

РАЗРАБОТКА НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ДИАГНОСТИКИ ПНЕВМОНИИ

А. И. Еникеева¹ [0009-0002-9192-952x], Р. А. Бурнашев² [0000-0002-1057-0328],

Р. Р. Фарахов³ [0000-0002-7993-3182]

¹⁻³Казанский (Приволжский) федеральный университет

¹adisen24@yandex.ru, ²r.burnashev@inbox.ru, ³rus-wing-dark@mail.ru

Аннотация

Статья посвящена разработке экспертной системы для диагностики пневмонии. Эта система основана на нечеткой логике и реализована с использованием алгоритма Мамдани. Рассмотрены основные этапы разработки системы, включая фаззификацию входных данных, определение нечетких правил на основе медицинских экспертных знаний, агрегирование нечетких выводов и их дефаззификацию для получения окончательного диагностического результата. Веб-интерфейс системы реализован с использованием фреймворка Django, что обеспечивает удобство взаимодействия для пользователей. Применение медицинской экспертной системы для диагностики пневмонии может сократить время, необходимое для установления диагноза, и улучшить качество диагностики за счет интеграции опыта медицинских экспертов и современных информационных технологий.

Ключевые слова: экспертная система, нечеткая логика, медицинская диагностика, алгоритм Мамдани, система поддержки принятия решений.

ВВЕДЕНИЕ

Пневмония остается одним из наиболее распространенных и опасных заболеваний дыхательной системы, поэтому точная и быстрая диагностика играет важную роль в успешном лечении. Своевременная диагностика пневмонии имеет решающее значение для эффективного лечения, однако традиционные методы диагностики требуют значительных ресурсов и экспертной интерпретации, что может быть недоступно в условиях ограниченных ресурсов. В связи с этим разработка медицинской экспертной системы, способной повысить эффективность и точность диагностики, становится актуальной задачей.

Экспертная система на основе нечеткой логики моделирует процесс рассуждения медицинских специалистов, предоставляя ценные инструменты для поддержки принятия решений, а также является эффективным инструментом современной медицины. Система позволяет повысить точность и скорость диагностики пневмонии, что в свою очередь способствует раннему началу лечения и улучшению прогноза для пациентов.

Целью системы является предоставление систематических и надежных средств оценки пневмонии путем интеграции различных клинических параметров в единую диагностическую структуру. Работа системы основана на сравнении введенных данных с базой правил, содержащей информацию о признаках пневмонии.

1. ОБЗОР МЕТОДОВ НЕЧЕТКОЙ ЛОГИКИ

Нечеткая логика является обобщением классической теории множеств и формальной логики и обеспечивает эффективные средства отображения, неопределенностей и неточностей реального мира [1]. В рамках нечеткой логики переменные могут принимать любые значения на интервале $[0, 1]$. На основе этого понятия вводятся различные логические операции над нечеткими множествами и формулируется понятие лингвистической переменной, в качестве значений которой выступают нечеткие множества [2].

Экспертные системы, основанные на нечеткой логике, используют различные алгоритмы нечеткого вывода для обработки данных [3]. Основными алгоритмами являются алгоритмы Мамдани, Сугено и Ларсен [4].

Для разработки экспертной системы был выбран алгоритм Мамдани, который является одним из наиболее популярных методов нечеткого вывода и отличается простотой реализации. Алгоритм примечателен тем, что он работает по принципу «черного ящика». На вход поступают количественные значения, на выход – они же. На промежуточных этапах используется аппарат нечеткой логики и теория нечетких множеств [5].

База нечетких правил Мамдани содержит лингвистические правила, использующие функции принадлежности для описания применяемых концепций [6].

Алгоритм Мамдани поддерживает добавление и изменение правил без значительных изменений в структуре системы [7]. Это особенно важно для медицинской диагностики, где постоянно появляются новые знания и методы лечения. В случае

необходимости новые правила могут быть легко добавлены, что обеспечивает долгосрочную актуальность и адаптивность системы.

В общем случае механизм логического вывода включает четыре этапа: введение нечеткости (фазификация), нечеткий вывод, композиция и приведение к четкости, или дефазификация (см. рисунок 1) [8].

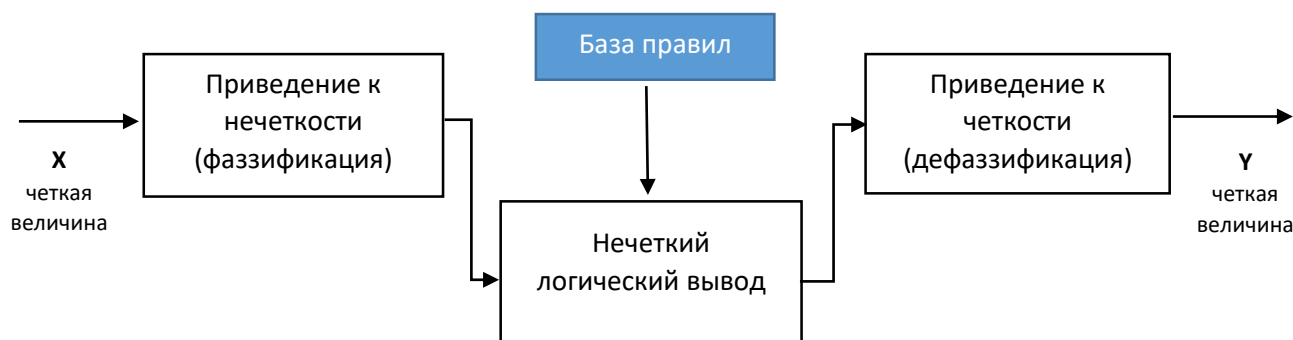


Рис. 1. Процесс обработки нечеткой логики

Алгоритм Сугено [9] является альтернативой алгоритму Мамдани и также широко используется в нечетких системах. Однако он чаще применяется в задачах управления и оптимизации, где выходные значения – линейные или полиномиальные функции входных переменных. Для медицинской диагностики, где важны интерпретируемость и простота правил, алгоритм Мамдани предпочтительнее.

Алгоритм Ларсена [10] схож с алгоритмом Мамдани, но отличается использованием произведения вместо минимума при агрегации правил. Это может привести к более сложным вычислениям и менее понятным результатам, что делает алгоритм Мамдани более подходящим для медицинских систем.

2. МОДУЛЬ НЕЧЕТКОЙ ЛОГИКИ

2.1. ОПРЕДЕЛЕНИЕ КЛИНИЧЕСКИХ ПАРАМЕТРОВ

Для разработки медицинской экспертной системы были выбраны следующие ключевые клинические параметры, указывающие на наличие пневмонии [11, 12]:

1. Возраст;
2. Кашель;
3. Одышка;
4. Температура;

5. Боль в грудной клетке;
6. Усталость;
7. Уровень кислорода в крови;
8. Уровень лейкоцитов в крови;
9. Уровень тромбоцитов в крови.

2.2. ФОРМИРОВАНИЕ БАЗЫ ПРАВИЛ

Основой для проведения операции нечеткого логического вывода является база правил, построенная на основе экспертных знаний. Каждое правило формулируется в виде утверждений «ЕСЛИ–ТО» и функции принадлежности для соответствующих лингвистических термов. Оценка условий каждого правила происходит с использованием логических операций над степенями принадлежности [13].

