wmean”* нечеткая система формируется таким образом, чтобы эффективно объединять агентов в коллектив.

Эффективность коллективного вывода зависит от выбора агентов. Выбор агентов определяется базой нечетких правил, интегрированных в FRBS. Работа БП определяется ее параметрами, например, типом и количеством правил, операторами агрегации правил и операторами вывода.

В работе [37] при построении коллективного решения БП построена вручную (размер БП = 7), хотя существует множество подходов к ее автоматическому формированию.

Автоматическое проектирование нечеткой системы во многих случаях можно рассматривать как процесс оптимизации или поиска. ГА является широко используемым методом глобального поиска, способный исследовать требуемое пространство с использованием доступных ресурсов. Известно, что ГА способен находить решения, удовлетворяющие заданному критерию качества, в сложных пространствах поиска.

Так как свойства поверхности отклика (целевой функции) заранее не известны, то предполагать, что они имеют удобные для классических алгоритмов качества (унимодальность, выпуклость и т.п.) заранее невозможно. Приходится исходить из того что эти свойства могут оказаться исключительно сложными (многоэкстремальность, несвязность, овражность и т.п.) а значит надо применять универсальные адаптивные алгоритмы глобальной оптимизации в ситуации типа «черный ящик». Это приводит к целесообразности использования стохастических адаптивных алгоритмов. Кроме того, в «наших задачах» встречаются переменные различного типа: бинарные, целочисленные, вещественные и т.п. В этой связи рациональным выбором являются генетические алгоритмы как стохастические адаптивные алгоритмы глобальной оптимизации с бинарным представлением решений, которое само по себе позволяет кодировать переменные всех типов.

Возможности поиска и способность ГА включать априорные знания расширили его использование при разработке большого спектра методов проектирования нечетких систем за последние несколько лет. В 1990-х были предложены эволюционные алгоритмы для автоматизации этапа получения знаний при проектировании нечетких систем [74] - [77].

Нечеткие системы, проектирование которых происходит с помощью эволюционных алгоритмов, называются генетическими нечеткими

системами (Genetic Fuzzy Systems, GFS) [78, 79]. Анализ литературы показывает, что наиболее эффективными типами GFS являются генетические системы с нечеткими правилами (Genetic Fuzzy Rule-Based Systems, GFRBS) [80], в которых настройка различных компонентов системы производится на основе нечетких правил (FRBS). Внутри GFRBS можно различать процессы оптимизации параметров и генерации правил.

В качестве алгоритма обучения как структуры ЛП, так и параметров нечетких моделей, рассматриваются иерархические генетические нечеткие системы (Hierarchical Genetic Fuzzy Systems, HGFS) [81]. В работе [82] предлагается подход на основе генетического программирования (ГП, Genetic Programming, GP) [83], который использует ЛП иерархическим образом (Genetic Programming based learning of COmpact and ACcurate fuzzy rule based classification systems for high dimensional problems, GP-COACHH). ГП также является мощным методом поиска, который работает путем моделирования эволюции структур с помощью естественного отбора на основе приспособленности и генетических операторов, таких как селекция, скрещивание и мутация.

Различные предложения могут быть найдены при использовании ГП для разработки нечетких наборов правил, которые внутренне представлены как синтаксические деревья с ограничениями по типу [84, 85, 86, 87]. В таких системах нечеткие правила представлены бинарными деревьями.

Нечеткий ГП, предложенный в [86], объединяет простой ГА, который работает с языком нечетких правил без контекста. Существуют специальные подходы, которые используют знание предметной области для определения функции и набора терминалов, которые являются составными блоками для нечетких правил [87].

Были также предложены подходы, при которых гибридизуются ГА или алгоритм имитации отжига (*Simulated annealing*) и ГП [88, 89].

В статье [90] предлагается коэволюционный подход к построению нечетких моделей Такаги – Сугено (TS). Нечеткая модель TS возникает из иерархического и совместного коэволюционного процесса оптимизации. Кроме того, в недавней литературе по GFS были представлены такие современные методы как рой частиц, дифференциальная эволюция и т. д., а также модификация стандартных компонентов ГА, такие как “niching” ГА, гибридные комбинации ГА и локального поиска. Niching - идея сегментирования популяции ГА на непересекающиеся множества (под этим подразумевается то, что покрывается несколько локальных оптимумов).

GFS были расширены с использованием многокритериальных эволюционных алгоритмов MOEGA [91, 92], для возможности учитывать сразу несколько критериев, а не один. Эти подходы известны как многокритериальные эволюционные нечеткие системы (Multi-Objective Evolutionary Fuzzy Systems, MOEFS), и они стали важной частью более общих GFS [93].

Основными критериями для нечетких систем являются:

- интерпретируемость, характеризующий различимость и понятность нечетких термов;

- точность, оцениваемая процентом правильно предсказанных объектов;

- сложность или компактность, выраженная числом нечетких правил.

При построении решения необходимо достигать компромисса между данными критериями, что приводит к набору альтернативных нечетких моделей. Эти гибридные подходы MOEFS [93] помимо трех вышеупомянутых критериев, могут включать в себя любые другие виды, такие как сложность системы, стоимость, время вычислений и дополнительные показатели производительности.

3.2 Генетический алгоритм многокритериальной оптимизации для отбора нечетких правил

При формировании БП для нечеткой системы необходимо решить две задачи: формирование исходного множества нечеткого набора правил и выбор правил из этого заданного (исходного) множества.

Если само генерирование нечетких правил может производиться даже случайным образом, то при отборе конечного множества нечеткой БП необходимо учитывать такие критерии как точность, выраженная путем вычисления среднеквадратичной ошибки правил (Mean Squared Error, MSE), сложность, выраженная в виде количества выбранных правил и другие.

В данной главе предлагается использовать двухступенчатый подход на основе эволюционных алгоритмов однокритериальной и многокритериальной оптимизации, именуемый Genetic Fuzzy Rule-Based Ensemble (GFRBE) [94]. Исходная генерация правил производится с помощью ГА однокритериальной оптимизации. В дальнейшем для отбора правил был использован и адаптирован многокритериальный алгоритм оптимизации NSGA-II [95].

Задача первой ступени, решаемая с помощью ГА однокритериальной оптимизации, сгенерировать максимально выразительные правила, это означает что одно правило кодирует в себе большое количество случаев.

Для этого используется специальный вид кодирования, который, во-первых, позволяет кодировать практически произвольное правило, а во-вторых, БП на первой ступени ГА состоят из малого количества правил, и в этом случае успешный индивид содержит хотя бы одно эффективное правило. В целом БП может плохо справляться с задачей, но если БП имеет хотя бы одно эффективное правило, то она уже может быть эффективной на второй ступени.

Задача же второй же ступени отыскать эффективные правила и сформировать из них единую БП с помощью MOEGA. При этом обе ступени должны работать последовательно.

Общая схема работы формирования и оптимизации базы правил с применением двухступенчатого подхода в процедуре нечеткого вывода GFRBE представлена на рисунке 3.1.

Рисунок 3.1. Общая схема коллективного принятия решения с помощью GFRBE.

Процесс оптимизации БП происходит в модуле логического вывода, которая выполняется в два этапа, как представлено на рисунке.

По окончанию работы первой ступени (ГА однокритериальной оптимизации), где каждый индивид представляет собой базу правил, в популяции из последнего поколения происходит ранжирование индивидов по пригодности, и пользователь выбирает N лучших, из которых будет производиться отбор правил с помощью NSGA-II. Количество правил Z в одной БП задается пользователем. В следствии чего, отбор правил

многокритериальным алгоритмом производится из множества (*Z* x *N)* правил. На выходе MOEGA, учитывая критерия эффективности, отбирает *M* правил.

Оценка эффективности ГА первой ступени производится на основе критерия - коэффициента корреляции согласованности ρ_c в случае задачи регрессии, и F – меры в случае классификации. Предлагаемый MOEGA учитывает два критерия: сложность, выраженная в виде количества выбранных правил и точность, выраженная соответствующим критерием эффективности.

Генетический алгоритм однокритериальной оптимизации для генерирования исходного множества правил. При проектировании системы на нечеткой логике возникают проблемы формирования (или выбора) базы правил. Для решения этой задачи успешно применяется ГА, который используется для нахождения базы нечетких правил, каждая из которых проверяется на эффективность путем определения ее функции пригодности согласно необходимым требованиям.

Для кодирования базы правил в ГА была выбрана бинарная система, количество правил в которой равно 7 (так же, как и в исходной БП [37]). Один индивид представляет собой базу правил. Количество индивидов и поколений также задается пользователем.

На рисунке 3.2. представлено кодирование хромосомы ГА (базы правил), в которой каждое правило состоит из предпосылки и выхода правила. Первая строка не является частью хромосомы, а представляет собой маску, позволяющую производить декодирование хромосомы и представления ее в виде готовой БП. Вторая строка представляет собой пример самой хромосомы. Кодирование хромосомы включает в себя кодирование связки в виде операторов «И» или «ИЛИ», кодирование номеров термов, включенных в предпосылку. 1- соответствующий терм включается, 0 – исключается.

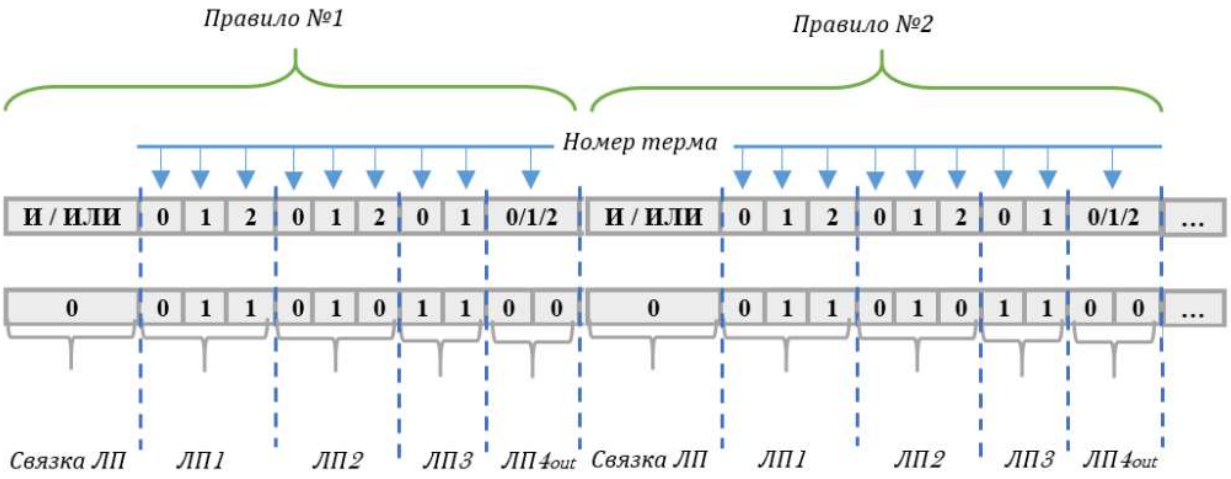

Рисунок 3.2. Структура кодирования хромосомы, которая представляет собой базу правил, для ГА однокритериальной оптимизации

ЛП входа *Distance (ЛП1)* и *Error (ЛП2)* описываются 3 термами, а *Weight_agent* (*ЛП3*) имеет 2 терма. На ЛП выхода *Confidence* (*ЛП4out*), которая имеет 3 терма, приходится 2 бита, то есть эта переменная может принимать следующие значения: 00 – 0 терм, 01 – 1 терм, 10 – 1 терм, 11 – 2 терм.

Количество бит, необходимое для представления 1 правила равно 11. Соответственно длина хромосомы (индивида), представляющего собой базу правил, имеет размерность (Z x 11) бит, где Z - количество правил.

В соответствии с рисунком 3.2. база правил интерпретируется следующим образом:

1. ЕСЛИ *Distance* = среднее или высокое И *Error* = средняя И *Weight_agent* = низкий или высокий ТО *Confidence* = низкая.
2. ЕСЛИ *Distance* = низкое ИЛИ *Error* = низкая или средняя ИЛИ *Weight_agent* = высокий ТО *Confidence* = низкая.

При вычислении функции пригодности в ГА используются элементы условной оптимизации, а именно применение «смертельного» штрафа к индивиду, т.е. метод, в котором недопустимые индивиды отбрасываются (т.е. пригодность данного индивида приравнивается нулю).

В предлагаемом подходе штраф назначается индивиду (то есть на всю базу правил), так как при случайном генерировании базы правил и в ходе эволюции могут получаться базы правил, которые невозможно вычислить. Например, когда для некоторых входящих значений (или для всех) в результате нечеткого вывода получается пустое нечеткое множество, к которому применить процедуру дефаззификации невозможно.

Генетический алгоритм многокритериальной оптимизации для выбора нечетких правил. Многие проблемы проектирования и поддержки принятия решений можно сформулировать как проблемы оптимизации. И в большинстве реальных задач необходимо принимать решение, основываясь не на одном критерии, или показателе качества, а на их совокупности.

В данной главе для выбора нечетких правил предлагается использовать эволюционный алгоритм многокритериальной оптимизации NSGA-II, система кодирования которого представлена на рисунке 3.3.

Бинарная система кодирования позволяет подобрать эффективные нечеткие правила для системы нечеткого вывода при коллективном принятии решений. Длина хромосомы (индивида), представляющая собой базу правил, равна Z x N бит, где Z – количество баз правил, отобранных после первой ступени. Из N баз происходит парсировка Z правил, каждому из которых соответствует один бит (значение 0 означает исключение правила из базы, а 1 – включение).

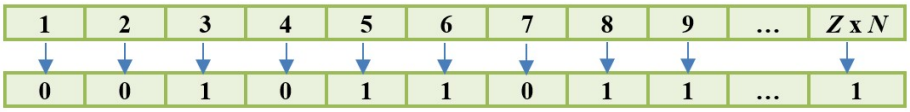

Рисунок 3.3. Структура кодирования индивида для MOEGA – NSGA-II

Каждый индивид MOEGA проверяется на эффективность на основе двух заданных критериев.

Критерием оптимизации NSGA-II является коэффициент корреляции согласованности ρ_c. (для задачи регрессии) или F-меры (для задачи классификации). Итоговая база правил также отбирается по соответствующему критерию, но если баз правил с максимальной точностью несколько, то из них выбирается база с наименьшим количеством правил.

3.3 Исследование эффективности автоматизированного формирования базы правил

В данной главе для генерирования базы правил был выбран ГА однокритериальной оптимизации со стандартными операторами: турнирная селекция, двухточечное скрещивание, средняя мутация. Параметры запусков для каждого этапа формирования БП: 10 поколений, 50 поколений. Параметры GFRBE: *nAgent* варьируется о 1 до 5, а *nPoints*=1.

Исследование эффективности двухступенчатого подхода к формированию БП на основе эволюционных алгоритмов однокритериальной и многокритериальной оптимизации при решении задач классификации представлен в таблицах 3.1 - 3.3, где **Ptest** – точность на тестовой выборке, а **Pvalid** – на контрольной (критерий эффективности – F - мера).

Результаты исследований представлены для исходной БП - RB_{basic} (7 правил), для сгенерированной БП на основе 1 ступени GFRBE - $RB_{ГА}$ (35 правил) и для БП, отобранной на основе 2 ступени GFRBE - $RB_{ГА+NSGA-II}$ (*NumRules* – количество правил, полученное по окончанию работы второй ступени GFRBE).

В качестве агентов GFRBE используются методы на основе библиотеки «Scikit-learn» (Python), описанные в первой главе.

Таблица 3.1. Результаты, полученные при решении задачи «Pageblocks» с использованием оптимизации БП в GFRBE

Коллективный метод	**«Pageblocks»**					
	RB_{basic} (7)		**$RB_{ГА}$ (35)**		**$RB_{ГА+NSGA-II}$**	
	Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
Лучший агент	0,958	0,957	0,958	0,957	0,958	0,957 NumRules = 7
Худший агент	0,884	0,892	0,884	0,892	0,884	0,892 NumRules = 7
Средний агент	0,924	0,927	0,924	0,927	0,924	0,927 NumRules = 7
mean	0,922	0,925	0,922	0,925	0,922	0,925 NumRules = 7
Wmean	0,938	0,930	0,938	0,930	0,938	0,930 NumRules = 7
GFRBE+M (nAgent=1)	0,950	**0,961**	0,960	**0,962**	0,960	**0,962** NumRules = 1
GFRBE+W (nAgent=1)	0,950	**0,961**	0,963	**0,962**	0,963	**0,962** NumRules = 4
GFRBE+M (nAgent=2)	0,949	0,950	0,954	0,953	0,954	0,953 NumRules = 2
GFRBE+W (nAgent=2)	0,950	**0,961**	0,955	0,954	0,955	0,930 NumRules = 2
GFRBE+M (nAgent=3)	0,957	0,955	0,964	**0,966**	0,964	**0,966** NumRules = 1
GFRBE+W (nAgent=3)	0,959	0,954	0,966	**0,965**	0,966	**0,965** NumRules = 2
GFRBE+M (nAgent=4)	0,951	0,945	0,958	**0,959**	0,958	**0,959** NumRules = 2
GFRBE+W (nAgent=4)	0,957	0,957	0,957	**0,964**	0,957	**0,964** NumRules = 3
GFRBE+M (nAgent=5)	0,948	0,942	0,957	0,953	0,957	0,953 NumRules = 2
GFRBE+W (nAgent=5)	0,952	0,946	0,953	0,953	0,953	0,953 NumRules = 3

На примере решения задачи классификации «Pageblocks» можно видеть, что гибридизация FRBS и формирования итогового решения с помощью среднего и взвешенного среднего позволяет найти решение точнее лучшего из агентов даже без оптимизации состава коллектива, а также позволяет улучшить результаты GFRBE при эффективном выборе количества агентов, принимающих решение. При этом, использование оптимизации БП в GFRBE позволяет уменьшить количество правил от одного до четырех без потери качества решения задачи.

