

Дата подання: 30.09.2022

Дата прийняття:

УДК 621.391

АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ НЕЧЁТКОЙ СИСТЕМЫ С МАТРИЧНЫМ ПРЕДСТАВЛЕНИЕМ ПРАВИЛ И КОРРЕКТНОЙ БАЗОЙ ЗНАНИЙ

Д. И. Егошкин, Н. А. Гук

АВТОМАТИЧНА ПОБУДОВА НЕЧІТКОЇ СИСТЕМИ З МАТРИЧНИМ ПРЕДСТАВЛЕННЯ ПРАВИЛ ТА КОРЕКТНОЮ БАЗОЮ ЗНАНЬ

Д. І. Єгошкін, Н. А. Гук

AUTOMATIC CONSTRUCTION OF A FUZZY SYSTEM WITH A MATRIX REPRESENTATION OF THE RULES AND A CORRECT KNOWLEDGE BASE

Danylo Yehoshkin, Natalia Guk

Объектом исследования является процесс автоматического формирования нечетких продукционных правил на основе обучающей выборки для решения задачи классификации. Решается проблема автоматического создания и последующей проверки правильности нечеткой модели логического вывода для задачи классификации. Результатом является автоматически построенная корректная база правил для решения задачи классификации. Анализ правильности базы знаний на корректность осуществляется при помощи критериев полноты, минимальности, связности и непротиворечивости. Для доказательства полноты базы правил применяется логика Хоара и метод резолюций. Проводится оценка качества классификации при помощи метрик: accuracy, precision, recall, f1-score. Рассмотрена зависимость результата классификации, от размера обучающей выборки.

Экспертная система обладает следующими особенностями: способностью обучаться на данных; высоким уровнем точности; корректной базой знаний. База знаний формируется с использованием объектов обучающей выборки на основании лингвистических переменных и терм-множеств. Применяется продукционная модель представления знаний, объединяющая модели Мамдани и Такаги-Сугено-Канг. Предполагается, что левые части продукционных правил описывают сочетания признаков объектов, а правые части соответствуют классам. Используется матричное представление антецедентов правил. Консеквенты представляются в виде вектора столбца. Для автоматического построения матрицы антецедентов предлагается использовать декартово произведение. Формирование вектора консеквентов

осуществляется автоматически с использованием данных о предметной области и процедуры обучения.

Полученная экспертная система может применяться для решения задач классификации, кластеризации, глубинного анализа данных, анализа больших данных.

Ключевые слова: экспертная система, нечеткая логика, нечеткая классификация, логика Хоара, метод резолюций.

1. Введение

Известно, что большинство современных подходов к машинному обучению оказали большое влияние на многие отрасли [1]. Развитие технологий позволило широко применять системы искусственного интеллекта для поддержки принятия решений в таких областях знаний как экономика, промышленность, медицина, наука, торговля, строительство, транспорт. Информация о предметных областях представляет собой большие массивы данных, которые необходимо обработать для принятия правильных решений. Поэтому для решения широкого класса практических задач применяются такие методы искусственного интеллекта как нечеткая логика, нейронные сети, генетические алгоритмы и другие. Это позволяет избежать применения точных математических моделей, основанных на аппарате математических уравнений и классической теории принятия решений. С помощью искусственного интеллекта создаются системы для имитации деятельности экспертов в различных областях. Задачи управления, идентификации, моделирования сложных физических явлений, классификации, распознавания образов успешно решаются с использованием элементов нечеткой логики [1]. Применение нечеткой логики обусловлено тем, что она на интуитивном уровне соответствует процессу человеческих рассуждений в условиях нечеткости и неполноты условий задачи. Поскольку система продукционных правил может быть неполной, несвязной и противоречивой, то после ее формирования необходимо проверить данную систему на корректность.

В ряде работ рассматриваются различные модели представления продукционных баз знаний и исследуется корректность экспертной системы после её формирования [2–7]. Однако предложенные подходы для проверки корректности базы правил не позволяют осуществлять проверку в автоматическом режиме с учетом информации об объекте исследования, что определяет необходимость теоретических и практических исследований в этой области знаний.

2. Анализ литературных данных и постановка проблемы

Автоматическое формирование продукционной базы правил в экспертных системах и доказательство ее корректности является актуальной задачей, поскольку позволяет обеспечить простоту разработки таких систем и высокое качество логического вывода. В литературе для решения подобных проблем развиваются разные подходы.