Процесс формирования базы правил включает следующие этапы:

1. Определение термов нечетких переменных. Для каждой входной и выходной переменных были определены термы на основе медицинских источников [11, 12] (см. таблицы 1 и 2).
2. Формулировка правил. На основе термов были сформулированы правила, отражающие причинно-следственные связи между симптомами и вероятностью пневмонии (см. таблицу 2).

Таблица 1. Определение термов нечетких входных переменных.

Входные значения	Лингвистические термы
Возраст (Age)	молодой (young)
	средний (middle)
	пожилой (elderly)
Кашель (Cough)	отсутствует (asymptomatic)
	легкий (mild)
	умеренный (moderate)
	сильный (severe)
Одышка (Shortness of breath)	отсутствует (asymptomatic)
	легкая (mild)
	умеренная (moderate)
	сильная (severe)

Температура (Temperature)	низкая (low)
	нормальная (normal)
	повышенная (elevated)
	высокая (high)
Боль в грудной клетке (Chest pain)	отсутствует (asymptomatic)
	легкая (mild)
	умеренная (moderate)
	сильная (severe)
Усталость (Fatigue)	отсутствует (asymptomatic)
	легкая (mild)
	выраженная (severe)
Уровень кислорода в крови (Saturation)	норма (normal)
	гипоксия (hypoxia)
	критическое состояние (critical)
Лейкоциты (Leukocytes)	пониженные (low)
	норма (normal)
	высокие (high)
Тромбоциты (Platelets)	пониженные (low)
	норма (normal)
	высокие (high)

Таблица 2. Определение термов нечетких выходных переменных.

Выходные значения	Выходные значения
Риск (Risk)	Низкий (low)
	Умеренный (moderate)
	Высокий (high)
	Очень высокий (very high)

Таблица 3. Пример правил экспертной системы.

	Правила
RULE 1:	IF Age IS Young AND Cough IS Asymptomatic AND Shortness of Breath IS Asymptomatic AND Temperature IS Normal AND Chest pain IS Asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal AND Platelets IS Normal THEN Risk IS LOW
RULE 2:	IF Age IS Middle AND Cough IS Asymptomatic AND Shortness of breath IS Asymptomatic AND Temperature IS Normal AND Chest pain IS Asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal AND Platelets IS Normal THEN Risk IS Low
RULE 3:	IF Age IS Elderly AND Cough IS Asymptomatic AND Shortness of breath IS Asymptomatic AND Temperature IS Normal AND Chest pain is asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal and Platelets IS Normal THEN Risk IS Moderate
RULE 4:	IF Age is Middle AND Cough IS Moderate AND Shortness of breath IS Mild THEN Risk IS High
RULE 5:	IF Age is Elderly AND Cough IS Moderate AND Shortness of breath IS Mild THEN Risk IS Very high
RULE 5:	IF Age IS Middle AND Temperature IS High AND Chest pain IS Mild THEN Risk IS High
RULE 6:	IF Age IS Elderly AND Temperature IS High AND Chest pain IS Mild THEN Risk IS Very high
RULE 7:	IF Age IS Middle AND Fatigue IS Severe AND Saturation IS Hypoxia THEN Risk IS High
RULE 8:	IF Age IS Elderly AND Fatigue IS Severe AND Saturation IS Hypoxia THEN Risk IS Very high
RULE 9:	IF Age IS Middle AND Leukocytes IS High AND Platelets IS Low THEN Risk IS High
RULE 10:	IF Age IS Elderly AND Leukocytes IS High AND Platelets IS Low THEN Risk IS Very high
RULE 11:	IF Age IS Young AND Cough IS Severe AND Shortness of breath IS Severe THEN Risk IS Very high
	...
RULE 54:	IF Saturation IS Critical THEN Risk IS Very high
RULE 55:	IF Temperature IS Elevated THEN Risk IS Moderate
RULE 56:	IF Chest pain IS Mild THEN Risk IS Moderate
RULE 57:	IF Fatigue IS Mild THEN Risk IS Moderate

2.3. ФАЗЗИФИКАЦИЯ

Фаззификация – процесс преобразования четких входных значений в нечеткие множества. Это важный этап в работе экспертных систем, основанных на нечеткой логике, так как он позволяет системе учитывать неопределенность и варьирование данных [14]. В контексте диагностики пневмонии фаззификация помогает обработать симптомы и диагностические показатели, которые могут иметь размытые границы между различными состояниями.

В процессе фаззификации четкие значения входных переменных преобразуются в степень принадлежности к одному или нескольким нечетким множествам [15]. Для этого используются функции принадлежности, которые определяют, насколько каждое четкое значение принадлежит соответствующему нечеткому множеству.

Рассмотрим пример фаззификации температуры тела. Предположим, что у пациента температура составляет 38 градусов Цельсия. Процесс фаззификации заключается в определении степени принадлежности этого значения к каждому из нечетких множеств: «низкая», «нормальная», «повышенная» и «высокая». Функции принадлежности для переменной “Temperature” задаются в виде уравнений (1)–(4). Графическое представление данных функций изображено на рисунке 2.

$$Temp_{low}(x) = \begin{cases} 1, & x \leq 35, \\ \frac{37-x}{2}, & 35 < x < 37, \\ 0, & x \geq 37. \end{cases} \quad (1)$$

$$Temp_{normal}(x) = \begin{cases} 0, & x \leq 35, \\ \frac{x-35}{2}, & 35 < x < 37, \\ 38-x, & 37 < x < 38, \\ 0, & x \geq 38. \end{cases} \quad (2)$$

$$Temp_{elevated}(x) = \begin{cases} 0, & x \leq 37, \\ x-37, & 37 < x < 38, \\ 39-x, & 38 < x < 39, \\ 0, & x \geq 39. \end{cases} \quad (3)$$

$$Temp_{high}(x) = \begin{cases} 0, & x \leq 38, \\ x-38, & 38 < x < 39, \\ 1, & x \geq 39. \end{cases} \quad (4)$$

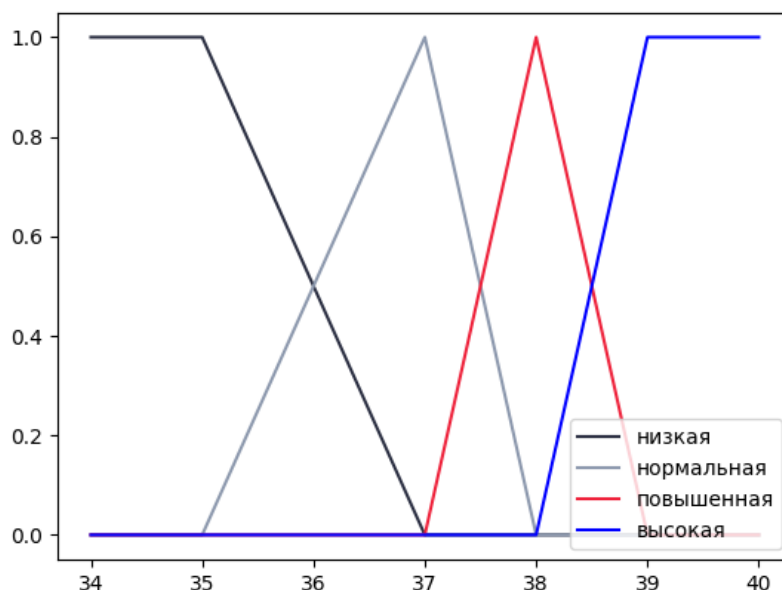


Рис. 2. Графическое представление функций принадлежности переменной “Temperature”