Таблица 3.2. Результаты, полученные при решении задачи «Phoneme» с использованием оптимизации БП в GFRBE

Коллективный метод	**«Phoneme»**					
	RB_{basic} (7)		**$RB_{ГА}$ (35)**		**$RB_{ГА+NSGA-II}$**	
	Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
Лучший агент	0,897	**0,912**	0,897	0,912	0,897	0,912 NumRules = 7
Худший агент	0,725	0,729	0,725	0,729	0,725	0,729 NumRules = 7
Средний агент	0,779	0,784	0,779	0,784	0,779	0,784 NumRules = 7
mean	0,775	0,769	0,775	0,769	0,775	0,769 NumRules = 7
Wmean	0,794	0,771	0,794	0,771	0,794	0,771 NumRules = 7
GFRBE+*M (nAgent=1)*	0,897	0,889	0,907	**0,919**	0,907	**0,919** NumRules = 1
GFRBE+*W (nAgent=1)*	0,897	0,889	0,922	**0,919**	0,922	**0,919** NumRules = 2
GFRBE+*M (nAgent=2)*	0,874	0,835	0,885	0,880	0,885	0,880 NumRules = 1
GFRBE+*W (nAgent=2)*	0,897	0,889	0,903	0,853	0,903	0,853 NumRules = 2
GFRBE+*M (nAgent=3)*	0,839	0,826	0,842	0,818	0,842	0,818 NumRules = 2
GFRBE+*W (nAgent=3)*	0,839	0,826	0,845	0,821	0,845	0,821 NumRules = 3
GFRBE+*M (nAgent=4)*	0,812	0,812	0,812	0,802	0,812	0,802 NumRules = 3
GFRBE+*W (nAgent=4)*	0,839	0,829	0,855	0,833	0,855	0,833 NumRules = 2
GFRBE+*M (nAgent=5)*	0,808	0,809	0,808	0,809	0,808	0,809 NumRules = 4
GFRBE+*W (nAgent=5)*	0,808	0,809	0,897	0,912	0,897	0,912 NumRules = 2

Также для набора данных «Phoneme» можно видеть, что при уменьшении количества правил с помощью оптимизации БП в GFRBE сохраняется точность решения задачи. При этом при увеличении количества агентов *nAgent* > 1 эффективность решения задачи регрессии снижается.

Таблица 3.3. Результаты, полученные при решении задачи «Satimage» с использованием оптимизации БП в GFRBE

Коллективный метод	**«Satimage»**					
	RB_{basic} (7)		**$RB_{ГА}$ (35)**		**$RB_{ГА+NSGA-II}$**	
	Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
Лучший агент	0,911	**0,895**	0,911	0,895	0,911	0,895 NumRules = 7
Худший агент	0,725	0,760	0,725	0,760	0,725	0,760 NumRules = 7
Средний агент	0,803	0,809	0,803	0,809	0,803	0,809 NumRules = 7
mean	0,833	0,817	0,833	0,817	0,833	0,817 NumRules = 7
Wmean	0,850	0,836	0,850	0,836	0,850	0,836 NumRules = 7
GFRBE+*M (nAgent=1)*	0,916	0,893	0,916	0,893	0,916	0,893 NumRules = 2
GFRBE+*W (nAgent=1)*	0,916	0,886	0,916	0,886	0,916	0,886 NumRules = 2
GFRBE+*M (nAgent=2)*	0,914	0,876	0,914	0,876	0,914	0,876 NumRules = 1
GFRBE+*W (nAgent=2)*	0,916	0,904	0,916	**0,904**	0,916	**0,904** NumRules = 2
GFRBE+*M (nAgent=3)*	0,880	0,853	0,880	0,853	0,880	0,853 NumRules = 1
GFRBE+*W (nAgent=3)*	0,903	0,877	0,903	0,877	0,903	0,877 NumRules = 3
GFRBE+*M (nAgent=4)*	0,857	0,856	0,857	0,856	0,857	0,856 NumRules = 3
GFRBE+*W (nAgent=4)*	0,880	0,869	0,880	0,869	0,880	0,869 NumRules = 2
GFRBE+*M (nAgent=5)*	0,850	0,826	0,850	0,826	0,850	0,826 NumRules = 2
GFRBE+*W (nAgent=5)*	0,879	0,869	0,879	0,869	0,879	0,869 NumRules = 3

На примере задачи «Satimage» также наибольшая эффективность решения задачи классификации достигается при малом количестве агентов.

3.4 Построение базы нечетких правил и настройка семантики лингвистических переменных с помощью эволюционного алгоритма

Одной из задач при проектировании нечетких систем является оптимизация БП и терм-множеств для ЛП. Структура терм-множества ЛП характеризуется количеством и типом функций принадлежности, которому соответствует свое количество параметров. БП представляет собой набор продукционных правил, связывающих ЛП, которые состоят из предпосылок и заключений. Количество предпосылок в каждом правиле может меняться, в таком случае они объединяются посредством логических связок «И» или «ИЛИ». Проблема идентификации параметров нечетких моделей решается в том числе методами оптимизации нелинейных функций.

Эволюционные алгоритмы представляют собой универсальный метод оптимизации, который имитирует генетическую адаптацию в естественной эволюции [96, 97]. В отличие от специализированных методов, разработанных для определенных типов задач оптимизации, они не требуют особых знаний о структуре проблемы, кроме самой целевой функции. Популяция возможных решений развивается с течением времени с помощью генетических операторов, таких как мутация, скрещивание и селекция.

Различные подходы, такие как ГА и ГП, эволюционные стратегии (ЭС, Evolution Strategies, ES) и эволюционное программирование (ЭП, Evolutionary Programming, EP), дифференциальная эволюция (ДЭ, Differential Evolution, DE) отличаются генетическими структурами, которые подвергаются адаптации, и обладают генетическими операторами, с помощью которых они генерируют новые варианты.

При построении GFRBE процесс формирования и БП и ЛП требует больших вычислительных затрат, связанных с применением алгоритмов оптимизации. В свою очередь встает вопрос об эффективном распределении

ресурсов на каждый из этих этапов проектирования нечеткой модели. При этом также необходимо определить последовательность выполнения этапов оптимизации БП и ЛП.

Двухступенчатый подход к формированию БП и ЛП для GFRBS. В данной главе предлагается двухступенчатый подход к оптимизации БП с помощью ГА однокритериальной безусловной оптимизации, а ЛП с помощью ДЭ [98]. Построение нечеткой системы предлагается рассматривать следующим образом: на первом шаге генерируется начальная база правил и первоначальный набор лингвистических переменных. Затем производится формирование базы правил при помощи ГА однокритериальной оптимизации. На втором шаге под каждую сформированную базу правил производится оптимизация лингвистических переменных с помощью ДЭ. Оценка БП и ЛП производится с помощью процедуры нечеткого коллективного вывода. БП или ЛП устанавливаются в систему GFRBS и проверяются на тестовой выборке.

Предлагается две принципиально различные схемы. В первой схеме, как представлено на рисунке 3.4, этап оптимизации ЛП производится после оптимизации БП.

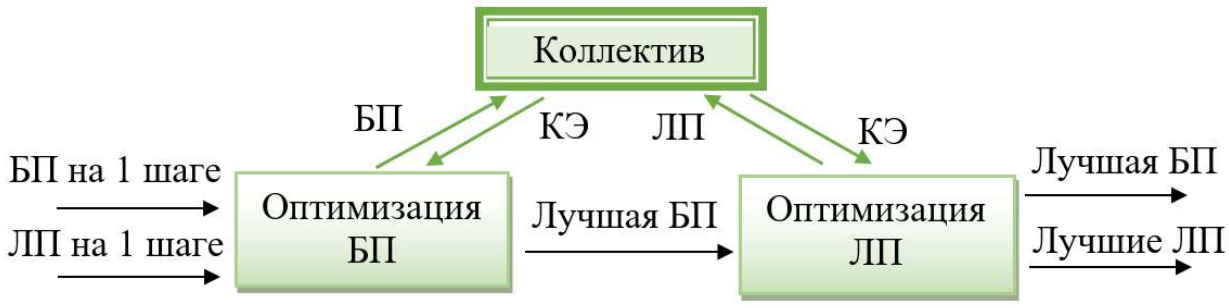

Рисунок 3.4. Пример схемы, в которой на 1 шаге производится оптимизация БП, а затем ЛП при проектировании системы на нечеткой логике для коллективного принятия решения (КЭ – критерий эффективности)

Во второй схеме построение нечеткой системы происходит наоборот, где этап оптимизации БП выполняется после оптимизации ЛП.

Оценивание эффективности индивида производится с помощью критерия качества формируемых БП и ЛП (ρ_c – в случае задачи регрессии, F-меры - для классификации).

На рисунке 3.5 представлено кодирование хромосомы ГА (БП) бинарной оптимизации, в которой каждое правило состоит из предпосылки и выхода правила. Первая строка указывает возможные варианты связки нечетких переменных, которая происходит с помощью операторов «И» или «ИЛИ», а вторая строка показывает бинарное представление, где значения 0 – означает оператор «И», а 1 – «ИЛИ». Данная связка применяется к каждому правилу своя.

Также первая строка указывает количество термов для каждой ЛП, на каждый из которых отводится по одному биту, соответственно во второй строке это отражается таким образом, что значение 0 означает исключение терма из правила, а 1 – включение.

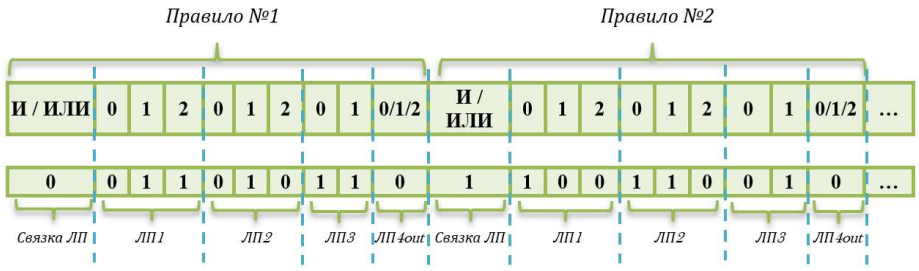

Рисунок 3.5. Структура кодирования хромосомы, которая представляет собой базу правил, для ГА однокритериальной оптимизации

ЛП входа *Distance* (ЛП1) и *Error* (ЛП2) описываются 3 термами, а *Weight_agent* (ЛП3) имеет 2 терма. На ЛП выхода *Confidence* (ЛП4out), которая имеет 3 терма, приходится 2 бита, то есть эта переменная может принимать следующие значения: 00 – 0 терм, 01 – 1 терм, 10 – 1 терм, 11 – 2 терм.

Количество бит, необходимое для представления 1 правила равно 11. Соответственно длина хромосомы (индивида), представляющего собой базу правил, имеет размерность (m x 11) бит, где m - количество правил. В данной работе m = 3.

Кодирование хромосомы ДЭ (ЛП) представлено на рисунке 3.6 (задача вещественной оптимизации), в которой содержатся параметры треугольной функции принадлежности *(a, b, c)* для каждого терма ЛП. Первая строка указывает номер терма для каждой ЛП, а вторая строка показывает вещественное представление параметров терма.

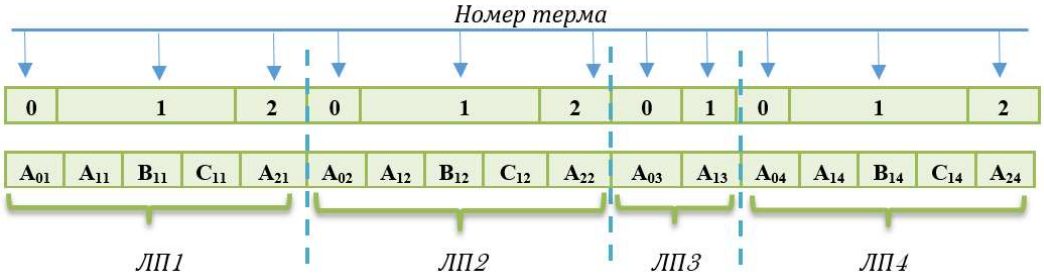

Рисунок 3.6. Структура кодирования хромосомы, которая представляет собой лингвистические переменные для ДЭ

Таким образом последовательное применение ГА для оптимизации БП, и ДЭ для оптимизации ЛП позволяет настраивать нечеткую систему коллективного вывода, в целом.

3.5 Исследование эффективности распределения ресурсов при оптимизации нечеткой системы управления

Исследование эффективности прямой (сначала БП, затем ЛП) и обратной схемы для формирования коллективного вывода с различной последовательностью этапов проектирования нечеткой системы проводилось

на основе трех тестовых задач: «House-16H», «MV» and «Elevators» (описание характеристик представлено в таблице 1.1.).

Каждая исходная выборка была разбита на 3 части: обучающая, контрольная и тестовая, в соотношении 60, 25 и 15 процентов.

Генетический алгоритм для БП и ЛП запускался с распределением ресурсов (количество индивидов) в разных соотношениях: 50/250, 100/200, 150/150, 200/100 и 250/50 для двух вариантов последовательностей проектирования системы на нечеткой логике для коллективного принятия решений.

Параметры ГА однокритериальной оптимизации: турнирная селекция, двухточечное скрещивание, средняя мутация. Параметры GFRBE: *nPoints*=1, *nAgent*=1.

Исследование эффективности двухступенчатого подхода к формированию БП и ЛП для GFRBE при решении задач регрессии представлен в таблице 3.4., где **Ptest** – точность на тестовой выборке, а **Pvalid** – на контрольной (критерий эффективности – коэффициент корреляции согласованности ρ_c).

На основе полученных результатов для набора данных «House-16H» можно сделать вывод о том, что порядок оптимизации (ЛП->БП или БП->ЛП) особенного значения не имеет. Вычислительных ресурсов на оптимизацию термов требуется больше, чем на оптимизацию правил. В первую очередь это объясняется различием в типах задач. Оптимизация БП является структурной оптимизацией и пространство поиска конечно. Оптимизация термов – параметрическая оптимизация. Пространство поиска хоть и ограничено, но бесконечно. Поэтому, схемы в которых на оптимизацию термов выделяется больше вычислительных ресурсов, оказываются эффективнее.

Таблица 3.4. Результаты решения тестовых задач регрессии при оптимизации БП и ЛП в GFRBE

Задача	Ресурсы для оптимизации ЛП	Ресурсы для оптимизации БП	БП->ЛП (Ptest/Pvalid)				ЛП->БП (Ptest/Pvalid)			
			БП		ЛП		ЛП		БП	
			1 этап		2 этап		1 этап		2 этап	
			Ptest	Pvalid	Ptest	Pvalid	Ptest	Pvalid	Ptest	Pvalid
House-16H	50	250	0,741	0,716	0,741	0,716	0,740	0,711	0,741	0,715
	100	200	**0,745**	**0,716**	**0,751**	**0,727**	0,743	0,714	0,745	0,715
	150	150	0,744	0,716	0,746	0,716	0,750	0,717	0,750	0,712
	200	100	**0,744**	**0,712**	**0,751**	**0,722**	0,751	0,724	0,751	0,724
	250	50	**0,738**	**0,708**	**0,753**	**0,727**	**0,750**	**0,717**	**0,756**	**0,729**
Elevators	50	250	0,915	0,908	0,915	0,908	0,915	0,908	0,915	0,908
	100	200	0,916	0,908	0,916	0,908	**0,916**	**0,906**	**0,915**	**0,909**
	150	150	0,915	0,908	0,915	0,908	**0,914**	**0,904**	**0,915**	**0,908**
	200	100	**0,916**	**0,901**	**0,924**	**0,918**	**0,916**	**0,908**	**0,924**	**0,919**
	250	50	**0,914**	**0,908**	**0,925**	**0,918**	**0,919**	**0,909**	**0,925**	**0,918**
MV	50	250	**0,965**	**0,962**	**0,974**	**0,977**	**0,954**	**0,963**	**0,976**	**0,976**
	100	200	**0,964**	**0,961**	**0,976**	**0,976**	**0,965**	**0,965**	**0,976**	**0,972**
	150	150	**0,960**	**0,966**	**0,976**	**0,976**	**0,964**	**0,971**	**0,976**	**0,975**
	200	100	**0,964**	**0,964**	**0,976**	**0,972**	**0,970**	**0,971**	**0,977**	**0,975**
	250	50	**0,970**	**0,969**	**0,975**	**0,976**	**0,968**	**0,971**	**0,976**	**0,976**

Для набора данных «Elevators» получены достаточно высокие показатели точности при обеих схемах оптимизации. Как видно из таблицы 3.4 точность итогового решения слабо зависит от порядка оптимизации (ЛП->БП или БП->ЛП). Однако, те схемы, в которых на оптимизацию лингвистических переменных выделяется больше вычислительных ресурсов, оказываются успешнее.

Набор данных «MV» является достаточно простой задачей для коллектива, поэтому при любой схеме и с любым распределением ресурсов получается достаточно высокий уровень точности. Выделить какие-то явные зависимости точности результата от схемы оптимизации или схемы распределения ресурсов, на данной задаче не получается.

ВЫВОДЫ

В рамках данной работы были предложены и исследованы две альтернативные схемы оптимизации нечеткой системы управления коллективным выводом. В результате исследования было показано, что порядок последовательного применения операторов оптимизации БП и ЛП существенного значения не имеет. Однако распределение ресурсов при структурной оптимизации БП и параметрической оптимизации ЛП имеет значение. На двух задачах из трех победили схемы, в которых больше вычислительных ресурсов выделялось на оптимизацию ЛП. Такой эффект в первую очередь объясняется различиями в структуре и размерности поисковых пространств в указанных задачах. На третьей задаче не выявлено существенного различия в схемах настройки нечеткой системы коллективного вывода. Это может быть связано с тем, что данные по этой задаче были синтетическими и природа зависимости между входами и выходами не является сложной.