Для анализа достоверности базы знаний нечеткого логического вывода в

работе [2] рассматривается представление системы правил в виде метаграфа. При этом нечеткая база знаний представляется с использованием модели Мамдани. После процедуры верификации база правил должна удовлетворять следующим условиям: избыточности, лингвистической непротиворечивости, отсутствия заикливания и лингвистической полноты. Предложено проводить статическую верификацию баз нечетких знаний на основе структуры метаграфа путем нахождения заикливания в структуре графа. В работе рассматривается база знаний, построенная экспертом, имеет смысл проверить данный подход на автоматически генерируемых базах знаний с большим количеством правил.

В [3] рассматривается вопрос автоматизации проверки корректности базы знаний продукционных правил. Предлагается квантовая модель кодирования символов многозначного алфавита при помощи кубического исчисления. В качестве алгоритма вывода используется алгоритм Мамдани. Разрабатывается компьютерная программа, которая позволяет выполнить формальную проверку базы продукционных правил на корректность при помощи метода резолюций. Теоретико-множественные операции над символами алфавита сводятся к соответствующим поразрядным логическим операциям над их кубитами. К недостаткам данного подхода можно отнести следующее: входные лингвистические переменные должны иметь одинаковое количество термов, а термы разных входных лингвистических переменных должны иметь одинаковые диапазоны значений. Кроме того, продукционные правила должны быть полностью определены, т. е. любому сочетанию значений термов входных лингвистических переменных соответствует определенное значение термов выходной лингвистической переменной.

Также для представления базы знаний применяются ациклические графы. В работе [4] для аккумуляции больших наборов знаний в реальном времени применяется алгоритм построения ориентированного ациклического графа. Последующая трансляция графа осуществляется при помощи специализированного языка TLC (Target Language Compiler). В работе предлагается проводить проверку базы знаний на корректность на этапе трансляции. Применяются как классические методы с использованием дизъюнктивных и конъюнктивных нормальных форм, так и с использованием нормальной формы отрицания NNF (Negation Normal Form). Это позволяет выводить пользователю контрпримеры, если база знаний является противоречивой или не полной. В работе рассматривается построение базы знаний в реальном времени, но не рассматривается эффект переобучения.

В случае, если необходимо представить данные в табличном виде для применения матричных и векторных операций обработки, используется таблица принятия решений (ТПР) [5]. В последующей работе автора были внесены изменения в алгоритм формирования, но представление при помощи таблиц принятия решений осталось неизменным [6]. Большим преимуществом ТПР является наглядность и удобное представление данных для эксперта в процессе анализа и заполнения базы правил. Также достоинством является то, что основными операциями продукционного цикла выступают логические (векторные и матричные) операции, позволяющие получить высокую скорость

обработки. Это позволяет весьма эффективно использовать табличную модель представления знаний. Для проверки базы знаний используется дизъюнктивная нормальная форма, удобна для автоматического доказательства теорем. Процесс доказательства основывается на логике высказываний и логике предикатов. В работах [5, 6] при использовании ТПР – антецеденты и консеквенты представлены композиционно, имеет место рассматривать антецеденты как отдельную матрицу, а консеквенты как вектор.

Для генерации базы правил типа Мамдани в работе [7] предлагается определение оптимальных консеквентов на основе использования мультиагентных оптимизационных алгоритмов, в частности муравьиных. Данный подход позволяет эффективно генерировать базы правил в следующих случаях:

- 1) при недостаточном объеме исходной информации;
- 2) при достаточно большом количестве правил, для которых составление базы нечетких правил на основе знаний экспертов не всегда эффективно;
- 3) при различном уровне квалификации экспертов.

При этом не рассматривается вопрос о достаточном количестве циклов генераций и времени для достижения необходимого уровня точности модели.

Указанные подходы в источниках [2–7], существенно повышают эффективность использования интеллектуальных систем. Позволяют формировать и проводить верификацию базы продукционных правил, однако:

- 1) накладывают ограничения на количество термов лингвистических переменных;
- 2) требуют наличие эксперта для анализа работы верификации;
- 3) накладывают ограничения на количество логических выводов.

Рассматривается задача классификации. Для решения поставленной задачи применяется подход, основанный на нечёткой логике.