Реализация фаззификации переменной “Temperature” изображена на рисунке 3.

```
class Temperature(Fuzzification):
    def __init__(self):
        super().__init__('Temperature')

    def fuzzify(self, temperature_value):
        low = self.trapezoidal_membership(temperature_value, 0, 0, 35, 37)
        normal = self.triangular_membership(temperature_value, 35, 37, 38)
        elevated = self.triangular_membership(temperature_value, 37, 38, 39)
        high = self.trapezoidal_membership(temperature_value, 38, 39, 41, 42)
        return {'Low': low, 'Normal': normal, 'Elevated': elevated, 'High': high}
```

Рис. 3. Реализация фаззификации

Фаззификация переменной “Age” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «молодой», «средний» и «пожилой». Функции принадлежности для этой переменной задаются в виде уравнений (5)–(7). Графическое представление данных функций представлено на рисунке 3.

$$Age_{young}, Y(x) = \begin{cases} 1, & 0 \leq x \leq 20, \\ \frac{35-x}{15}, & 20 < x < 35, \\ 0, & x \geq 35. \end{cases} \quad (5)$$

$$Age_{middle}, M(x) = \begin{cases} 0, & x \leq 20, \\ \frac{x-20}{15}, & 20 < x < 35, \\ 1, & 35 \leq x \leq 45, \\ \frac{60-x}{15}, & 45 < x < 60, \\ 0, & x \geq 60. \end{cases} \quad (6)$$

$$Age_{elderly}, E(x) = \begin{cases} 0, & x \leq 45, \\ \frac{x-45}{15}, & 45 < x < 60, \\ 1, & x \geq 60. \end{cases} \quad (7)$$

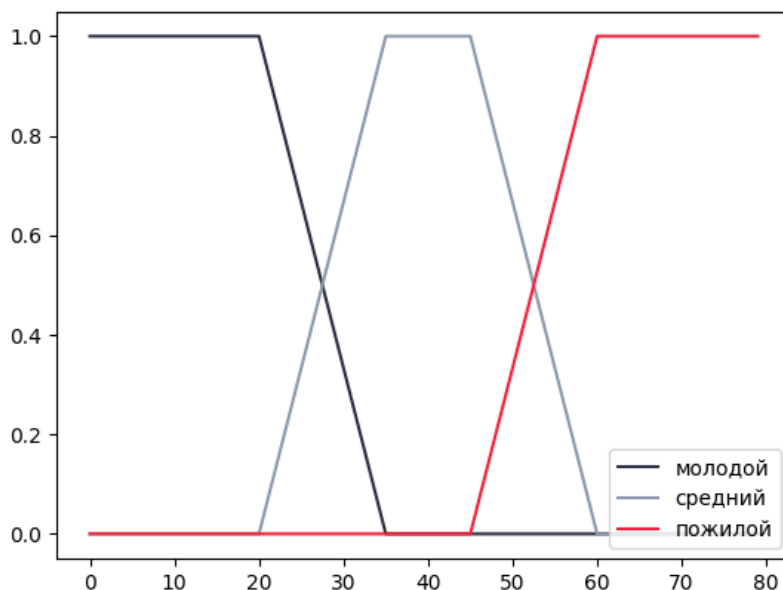


Рис. 4. Графическое представление функций принадлежности переменной “Age”

Фаззификация переменной “Cough” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «отсутствует», «легкий», «умеренный» и «сильный». Функции принадлежности для этой переменной задаются в виде уравнений (8)–(11). Графическое представление данных функций изображено на рисунке 4.

$$Cough_{asympt}(x) = \begin{cases} 1, & x \leq 1, \\ \frac{3-x}{2}, & 1 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (8)$$

$$Cough_{mild}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{2}, & 1 < x < 3, \\ \frac{5-x}{2}, & 3 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (9)$$

$$Cough_{moderate}(x) = \begin{cases} 0, & x \leq 3, \\ \frac{x-3}{2}, & 3 < x < 5, \\ \frac{7-x}{2}, & 5 < x < 7, \\ 0, & x \geq 7. \end{cases} \quad (10)$$

$$Cough_{severe}(x) = \begin{cases} 0, & x \leq 5, \\ \frac{x-5}{2}, & 5 < x < 7, \\ 1, & x \geq 7. \end{cases} \quad (11)$$

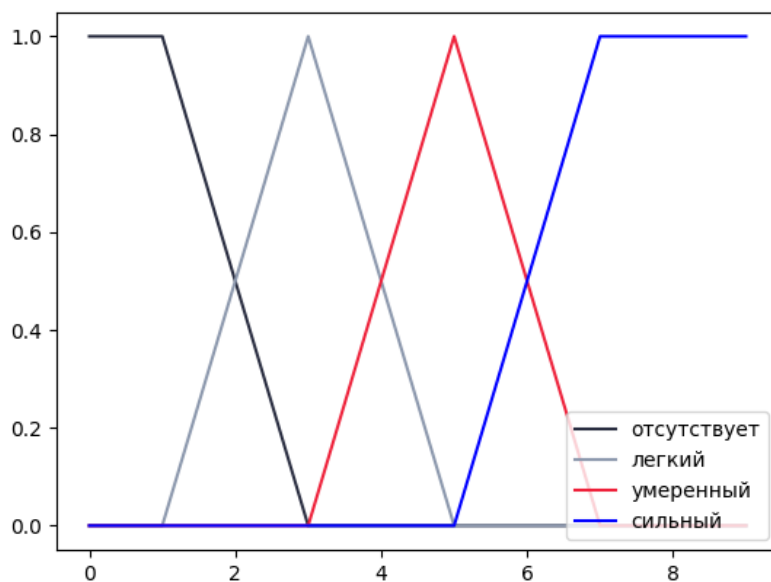


Рис. 5. Графическое представление функций принадлежности переменной "Cough"

Фаззификация переменной “Shortness of breath” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «отсутствует», «легкая», «умеренная» и «сильная». Функции принадлежности для этой переменной задаются в виде уравнений (12)–(15). Графическое представление данных функций изображено на рисунке 5.

$$SoB_{asympt}(x) = \begin{cases} 1, & x \leq 1, \\ \frac{3-x}{2}, & 1 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (12)$$

$$SoB_{mild}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{2}, & 1 < x < 3, \\ \frac{5-x}{2}, & 3 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (13)$$

$$SoB_{moderate}(x) = \begin{cases} 0, & x \leq 3, \\ \frac{x-3}{2}, & 3 < x < 5, \\ \frac{7-x}{2}, & 5 < x < 7, \\ 0, & x \geq 7. \end{cases} \quad (14)$$

$$SoB_{severe}(x) = \begin{cases} 0, & x \leq 5, \\ \frac{x-5}{2}, & 5 < x < 7, \\ 1, & x \geq 7. \end{cases} \quad (15)$$