В результате проведенных исследований эффективности применения двухступенчатого подхода к оптимизации БП в GFRBSE можно сделать вывод о том, что оптимизация БП позволяет получать высокую точность работы коллектива.

Применение же второй ступени ГА не позволяет повысить точность коллективного решения, однако позволяет существенно сократить количество правил в итоговой базе. Все полученные результаты позволяют сделать вывод об увеличении точности работы коллектива при оптимизации состава правил с помощью GFRBE.

4 Практическая реализация и исследование эффективности коллективной системы принятия решения на основе нечеткой логики

4.1 Исследование эффективности коллективных систем на нечеткой логике в задаче распознавания лиц по их изображению

Автоматизированная человеко-машинная система, работающая в режиме диалога, называется диалоговой системой. В режиме диалога такая система отвечает на все команды пользователя, а также обращается к нему за подтверждением выполнения конкретных действий или за получением дополнительной информации [99].

В настоящее время диалоговые системы применяются в самых разных областях человеческой деятельности. В результате человеко-машинной коммуникации можно определять многие параметры, например, возраст, пол, эмоции и другие характеристики человека [100].

Задача распознавания образов является основной задачей интеллектуального анализа данных.

Человеческое лицо играет центральную роль в социальном взаимодействии, поэтому не удивительно, что автоматическая обработка информации для лица является важной и очень активной областью исследования. Распознавание лиц представляет из себя частный случай задачи распознавания образов [101, 102].

Решение задачи распознавания лиц производится во многих системах компьютерного зрения. Самыми распространенными системами являются системы биометрической верификации и идентификации. Область применения таких систем – биометрические сканеры и паспорта, системы технического зрения роботов и т. д.

Распознавание лиц по их изображению производится в два этапа:

1. Определение локальной области лица человека на изображении;
2. Идентификация личности на основе изображения лица.

Сложность этой задачи обусловлена некоторыми факторами:

1. Изменчивость изображения от освещения (направленности, типа и количества источников света);
2. Различные пространственные положения (относительно камеры) и масштабы лиц на их изображениях;
3. Высокая вариативность лиц из-за анатомических особенностей людей и изменчивости типа внешности (за какой - либо период времени).

В связи с этим, чтобы получить достоверные результаты для алгоритмов распознавания лиц по их изображению, необходимо выполнить несколько требований для изображений [103]:

1. Все изображения должны быть преобразованы из модели RGB (цветовая модель) в градации серого;
2. Изображения должны быть центрированы;
3. Изображения должны иметь одинаковые размеры.

Классическая схема алгоритма распознавания лиц представлена на рисунке 4.1.:

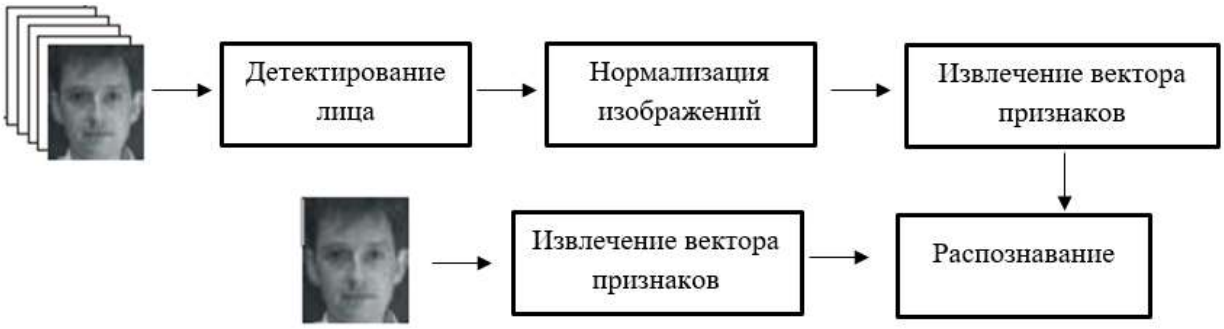

Рисунок 4.1. Этапы классической схемы алгоритма распознавания лиц

Извлечение вектора признаков происходит на основе изображения. Каждый пиксель анализируемого изображения становится компонентом

вектора, трансформируя черно-белое изображение в вектор пространства [103] (рисунок 4.2.).

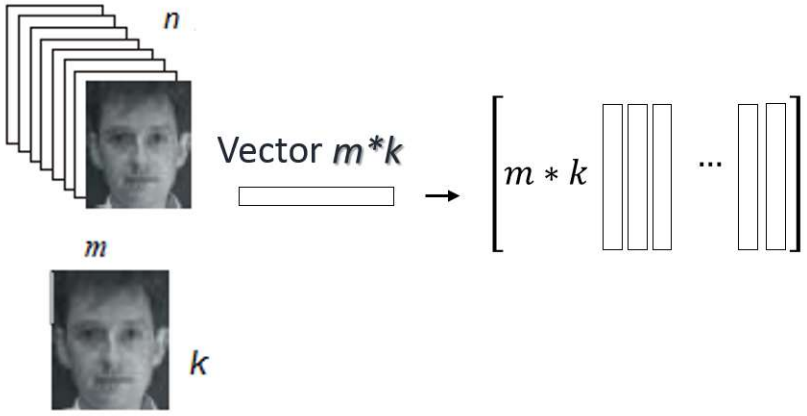

Рисунок 4.2. Извлечение вектора признаков из изображения размерности $m \times k$

Использование всего набора численных признаков в процессе решения задачи классификации может существенно замедлить работу алгоритма и снизить точность получаемого решения. Поэтому важным для снижения размерности в процессе решения задач, связанных с обработкой изображений, является извлечение наиболее информативных признаков, используемых алгоритмами распознавания. Для того, чтобы выбрать наиболее подходящие характеристики, могут быть использованы статистические методы, такие как факторный анализ, а также более сложные, к примеру, основанные на ГА.

В работах [104, 105] рассматривается процедура извлечения информативных признаков, основанная на адаптивном многокритериальном генетическом алгоритме, исследуется ее эффективность в сочетании с различными классификационными моделями.

Также для отбора информативных признаков может применяться метод главных компонент (Principal Component Analysis, PCA). Анализ главных компонент основан на определении минимального числа факторов, которые

вносят наибольший вклад в дисперсию данных. Они называются главными компонентами.

Существующие подходы к задаче распознавания лиц по их изображению. Для решения задачи распознавания лиц существует множество инструментов, из которых наиболее распространенным является библиотека с открытыми исходными кодами OpenCV, а именно – класс FaceRecognition.

OpenCV (Open Source Computer Vision) это популярная библиотека компьютерного зрения, разработанная компанией Intel в 1999 году. Кросс-платформенная библиотека нацелена на обработку изображений в режиме реального времени, и включает в себя свободную реализацию новейших алгоритмов компьютерного зрения [106].

Данная библиотека значительно упрощает программирование в области компьютерного зрения, предоставляя удобный интерфейс для детектирования, отслеживания и распознавания лиц. Особо ценным в OpenCV является математический аппарат и функционал по обработке изображений.

Методы обнаружения лиц могут быть представлены двумя категориями:

1. Методы, основанные на построении некоторого набора правил для обнаружения лица на изображении [107], например, Метод Виолы – Джонса [108]. Эти методы производят идентификацию элементов и особенностей, характерных для изображения лица, а также анализ количества и пространственного положения лиц;
2. Методы, в которых каждому изображению ставится в соответствии вектор признаков, компоненты которого представляют каждый пиксель анализируемого изображения [109]. К таким методам относятся методы опорных векторов [110] и линейный дискриминантный анализ [111].

В OpenCV используется несколько алгоритмов для решения задачи распознавания лиц.

Один из самых распространенных является алгоритм из класса методов принятия решений через сопоставление образов под названием - Eigenfaces [112, 113]. Метод Eigenfaces основан на использовании фундаментальных статистических характеристик: математического ожидания и ковариационной матрицы, использовании метода главных компонент.

Существует проблема представления изображения в векторном пространстве из-за его высокой размерности. Двумерное $p \times q$полутоновое изображение охватывает векторное пространство размерностью $m = p \times q$. Таким образом, изображение с количеством 100×100 пикселей уже находится в 10,000 – мерном пространстве изображений.

Отбор главных компонент, которые находятся из нормализованных изображений обучающего множества, приводит к понижению размерности обучающей выборки для упрощения работы классификатора.

В алгоритме Eigenfaces находится такой базис меньшей размерности, после проекции в который максимально сохраняется информация по осям с большой дисперсией и теряется информацию по осям с маленькой дисперсией. Это нужно для того, чтобы оставить только ту информацию, которая бы характеризовала различия лиц и удалить ненужную информацию, которая может помешать правильно идентифицировать человека.

Процедура идентификации выполняется в новом базисе с использованием Евклидовой метрики. Идентификация производится с помощью поиска ближайшего соседа между обучающим объектом тренировочных образов и проецируемого изображения запроса.

Особенностью алгоритма Eigenfaces является то, что он кодирует не только черты лица, но и освещение в изображениях. Поэтому он является неустойчивым к изменению освещения.

В библиотеке OpenCVпредставлены еще 2 алгоритма распознавания изображений: Fisherfaces [114] и локальные бинарные шаблоны (Local Binary Patterns Histograms, LBPH) [115].

FisherFaces - модификация алгоритма EigenFaces. Использует нормированную яркость пикселей. Более устойчив к изменению освещения. В алгоритме Fisherfaces предполагается поиск такой комбинации признаков, которая позволила бы максимизировать дисперсию между множествами изображений лиц и одновременно минимизировать дисперсию внутри каждого множества. Уменьшения признакового пространства в алгоритме Fisherfaces производится с помощью линейного дискриминантного анализа.

Основная идея LBHP – обобщить локальную структуру в изображении, сравнивая каждый пиксель с его окрестностью. Проверяемый пиксель берется как центр, а его соседи напротив являются порогом. Если интенсивность центрального пикселя больше или равна его соседнему пикселю, то области присваивается значение 1 и 0, если нет. 8-битное число, получаемое обходом пикселей по часовой стрелке, называется Binary Pattern, и используется для сравнения со значениями в классификаторе (рисунок 4.3).

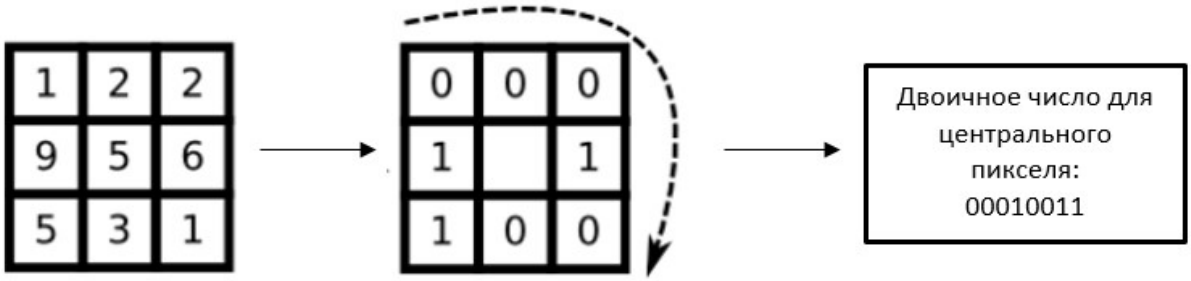

Рисунок 4.3. Локальный бинарный шаблон для центрального пикселя

Метод LBHP является универсальным, так как этот подход можно использовать для обнаружения произвольных объектов, не только лиц.

Наборы данных с изображениями лиц. Для проведения экспериментальных исследований в качестве тестовых баз изображений было использовано 2 базы данных:

1. AT&T Facedatabase [116].
2. Faces95 [117].

Набор данных «AT&T Facedatabase» содержит изображения лиц 40 спикеров, по 10 изображений на каждого. Изображения лиц представлены в разрешении 92×112 (10304 атрибутов). Для каждого спикера изображения получены с различной освещенностью, положением головы, открытым и закрытым ртом, с очками и без, с разным выражением лица. Количество изображений для каждого спикера различно. Объем обучающей выборки – 280 объектов, а тестовой – 120.

Набор данных «Faces95» содержит изображения 74 спикеров, по 20 изображений на каждого (1480 изображений). Изображения представлены в разрешении 180 × 200 pixels (36000 атрибутов). Объем обучающей выборки – 1036 объектов, а тестовой – 444.

Решение задач распознавания лиц по их изображению. В качестве методов анализа данных, из которых будет сформирован коллектив, были выбраны следующие алгоритмы, представленные в программной системе RapidMiner Studio [118]:

1. Искусственная нейронная сеть (Artificial Neural Networks, ANN);
2. Метод деревьев принятия решений (Decision Trees, DT);
3. Метод опорных векторов (Support Vector Machine, SVM).

Для уменьшения признакового пространства применяется метод главных компонент.

Разбиение на тестовое и обучающее множество производилось случайным образом в соотношении 70/30. При тестировании, обучение всех алгоритмов, входящих в коллектив, производилось на всем обучающем множестве. На всех задачах алгоритмы запускались 10 раз.

Все результаты усреднялись. Так же был применен оператор T-Test (уровень значимости – 0.05) по критерию Стьюдента. Проведем сравнительный анализ эффективности каждого метода в отдельности с коллективом, включающем в себя все 3 алгоритма.

Критерием эффективности является точность классификации - доля правильно распознанных элементов из выборки.

В задаче №1 после применения PCA число признаков уменьшилось с 10304 до 399. Сравнительный анализ эффективности FRBS и отдельных агентов представлен в таблице 4.1.

Таблица 4.1. Сравнительный анализ эффективности для набора данных «AT&T Facedatabase»

T-test		77.42%	61.83%	77.42%
		ANN	**DT**	**FRBS**
72.25%	**SVM**	0.00493818	0.000159513	0.004938181
77.42%	**ANN**		3.02454E-05	-
61.83%	**DT**			3.02454E-05

По результатам, представленным в таблице 4.1, можно сделать выводы о том, что FRBS имеет то же значение средней точности, что и нейронная сеть. Сравнение результатов, полученных FRBS и нейронной сетью, показало, что различия не являются статистически значимыми.

В работе [119] представлены результаты распознавания методов Eigenfaces и нелинейного метода опорных векторов (Nonlinear Principal Component Analysis, N-PCA) для базы данных «AT&T Facedatabase» (проведены исследования с различным процентным соотношением обучающей и тестовой выборки), в работе [120] представлены результаты на основе коллективных методов, а в [121] результаты коллективного алгоритма AdaBoost.

По результатам, представленным в таблице 4.2, можно сделать вывод, что FRBS и искусственная нейронная сеть имеют наилучшее значение средней точности.

Таблица 4.2. Сравнительный анализ эффективности для набора данных «AT&T Facedatabase» для различных алгоритмов

Точность	
SVM	93,667%
ANN	**95,966%**
DT	62,167%
FRBS	**95,666%**
Eigenfaces	92,50%
N-PCA	93,75%
majority voting	80,6%
AdaBoost	85,98%

Для набора данных «Faces95» после использования метода PCA количество компонентов уменьшилось с 36000 до 432. Результаты показаны в таблице 4.3.

Таблица 4.3. Сравнительный анализ эффективности для набора данных «Faces95»

T-test		83.262%	71.203%	83.2638%
		ANN	**DT**	**FRBS**
76.065%	**SVM**	3.60889E-08	0.001098475	3.5693E-08
83.262%	**ANN**		6.53248E-07	0.022045801
71.203%	**DT**			6.5058E-07

Согласно таблице 4.3 FRBS показал лучший результат. Анализ по критерию Стьюдента показывает, что FRBS имеет статистически значимые различия по сравнению со всеми другими алгоритмами.

В работе [122] представлены результаты распознавания с помощью таких методов как Eigenfaces, линейный дискриминантный анализ (Linear Discriminant Analysis, LDA) и метод опорных векторов с использованием фильтров Gabor (Gabor Support Vector Machine, Gabor SVM) для базы данных «Faces95» (также проведены исследования с различным процентным соотношением обучающей и тестовой выборки).

Таблица 4.4. Сравнительный анализ эффективности для набора данных «Faces95» для различных алгоритмов

	Точность
SVM	89.027%
ANN	92.386%
DT	71.992%
FRBS	**92.454%**
Eigenfaces	82.7%
LDA	92.6%
Gabor SVM	90.2%

По результатам, представленным в таблице 4.4, можно сделать вывод, что FRBS имеет наилучшее значение средней точности. Согласно представленным выше результатам, можно сделать вывод, что коллективное принятие решений позволяет повысить точность решения.

4.2. Исследование эффективности коллективных систем на нечеткой логике в задаче прогнозирования эмоционального состояния человека по аудиоданным

Одной из важнейших задач современного этапа информатизации общества является развитие систем человеко-машинного интерфейса, в том числе систем автоматизированного распознавания эмоций человека.

В пространственном подходе к определению эмоций используется n-мерное пространство, состоящее из «факторов» эмоций. Любая эмоция может быть представлена как некоторая комбинация этих факторов. Для одной из таких комбинаций используются два измерения: «Valence» и «Arousal», где первое измерение - насколько положительна или отрицательна эмоция (валентность), а второе измерение - насколько интенсивно физическое возбуждение эмоции (возбуждение) [123]. Например, «Happy» - это состояние с высокой валентностью, сильное возбуждение, а «Stressed» - это состояние с низкой валентностью и сильное возбуждение.

Системы автоматического распознавания эмоций, основанные на контролируемом машинном обучении, требуют надежной аннотации эмоционального поведения для построения полезных моделей.

Задача распознавания эмоций является задачей классификации, для решения которой используют традиционные классификаторы, такие как: смесь гаусовских распределений (Generalized Method of Moments, GMM), скрытые марковские модели (Hidden Markov Model, HMM), ANNs, SVM, обобщенно-регрессионные нейронные сети (Generalized Regression Neural Network, GRNN), глубокие нейронные сети (Deep Neural Network, DNN).