Информация о предметной области представляется в виде продукционной модели в предположении, что левые части правил продукций описывают сочетания признаков объектов, а правые соответствуют классам.

При формировании левых частей продукций используются признаки объектов из некоторого конечного (квазиконечного, если допустимо пополнение модели) множества, конъюнкция истинных значений которых и определяет условия применимости продукции. В правой части указываются классы из некоторого конечного допустимого множества классов объектов.

Для обеспечения достоверности логического вывода на основе сформулированной базы знаний необходимо, чтобы база знаний обладала свойствами полноты, минимальности (не избыточности), непротиворечивости и связанности. Разрабатывается подход, который обеспечивает проверку этих свойств.

3. Цель и задачи исследования

Целью исследования является разработка подхода к автоматической генерации нечетких продукционных правил на основе обучающей выборки для решения задачи классификации с последующей проверкой корректности

построенной модели. Это позволит создать нечёткие экспертные системы, способные обучаться на тестовых наборах данных. Продукционные базы правил, таких обученных систем, позволят экспертам находить скрытые зависимости между признаками объектов и их классами.

Для достижения цели были поставлены следующие задачи:

- создать продукционную модель представления знаний об объектах предметной области, объединяющую модели Мамдани и Такаги-Сугено-Канг, на основе результатов наблюдения за объектами предметной области.
- разработать алгоритм автоматического формирования и выбрать способ представления базы правил;
- разработать подход для проверки корректности базы правил с использованием логики Хоара и метода резолюций. Применить программное обеспечение Simplify для автоматического доказательства непротиворечивости;
- выбрать метрики для оценки качества полученной нечёткой базы правил;
- применить разработанный подход для проверки корректности базы правил для решения задачи классификации.

4. Материалы и методы исследования

Объектом исследования является процесс автоматического формирования нечетких продукционных правил на основе обучающей выборки для решения задачи классификации. Формулируется задача следующим способом: необходимо на основе обучающей выборки построить нечеткие продукционные правила для MISO системы. При этом основания гипотеза формирования правил, заключается в том, что правила строятся на основе лингвистических переменных и их терм-множеств.

Для построения антецедентов правил использованы признаки объектов из конечного терм-множества, конъюнкция истинных значений которых определяет условия применимости продукции. Антецеденты правил представляются в виде матрицы, это необходимо для автоматического формирования матрицы при помощи декартового произведения. Для консеквентов указываются классы из некоторого конечного допустимого множества классов объектов. Консеквенты правил представляются в виде вектора столбца, значения элементов вектора зависят от матрицы антецедентов и объектов обучающей выборки. Формирование вектора консеквентов осуществляется при помощи процедуры обучения базы знаний.

Для проверки полученной базы знаний на корректность, используются критерии: полноты, минимальности, связности и непротиворечивости. Для доказательства полноты применяется логика Хоара, метод резолюций и приложение Simplify, которое осуществляет автоматическую проверку системы на непротиворечивость с использованием логики первого порядка.

Разработанный подход применяется к решению задачи классификации, оценка качества полученной нечёткой базы правил проводится при помощи метрик accuracy, precision, recall, f1-score.

5. Результаты разработки методики анализа корректности

продукционной модели

5. 1. Продукционная модель представления знаний и метод логического вывода

Объекты предметной области в задаче классификации описываются системой признаков $\hat{k}_1, \hat{k}_2, \dots, \hat{k}_L$ для каждого объекта $x_i \in X$, признаки соответствуют лингвистическим переменным k_1, k_2, \dots, k_L , при этом каждая лингвистическая переменная содержит терм множество A_{lt} , где l номер признака $l = \overline{1, L}$, а $t = \overline{1, T_l}$; T_l – количество термов для признака k_l . Выходная переменная y соответствует классу, которому принадлежит объект предметной области. Для решения задачи классификации построим продукционную модель представления знаний, объединяющую модели Мамдани и Такаги-Сугено-Канг [8].