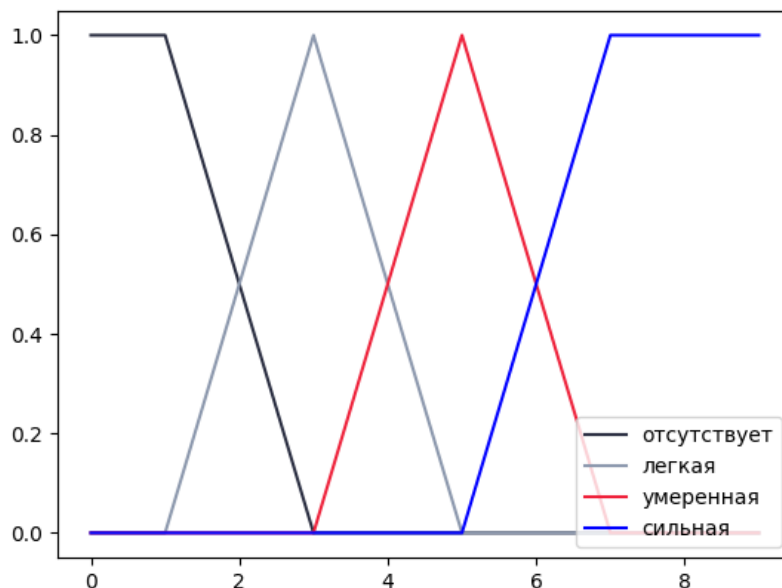


Рис. 6. Графическое представление функций принадлежности переменной «Shortness of breath»

Фаззификация переменной «Chest pain» заключается в определении степени принадлежности значения к каждому из нечетких множеств: «отсутствует», «легкая», «умеренная» и «сильная». Функции принадлежности для этой переменной задаются в виде уравнений (16)–(19). Графическое представление данных функций изображено на рисунке 6.

$$CP_{asympt}(x) = \begin{cases} 1, & x \leq 1, \\ \frac{3-x}{2}, & 1 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (16)$$

$$CP_{mild}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{2}, & 1 < x < 3, \\ \frac{5-x}{2}, & 3 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (17)$$

$$CP_{moderate}(x) = \begin{cases} 0, & x \leq 3, \\ \frac{x-3}{2}, & 3 < x < 5, \\ \frac{7-x}{2}, & 5 < x < 7, \\ 0, & x \geq 7. \end{cases} \quad (18)$$

$$CP_{severe}(x) = \begin{cases} 0, & x \leq 5, \\ \frac{x-5}{2}, & 5 < x < 7, \\ 1, & x \geq 7. \end{cases} \quad (19)$$

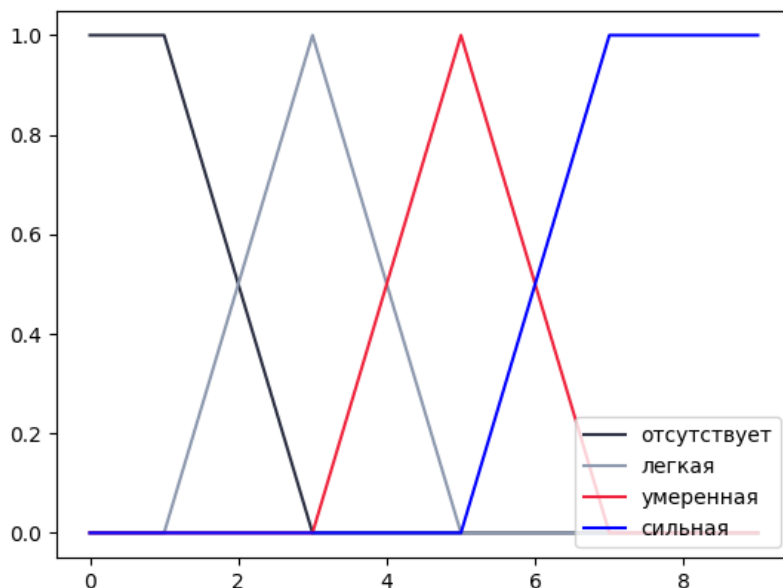


Рис. 7. Графическое представление функций принадлежности переменной “Chest pain”

Фаззификация переменной “Fatigue” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «отсутствует», «легкая» и «выраженная». Функции принадлежности для этой переменной задаются в виде уравнений (20)–(22). Графическое представление данных функций изображено на рисунке 8.

$$Fatigue_{asympt}(x) = \begin{cases} 1, & x \leq 1, \\ \frac{3-x}{2}, & 1 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (20)$$

$$Fatigue_{mild}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{2}, & 1 < x < 3, \\ \frac{5-x}{2}, & 3 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (21)$$

$$Fatigue_{severe}(x) = \begin{cases} 0, & x \leq 4, \\ x - 4, & 4 < x < 5, \\ 1, & x \geq 5. \end{cases} \quad (22)$$

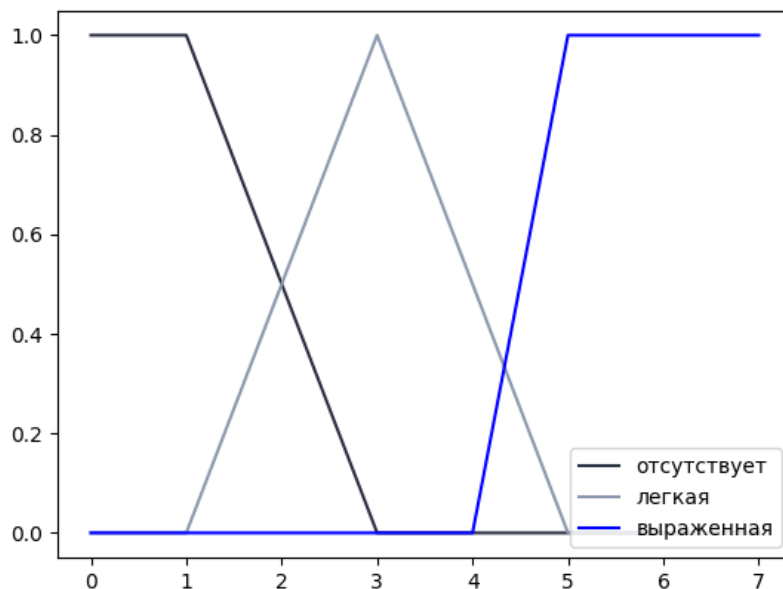


Рис. 8. Графическое представление функций принадлежности переменной "Fatigue"

Фаззификация переменной "Saturation" заключается в определении степени принадлежности значения к каждому из нечетких множеств: «норма», «гипоксия» и «критическое состояние». Функции принадлежности для этой переменной задаются в виде уравнений (23)–(25). Графическое представление данных функций изображено на рисунке 9.

$$Saturation_{normal}(x) = \begin{cases} 0, & x \leq 94, \\ x - 94, & 94 < x < 95, \\ 1, & 95 < x < 100, \\ 0, & x \geq 100. \end{cases} \quad (23)$$

$$Saturation_{hypoxia}(x) = \begin{cases} 0, & x \leq 89, \\ x - 89, & 89 < x < 90, \\ 95 - x, & 90 < x < 94, \\ 0, & x \geq 94. \end{cases} \quad (24)$$

$$Saturation_{critical}(x) = \begin{cases} 1, & x \leq 89, \\ 90 - x, & 89 < x < 90, \\ 0, & x \geq 90. \end{cases} \quad (25)$$