В последние годы все больше усилий уделяется автоматическому и непрерывному прогнозированию эмоций спонтанного поведения людей [124]. Для такой задачи были предложены и исследованы различные регрессионные модели, такие как SVR [125], регрессия вектора релевантности (RVR) [126], нейронные сети прямой проводимости (FNNs) [127], рекуррентные нейронные сети (RNN) [128], и другие. При использовании разных размеров одного признакового пространства для непрерывного прогнозирования возбуждения и валентности в различных базах данных результаты показывают, что одни алгоритмы дают лучше результат при использовании наименьшего признакового пространства, а другие алгоритмы – на всем признаковом пространстве.

К примеру, в работе [129] сделан вывод о превосходстве алгоритма SVR над BLSTM-RNN, а в работе [130] наоборот, о превосходстве второго метода. Разумным объяснением этого может быть то, что каждая модель прогнозирования имеет свои плюсы и минусы. Например, BLSTM-RNN очень чувствительны к переоснащению, но SVR не может явно моделировать контекстные зависимости.

Многие алгоритмы машинного обучения были применены к проблеме прогнозирования эмоционального состояния человека. Чтобы объединить преимущества различных регрессионных моделей применяются различные коллективные подходы для дальнейшего улучшения непрерывного предсказания эмоций.

В данной главе исследуется эффективность использования алгоритмов машинного обучения и их коллективов, способных интегрировать контекстную информацию в моделирование, для автоматического прогнозирования эмоций от нескольких (асинхронных) оценщиков в непрерывных временных областях, т.е. возбуждение и валентность [37]. Эмоции человека выражены в виде комбинации двух показателей: «Valence» – направленность эмоции (отрицательные или положительные эмоции) и «Arousal» – выраженность эмоции (степень возбужденности). Эти показатели представлены вещественными числами.

Набор данных «RECOLA». Оценка производится на базе данных «RECOLA» [131], которая содержит модуль Annotation. Этот модуль содержит аннотации (данные об аффективном поведении), выполненные шестью ассистентами (три мужчины, три женщины) с помощью веб-инструмента аннотации ANNEMO. Аннотации представлены для каждой аудиозаписи, записанной отдельно для каждого участника с частотой кадров 40 мс для аффективного (эмоционального) поведения.

База данных представлена 23 субъектами, аудиозапись каждого из которых длится 5 минут. На основе этого необходимо решить задачу

прогнозирования аффективного (эмоционального) поведения. Для извлечения акустических характеристик использовалась система OpenSMILE с набором функций ComParE, которая извлекает 130 акустических признаков.

Однако автоматическое распознавание эмоций от непрерывных по времени меток требует определить соответствующую длину временного окна, используемого для предсказания эмоций, которое зависит от модальности и эмоций.

В литературе нет четкого консенсуса относительно наилучшей длины временного окна, которое можно использовать для данной модальности и эмоций. В то время как предполагается, что общая продолжительность эмоции падает от 0,5 до 4 секунд, длина окна анализа, используемого для предсказания эмоций, может значительно варьироваться в зависимости от модальности; аудио-сигналы обычно меняются со временем быстрее, чем видеосигналы, и даже больше, чем физиологические сигналы. Выбранная последовательность преобразуется во входы с помощью скользящего окна, после чего, данные нормируются.

При увеличении размера окна существенно увеличивается размерность признакового пространства входов. В задаче «RECOLA» в качестве входа необработанный сигнал сегментировался на последовательности продолжительностью 1,5 секунды, что соответствует размерности вектора – 5200 признаков, а размерности выборки – 170000 объектов.

В качестве переменной выхода для последовательности использовалось усреднение значений аннотаций в каждый период времени, предоставленных всеми 6 аннотаторами.

Выбор размера временного окна был сделан на основе результатов исследований в работе [132], в которой наибольшее значение точности решения задачи по аудиоданным достигалась с периодом окна в 1-2 секунды.

Для уменьшения признакового пространства предлагается применить PCA с сокращением размерности до 200 компонент.

Анализ эффективности эволюционной процедуры для автоматизированного формирования состава коллектива. В настоящей работе для формирования состава коллектива с учетом критериев предлагается применять многокритериальные эволюционные алгоритмы NSGA-II и SPEA-II [133]. Эти алгоритмы позволят автоматизировать формирование состава коллектива, тем самым экономя вычислительные ресурсы (позволят минимизировать число агентов) и решая поставленные задачи достаточно качественно (повышая способность к обобщению результата).

Для формирования коллектива в состав были включены следующие агенты:

1. GBR.
2. KNR.
3. LLasso.
4. LR.
5. LRidge.

6-7. MLP (структура сети: 200x100x50x20, 200x200 нейронов на соответствующих слоях, сигмоидальная активационная функция).

8-10. Искусственная нейронная сеть (многослойный персептрон - NNKeras, построенная на основе библиотеки «Keras» и «TensorFlow» (Python); структура сети: 200x100x20, 200x100x50x20, 200x200 нейронов на соответствующих слоях, сигмоидальная активационная функция).

11-12. RFR (Количество деревьев в коллективе: 10, 50; глубина дерева: 18; количество признаков, используемых одним деревом: 100).

13. SVR.

Критерием эффективности работы предлагаемого подхода на основе нечеткой логики является коэффициент корреляции согласованности ρ_c (**Ptest** – точность на тестовой выборке, а **Pvalid** – на контрольной).

Результаты худшего и лучшего агентов для набора данных «RECOLA» представлены в таблице 4.5.

Таблица 4.5. Результаты наихудшего и наилучшего агента для меток «Valence» и «Arousal»

Метка класса	**Агент**	**Ptest**	**Pvalid**
«Valence»	Худший агент	GBR:0,2914	GBR:0,258
	Лучший агент	NNKeras:0,796	NNKeras:0,796
«Arousal»	Худший агент	GBR:0,4617	GBR:0,4617
	Лучший агент	NNKeras:0,8257	NNKeras:0,8236

Для текущего решения с помощью системы на нечеткой логике применялось 2 конфигурации: итоговое решение принималось средним (*FRBS+M*) и взвешенным средним (*FRBS+W*). Количество агентов *nAgent*, композиция которых применялась для получения решения для объекта из тестовой выборки варьировалось от 1 до 10, а параметр *nPoints* = 1.

После применения работы PCA исходная выборка делится на 3 множества: *learning* - 60% от общего числа точек, *testing* - 25%, *validation* - 15%. Параметры запусков:100 поколений, 100 индивидов. Результаты тестирования представлены в таблицах 4.6 и 4.7.

В результате проведенных исследований, представленных в таблице 4.6 можно сделать выводы о том, что коллектив с выводом *mean* и *Wmean* без процедуры FRBS отработали хуже лучшего агента, даже с учетом оптимизации состава коллектива. А гибридизация FRBS и *mean/Wmean* позволяет найти решение точнее лучшего из агентов даже без оптимизации состава.

Выбор агентов в состав коллектива с помощью многокритериальной эволюционной процедуры позволяет существенно повысить точность работы коллектива. При этом SPEA-II дает результаты не хуже, а в ряде случаев лучше, чем NSGA-II,

Таблица 4.6. Анализ эффективности коллективного принятия решения на основе нечеткой логики с оптимизацией состава коллектива для прогнозирования метки «Valence»

nAg	Метод	Без оптимизации		NSGA-2			SPEA-2		
		Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Маска MEA**	**Ptest**	**Pvalid**	**Маска MEA**
1	*FRBS*+M	0,810	0,822	0,842	0,840	0000110101110	0,846	0,842	1001010101011
	FRBS+W	0,810	0,822	0,842	0,840	0000110101110	0,846	0,842	1001010101011
2	*FRBS*+M	0,863	0,851	0,870	0,867	0101101111110	0,873	**0,872**	1011111111000
	FRBS+W	0,863	0,851	0,870	0,867	0101101111110	0,875	0,870	1010111111110
3	*FRBS*+M	0,865	**0,858**	0,866	**0,872**	1001101111101	0,869	**0,871**	1010101111000
	FRBS+W	0,865	**0,859**	0,866	**0,872**	1001101111101	0,870	**0,874**	0110110111010
4	*FRBS*+M	0,840	0,832	0,841	0,843	0100101111101	0,838	0,840	0110101111110
	FRBS+W	0,842	0,835	0,839	0,843	0101100111110	0,840	0,844	0111101111111
5	*FRBS*+M	0,809	0,797	0,807	0,805	0111010111110	0,806	0,803	0101010111101
	FRBS+W	0,813	0,801	0,807	0,807	1101010111110	0,812	0,812	0101001111111
6	*FRBS*+M	0,779	0,762	0,773	0,767	0111010111110	0,773	0,768	0110100111111
	FRBS+W	0,784	0,769	0,778	0,773	0111010111110	0,775	0,769	1110010111101
7	*FRBS*+M	0,749	0,731	0,731	0,724	1110110111011	0,742	0,736	1111101111111
	FRBS+W	0,755	0,738	0,740	0,735	0111010111110	0,739	0,734	0111101111110
8	*FRBS*+M	0,718	0,699	0,699	0,691	1101101111110	0,700	0,689	0101100111111
	FRBS+W	0,725	0,708	0,710	0,704	1111101111110	0,787	0,779	0100000111010
9	*FRBS*+M	0,689	0,669	0,665	0,654	1101101111110	0,665	0,656	0111101111110
	FRBS+W	0,696	0,678	0,729	0,722	0000001101001	0,697	0,692	0101000110010
10	*FRBS*+M	0,659	0,641	0,645	0,633	1111100111111	0,633	0,623	1111101111011
	FRBS+W	0,668	0,651	0,733	0,720	1000010011000	0,706	0,689	1000000011110
	mean	0,580	0,559	0,723	0,709	1000010011000	0,706	0,688	0000001011110
	Wmean	0,591	0,570	0,733	0,720	1000010011000	0,734	0,729	0100100110000

Таблица 4.7. Анализ эффективности коллективного принятия решения на основе нечеткой логики с оптимизацией состава коллектива для прогнозирования метки «Arousal»

nAg	Метод	Без оптимизации		NSGA-2			SPEA-2		
		Ptest	Pvalid	Ptest	Pvalid	Маска MEA	Ptest	Pvalid	Маска MEA
1	*FRBS*+M	0,848	0,843	0,759	0,757	1000000010011	0,864	0,854	0010100100001
	FRBS+W	0,848	0,843	0,865	0,851	0000001101001	0,864	0,854	0010100100001
2	*FRBS*+M	0,876	**0,868**	0,894	**0,883**	0111101110111	0,893	**0,883**	0011010111101
	FRBS+W	0,876	**0,868**	0,894	**0,883**	0111101110111	0,893	**0,883**	0011010111101
3	*FRBS*+M	0,874	**0,868**	0,874	0,867	0111101110111	0,875	0,866	0110100110001
	FRBS+W	0,875	**0,868**	0,875	0,869	0111101110111	0,875	0,866	0110100110001
4	*FRBS*+M	0,858	0,850	0,855	0,847	0010011110101	0,850	0,848	1011011110111
	FRBS+W	0,859	0,852	0,855	0,851	0010011110101	0,850	0,848	1011011110111
5	*FRBS*+M	0,838	0,833	0,832	0,831	1111010111101	0,827	0,821	1100100110111
	FRBS+W	0,840	0,835	0,831	0,827	1111010111010	0,827	0,821	1100100110111
6	*FRBS*+M	0,820	0,817	0,811	0,806	1111010111110	0,804	0,798	1101100111110
	FRBS+W	0,822	0,819	0,810	0,807	1111010111100	0,804	0,798	1101100111110
7	*FRBS*+M	0,802	0,800	0,775	0,768	1100100111111	0,774	0,774	1010111110011
	FRBS+W	0,805	0,803	0,782	0,777	0101111111110	0,774	0,774	1010111110011
8	*FRBS*+M	0,785	0,783	0,765	0,759	0100110111011	0,765	0,759	1111101110011
	FRBS+W	0,788	0,786	0,770	0,763	1101101111110	0,765	0,759	1111101110011
9	*FRBS*+M	0,768	0,766	0,741	0,735	1101101111110	0,740	0,735	1110111110011
	FRBS+W	0,772	0,770	0,757	0,752	0000001101001	0,740	0,735	1110111110011
10	*FRBS*+M	0,752	0,748	0,714	0,707	1100100111111	0,708	0,703	1100100111001
	FRBS+W	0,756	0,752	0,757	0,752	0000001101001	0,708	0,703	1100100111001
	mean	0,679	0,676	0,766	0,756	100100111010	0,787	0,778	0000010100000
	Wmean	0,687	0,683	0,759	0,757	1000000010011	0,782	0,775	0000100110110

Аналогичные результаты получены при решении задачи прогнозирования метки «Arousal». При этом отдельные процедуры *mean* и *Wmean* отработали хуже лучшего агента в коллективе, даже с учетом оптимизации состава коллектива. Гибридизация FLS и *mean/Wmean* позволяет найти решение точнее лучшего из агентов даже без оптимизации состава коллектива.

По результатам, приведенным в таблице 4.7, невозможно выбрать победителя между SPEA-II и NSGA-II. Данные алгоритмы показали сравнимую эффективность.

Сравнительный анализ решения задач «Valence» и «Arousal» в соответствии со значениями лучшего и худшего агента представлены в таблице 4.8.

Таблица 4.8. Результаты решения задач «Valence» и «Arousal»

Задача	***nAgent***	**Метод**	**Без оптимизации**		**NSGA-2**		**SPEA-2**	
			Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
«Valence»	1	FRBS	0,81	**0,822**	**0,842**	**0,84**	**0,846**	**0,842**
	5	FRBS+M	0,809	0,797	0,807	0,805	0,806	0,803
		FRBS+W	**0,813**	0,801	0,807	0,807	0,812	0,812
	mean		0,58	0,559	0,723	0,709	0,706	0,688
	Wmean		0,591	0,57	0,733	0,72	0,734	0,729
	Худший агент		0,291	0,258	0,291	0,258	0,291	0,258
	Лучший агент		0,796	0,796	0,796	0,796	0,796	0,796
«Arousal»	1	FRBS	**0,848**	**0,843**	0,759	0,757	**0,864**	**0,854**
	5	FRBS+M	0,838	0,833	**0,832**	**0,831**	0,827	0,821
		FRBS+W	0,84	0,835	0,831	0,827	0,827	0,821
	mean		0,679	0,676	0,766	0,756	0,787	0,778
	Wmean		0,687	0,683	0,759	0,757	0,782	0,775
	Худший агент		0,462	0,462	0,462	0,462	0,462	0,462
	Лучший агент		0,826	0,824	0,826	0,824	0,826	0,824

Можно сделать выводы о том, что коллективный метод принятия решения нечеткой системой показывает большее значение точности, чем методы *mean* и *Wmean*.

Очевидна зависимость точности предсказания эмоций человека в задачах «Valence» и «Arousal» от количества агентов, согласованное решение которых используется в коллективном выводе при принятии решения FRBS (рисунок 4.4. и 4.5.). Можно видеть, что при малом количестве агентов, то

есть от 2 до 4 достигается наибольшая эффективность решения задачи регрессии.

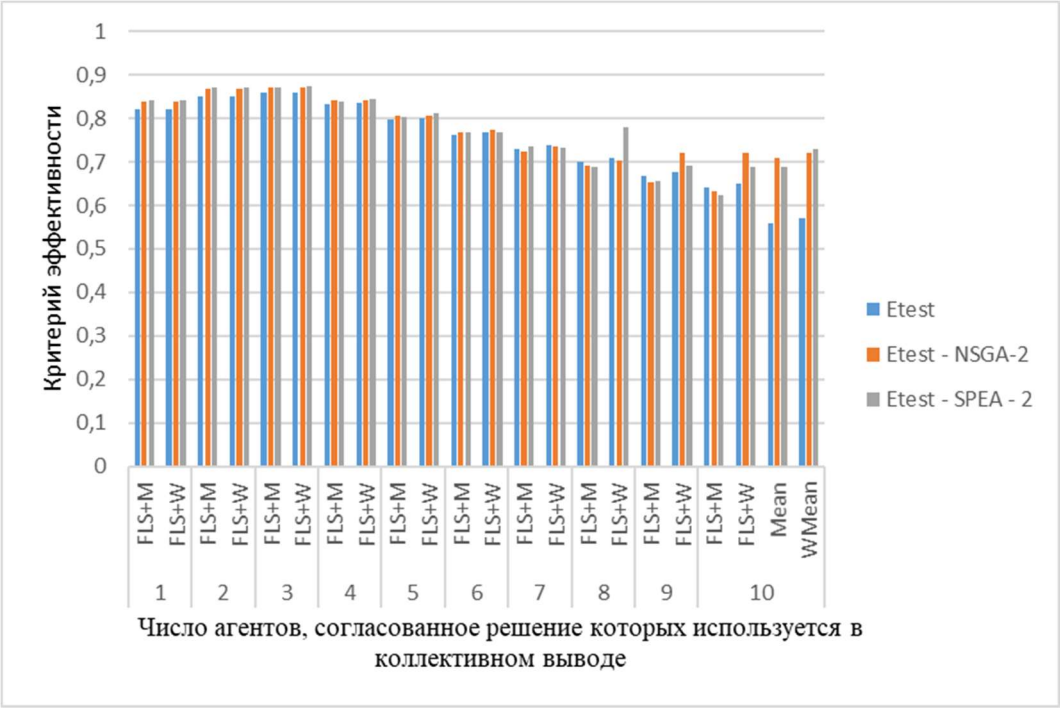

Рисунок 4.4. Анализ эффективности в зависимости от количества агентов для задачи Valence

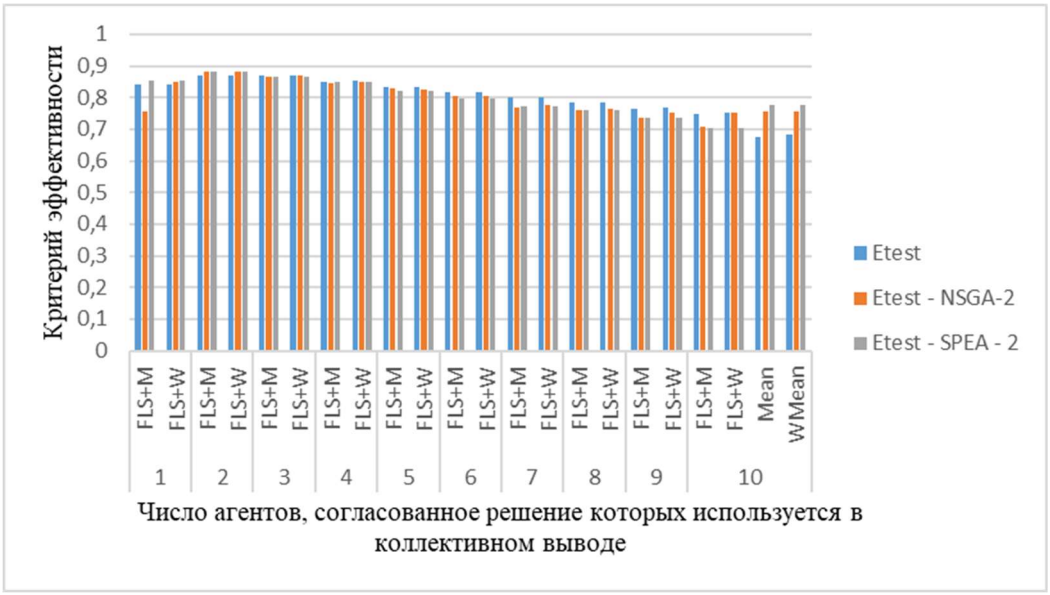

Рисунок 4.5. Анализ эффективности в зависимости от количества агентов для задачи «Arousal»

Число агентов при оптимизации состава коллектива, а также без нее при выводе существенно влияет на результат, Начиная с *nAgent* = 3 точность ухудшается, соответственно параметр *nAgent* должен быть не слишком маленьким и не слишком большим.

В ходе исследований показано, что выбор эффективных агентов с помощью многокритериального ГА позволяет существенно повысить точность работы и обобщающую способность коллектива. Для формирования эффективного коллектива рекомендуется использовать SPEA-II, т.к. SPEA-II в общем случае не уступает NSGA-II, а в некоторых конкретных случаях превосходит его.

Исследование эффективности применения генетического алгоритма многокритериальной оптимизации для отбора нечетких правил. Для формирования и отбора нечетких правил проводится сравнительный анализ эффективности двухступенчатого подхода на основе ГА однокритериальной оптимизации (RB_{GA} = 35 правил) и многокритериальной оптимизации NSGA-II ($RB_{GA+NSGA-II}$) в GFRBS к оптимизации БП с исходной БП, представленной семью правилами (RB_{basic} = 7 правил).

В качестве алгоритма формирования опорного множества использовался ГА однокритериальной оптимизации. Тестирование GFRBS производилось со значениями параметров Z = 7; N = 5, и запуск каждой ступени производился с ресурсами в 50 индивидов и 10 поколений.

Для решения задачи анализа эмоций человека по аудиоданным при формировании коллектива использовалось так же 13 агентов.

Критерием эффективности работы GFRBS является коэффициент корреляции согласованности ρ_c. Количество агентов, композиция которых применялась для получения решения для объекта из тестовой выборки *nAgent* = 2.

Для текущего решения с помощью системы на нечеткой логике применялась конфигурация, в которой итоговое решение принималось взвешенным средним.

В таблице 4.9 представлены результаты работы GFRBS с указанием эффективности как лучшего, среднего и худшего агента на полной выборке, так и всего коллектива в целом на тестовой (**Ptest**) и на контрольной выборке (**Pvalid**) для метки «Arousal».

Для формирования опорного множества применялся метод RIS и ГА однокритериальной оптимизации для отбора 10, 50, 100 и 200 точек.

Также в таблице указано количество правил *NumRules*, полученное по окончанию работы второй ступени MOEGA. В некоторых случаях приведены маски полученных БП.

В качестве метода для отбора информативных признаков применяется метод PCA с сокращением размерности до 200 компонент. После применения работы PCA исходная выборка делится на 3 множества: объем обучающей выборки 60% от общего числа точек, тестовой 25%, а контрольной 15%.

На основе полученных результатов можно сделать выводы о том, что оптимизация БП позволяет существенно повысить точность работы коллектива. На исходной БП, точность работы коллектива на отобранных точках существенно выше, чем на случайных точках.

Но при оптимизации БП точность работы коллектива на отобранном опорном множестве сравнима с точностью коллектива на опорном множестве, выбранном случайно. При этом, точность работы коллектива на опорном множестве сравнима с точностью работы коллектива с использованием всей обучающей выборки. Соответственно, опорное множество позволяет существенно экономить вычислительные ресурсы.

Таблица 4.9. Результаты, полученные при решении задачи регрессии с использованием оптимизации БП в GFRBE для метки «Arousal».

«Arousal»	**Rule Base**		
	RB_{basic} (7)	**$RB_{ГА}$ (35)**	**$RB_{ГА+NSGA-II}$**
Лучший агент (**Ptest**)	0,832		
Лучший агент (**Pvalid**)	0,824		
Средний агент (**Ptest**)	0,587		
Средний агент (**Pvalid**)	0,582		
Худший агент (**Ptest**)	0,462		
Худший агент (**Pvalid**)	0,432		
GFRBE (**Ptest**)	0,820	0,875	0,875
GFRBE (**Pvalid**).	0,832	0,864	0,864 *NumRules* =4
$GFRBE_{200\ ГА}$ (**Ptest**)	0,823	0,877	0,877
$GFRBE_{200\ ГА}$(**Pvalid**)	0,802	0,866	0,866 *NumRules* =2: 00000011110 11111111110
$GFRBE_{200\ RIS}$ (**Ptest**)	0,784	0,877	0,877 *NumRules* =4
$GFRBE_{200\ RIS}$ (**Pvalid**)	0,777	0,866	0,866 *NumRules* =4: 11010100101 11100111100 00011110111 00110001010
$GFRBE_{100\ ГА}$ (**Ptest**)	0,832	0,877	0,877 *NumRules* =2
$GFRBE_{100\ ГА}$ (**Pvalid**)	0,814	0,866	0,866 *NumRules* = 2
$GFRBE_{100\ RIS}$ (**Ptest**)	0,770	0,877	0,877 *NumRules* =4
$GFRBE_{100\ RIS}$ (**Pvalid**)	0,758	0,866	0,866 *NumRules* =4
$GFRBE_{50\ ГА}$ (**Ptest**)	0,844	0,877	0,877 *NumRules* = 4
$GFRBE_{50\ ГА}$ (**Pvalid**)	0,832	0,867	0,867 *NumRules* =4: 00100100000 00011000101 01011000010 00001101111
$GFRBE_{50\ RIS}$ (**Ptest**)	0,776	0,877	0,877 *NumRules* =1
$GFRBE_{50\ RIS}$ (**Pvalid**)	0,768	0,866	0,866 *NumRules* =1: 00001111110
$GFRBE_{10\ ГА}$ (**Ptest**)	0,873	0,877	0,877 *NumRules* =2
$GFRBE_{10\ ГА}$ (**Pvalid**)	0,860	0,866	0,866 *NumRules* =2: 00000010111 00000000111
$GFRBE_{10\ RIS}$ (**Ptest**)	0,834	0,877	0,877 *NumRules* =3
$GFRBE_{10\ RIS}$ (**Pvalid**)	0,830	0,866	0,866 *NumRules* =3

С уменьшением опорного множества, существенного уменьшения точности работы коллектива выявлено не было. В данной задаче можно получить высокую точность работы коллектива даже на опорном множестве из 10 точек. В свою очередь, применение второй ступени ГА позволяет существенно сократить количество правил в базе правил. Однако, не позволяет повысить точность работы коллектива.

В таблице 4.10 представлены результаты работы GFRBSE для метки «Valense».

Таблица 4.10. Результаты, полученные при решении задачи регрессии с использованием оптимизации БП в GFRBSE для метки «Valense».

«Valense»	**Rule Base**			
	RB_{basic} (7)	**$RB_{ГА}$ (35)**	**$RB_{ГА+NSGA-II}$**	
Лучший агент (**Ptest)**	0,799			
Лучший агент (**Pvalid**)	0,807			
Средний агент (**Ptest)**	0,472			
Средний агент (**Pvalid**)	0,454			
Худший агент (**Ptest)**	0,289			
Худший агент (**Pvalid**)	0,270			
GFRBE (**Ptest**)	0,832	0,852	0,850	*NumRules* =5
GFRBE (**Pvalid**).	0,808	0,85	0,850	
$GFRBE_{200\ ГА}$ (**Ptest**)	0,84	0,851	0,851	
$GFRBE_{200\ ГА}$ (**Pvalid**)	0,811	0,851	0,851	*NumRules=3*
$GFRBE_{200\ RIS}$ (**Ptest**)	0,737	0,845	0,845	
$GFRBE_{200\ RIS}$ (**Pvalid**)	0,746	0,850	0,850	*NumRules=3*
$GFRBE_{100\ ГА}$ (**Ptest**)	0,838	0,848	0,848	
$GFRBE_{100\ ГА}$ (**Pvalid**)	0,819	0,841	0,841	*NumRules=4*
$GFRBE_{100\ RIS}$ (**Ptest**)	0,759	0,845	0,845	
$GFRBE_{100\ RIS}$ (**Pvalid**)	0,768	0,845	0,845	*NumRules=3*
$GFRBE_{50\ ГА}$ (**Ptest**)	0,839	0,847	0,847	
$GFRBE_{50\ ГА}$ (**Pvalid**)	0,828	0,842	0,842	*NumRules=3*
$GFRBSE_{50\ RIS}$ (**Ptest**)	0,76	0,848	0,848	*NumRules=3*
$GFRBE_{50\ RIS}$ (**Pvalid**)	0,756	0,84	0,840	*NumRules=3*
$GFRBE_{10\ ГА}$ (**Ptest**)	0,849	0,847	0,847	
$GFRBE_{10\ ГА}$ (**Pvalid**)	0,842	0,844	0,844	
$GFRBE_{10\ RIS}$ (**Ptest**)	0,766	0,844	0,844	*NumRules=5*
$GFRBE_{10\ RIS}$ (**Pvalid**)	0,776	0,841	0,841	*NumRules=2*

Большинство выводов, выполненных по задаче «Arousal» справедливы и для «Valense». Однако в отличии от предыдущей задачи, можем заметить некоторое снижение точности работы коллектива при сокращении опорного множества. По результатам обоих задач сокращение количества правил происходит от 35 до 1 или 2, что дает существенное преимущество GFRBSE, значение точности которого больше чем в работе [116], исследования в которой производились на аналогичном наборе аудиоданных.

В заключение к проведенным исследованиям об эффективности применения двухступенчатого подхода к оптимизации БП в GFRBSE можно сделать вывод о том, что оптимизация БП позволяет получать высокую точность работы коллектива, как на отобранных точках, так и на точках, выбранных случайно. При этом, в некоторых случаях, оптимизация БП на отобранных точках позволяет найти БП с меньшим количеством правил, чем на случайно отобранных точках.

Однако по точности работы коллектива, существенного преимущества оптимизации БП на заранее отобранном опорном множестве – не выявлено. Точность коллектива на отобранных точках и исходной базе правил значительно выше, чем на случайных точках и исходной базе правил. Соответственно применение случайного отбора точек в опорное множество позволяет сократить общие вычислительные затраты.

Также оптимизация БП позволяет получать на опорном множестве точность близкую к точности на всей обучающей выборке. При этом, выбор опорного множества позволяет существенно экономить вычислительные ресурсы.

Применение же второй ступени ГА не позволяет повысить точность коллективного решения, однако позволяет существенно сократить количество правил в итоговой базе. Все полученные результаты позволяют сделать вывод об увеличении точности работы коллектива при оптимизации состава правил с помощью GFRBE.

4.3 Исследование эффективности коллективных систем на нечеткой логике в задаче восстановления криолитового соотношения

Оптимальное регулирование содержаний компонентов в целях получения расплава с заданными свойствами позволяет оптимизировать процесс восстановления алюминия и достичь высоких технико-экономических показателей (ТЭП). Состав электролита определяют по значениям трех параметров – криолитового отношения (КО) и содержаний фторидов кальция и магния.

Корректировка состава электролита производится на основе подбора оптимального КО: отношение содержания фторида алюминия к фториду натрия (NaF/AlF_3). Сложность задачи, стоящей перед аналитиками, заключается в том, что КО не является измеряемой величиной, а вычисляется из измеряемых количеств фторидов натрия, алюминия, кальция, магния, лития. Анализ отобранных из ванн закристаллизованных проб выполняется химическим [134] или рентгеновским дифрактометрическим методами [135] в лабораторных условиях после отбора проб.

Согласно заводским инструкциям пробы электролита отбираются 1 раз в 3 суток специальными пробоотборниками из отверстия в криолитоглиноземной корке алюминиевого электролизера [136].

Одним из основных аналитических показателей является содержание фтористого алюминия (AlF3). Известны 3 метода определения содержания AlF3: титрование с Th(NO3)4, титрование с AlCl3; титрование KOH и HCl [137]. К сожалению, эти методы не подходят для оперативного контроля из-за длительности подготовки проб к анализу. В случае определения КО с помощью дифрактометрического метода используется соотношение интенсивностей аналитических пиков главных составляющих электролита: хиолита – Na5Al3F14 и криолита – Na3AlF6.

«Недостатком способа является то, что отбор образцов электролита для анализа его химического состава обычно осуществляется один раз в три дня, что является недостаточным с точки зрения оперативности контроля, так как величина криолитового отношения может существенно изменяться в течение нескольких часов. В связи с этим электролизер длительное время работает с отклонением параметров от заданных значений, что влечет за собой снижение показателей эффективности его работы» [138] Таким образом была поставлена задача моделирования процесса определения криолитового отношения для дальнейшего прогнозирования значений показателя в моменты, когда снять показания с оборудования, напрямую (истинные значения), не представляется возможным.

В рамках решения задачи восстановления КО предлагается 2 схемы:

1. Обучение агентов производится на основе имеющегося набора данных для решения задачи восстановления КО. Проводится сравнительный анализ эффективности коллективной системы на основе нечеткой логики FRBS с моделью, имеющейся на производстве алюминия.
2. Обучение агентов производится на таких же входах, как и в 1 схеме, но в состав коллектива включается модель с производства.

Параметры запуская ГА: 100 индивидов, 100 поколений. Критерий эффективности – коэффициент корреляции согласованности (**Ptest** – точность на тестовой выборке, а **Pvalid** – на контрольной).

Параметры FRBS: *nPoints* = 1, *nAgent* = варьируется от 1 до 5. Коллектив состоит из базовых агентов, описанных в первой главе.

В таблице 4.13 представлены результаты решения задачи восстановления КО с помощью FRBS с оптимизацией агентов многокритериальным алгоритмом NSGA-II в сравнении с базовой версией алгоритма FRBS без оптимизации, описанным в первой главе.

В результате оптимизации агентов в FRBS на 1 схеме можно сделать выводы о том, что в базовой версии коллектива, коллектив проигрывает лучшему агенту. В свою очередь оптимизация состава агентов несколько улучшает результат, но по-прежнему коллектив проигрывает лучшему решению.

Таблица 4.13. Исследование эффективности оптимизации агентов в FRBS для схемы №1

№ п/п	Схема №1	«Восстановление КО»			
		Без оптимизации		NSGA-II	
		Ptest	Pvalid	Ptest	Pvalid
1	Лучший агент	0,533	**0,508**	0,533	**0,508**
2	Худший агент	0,348	0,339	0,348	0,339
3	Средний агент	0,477	0,459	0,471	0,454
4	*mean*	0,501	0,481	0,528	0,481
5	*Wmean*	0,500	0,482	0,526	0,479
6	*FRBS+M (nAgent=1)*	0,360	0,449	0,528	0,499
7	*FRBS+W (nAgent=1)*	0,360	0,449	0,528	0,499
8	*FRBS+M (nAgent=2)*	0,415	0,440	0,526	0,484
9	*FRBS+W (nAgent=2)*	0,415	0,440	0,507	0,441
10	*FRBS+M (nAgent=3)*	0,434	0,456	0,514	0,459
11	*FRBS+W (nAgent=3)*	0,434	0,456	0,529	0,478
12	*FRBS+M (nAgent=4)*	0,448	0,470	0,530	0,489
13	*FRBS+W (nAgent=4)*	0,447	0,470	0,517	0,471
14	*FRBS+M (nAgent=5)*	0,465	0,479	0,520	0,475
15	*FRBS+W (nAgent=5)*	0,465	0,479	0,520	0,475

В таблице 4.14 представлены результаты исследования влияния размера опорного множества на точность решения задачи восстановления КО. Формирование опорного множества проводится с помощью ГА.

По результатам исследований можно сделать выводы о том, что для схемы №1 оптимизация опорного множества дает улучшение результата. Коллективный вывод дает результат лучше лучшего агента.