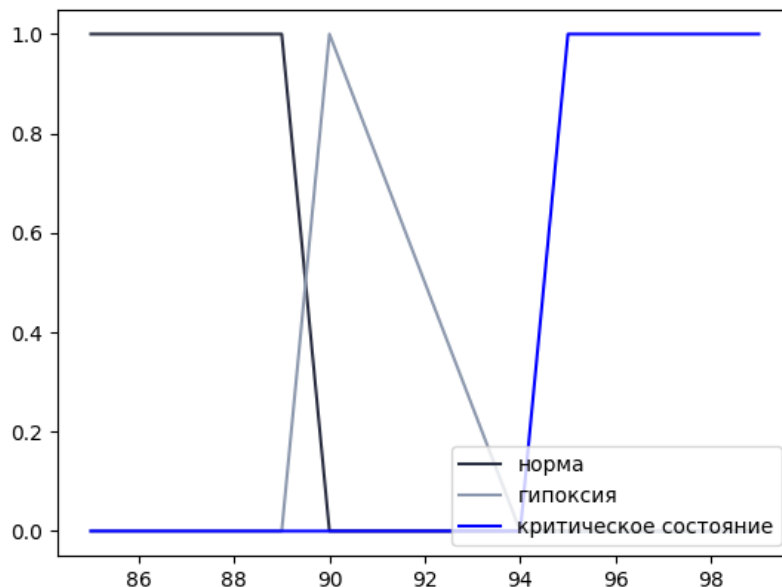


Рис. 9. Графическое представление функций принадлежности переменной “Saturation”

Фаззификация переменной “Leukocytes” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «пониженные», «норма» и «повышенные». Функции принадлежности для этой переменной задаются в виде уравнений (26)–(28). Графическое представление данных функций изображено на рисунке 10.

$$Leukocytes_{low}(x) = \begin{cases} 1, & x \leq 4.5, \\ \frac{5-x}{0.5}, & 4.5 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (26)$$

$$Leukocytes_{normal}(x) = \begin{cases} 0, & x \leq 4.5, \\ \frac{x-4.5}{0.5}, & 4.5 < x < 5, \\ 1, & 5 \leq x \leq 10.5, \\ \frac{11.3-x}{0.8}, & 10.5 < x < 11.3, \\ 0, & x \geq 11.3. \end{cases} \quad (27)$$

$$\text{Leukocytes}_{high}(x) = \begin{cases} 0, & x \leq 10.5, \\ \frac{x-10.5}{0.8}, & 10.5 < x < 11.3, \\ 0, & x \geq 11.3. \end{cases} \quad (28)$$

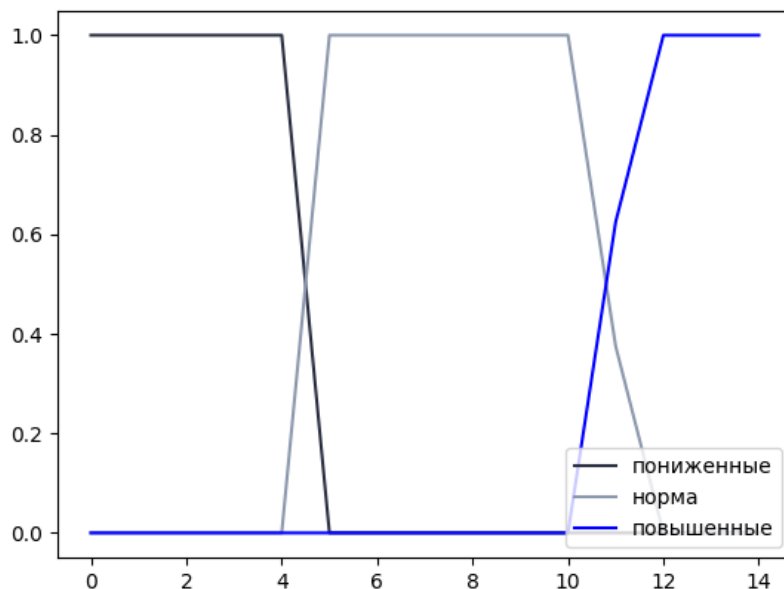


Рис. 10. Графическое представление функций принадлежности переменной “Leukocytes”

Фаззификация переменной “Platelets” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «пониженные», «норма» и «повышенные». Функции принадлежности для этой переменной задаются в виде уравнений (29)–(31). Графическое представление этих функций изображено на рисунке 11.

$$\text{Platelets}_{low}(x) = \begin{cases} 1, & x \leq 180, \\ \frac{200-x}{20}, & 180 < x < 200, \\ 0, & x \geq 200. \end{cases} \quad (29)$$

$$\text{Platelets}_{normal}(x) = \begin{cases} 0, & x \leq 180, \\ \frac{x-180}{20}, & 180 < x < 200, \\ 1, & 200 \leq x \leq 300, \\ \frac{320-x}{20}, & 300 < x < 320, \\ 0, & x \geq 320. \end{cases} \quad (30)$$

$$\text{Platelets}_{high}(x) = \begin{cases} 0, & x \leq 300, \\ \frac{x-300}{20}, & 300 < x < 320, \\ 1, & x \geq 320. \end{cases} \quad (31)$$

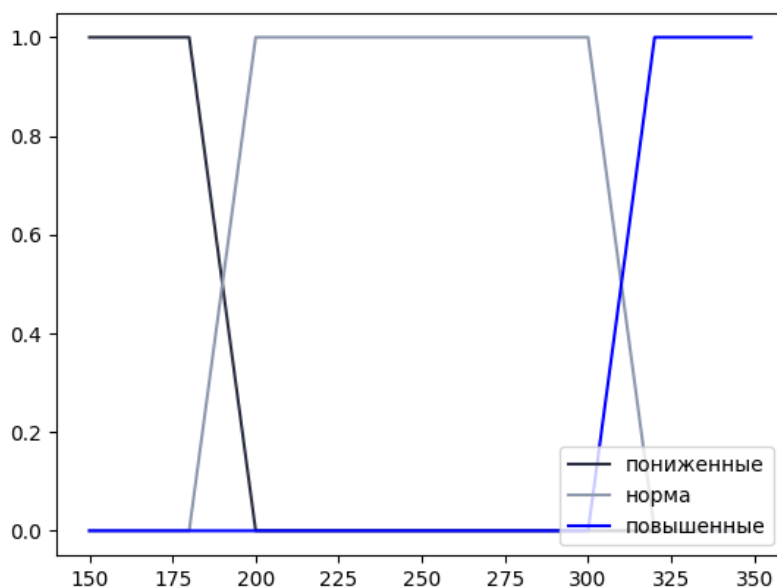


Рис. 11. Графическое представление функций принадлежности переменной “Age”

Фаззификация переменной “Risk” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «низкий», «средний», «высокий» и «очень высокий». Функции принадлежности для этой переменной задаются в виде уравнений (32)–(35). Графическое представление этих функций изображено на рисунке 12.

$$\text{Risk}_{low}(x) = \begin{cases} 1, & x \leq 1, \\ 2 - x, & 1 < x < 2, \\ 0, & x \geq 2. \end{cases} \quad (32)$$

$$\text{Risk}_{moderate}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{0.5}, & 1 < x < 1.5, \\ \frac{2-x}{0.5}, & 1.5 < x < 2, \\ 0, & x \geq 2. \end{cases} \quad (33)$$

$$Risk_{high}(x) = \begin{cases} 0, & x \leq 2, \\ \frac{x-2}{0.5}, & 2 < x < 2.5, \\ \frac{3-x}{0.5}, & 2.5 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (34)$$

$$Risk_{very\ high}(x) = \begin{cases} 0, & x \leq 3, \\ x - 3, & 3 < x < 4, \\ 1, & x \geq 4. \end{cases} \quad (35)$$

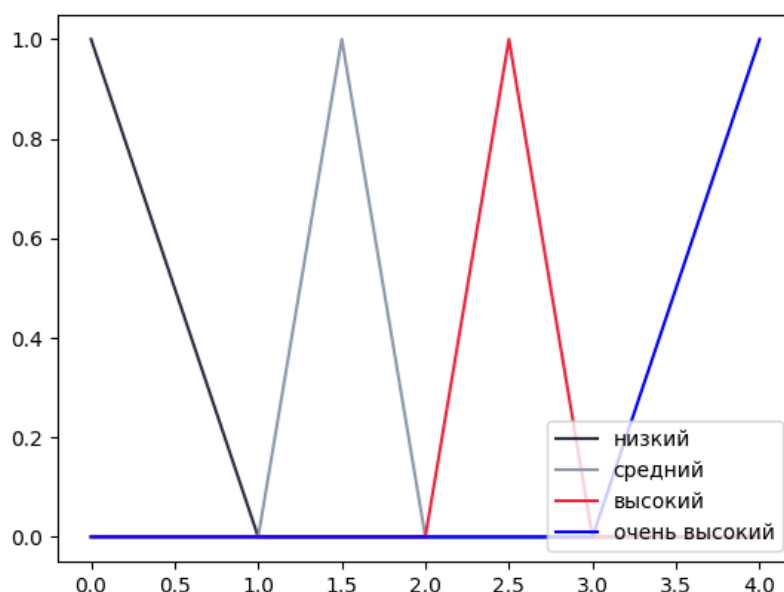


Рис. 12. Графическое представление функций принадлежности переменной “Risk”

2.4. ПРИМЕНЕНИЕ НЕЧЕТКИХ ПРАВИЛ

В рамках нечеткой логики, особенно в алгоритме Мамдани, применение нечетких правил является одним из ключевых этапов. Этот этап включает использование набора заранее определенных нечетких правил для оценки степени выполнения каждого правила на основе фаззифицированных данных [16]. Применение нечетких правил позволяет учитывать множество условий и выводить соответствующие нечеткие результаты, что является основой для принятия решений в экспертных системах.

Для каждого условия правила определяется степень принадлежности входного значения к соответствующему нечеткому множеству. Эта степень принадлежности

указывает, насколько входное значение соответствует данному нечеткому множеству.

Если правило содержит несколько условий, необходимо комбинировать степени выполнения всех условий для определения общей степени выполнения правила [17]:

- Для комбинации “AND” используется минимум значений степеней принадлежности.
- Для комбинации “OR” используется максимум значений степеней принадлежности.

Реализация применения нечетких правил представлена на рисунке 3, где:

- ‘fuzzy_inputs’ – словарь фаззифицированных значений входных переменных.
- ‘rules’ – список правил, где каждое правило состоит из условия и заключения.
- Для каждого правила вычисляется степень выполнения условий, которая определяется как минимальное значение степеней принадлежности условий правила.
- Результаты всех правил возвращаются в виде списка степеней выполнения правил и соответствующих заключений.

```
def apply_fuzzy_rules(fuzzy_inputs, rules):  
    rule_strengths = []  
    for condition, conclusion in rules:  
        strengths = []  
        for cond in condition:  
            variable, label = cond.split(' IS ')  
            strengths.append(fuzzy_inputs[variable][label])  
        rule_strengths = min(strengths)  
        rule_strengths.append((rule_strengths, conclusion))  
    return rule_strengths
```

Рис. 13. Реализация применения нечетких правил

2.5. АГРЕГИРОВАНИЕ

Агрегирование является одним из ключевых этапов алгоритма Мамдани, который выполняется после применения нечетких правил. На этом этапе происходит комбинация (агрегация) всех активированных правил для каждой выходной переменной, чтобы определить общий нечеткий вывод системы [18].

Цель агрегирования состоит в том, чтобы объединить результаты нескольких правил, которые были активированы на предыдущем этапе. Каждое правило имеет свою степень выполнения, которая отражает вклад этого правила в результирующий нечеткий вывод системы [19].

После применения нечетких правил получаем список или словарь, содержащий степень выполнения каждого активированного правила и соответствующее заключение (выходная переменная и ее нечеткое множество). Результаты правил группируются по выходным категориям. Например, все правила, активированные для выходной категории "low", объединяются в одну группу.

Для каждой выходной переменной суммируются степени выполнения всех правил, активированных для этой переменной. Обычно используется операция «максимум» из степеней выполнения для каждого множества принадлежности выходной переменной. Это позволяет учесть различные аспекты, представленные различными правилами [20].

В результате агрегирования получаем агрегированные значения для каждого множества, связанного с выходной переменной. Эти значения представляют собой нечеткий вывод системы после учета всех активированных правил.

Реализация агрегирования приведена на рисунке 14, где:

- Функция принимает список кортежей 'rule_strengths', где каждый кортеж состоит из степени выполнения правила и его вывода.
- Степень выполнения 'strength' – значение от 0 до 1, показывающее, насколько сильно выполнено правило.
- Функция возвращает словарь 'aggregated', где ключи – это метки выходных переменных, а значения – максимальная степень выполнения для каждой категории.

```
def aggregate(rule_strengths):
    aggregated = {}
    for strength, conclusion in rule_strengths:
        if conclusion in aggregated:
            aggregated[conclusion] = max(aggregated[conclusion], strength)
        else:
            aggregated[conclusion] = strength
    return aggregated
```

Рис. 14. Реализация агрегирования

2.6. ДЕФАЗЗИФИКАЦИЯ

Дефаззификация является заключительным этапом в алгоритме Мамдани, где нечеткий вывод преобразуется в четкое значение. На этом этапе агрегированные нечеткие выводы, полученные после применения правил и агрегирования, конвертируются в одно четкое значение, которое можно использовать для принятия решений [21, 22].

Существует несколько методов дефаззификации, но наиболее популярным является метод центра тяжести (центроидный метод).

Метод центра тяжести предполагает вычисление центра тяжести объединенного нечеткого множества. Этот метод дает наиболее сбалансированный результат и широко используется в различных приложениях нечеткой логики.

Основные шаги дефаззификации:

1. Интеграция функции принадлежности: Вычисление взвешенной суммы всех возможных значений.
2. Выбор точки центра тяжести: Деление этой суммы на суммарную функцию принадлежности.

Реализация дефаззификации показана на рисунке 15. Здесь 'numerator' представляет сумму всех значений, а 'denominator' – сумму функций принадлежности. Результат деления – четкое значение, представляющее дефаззифицированный выход.

```
def defuzzify(aggreagated, output_functions, resolution=1000):
    x = np.linspace( start: 0, stop: 1, resolution)
    numerator = 0.0
    denominator = 0.0

    for label, strength in aggreagated.items():
        membership_function = output_functions[label]
        for xi in x:
            mu = min(strength, membership_function.fuzzify(xi))
            numerator += xi * mu
            denominator += mu

    return numerator / denominator if denominator != 0 else 0
```

Рис. 15. Реализация дефаззификации

3. РЕЗУЛЬТАТЫ

Для реализации экспертной системы были выбраны фреймворк Django, предназначенный для высоконадежной и динамичной разработки веб-сайтов, и база данных PostgreSQL.

Выбор фреймворка Django обусловлен тем, что он реализует архитектурный паттерн MVC (Model – View – Controller), и такая архитектура позволяет работать с визуальным представлением и бизнес-логикой приложения отдельно.

Разработанная экспертная система состоит из следующих компонентов:

1. Веб-интерфейс: обеспечивает взаимодействие пользователя с системой.
2. Серверная часть: обрабатывает запросы пользователей, выполняет логические операции и взаимодействует с базой данных. Основные компоненты Django:
 - Модели: определяют структуру данных, хранящихся в базе данных.
 - Представления: обрабатывают запросы пользователей и возвращают ответы.
 - Шаблоны: отображают HTML-страницы, интегрируя данные из представлений.
3. База данных: хранит базу правил и результаты диагностики.