Таблица 4.14. Исследование влияния размера опорного множества, формируемого с помощью ГА для схемы №1

«Восстановление КО» (Схема №1)	***nPoints***							
	10		**20**		**30**		**50**	
	Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
Лучший агент	0,533	0,508	0,533	0,508	0,533	0,508	0,533	0,508
Худший агент	0,348	0,339	0,348	0,339	0,348	0,339	0,348	0,339
Средний агент	0,477	0,459	0,477	0,459	0,477	0,459	0,477	0,459
mean	0,501	0,481	0,501	0,481	0,501	0,481	0,501	0,481
Wmean	0,500	0,482	0,500	0,482	0,500	0,482	0,500	0,482
FRBS+M (nAgent=1)	0,583	0,486	0,591	0,505	0,593	0,487	0,613	**0,571**
FRBS+W (nAgent=1)	0,575	0,383	0,576	0,456	0,616	0,440	0,618	0,376
FRBS+M (nAgent=2)	0,574	0,466	0,594	0,469	0,607	**0,547**	0,589	0,426
FRBS+W (nAgent=2)	0,568	**0,527**	0,586	0,510	0,614	0,491	0,615	0,436
FRBS+M (nAgent=3)	0,564	0,479	0,592	0,486	0,595	0,512	0,600	0,497
FRBS+W (nAgent=3)	0,561	**0,527**	0,583	0,480	0,600	0,480	0,602	0,511
FRBS+M (nAgent=4)	0,554	0,507	0,538	0,472	0,571	0,481	0,592	0,492
FRBS+W (nAgent=4)	0,563	0,497	0,573	0,507	0,580	0,512	0,562	0,494
FRBS+M (nAgent=5)	0,557	0,491	0,561	0,500	0,559	0,492	0,571	0,503
FRBS+W (nAgent=5)	0,543	0,480	0,547	**0,512**	0,570	0,492	0,568	0,508
FRBS+M (nAgent=6)	0,552	0,489	0,551	0,494	0,544	0,497	0,551	0,486
FRBS+W (nAgent=6)	0,552	0,491	0,549	0,510	0,557	0,489	0,561	0,487
FRBS+M (nAgent=7)	0,539	0,470	0,544	0,502	0,541	0,473	0,549	0,493
FRBS+W (nAgent=7)	0,542	0,478	0,547	0,499	0,541	0,473	0,549	0,470
FRBS+M (nAgent=8)	0,532	0,480	0,535	0,494	0,530	0,474	0,539	0,483
FRBS+W (nAgent=8)	0,524	0,488	0,539	0,478	0,540	0,491	0,541	0,493
FRBS+M (nAgent=9)	0,523	0,491	0,524	0,489	0,533	0,485	0,529	0,486
FRBS+W (nAgent=9)	0,527	0,486	0,529	0,498	0,527	0,492	0,534	0,492

В таблице 4.15 представлены результаты сравнительного анализа эффективности FRBS на основе 1 схемы с исходным количеством правил в БП (RBbasic = 7), сформированным с помощью ГА однокритериальной оптимизации ($RB_{ГА}$ = 35) и с количеством правил, отобранных с помощью многокритериального алгоритма NSGA-II ($RB_{ГА+NSGA-II}$) .

Таблица 4.15. Исследование эффективности оптимизации БП в FRBS для схемы №1

№ п/п	«Восстановление КО» (Схема №1)	Rule Base					
		RB_{basic} (7)		$RB_{ГА}$ (35)		$RB_{ГА+NSGA-II}$	
		Ptest	Pvalid	Ptest	Pvalid	Ptest	Pvalid
1	*Лучший агент*	0,533	**0,508**	0,533	0,508	0,533	0,508
2	*Худший агент*	0,348	0,339	0,348	0,339	0,348	0,339
3	*Средний агент*	0,477	0,459	0,477	0,459	0,477	0,459
4	*mean*	0,501	0,481	0,501	0,481	0,501	0,481
5	*Wmean*	0,500	0,482	0,500	0,482	0,500	0,482
6	*GFRBE+M (nAgent=1)*	0,360	0,449	0,535	0,455	0,535	0,455
7	*GFRBE+W (nAgent=1)*	0,360	0,449	0,535	0,455	0,535	0,455
8	*GFRBE+M (nAgent=2)*	0,415	0,440	0,517	**0,522**	0,520	**0,522**
9	*GFRBE+W (nAgent=2)*	0,415	0,440	0,509	0,423	0,509	0,423
10	*GFRBE+M (nAgent=3)*	0,434	0,456	0,534	0,519	0,534	0,519
11	*GFRBE+W (nAgent=3)*	0,434	0,456	0,534	0,493	0,534	0,493
12	*GFRBE+M (nAgent=4)*	0,448	0,470	0,524	0,472	0,519	0,472
13	*GFRBE+W (nAgent=4)*	0,447	0,470	0,522	0,493	0,522	0,493
14	*GFRBE+M (nAgent=5)*	0,465	0,479	0,511	0,477	0,522	0,477
15	*GFRBE+W (nAgent=5)*	0,465	0,479	0,524	0,505	0,524	0,505
16	*GFRBE+M (nAgent=6)*	0,484	0,477	0,529	0,467	0,533	0,467
17	*GFRBE+W (nAgent=6)*	0,483	0,476	0,531	0,478	0,531	0,478
18	*GFRBE+M (nAgent=7)*	0,485	0,484	0,532	0,486	0,532	0,486
19	*GFRBE+W (nAgent=7)*	0,484	0,484	0,533	0,486	0,537	0,486

В результате работы FRBS на основе первой схемы можно сделать вывод о том, что оптимизация базы правил и лингвистических переменных так же позволяет получить результат лучше лучшего агента.

В таблице 4.16 представлено исследование эффективности FRBS на основе схемы №1 в зависимости от различной последовательности этапов оптимизации: формирование состава коллектива (Ag), выбор опорного множество (NP), генерирование (RB1) и отбор правил (RB2), формирование лингвистических переменных (Linguistic variables, LV).

Таблица 4.16. Исследование эффективности последовательности этапов проектирования и формирования FRBS на основе схемы №1

Схема №1		**1**		**2**		**3**		**4**		**5**	
		Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
Этапы формирования и оптимизации FRBS	Лучший агент	0,533	0,508	0,533	0,508	0,533	0,508	0,533	0,508	0,533	0,508
	Худший агент	0,348	0,339	0,348	0,339	0,348	0,339	0,348	0,339	0,348	0,339
	Средний агент	0,477	0,459	0,477	0,459	0,477	0,459	0,477	0,459	0,477	0,459
	mean	0,501	0,481	0,501	0,481	0,501	0,481	0,501	0,481	0,501	0,481
	Wmean	0,500	0,482	0,500	0,482	0,500	0,482	0,500	0,482	0,500	0,482
RB1, RB2, LV, Ag, Dat	+Wmean	0,531	0,441	0,531	0,441	0,549	0,434	0,528	0,481	0,548	0,483
Ag, RB1, RB2, LV, Dat		0,528	0,481	0,546	0,490	0,546	0,490	0,547	0,490	0,546	0,481
Ag, Dat, RB1, RB2, LV		0,528	0,481	0,547	0,484	0,528	0,481	0,528	0,481	0,528	0,481
RB1, RB2, LV, Dat, Ag		0,542	0,465	0,542	0,465	0,549	0,472	0,549	0,472	0,528	0,481
Dat, RB1, RB2, LV, Ag		0,554	0,499	0,555	0,497	0,555	0,497	0,559	0,491	0,528	0,481
Dat, Ag, RB1, RB2, LV		0,540	0,471	0,528	0,481	0,541	0,485	0,541	0,485	0,541	0,485
LV, RB1, RB2, Ag, Dat		0,541	0,464	0,541	0,464	0,541	0,464	0,528	0,481	0,545	0,489
Ag, LV, RB1, RB2, Dat		0,528	0,481	0,541	0,487	0,544	0,483	0,544	0,483	0,550	0,483
Ag, Dat, LV, RB1, RB2		0,528	0,481	0,546	0,491	0,548	0,491	0,548	0,491	0,548	0,491
LV, RB1, RB2, Dat, Ag		0,521	0,505	0,540	0,521	0,540	0,521	0,540	0,521	0,540	0,521
Dat, LV, RB1, RB2, Ag		0,556	0,489	0,556	0,489	0,556	0,489	0,556	0,489	0,556	0,489
Dat, Ag, LV, RB1, RB2		0,559	0,480	0,528	0,481	0,544	0,497	0,545	0,481	0,545	0,481

Комбинация процедур настройки и автоматизированного формирования не позволяют существенно улучшить результаты по сравнению с оптимизацией БП в FRBS, однако позволяют получить максимальную точность из всех исследованных ранее процедур настройки.

Проведем исследование эффективности применения схемы №2 при проектировании FRBS. В таблице 4.17 представлены результаты решения

задачи восстановления КО с применением процедуры оптимизации агентов для включения в состав коллектива с помощью ГА.

Таблица 4.17. Исследование эффективности оптимизации агентов в FRBS для схемы №2

№ п/п	Схема №2	«Восстановление КО»			
		Без оптимизации		NSGA-II	
		Ptest	Pvalid	Ptest	Pvalid
1	Лучший агент	0,546	**0,509**	0,546	0,509
2	Худший агент	0,348	0,339	0,348	0,339
3	Средний агент	0,479	0,460	0,479	0,460
4	*mean*	0,552	0,525	0,587	0,530
5	*Wmean*	0,545	0,522	0,588	0,530
6	*FRBS+M (nAgent=1)*	0,386	0,469	0,821	0,431
7	*FRBS+W (nAgent=1)*	0,386	0,469	0,822	0,430
8	*FRBS+M (nAgent=2)*	0,406	0,438	0,774	0,461
9	*FRBS+W (nAgent=2)*	0,406	0,438	0,775	0,461
10	*FRBS+M (nAgent=3)*	0,433	0,464	0,733	0,472
11	*FRBS+W (nAgent=3)*	0,432	0,464	0,732	0,490
12	*FRBS+M (nAgent=4)*	0,460	0,470	0,696	**0,509**
13	*FRBS+W (nAgent=4)*	0,459	0,470	0,696	**0,508**
14	*FRBS+M (nAgent=5)*	0,474	0,481	0,665	0,503
15	*FRBS+W (nAgent=5)*	0,473	0,481	0,666	0,502

По результатам, представленным в таблице 4.17 можно сделать выводы о том, что базовая версия модели коллективного принятия решения дает результат хуже лучшего. При этом процедуры *mean* и *Wmean* дает результат лучше лучшего. В свою очередь оптимизация агентов позволяет улучшить результат. При этом коллективный вывод с помощью процедур *mean*, *Wmean* улучшается по сравнению с базовым алгоритмом FRBS, а коллективный вывод с помощью FRBS дает значение хуже лучшего агента.

В таблице 4.18 представлены результаты решения задачи восстановления КО с помощью FRBS в зависимости от размера опорного

множества, отбор которого производится с помощью ГА (*nPoints* = 10, 20, 30, 50) на основе схемы №2.

Таблица 4.18. Исследование влияния размера опорного множества, формируемого с помощью ГА для схемы №2

«Восстановление КО» (Схема №2)	***nPoints***							
	10		**20**		**30**		**50**	
	Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
Лучший агент	0,546	0,509	0,546	0,509	0,546	0,509	0,546	0,509
Худший агент	0,348	0,339	0,348	0,339	0,348	0,339	0,348	0,339
Средний агент	0,485	0,464	0,485	0,464	0,485	0,464	0,485	0,464
mean	0,552	0,525	0,552	0,525	0,552	0,525	0,552	0,525
Wmean	0,545	0,522	0,545	0,522	0,545	0,522	0,545	0,522
FRBS+M (nAgent=1)	0,670	0,422	0,651	**0,547**	0,699	0,504	0,649	0,500
FRBS+W (nAgent=1)	0,659	**0,523**	0,685	0,525	0,691	0,533	0,643	0,391
FRBS+M (nAgent=2)	0,617	0,481	0,656	0,512	0,719	0,518	0,676	0,453
FRBS+W (nAgent=2)	0,652	0,462	0,722	0,457	0,703	0,493	0,700	0,493
FRBS+M (nAgent=3)	0,665	0,460	0,654	0,425	0,670	0,459	0,682	**0,526**
FRBS+W (nAgent=3)	0,683	0,493	0,675	0,571	0,680	0,491	0,691	0,494
FRBS+M (nAgent=4)	0,700	0,490	0,678	0,500	0,654	0,503	0,656	0,506
FRBS+W (nAgent=4)	0,672	0,496	0,677	0,528	0,714	0,497	0,697	0,466
FRBS+M (nAgent=5)	0,662	0,517	0,665	0,522	0,652	0,501	0,661	0,499
FRBS+W (nAgent=5)	0,665	0,507	0,645	0,529	0,661	**0,549**	0,639	0,514
FRBS+M (nAgent=6)	0,645	0,491	0,646	0,509	0,645	0,513	0,655	0,507
FRBS+W (nAgent=6)	0,633	0,485	0,657	0,530	0,632	0,523	0,645	0,480
FRBS+M (nAgent=7)	0,627	0,509	0,625	0,507	0,628	0,508	0,624	0,508
FRBS+W (nAgent=7)	0,630	0,483	0,657	0,531	0,646	0,517	0,622	0,480
FRBS+M (nAgent=8)	0,603	0,476	0,625	0,526	0,621	0,519	0,609	0,480
FRBS+W (nAgent=8)	0,620	0,505	0,641	0,527	0,608	0,484	0,636	0,484
FRBS+M (nAgent=9)	0,608	0,502	0,615	0,510	0,627	0,490	0,617	0,493
FRBS+W (nAgent=9)	0,626	0,499	0,598	0,503	0,609	**0,575**	0,612	0,481

В результате работы FRBS на основе второй схемы можно сделать выводы о том, что оптимизация опорного множества дает результат лучше лучшего агента, но хуже процедур *mean, Wmean*.

В таблице 4.19. представлены результаты исследования эффективности формирования БП в FRBS на основе схемы №2 с помощью двухступенчатого подхода к генерированию и отбора правил.

Таблица 4.19. Исследование эффективности оптимизации БП в FRBS для схемы №2

№ п/п	«Восстановление КО» (Схема №2)	«Rule Base»					
		RB_{basic} (7)		**$RB_{ГА}$ (35)**		**$Rb_{ГА+NSGA-II}$**	
		Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
1	Лучший агент	0,546	0,509	0,546	0,509	0,546	0,509
2	Худший агент	0,348	0,339	0,348	0,339	0,348	0,339
3	Средний агент	0,479	0,460	0,485	0,464	0,485	0,464
4	*mean*	0,552	0,525	0,552	0,525	0,552	0,525
5	*Wmean*	0,545	0,522	0,545	0,522	0,545	0,522
6	*GFRBE+M (nAgent=1)*	0,386	0,469	0,542	0,468	0,535	0,455
7	*GFRBE+W (nAgent=1)*	0,386	0,469	0,548	0,475	0,535	0,455
8	*GFRBE+M (nAgent=2)*	0,406	0,438	0,512	0,448	0,520	**0,522**
9	*GFRBE+W (nAgent=2)*	0,406	0,438	0,545	0,414	0,509	0,423
10	*GFRBE+M (nAgent=3)*	0,433	0,464	0,628	0,556	0,534	0,519
11	*GFRBE+W (nAgent=3)*	0,432	0,464	0,562	0,442	0,534	0,493
12	*GFRBE+M (nAgent=4)*	0,460	0,470	0,616	0,507	0,519	0,472
13	*GFRBE+W (nAgent=4)*	0,459	0,470	0,612	0,571	0,522	0,493
14	*GFRBE+M (nAgent=5)*	0,474	0,481	0,635	0,574	0,522	0,477
15	*GFRBE+W (nAgent=5)*	0,473	0,481	0,591	0,548	0,524	0,505
16	*GFRBE+M (nAgent=6)*	0,502	0,490	0,650	0,563	0,533	0,467
17	*GFRBE+W (nAgent=6)*	0,502	0,490	0,628	**0,582**	0,531	0,478
18	*GFRBE+M (nAgent=7)*	0,506	0,499	0,628	0,535	0,532	0,486
19	*GFRBE+W (nAgent=7)*	0,505	0,498	0,612	0,589	0,537	0,486

По полученным результатам можно сделать вывод о том, что оптимизация базы правил для FRBS на основе схемы №2 дает существенное улучшение результата. При этом результат получается значительно лучше процедур *mean, wmean*.

Таблица 4.20. Исследование эффективности последовательности этапов проектирования и формирования FRBS на основе схемы №2

Схема №2		**1**		**2**		**3**		**4**		**5**	
		Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
Этапы формирования и оптимизации FRBS	Лучший агент	0,546	0,509	0,546	0,509	0,546	0,509	0,546	0,509	0,546	0,509
	Худший агент	0,348	0,339	0,348	0,339	0,348	0,339	0,348	0,339	0,348	0,339
	Средний агент	0,485	0,464	0,485	0,464	0,485	0,464	0,485	0,464	0,485	0,464
	mean	0,552	0,525	0,552	0,525	0,552	0,525	0,552	0,525	0,552	0,525
	Wmean	0,545	0,522	0,545	0,522	0,545	0,522	0,545	0,522	0,545	0,522
RB1, RB2, LV, Ag, Dat	+Wmean	0,641	0,584	0,641	0,584	0,676	0,598	0,676	0,598	0,676	0,598
Ag, RB1, RB2, LV, Dat		0,587	0,560	0,587	0,560	0,587	0,560	0,587	0,560	0,587	0,560
Ag, Dat, RB1, RB2, LV		0,587	0,530	0,606	0,484	0,623	0,499	0,623	0,499	0,623	0,499
RB1, RB2, LV, Dat, Ag		0,697	0,585	0,697	0,585	0,697	0,585	0,697	0,585	0,697	0,585
Dat, RB1, RB2, LV, Ag		0,672	0,506	0,672	0,506	0,672	0,506	0,689	0,510	0,689	0,510
Dat, Ag, RB1, RB2, LV		0,632	0,520	0,587	0,530	0,617	0,492	0,617	0,492	0,611	0,484
LV, RB1, RB2, Ag, Dat		0,612	0,580	0,663	0,521	0,663	0,521	0,587	0,530	0,587	0,530
Ag, LV, RB1, RB2, Dat		0,587	0,530	0,587	0,530	0,587	0,530	0,587	0,530	0,587	0,530
Ag, Dat, LV, RB1, RB2		0,587	0,530	0,587	0,530	0,587	0,530	0,587	0,530	0,587	0,530
LV, RB1, RB2, Dat, Ag		0,640	0,490	0,632	0,531	0,632	0,531	0,664	0,523	0,587	0,530
Dat, LV, RB1, RB2, Ag		0,642	0,467	0,631	0,478	0,631	0,478	0,631	0,478	0,587	0,530
Dat, Ag, LV, RB1, RB2		0,669	0,509	0,587	0,530	0,587	0,530	0,587	0,530	0,587	0,530

Эффективность модели, имеющейся на производстве для решения задачи моделирования технологического процесса металлургического производства, а именно восстановления КО – 54% на тестовой выборке и 50% на контрольной выборке. Максимальная эффективность полученная на основе FRBS – 62,8% на тестовой и 58,2% на контрольной, что является существенным увеличением точности решения задачи.