Форма для диагностики пневмонии представляет собой интерфейс, который позволяет медицинским специалистам или пользователям вводить различные клинические данные пациента для оценки вероятности наличия пневмонии (см. рисунок 16). После обработки введенных результатов система выдает результат в виде степени риска наличия пневмонии.

The image shows a web form titled "Pneumonia Diagnosis Form". It contains the following fields from top to bottom:

- Age: Enter age
- Cough (0-10): Enter cough severity
- Shortness Of Breath (0-10): Enter shortness of breath severity
- Temperature (°C): Enter temperature
- Chest Pain (0-10): Enter chest pain severity
- Fatigue (0-10): Enter fatigue level
- Oxygen Level (%): Enter oxygen level
- Leukocyte Count ($\times 10^9/L$): Enter white blood cell count
- Platelet Count ($\times 10^9/L$): Enter platelet count

At the bottom of the form is a blue "Submit" button.

Рис. 16. Форма диагностики

Форма для добавления правил экспертной системы позволяет эксперту добавлять, редактировать и удалять правила, используемые для диагностики пневмонии (см. рисунок 17). Правила состоят из условий (IF) и заключений (THEN), которые описываются экспертом.

Expert Rules Form

IF Condition

THEN Conclusion

[Add Rule](#)

Existing Rules

#	Condition	Conclusion	Actions
1	IF Age IS Middle AND Temperature IS High	THEN Risk IS High	Edit Delete

Рис. 17. Форма для ввода правил

ЗАКЛЮЧЕНИЕ

Представлена разработанная экспертная система для диагностики пневмонии. Описан и реализован алгоритм Мамдани, который включал в себя формирование базы правил, фаззификацию входных данных, применение нечетких правил, агрегирование результатов и дефаззификацию. Экспертные системы на основе нечеткой логики представляют собой инструмент для обработки и анализа нечеткой информации в медицинской практике. Разработанная система демонстрирует принципы и применения нечеткой логики в диагностике пневмонии, предоставляя врачам ценные инструменты для принятия обоснованных и информированных решений, позволяющие автоматизировать процесс диагностики.

СПИСОК ЛИТЕРАТУРЫ

1. Григорьева Д.Р., Гареева Г.А., Басыров Р.Р. Основы нечеткой логики: Учебно-методическое пособие к практическим занятиям и лабораторным работам. Набережные Челны: Изд-во НЧИ КФУ. 2018. 42 с.
2. Fuzzy logic.
URL: https://en.wikipedia.org/wiki/Fuzzy_logic, last accessed 27/05/2024.
3. Zadeh L.A. The role of fuzzy logic in the management of uncertainty in expert systems // Fuzzy sets and systems, 1983. 11(1-3), P. 199-227.
4. Negnevitsky M. Artificial Intelligence: A guide to Intelligent Systems (2nd ed.). Addison-Wesley, 2005. 435 p.

5. *Demetriou D., See L.M., Stillwell J.* Expert Systems for Planning and Spatial Decision Support // *GeoComputation*, Chapter 11, 2014. P. 257–274. <https://doi.org/10.1201/b17091-12>.
6. *Zadeh L.A.* Fuzzy sets // In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, 1996. P. 394–432.
7. *Pedrycz W., Gomide F.* *Fuzzy Systems Engineering: Toward Human-Centric Computing* // John Wiley & Sons, 2007. 549 p.
8. *Umair Ahmed, Ghulam Rasool, Saqib Zafar, Farhan Maqbool* Fuzzy Rule Based Diagnostic System to Detect the Lung Cancer // *International conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 2018. <https://doi.org/10.1109/ICECUBE.2018.8610976>.
9. Tools To Generate Fuzzy Takagi Sugeno Method by Python. <https://medium.com/@zekkelar1337/tools-to-generate-fuzzy-takagi-sugeno-method-by-python-748f978e3673>, last accessed 27/05/2024.
10. Comparative analysis of Mamdani, Larsen and Tsukamoto methods of fuzzy inference system for students' academic performance evaluation. URL: <https://ijsra.net/sites/default/files/IJSRA-2023-0443.pdf>, last accessed 25/05/2024.
11. Pneumonia Symptoms and Diagnosis. URL: <https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis>, last accessed 27/05/2024.
12. Pneumonia Symptoms. URL: <https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204>, last accessed 27/05/2024.
13. *Arani L.A., Sadoughi F., Langarizadeh M.* An Expert System to Diagnose Pneumonia Using Fuzzy Logic // *Acta Inform Med*, 2019. P. 103–107. <https://doi.org/10.5455/aim.2019.27.103-101>
14. *Fazel Zarandi M.H., Zolnoori M., Moin M., Taherian M.* Fuzzy Rule-Based Expert System for Evaluating Level of Asthma Control // *J Med Syst*. 2012. V. 36, no. 5. P. 2947–2958. <https://doi.org/10.1007/s10916-011-9773-3>.
15. *Azaab S., Abu Naser S., Sulisel O.* A proposed expert system for selecting exploratory factor analysis procedures // *Journal of the college of education*. 2000 . V. 4, no. 2. P. 9–26.
16. *Hasan M.A., Sher-e-Alam K., Chowdhury A.R.* Human Disease Diagnosis Using

a Fuzzy Expert System // Journal of Computing. 2010. V. 2, no. 6.. P. 66–70.
<https://doi.org/10.48550/arXiv.1006.4544>

17. *Febria Anjara, Adam Amril Jaharadak.* Expert system for disease diagnosis in living things: A narrative review // Journal of Physics: Conference series. 2019. V. 1167.
<https://doi.org/10.1088/1742-6596/1167/1/012070>.

18. *Pavate A., Nerurkar P., Ansari N., Bansode R.* Early Prediction of Five Major Complications Ascends in Diabetes Mellitus Using Fuzzy Logic // Soft Computing in Data Analytics, Advances in Intelligent Systems and Computing. 2019. P. 759–768.
https://doi.org/10.1007/978-981-13-0514-6_72

19. *Russell S., Norvig P.* Artificial Intelligence: A Modern Approach (3rd ed.). Pearson Education, 2010. 1152 p.

20. *Buchanan B.G., Shortliffe E.H.* Rule-based expert systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, 1984. 503 p.

21. *Giarratano J.C., Riley G.* Expert Systems: Principles and Programming (4th ed.) Course Technology, 2005. 868 p.

22. *Waterman D.A.* A Guide to Expert Systems. Addison-Wesley, 1986. 419 p.

23. *Burnashev R.A., Khairullin B.M., Prokopyev N.A., Farahov R.A., Bol-sunovskaya M.V., Enikeev A.I.* Design and Development of a Research Integrated Geoinformation System with a Fuzzy Expert System // 2023 IEEE 24th International Conference of Young Professionals in Electron Devices and Materials (EDM), Novosibirsk, Russian Federation, 2023. P. 1310–1313. <https://doi.org/10.1109/EDM58354.2023.10225118>.

24. *Burnashev R.* Designing a Prototype of an Adaptive Expert System Using Fuzzy Logic and a Genetic Algorithm // 2023 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2023. P. 1168–1172.
<https://doi.org/10.1109/RusAutoCon58002.2023.10272949>.