Модель восстановления КО, на основе которой производство получает соотношение КО может нести в себе некую дополнительную и важную информацию. Соответственно, предлагается исследовать ситуацию (схема №3), когда обучение агентов производится на основе имеющегося набора данных для решения задачи восстановления КО и на основе выхода модели с производства, т.е. выход модели также является входом агентов. Также модель с производства входит в состав коллектива как отдельный агент.

Для промышленного внедрения коллективной системы на основе нечеткой логике FRBS рассмотрим последовательные комбинации различных этапов построения и формирования FRBS. В таблице 4.21 представлено исследование эффективности включения модели восстановления КО с производства в состав коллектива и в качестве входа в FRBS (по схеме №3) с различной последовательностью этапов проектирования и оптимизации FRBS.

По полученным результатам можно сделать выводы о том, что включение выхода модели в обучающие данные агентов позволяют существенно повысить точность самих агентов и точность коллективного вывода в целом.

Например, в последовательности этапов «LV, RB1, RB2, Ag, NP» при проектировании FRBS с комбинацией вывода *mean* каждый последующий этап проектирования FRBS не улучшает решение. А при комбинации последовательностей этапов «RB1, RB2, LV, Ag, NP» с *Wmean* каждый последующий этап повышает эффективность FRBS.

Таблица №4.21 Исследование эффективности последовательности этапов проектирования и формирования FRBS на основе схемы №3

Схема №3		**Этап оптимизации**									
		1		**2**		**3**		**4**		**5**	
		Ptest	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**	**Ptest**	**Pvalid**
Этапы формирования и оптимизации FRBS	Лучший агент	0,687	0,577	0,687	0,577	0,687	0,577	0,687	0,577	0,687	0,577
	Худший агент	0,298	0,340	0,298	0,340	0,298	0,340	0,298	0,340	0,298	0,340
	Средний агент	0,579	0,517	0,579	0,517	0,579	0,517	0,579	0,517	0,579	0,517
RB1, RB2, LV, Ag, NP	+*mean*	0,652	0,560	0,652	0,560	0,695	0,571	0,667	0,572	0,692	0,575
	+*Wmean*	0,671	0,557	0,671	0,557	0,656	0,564	0,667	0,572	0,690	0,574
Ag, RB1, RB2, LV, NP	+*mean*	0,667	0,572	0,688	0,577	0,680	0,577	0,692	0,570	0,692	0,570
	+*Wmean*	0,667	0,572	0,688	0,572	0,688	0,572	0,687	0,570	0,687	0,570
Ag, NP, RB1, RB2, LV	+*mean*	0,663	0,559	0,692	0,563	0,692	0,563	0,692	0,563	0,693	0,563
	+*Wmean*	0,667	0,572	0,692	0,570	0,693	0,570	0,693	0,570	0,693	0,570
RB1, RB2, LV, NP, Ag	+*mean*	0,695	0,597	0,695	0,597	0,695	0,597	0,695	0,597	0,695	0,597
	+*Wmean*	0,673	0,591	0,673	0,591	0,673	0,591	0,673	0,591	0,673	0,591
NP, RB1, RB2, LV, Ag	+*mean*	0,716	0,544	0,711	0,552	0,714	0,552	0,586	0,599	0,586	0,599
	+*Wmean*	0,704	0,560	0,704	0,551	0,704	0,551	0,704	0,555	0,667	0,572
NP, Ag, RB1, RB2, LV	+*mean*	0,718	0,522	0,667	0,572	0,689	0,577	0,689	0,577	0,684	0,572
	+*Wmean*	0,721	0,592	0,721	0,592	0,721	0,592	0,721	0,592	0,721	0,592
LV, RB1, RB2, Ag, NP	+*mean*	0,720	0,603	0,720	0,603	0,720	0,603	0,720	0,603	0,720	0,603
	+*Wmean*	0,724	0,591	0,724	0,591	0,724	0,591	0,724	0,591	0,724	0,591
Ag, LV, RB1, RB2, NP	+*mean*	0,667	0,572	0,667	0,572	0,667	0,572	0,667	0,572	0,667	0,572
	+*Wmean*	0,667	0,572	0,692	0,578	0,693	0,573	0,693	0,573	0,687	0,571
Ag, NP, LV, RB1, RB2	+*mean*	0,667	0,572	0,691	0,565	0,693	0,572	0,689	0,571	0,689	0,571
	+*Wmean*	0,663	0,559	0,694	0,558	0,694	0,563	0,694	0,564	0,694	0,564
LV, RB1, RB2, NP, Ag	+*mean*	0,694	0,564	0,691	0,562	0,691	0,562	0,733	0,553	0,667	0,572
	+*Wmean*	0,680	0,565	0,686	**0,615**	0,686	**0,615**	0,686	**0,615**	0,686	**0,615**
NP, LV, RB1, RB2, Ag	+*mean*	0,723	0,600	0,722	0,605	0,724	0,596	0,721	0,596	0,667	0,572
	+*Wmean*	0,719	0,557	0,723	0,574	0,725	0,601	0,725	0,601	0,725	0,601
NP, Ag, LV, RB1, RB2	+*mean*	0,723	**0,606**	0,667	0,572	0,681	0,555	0,675	0,580	0,675	0,580
	+*Wmean*	0,711	0,543	0,682	0,612	0,682	0,612	0,682	0,612	0,682	0,612

Максимальное значение эффективности решения задачи восстановления КО, полученное на основе 3 схемы, - 68,6% на тестовой выборке и 61,5% на контрольной.

Таким образом, коллективный вывод на основе общих методов машинного обучения позволяет получить результат на уровне модели разработанной специалистами отрасли. При этом включение в состав коллектива такой модели позволяет существенно повысить точность прогноза. А включение данных модели для обучения общих моделей машинного обучения и включение модели в состав коллектива позволяет еще больше повысить точность прогноза.

ВЫВОДЫ

В четвертой главе было произведено исследование эффективности FRBS на задачах распознавания лиц по изображению и прогнозирования аффективного (эмоционального) поведения человека по голосу, а также на задаче моделирования технологического процесса металлургического производства. На предложенных задачах были проверены базовые схемы работы коллективного вывода и схемы с применением процедур оптимизации.

В результате проведенных исследований можно заключить, что базовая схема коллективного вывода успешно справляется с поставленной задачей. Не всегда удается получить решение лучше, чем лучший агент, но всегда базовая схема работает лучше, чем процедуры голосования и взвешенного голосования.

Применение процедур автоматизированной настройки и формирования опорного множества, базы правил, лингвистических переменных и состава коллектива позволяют получать решения лучше лучшего. При этом

предложенные процедуры коллективного вывода позволяют улучшить результаты специализированных или отраслевых моделей разработанных специально под изучаемую задачу.

ЗАКЛЮЧЕНИЕ

В диссертационной работе получены следующие основные результаты.

1. Разработана новая схема формирования коллективного вывода на основе нечеткой логики, отличающаяся иерархической процедурой интеграции правил коллективного вывода.
2. Разработана новая эволюционная процедура выбора агентов для формирования эффективных коллективов, отличающаяся от известных использованием нескольких критериев эффективности.
3. Разработана новая эволюционная процедура автоматизированного формирования базы правил, отличающаяся от известных применением двух уровней эволюции и способом представления решения в бинарном пространстве поиска.
4. Разработана новая система на основе нечеткой логики для формирования коллективов моделей и алгоритмов анализа данных для решения задач классификации и регрессии, отличающаяся от известных адаптированной процедурой формирования коллективного решения.
5. Разработана комплексная процедура автоматизированного формирования системы коллективного вывода на основе нечеткой логики, отличающаяся возможностью эффективного перераспределения вычислительных ресурсов.

Таким образом, в диссертации разработан новый алгоритм коллективного вывода на основе нечеткой логики и эволюционных процедур автоматизированной настройки, позволяющий решать сложные промышленные задачи и улучшать результаты специализированных моделей, что имеет существенное значение для теории и практики интеллектуального анализа данных

Список использованной литературы

1. Breiman, L. Bagging predictors / L. Breiman // Machine learning. – 1996. – Т. 24. – №. 2. – С. 123-140.
2. Freund, Y. A decision-theoretic generalization of on-line learning and an application to boosting / Y. Freund, R. E. Schapire // Journal of computer and system sciences. – 1997. – Т. 55. – №. 1. – С. 119-139.
3. Quinlan J. R. et al. Bagging, boosting, and C4. 5 //AAAI/IAAI, Vol. 1. – 1996. – С. 725-730.
4. Kearns M. J. Thoughts on hypothesis boosting, 1988 //ML class project. – Т. 319. – С. 320.
5. Breiman L. Random forests //Machine learning. – 2001. – Т. 45. – №. 1. – С. 5-32.
6. Wolpert D. H. Stacked generalization //Neural networks. – 1992. – Т. 5. – №. 2. – С. 241-259.
7. Schapire R. E. The boosting approach to machine learning: An overview //Nonlinear estimation and classification. – Springer, New York, NY, 2003. – С. 149-171.
8. Friedman J. H. Greedy function approximation: a gradient boosting machine //Annals of statistics. – 2001. – С. 1189-1232.
9. Skurichina M., Duin R. P. W. Bagging, boosting and the random subspace method for linear classifiers //Pattern Analysis & Applications. – 2002. – Т. 5. – №. 2. – С. 121-135.
10. Töscher A., Jahrer M., Bell R. M. The bigchaos solution to the netflix grand prize //Netflix prize documentation. – 2009. – С. 1-52.
11. Dietterich, T. G. Ensemble methods in machine learning / T. G. Dietterich // International workshop on multiple classifier systems. – 2000. - С. 1-15.
12. Kuncheva, L. I. Combining pattern classifiers: methods and algorithms. / L. I. Kuncheva. // New York: John Wiley and Sons. – 2014. – P. 382.

13. Lam L., Suen C. Y. Optimal combinations of pattern classifiers / L. Lam, C. Y. Suen // Pattern Recognition Letters. – 1995. – Т. 16. – №. 9. – P. 945-954.
14. Kittler, J. On combining classifiers / J. Kittler, M. Hatef, R. P. Duin, J. Matas // IEEE transactions on pattern analysis and machine intelligence. – 1998. – Т. 20. – №. 3. – P. 226-239.
15. Jain, A. K. Statistical pattern recognition: A review / A. K. Jain, R. P. W. Duin, J. Mao //IEEE Transactions on pattern analysis and machine intelligence. – 2000. – Т. 22. – №. 1. – P. 4-37.
16. Cho, S. B. Multiple network fusion using fuzzy logic / S. B. Cho, J. H. Kim // IEEE Transactions on Neural Networks. – 1995. – Т. 6. – №. 2. – С. 497-501.
17. Drucker, H. Boosting and other ensemble methods / H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, V. Vapnik // Neural Computation. – 1994. – Т. 6. – №. 6. – С. 1289-1301.
18. Filippi, E. Multi-layer perceptron ensembles for increased performance and fault-tolerance in pattern recognition tasks / E. Filippi, M. Costa, E. Pasero // Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94). – IEEE, 1994. – Т. 5. – С. 2901-2906.
19. Duch W. What is Computational Intelligence and where is it going? //Challenges for computational intelligence. – Springer, Berlin, Heidelberg, 2007. – С. 1-13.
20. Joines, J. A. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's / J. A. Joines, C. R. Houck // Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence. – IEEE, 1994. – С. 579-584.
21. Karr, C. Genetic algorithms for fuzzy controllers / C. Karr // Ai Expert. – 1991. – Т. 6. – №. 2. – С. 26-33.
22. Тарасенко Ф.П. Прикладной системный анализ. Наука и искусство решения проблем: учебное пособие. — Томск: ТГУ, 2004.

23. Valentini, G. Ensembles of learning machines / G. Valentini, F. Masulli // Italian workshop on neural nets. – Springer, Berlin, Heidelberg, 2002. – С. 3-20.
24. Кашницкий, Ю. С. Ансамблевый метод машинного обучения, основанный на рекомендации классификаторов / Ю. С. Кашницкий, Д. И. Игнатов // Интеллектуальные системы. Теория и приложения. – 2015. – Т. 19. – № 4. – С. 37-55.
25. Borchert, M. Emotions in speech-experiments with prosody and quality features in speech for use in categorical and dimensional emotion recognition environments / M. Borchert, A. Dusterhoft // 2005 International Conference on Natural Language Processing and Knowledge Engineering. – IEEE, 2005. – С. 147-151.
26. Friedman, J. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors) / J. Friedman, T. Hastie, R. Tibshirani // The annals of statistics. – 2000. – Т. 28. – №. 2. – С. 337-407.
27. Sahu, A. Image denoising with a multi-phase kernel principal component approach and an ensemble version / A. Sahu, G. Runger, D. Apley // 2011 IEEE applied imagery pattern recognition workshop (AIPR). – IEEE, 2011. – С. 1-7.
28. Shinde, A. Preimages for variation patterns from kernel PCA and bagging / A. Shinde, A. Sahu, D. Apley, G. Runger // IIE Transactions. – 2014. – Т. 46. – №. 5. – С. 429-456.
29. Menahem, E. Troika–An improved stacking schema for classification tasks / E. Menahem, L. Rokach, Y. Elovici // Information Sciences. – 2009. – Т. 179. – №. 24. – С. 4097-4122.
30. Seewald, A. K. How to make stacking better and faster while also taking care of an unknown weakness / A. K. Seewald // Proceedings of the nineteenth international conference on machine learning. – Morgan Kaufmann Publishers Inc., 2002. – С. 554-561.

31. Sill J. et al. Feature-weighted linear stacking //arXiv preprint arXiv:0911.0460. – 2009.
32. Deng, L. Scalable stacking and learning for building deep architectures / L. Deng, D. Yu, J. Platt // 2012 IEEE International conference on Acoustics, speech and signal processing (ICASSP). – IEEE, 2012. – С. 2133-2136.
33. Круглов В. В., Дли М. И., Голунов Р. Ю. Нечеткая логика и искусственные нейронные сети. – М. : Физматлит, 2001.
34. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – Горячая линия–Телеком, 2013.
35. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. – М.: - Телеком, 2007. – 228с.
36. Polyakova A., Lipinskiy L. A study of fuzzy logic ensemble system performance on face recognition problem //IOP Conference Series: Materials Science and Engineering. – IOP Publishing, 2017. – Т. 173. – №. 1. – С. 012013.
37. Полякова, А.С. Формирование коллектива решающих правил многокритериальным эволюционным алгоритмом в задаче анализа эмоций человека по аудиоданным / А.С. Полякова, Л.В. Липинский // Вестник МГТУ им. Н.Э. Баумана. Серия «Приборостроение». – 2018. – Т.18. – №4. – С. 744-747.
38. Asuncion A., Newman D. UCI machine learning repository. – University of California, Irvine, School of Information and Computer Sciences [Электронный ресурс]. Режим доступа: http://www.ics.uci.edu/~mlearn/MLRepository.html – 2007
39. Alcalá-Fdez, J. KEEL: a software tool to assess evolutionary algorithms for data mining problems / J. Alcalá-Fdez, L. Sanchez, S. Garcia, M. J. Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J. C. Fernández // Soft Computing. – 2009. – Т. 13. – №. 3. – С. 307-318.

40. Polyakova A.S., Lipinskiy L.V., Semenkin E.S. Investigation of Reference Sample Reduction Methods for Ensemble Output with Fuzzy Logic-Based Systems // 8th International Congress on Advanced Applied Informatics "7th International Conference on Smart Computing and Artificial Intelligence" (SCAI 2019), Toyama, Japan, 2019 (Web of Science, Scopus).
41. Kordos, M. Instance selection in logical rule extraction for regression problems / M. Kordos, S. Białka, M. Blachnik // International Conference on Artificial Intelligence and Soft Computing. – Springer, Berlin, Heidelberg, 2013. – С. 167-175.
42. Kordos, M. Instance selection with neural networks for regression problems / M. Kordos, M. Blachnik // International Conference on Artificial Neural Networks. – Springer, Berlin, Heidelberg, 2012. – С. 263-270.
43. Jankowski, N. Comparison of instances seletion algorithms i. Algorithms survey / N. Jankowski, M. Grochowski // International conference on artificial intelligence and soft computing. – Springer, Berlin, Heidelberg, 2004. – С. 598-603.
44. Reinartz, T. A unifying view on instance selection / T. Reinartz // Data Mining and Knowledge Discovery. – 2002. – Т. 6. – №. 2. – С. 191-210.
45. Brighton H. Advances in instance selection for instance-based learning algorithms / H. Brighton, C. Mellish // Data mining and knowledge discovery. – 2002. – Т. 6. – №. 2. – С. 153-172.
46. Wilson D. R. Instance pruning techniques / D. R. Wilson, T. R. Martinez // ICML. – 1997. – Т. 97. – №. 1997. – С. 400-411.
47. Wilson, D. L. Asymptotic properties of nearest neighbor rules using edited data / D. L. Wilson // IEEE Transactions on Systems, Man, and Cybernetics. – 1972. – №. 3. – С. 408-421.
48. Wilson, D. R. Reduction techniques for instance-based learning algorithms / D. R. Wilson, T. R. Martinez // Machine learning. – 2000. – Т. 38. – №. 3. – С. 257-286.