DEVELOPMENT OF A FUZZY LOGIC-BASED EXPERT SYSTEM FOR PNEUMONIA DIAGNOSTICS

A. I. Enikeeva¹ [0009-0002-9192-952x], R. A. Burnashev² [0000-0002-1057-0328],

R. R. Farahov³ [0000-0002-7993-3182]

¹⁻³Kazan (Volga Region) Federal University

¹adisen24@yandex.ru, ²r.burnashev@inbox.ru, ³rus-wing-dark@mail.ru

Abstract

The article is devoted to the development of an expert system for diagnosing pneumonia. This system is based on fuzzy logic and implemented using the Mamdani algorithm. The main stages of system development are considered, including the fuzzification of input data, the definition of fuzzy rules based on medical expert knowledge, the aggregation of fuzzy inferences, and their defuzzification to obtain the final diagnostic result. The system's web interface is implemented using the Django framework, ensuring ease of interaction for users. The use of a medical expert system for pneumonia diagnostics can reduce the time required to establish a diagnosis and improve the quality of diagnosis by integrating the experience of medical experts and modern information technologies.

Keywords: *expert system, fuzzy logic, medical diagnostics, Mamdani algorithm, decision support system.*

REFERENCES

1. Grigoreva D.R., Gareeva G.A., Basyrov R.R. *Osnovy nechetkoj logiki: Uchebno-metodicheskoe posobie k prakticheskim zanyatiyam I laboratornym rabotam.* Naberezhnye Chelny: Izd-vo NChI KFU, 2018. 42 s.
2. Fuzzy logic.
URL: https://en.wikipedia.org/wiki/Fuzzy_logic, last accessed 27/05/2024.
3. Zadeh L.A. The role of fuzzy logic in the management of uncertainty in expert systems // *Fuzzy sets and systems*, 1983. 11(1-3), P. 199-227.
4. Negnevitsky M. *Artificial Intelligence: A guide to Intelligent Systems* (2nd ed.). Addison-Wesley, 2005. 435 p.
5. Demetriou D., See L.M., Stillwell J. *Expert Systems for Planning and Spatial Decision Support* // *GeoComputation*, Chapter 11, 2014. P. 257–274.

<https://doi.org/10.1201/b17091-12>.

6. Zadeh L.A. Fuzzy sets // In Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh, 1996. P. 394–432.

7. Pedrycz W., Gomide F. Fuzzy Systems Engineering: Toward Human-Centric Computing // John Wiley & Sons, 2007. 549 p.

8. Umair Ahmed, Ghulam Rasool, Saqib Zafar, Farhan Maqbool Fuzzy Rule Based Diagnostic System to Detect the Lung Cancer // International conference on Computing, Electronic and Electrical Engineering (ICE Cube), 2018. <https://doi.org/10.1109/ICECUBE.2018.8610976>.

9. Tools To Generate Fuzzy Takagi Sugeno Method by Python.
URL: <https://medium.com/@zekkelar1337/tools-to-generate-fuzzy-takagi-sugeno-method-by-python-748f978e3673>, last accessed 27/05/2024.

10. Comparative analysis of Mamdani, Larsen and Tsukamoto methods of fuzzy inference system for students' academic performance evaluation. <https://ijsra.net/sites/default/files/IJSRA-2023-0443.pdf>, last accessed 25/05/2024.

11. Pneumonia Symptoms and Diagnosis.
URL: <https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis>, last accessed 27/05/2024.

12. Pneumonia Symptoms.
URL: <https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204>, last accessed 27/05/2024.

13. Leila Akramian Arani, Frahnaz Sadoughi, Mustafa Langarizadeh. An Expert System to Diagnose Pneumonia Using Fuzzy Logic // Acta Inform Med, 2019. P. 103–107. <https://doi.org/10.5455/aim.2019.27.103-101>

14. Fazel Zarandi M.H., Zolnoori M., Moin M., Taherian M. Fuzzy Rule-Based Expert System for Evaluating Level of Asthma Control // J Med Syst. 2012. V. 36, no. 5. P. 2947–2958. <https://doi.org/10.1007/s10916-011-9773-3>.

15. Azaab S., Abu Naser S., Sulisel O. A proposed expert system for selecting exploratory factor analysis procedures // Journal of the college of education. 2000. V. 4, no. 2. P. 9–26.

16. Hasan M.A., Sher-e-Alam K., Chowdhury A.R. Human Disease Diagnosis Using a Fuzzy Expert System // Journal of Computing. 2010. V. 2, no. 6.. P. 66–70.

<https://doi.org/10.48550/arXiv.1006.4544>

17. *Febria Anjara, Adam Amril Jaharadak.* Expert system for disease diagnosis in living things: A narrative review // *Journal of Physics: Conference series.* 2019. V. 1167. <https://doi.org/10.1088/1742-6596/1167/1/012070>.
18. *Pavate A., Nerurkar P., Ansari N., Bansode R.* Early Prediction of Five Major Complications Ascends in Diabetes Mellitus Using Fuzzy Logic // *Soft Computing in Data Analytics, Advances in Intelligent Systems and Computing.* 2019. P. 759–768. https://doi.org/10.1007/978-981-13-0514-6_72
19. *Russell S., Norvig P.* *Artificial Intelligence: A Modern Approach* (3rd ed.). Pearson Education, 2010. 1152 p.
20. *Buchanan B.G., Shortliffe E.H.* *Rule-based expert systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, 1984. 503 p.
21. *Giarratano J.C., Riley G.* *Expert Systems: Principles and Programming* (4th ed.) Course Technology, 2005. 868 p.
22. *Waterman D.A.* *A Guide to Expert Systems.* Addison-Wesley, 1986. 419 p.
23. *Burnashev R.A., Khairullin B.M., Prokopyev N.A., Farahov R.A., Bol-sunovskaya M.V., Enikeev A.I.* Design and Development of a Research Integrated Geoinformation System with a Fuzzy Expert System // 2023 IEEE 24th International Conference of Young Professionals in Electron Devices and Materials (EDM), Novosibirsk, Russian Federation, 2023. P. 1310–1313. <https://doi.org/10.1109/EDM58354.2023.10225118>.
24. *Burnashev R.* Designing a Prototype of an Adaptive Expert System Using Fuzzy Logic and a Genetic Algorithm // 2023 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2023. P. 1168–1172. <https://doi.org/10.1109/RusAutoCon58002.2023.10272949>.

СВЕДЕНИЯ ОБ АВТОРАХ



ЕНИКЕЕВА Аделя Искандеровна – аспирант Института информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета, ассистент кафедры анализа данных и технологий программирования Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

Adelya ENIKEEVA – Post-graduate Student, Institute of Information Technologies and Intelligent Systems, Assistant at the Department of Data Analysis and Programming technologies, Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University. Current scientific interests: human-computer interaction, software engineering, data management systems.

email: adisen24@yandex.ru;

ORCID: 0009-0002-9192-952x



БУРНАШЕВ Рустам Арифович – доцент кафедры анализа данных и технологий программирования Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

Rustam BURNASHEV – Associate Professor, Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University.

Current scientific interests: expert systems, data analysis, fuzzy logic, machine learning.

email: r.burnashev@inbox.ru;

ORCID: 0000-0002-1057-0328



ФАРАХОВ Рустам Ринатович – ассистент кафедры анализа данных и технологий программирования Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

Rustam FARANOV – Assistant at the Department of Data Analysis and Programming Technologies, Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University.

Current scientific interests: expert systems, internet of things.

email: rus-wing-dark@mail.ru;

ORCID: 0000-0002-7993-3182

Материал поступил в редакцию 15 мая 2024 года