49. Ritter, G. An algorithm for a selective nearest neighbor decision rule / G. Ritter, H. Woodruff, S. Lowry, T. Isenhour //IEEE Transactions on Information Theory. – 1975. – Т. 21. – №. 6. – С. 665-669.
50. Li, X. Data reduction via adaptive sampling [Text] / X. Li // Communications in Information and Systems. – 2002. – Vol. 2, Issue 1. – P. 5–38. doi:10.4310/cis.2002.v2.n1.a3
51. Evans, R. Clustering for classification: using standard clustering methods to summarise datasets with minimal loss of classification accuracy [Text] / R. Evans. – Saarbrücken: VDM Verlag, 2008. – 108 p.
52. Полякова А.С., Сидоров М. Ю, Семенкин Е. С. Комбинирование подходов кластеризации и классификации для задачи распознавания эмоций по речи // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2016. Т. 17. № 2(59). С. 335-342
53. Madigan, D. Likelihood-based data squashing: A modeling approach to instance construction / D. Madigan, N. Raghavan, W. Dumouchel, M. Nason, C. Posse, G. Ridgeway // Data Mining and Knowledge Discovery. – 2002. – Т. 6. – №. 2. – С. 173-190.
54. Sane, S. S. A Novel supervised instance selection algorithm / S. S. Sane, A. A. Ghatol // International Journal of Business Intelligence and Data Mining. – 2007. – Т. 2. – №. 4. – С. 471-495.
55. Chaudhuri A., Stenger H. Survey sampling theory and methods. New York, Chapman & Hall, 2005, 416 p.
56. Encyclopedia of survey research methods. Ed. P. J. Lavrakas. Thousand Oaks, Sage Publications, 2008, Vol. 1–2, 968 p. DOI: 10.1108/09504121011011879
57. Субботин, С. А. Быстрый метод выделения обучающих выборок для построения нейросетевых моделей принятия решений по прецедентам [Электронный ресурс] / С. А. Субботин // Радіоелектроніка, інформатика, управління. – 2015. – №. 1 (32). Режим доступа: URL http://nbuv.gov.ua/UJRN/riu_2015_1_8 (дата обращения: 18.12.2018).

58. Bernard H. R. Social research methods: qualitative and quantative approaches. Thousand Oaks, Sage Publications, 2006, 784 p.
59. Банди, Б. Методы оптимизации. Вводный курс / Б. Банди. – М.: Радио и связь, 1988. – 128 с.
60. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. М.: ФИЗМАТЛИТ, 2003.-432 с.
61. Сергиенко, А.Б, Галушин, П.В, Бухтояров, В.В., Сергиенко, Р.Б., Сопов, Е.А., Сопов, С.А., Генетический алгоритм. Стандарт: [Электронный ресурс] – URL: http://www.harrix.org/main/project_standart_ga.php
62. Tolvi, J. Genetic algorithms for outlier detection and variable selection in linear regression models / J. Tolvi // Soft Computing. – 2004. – Т. 8. – №. 8. – С. 527-533.
63. Antonelli, M. Genetic training instance selection in multiobjective evolutionary fuzzy systems: A coevolutionary approach / M. Antonelli, P. Ducange, F. Marcelloni // IEEE Transactions on fuzzy systems. – 2011. – Т. 20. – №. 2. – С. 276-290.
64. Herrera, F. Genetic fuzzy systems: taxonomy, current research trends and prospects / F. Herrera // Evolutionary Intelligence. – 2008. – Т. 1. – №. 1. – С. 27-46.
65. García-Pedrajas, N. Boosting instance selection algorithms / N. García-Pedrajas, A. De Haro-García // Knowledge-Based Systems. – 2014. – Т. 67. – С. 342-360.
66. Blachnik, M. Bagging of instance selection algorithms / M. Blachnik, M. Kordos // International Conference on Artificial Intelligence and Soft Computing. – Springer, Cham, 2014. – С. 40-51.
67. Deb, K. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II / K. Deb, S. Agrawal, A. Pratap, T. Meyarivan // International conference on parallel problem solving from nature. – Springer, Berlin, Heidelberg, 2000. – С. 849-858.

68. Chi, Z. Fuzzy algorithms: with applications to image processing and pattern recognition. / Z. Chi, H. Yan, T. Pham – World Scientific, 1996. – (Advances in Fuzzy Systems — Applications and Theory: 10 vol. / Chi, Z. Vol. 10.)
69. Pedrycz W. (ed.). Fuzzy modelling: paradigms and practice. – Springer Science & Business Media, 2012. – T. 7.
70. Driankov, D. An introduction to fuzzy control. / D. Driankov, H. Hellendoorn, M. Reinfrank – Springer Science & Business Media, 2013.
71. Mamdani E. H., Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller //International journal of man-machine studies. – 1975. – T. 7. – №. 1. – C. 1-13.
72. Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control //Readings in fuzzy sets for intelligent systems. – Morgan Kaufmann, 1993. – C. 387-403.
73. Shoureshi R., Hu Z. Tsukamoto-type neural fuzzy inference network //Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334). – IEEE, 2000. – T. 4. – C. 2463-2467.
74. Cord O. et al. Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases. – World Scientific, 2001. – T. 19.
75. Karr, C. Genetic algorithms for fuzzy controllers / C. Karr //Ai Expert. – 1991. – T. 6. – №. 2. – C. 26-33.
76. Lee, M. A. Automatic design and adaptation of fuzzy systems and genetic algorithms using soft computing techniques: Ph.D. dissertation, Univ. California, Berkeley – 1994.
77. Park, D. Genetic-based new fuzzy reasoning models with application to fuzzy control / D. Park, A. Kandel, G. Langholz // IEEE Transactions on Systems, Man, and Cybernetics. – 1994. – T. 24. – №. 1. – C. 39-47.
78. Angelov P. P. Evolving rule-based models: a tool for design of flexible adaptive systems. – Physica, 2013. – T. 92.

79. Herrera, F. Genetic fuzzy systems: taxonomy, current research trends and prospects / F. Herrera // Evolutionary Intelligence. – 2008. – Т. 1. – №. 1. – С. 27-46.
80. Cord O. et al. Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases. – World Scientific, 2001. – Т. 19.
81. Delgado, M. R. Hierarchical genetic fuzzy systems / M. R. Delgado, F. Von Zuben, F. Gomide // Information Sciences. – 2001. – Т. 136. – №. 1-4. – С. 29-52.
82. LóPez, V. A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets / V. LóPez, A. FernáNdez, M. J. Del Jesus, F. Herrera // Knowledge-Based Systems. – 2013. – Т. 38. – С. 85-104.
83. Koza J. R., Koza J. R. Genetic programming: on the programming of computers by means of natural selection. – MIT press, 1992. – Т. 1.
84. Alba, E. Evolutionary design of fuzzy logic controllers using strongly-typed GP / E. Alba, C. Cotta, J. M. Troya // Mathware and Soft Computing. – 1999. – Т. 6. – №. 1. – С. 109-124.
85. Chien, B. C. Learning discriminant functions with fuzzy attributes for classification using genetic programming / B. C. Chien, J. Y. Lin, T. P. Hong //Expert Systems with Applications. – 2002. – Т. 23. – №. 1. – С. 31-37.
86. Geyer-Schulz A. Fuzzy rule-based expert systems and genetic machine learning. – Physica Verlag, 1997. – Т. 3.
87. Hoffmann, F. Genetic programming for model selection of TSK-fuzzy systems / F. Hoffmann, O. Nelles // Information Sciences. – 2001. – Т. 136. – №. 1-4. – С. 7-28.
88. Ramos, L. S. A niching scheme for steady state GA-P and its application to fuzzy rule based classifiers induction / L. S. Ramos, J. A. C. González // Mathware and Soft Computing. – 2000. – Т. 7. – №. 2-3. – С. 337-350.

89. Sánchez, L. Combining GP operators with SA search to evolve fuzzy rule based classifiers / L. Sánchez, I. Couso, J. A. Corrales // Information Sciences. – 2001. – Т. 136. – №. 1-4. – С. 175-191.
90. Delgado, M. R. Coevolutionary genetic fuzzy systems: a hierarchical collaborative approach / M. R. Delgado, F. Von Zuben, F. Gomide // Fuzzy sets and systems. – 2004. – Т. 141. – №. 1. – С. 89-106.
91. Coello, C. A. C. Evolutionary algorithms for solving multi-objective problems. / C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen // New York: Springer, 2007. – Т. 5. – С. 79-104.
92. Fernandez, A. Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges / A. Fernandez, V. Lopez, M. J. del Jesus, F. Herrera //Knowledge-Based Systems. – 2015. – Т. 80. – С. 109-121.
93. Fazzolari, M. A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions / M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, F. Herrera // IEEE Transactions on Fuzzy systems. – 2012. – Т. 21. – №. 1. – С. 45-65.
94. Полякова А.С., Липинский Л.В., Семенкин Е.С. Эволюционный алгоритм автоматизированного формирования базы правил в процедуре нечеткого вывода при коллективном принятии решений// Системы управления и информационные технологии, №2(76), 2019. – С. 29-36.
95. Deb, K. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II / K. Deb, S. Agrawal, A. Pratap, T. Meyarivan // International conference on parallel problem solving from nature. – Springer, Berlin, Heidelberg, 2000. – С. 849-858.
96. Back, T. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. – Oxford university press, 1996.
97. Goldberg, D. E. Genetic Algorithms in Search. Optimization and Machine Learning. Addism1—Wesley //Reading, MA. – 1989.

98. Polyakova A. S., Lipinskiy L. V., Semenkin E. S. Investigation of resource allocation efficiency in optimization of fuzzy control system //IOP Conference Series: Materials Science and Engineering. – IOP Publishing, 2019. – Т. 537. – №. 5. – С. 052036.
99. Special Interest Group on Discourse and Dialogue [Электронный ресурс] / Режим доступа: URL http://www.sigdial.org/ (дата обращения: 24.01.2017)
100. Pantic, M. Toward an affect-sensitive multimodal human-computer interaction / M. Pantic, L. J. M. Rothkrantz //Proceedings of the IEEE. – 2003. – Т. 91. – №. 9. – С. 1370-1390.
101. Zhao, W. Face recognition: A literature survey / W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld // ACM computing surveys (CSUR). – 2003. – Т. 35. – №. 4. – С. 399-458.
102. Emami, S. Face Recognition using Eigenfaces or Fisherfaces // Mastering OpenCV with Practical Computer Vision Projects. Pact Publishing. – 2012.
103. Yang, M. H. Detecting faces in images: A survey / M. H. Yang, D. J. Kriegman, N. Ahuja // IEEE Transactions on pattern analysis and machine intelligence. – 2002. – Т. 24. – №. 1. – С. 34-58.
104. Брестер, К. Ю. Распознавание психоэмоционального состояния дистанционного студента по устной речи адаптивными интеллектуальными информационными технологиями / К. Ю. Брестер, С. Р. Вишневская, О. Э. Семенкина //Сибирский журнал науки и технологий. – 2014. – №. 3 (55). - С. 35–41.
105. Семенкин, Е. С. Система автоматического извлечения информативных признаков для распознавания эмоций человека в речевой коммуникации / Е.С. Семенкин, М.Ю. Сидоров //Программные продукты и системы. – 2014. – №. 4 (108). С. 127–131.
106. Face Recognition with OpenCV [Электронный ресурс] / Режим доступа: URL

http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html (дата обращения: 07.03.2017)

107. Kotropoulos, C. Rule-based face detection in frontal views / C. Kotropoulos, I. Pitas // 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing. – IEEE, 1997. – Т. 4. – С. 2537-2540.
108. Viola, P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // CVPR (1). – 2001. – Т. 1. – №. 511-518. – С. 3.
109. Yang, M. H. Detecting faces in images: A survey / M. H. Yang, D. J. Kriegman, N. Ahuja // IEEE Transactions on pattern analysis and machine intelligence. – 2002. – Т. 24. – №. 1. – С. 34-58.
110. Maydt J., Lienhart R. A fast method for training support vector machines with a very large set of linear features //Proceedings. IEEE International Conference on Multimedia and Expo. – IEEE, 2002. – Т. 1. – С. 309-312.
111. Rahat, M. Improving 2D Boosted Classifiers Using Depth LDA Classifier for Robust Face Detection / M. Rahat, M. Nazari, A. Bafandehkar, S. S. Ghidary // International Journal of Computer Science Issues (IJCSI). – 2012. – Т. 9. – №. 3. – С. 35.
112. Cendrillon, R. Real-time face recognition using eigenfaces / R. Cendrillon, B. Lovell // Visual Communications and Image Processing 2000. – International Society for Optics and Photonics, 2000. – Т. 4067. – С. 269-276.
113. Turk, M. Eigenfaces for recognition / M. Turk, A. Pentland //Journal of cognitive neuroscience. – 1991. – Т. 3. – №. 1. – С. 71-86.
114. Belhumeur, P. N. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection / P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman //IEEE Transactions on Pattern Analysis & Machine Intelligence. – 1997. – №. 7. – С. 711-720.
115. Ahonen, T. Face recognition with local binary patterns / T. Ahonen, A. Hadid, M. Pietikäinen //European conference on computer vision. – Springer, Berlin, Heidelberg, 2004. – С. 469-481.

116. AT&T Facedatabase [Электронный ресурс] / Режим доступа: URL http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
117. Collection of Facial Images: Faces95 [Электронный ресурс] / Режим доступа: URL http://cswww.essex.ac.uk/mv/allfaces/faces95.html
118. RapidMiner Studio [Электронный ресурс] / Режим доступа: URL https://rapidminer.com/
119. Bansal, A. K. Performance evaluation of face recognition using PCA and N-PCA / A. K. Bansal, P. Chawla // International Journal of Computer Applications. – 2013. – Т. 76. – №. 8.
120. Zhang C., Liang X., Matsuyama T. Generic learning-based ensemble framework for small sample size face recognition in multi-camera networks //Sensors. – 2014. – Т. 14. – №. 12. – С. 23509-23538.
121. Guo G. D., Zhang H. J. Boosting for fast face recognition //Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. – IEEE, 2001. – С. 96-100.
122. Al Daoud, E. Face Recognition Using Fuzzy Clustering and Kernel Least Square / E. Al Daoud //Journal of Computer and Communications. – 2015. – Т. 3. – №. 03. – С. 1.
123. Russell, J. A. A circumplex model of affect / J. A. Russell //Journal of personality and social psychology. – 1980. – Т. 39. – №. 6. – С. 1161.
124. Gunes, H. Automatic, dimensional and continuous emotion recognition / H. Gunes, M. Pantic // International Journal of Synthetic Emotions (IJSE). – 2010. – Т. 1. – №. 1. – С. 68-99.
125. Drucker, H. Support vector regression machines / H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, V. Vapnik // Advances in neural information processing systems. – 1997. – С. 155-161.
126. Tipping, M. E. Sparse Bayesian learning and the relevance vector machine / M. E. Tipping //Journal of machine learning research. – 2001. – Т. 1. – №. Jun. – С. 211-244.

127. Kwok, T. Y. Constructive algorithms for structure learning in feedforward neural networks for regression problems / T. Y. Kwok, D. Y. Yeung //IEEE transactions on neural networks. – 1997. – Т. 8. – №. 3. – С. 630-645.
128. Williams, R. J. A learning algorithm for continually running fully recurrent neural networks / R. J. Williams, D. Zipser //Neural computation. – 1989. – Т. 1. – №. 2. – С. 270-280.
129. Tian, L. Emotion recognition in spontaneous and acted dialogues / L. Tian, J. D. Moore, C. Lai //2015 International Conference on Affective Computing and Intelligent Interaction (ACII). – IEEE, 2015. – С. 698-704.
130. Nicolaou, M. A. Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space / M. A. Nicolaou, H. Gunes, M. Pantic /IEEE Transactions on Affective Computing. – 2011. – Т. 2. – №. 2. – С. 92-105.
131. Ringeval F. Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions / F. Ringeval, A. Sonderegger, J. Sauer, D. Lalanne //2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG). – IEEE, 2013. – С. 1-8.
132. Ringeval, F. Prediction of asynchronous dimensional emotion ratings from audiovisual and physiological data / F. Ringeval, F. Eyben, E. Kroupi, A. Yuce, J. P. Thiran, T. Ebrahimi, D. Lalanne, B. Schuller // Pattern Recognition Letters. – 2015. – Т. 66. – С. 22-30.
133. Zitzler, E. SPEA2: Improving the strength Pareto evolutionary algorithm / E. Zitzler, M. Laumanns, L. Thiele // TIK-report. – 2001. – Т. 103.
134. Куликов Б.П., Истомин С.П. Переработка отходов алюминиевого производства. // Изд. МАНЭБ. С. Петербург. - 2004. - 478 с.
135. Никаноров А.В., Седых В.И., Полонский С.Б., Ершов П.Р. Возможности переработки техногенного сырья в колонных аппаратах с нисходящим пульповоздушным потоком// Известия ВУЗов. Цветная металлургия, 2003, № 4, с.4-8.

136. Янко Э.А. Производство алюминия. Пособие для мастеров и рабочих цехов электролиза алюминиевых заводов. С-Петербург, 2007. 305 с.
137. Панченко Т. В. Генетические алгоритмы: учеб. -метод. пособие/под ред // ЮЮ Тарасевича/Астрахан. ун-т. Астрахань. – 2007.
138. [патент РУСАЛ] Патент РФ № 2013128057/02, 18.06.2013. Способ автоматического контроля криолитового отношения// Патент России № 2540248. 2013. Бюл. № 36. / Пузанов И. И., Завадяк А. В., Клыков В. А., [и др.].