Правила Π_p типа MISO имеют вид:

$$\Pi_p : \text{IF } k_1 \text{ is } \tilde{a}_{p1} \text{ AND } k_2 \text{ is } \tilde{a}_{p2} \text{ AND } \dots \text{ AND } k_L \text{ is } \tilde{a}_{pL}, \quad (1)$$

$$\text{THEN } y \text{ is } f(k_1, \dots, k_L, p) \cdot c_p,$$

где k_1, k_2, \dots, k_L – входные лингвистические переменные; \tilde{a}_{pl} – переменная, принимающая нечеткие значения A_{lt} для правила p ; A_{lt} – нечеткое терм-множество лингвистической переменной k_l для правила p ; t – номер терма для k_l , $t = \overline{1, T_l}$; T_l – количество термов для признака k_l ; y – выходная переменная; p – номер правила в базе правил, $p = \overline{1, P}$; P – общее количество правил Π ; c_p – переменная, принимающая нечеткие значения C_m для правила p ; C_m – метка нечеткого класса, к которому принадлежит объект $x_i \in X$; m – номер класса C , $m = \overline{1, M}$; M – количество классов C .

Вещественная, неотрицательная, нормированная функция $f(k_1, \dots, k_L, p)$ непрерывна на отрезке $[a, b]$ и используется в качестве весового коэффициента для консеквента правила. Она определяет степень принадлежности выходной переменной y к терм-множеству C_m и вычисляется следующим образом:

$$f(k_1, \dots, k_L, p) = \frac{\sum_{l=1}^L \mu_{a_{pl}}(k_l)}{L}. \quad (2)$$

Процедура фаззификации определяется следующим образом:

$$a_{pl} = \int_{\tilde{k}_l}^{\tilde{k}_l} \mu_{a_{pl}}(k_l) |k_l| dk_l, \quad (3)$$

где $\mu_{a_{pl}}(k_l)$ – функция принадлежности четкого значения входной переменной k_l нечеткому терму a_{pl} .

Для автоматизации процесса построения правил, удобно представить antecedentes правил в виде матрицы A :

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1l} \\ a_{21} & a_{22} & \dots & a_{2l} \\ \dots & \dots & \dots & \dots \\ a_{p1} & a_{p2} & \dots & a_{pl} \end{pmatrix}, \quad (4)$$

где a_{pl} – элемент матрицы, принимающий значение \tilde{a}_{pl} – переменной из системы правил, и равный нечеткому значению A_{lt} для правила p ; P – количество правил; L – размерность вектора признаков $\bar{K}=(k_1, k_2, \dots, k_L)$ объекта $x_i \in X$.

Консеквенты правил будем представлять в виде вектора столбца \tilde{C} :

$$\tilde{C} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \dots \\ c_p \end{pmatrix}. \quad (5)$$

Далее, нечеткое множество \tilde{C} классов, что отвечают выходной переменной y , определяется в виде:

$$\tilde{C}_m = \langle \sum_{p=1}^P f(k_1, \dots, k_L, p) | C_m \in \Pi_p, C_m \rangle.$$

Таким образом, множество \tilde{C} – будет состоять из упорядоченных пар $\langle f_m / C_m \rangle$, таких что:

$$f_m = \sum_{p=1}^P f(k_1, \dots, k_L, p) | C_m \in \Pi_p,$$

где пара $\langle f_m / C_m \rangle$ – определяет степень принадлежности f_m объекта y_i классу C_m .

Для дефазификации \tilde{C} применяется алгоритм Мамдани с центроидным методом. Нечеткое множество \tilde{C} классов, соответствующее входному объекту X определяется в виде:

$$\tilde{C} = \text{agg}_{m=1, M} \left(\int_{\tilde{C}} \text{imp}(f_m \cdot \mu_{C_m}(c), \mu_{C_m}(c) / c) dc \right),$$

где операции импликации imp и агрегирования agg реализуются путем нахождения \min и \max соответственно.

$$C^* = \int_{\tilde{C}} C \cdot \mu_{\tilde{C}}(c) dc / \int_{\tilde{C}} \mu_{\tilde{C}}(c) dc.$$

Четкое значение логического вывода C^* определяется в результате дефазификации нечеткого множества \tilde{C} по методу центра тяжести.

5. 2. Алгоритм автоматического формирования базы правил

Ниже рассмотрен процесс автоматического формирования модели представления знаний в виде матрицы антецедентов A и вектора консеквентов \vec{c} .

Для каждой лингвистической переменной k_l , существует конечное (квазиконечного, если допустимо пополнение модели) нечеткое терм-множество $A_l=(A_{l1}, A_{l2}, A_{l3},..., A_{lt})$ – определяющее конечный алфавит для описания состояний или множества состояний лингвистической переменной.

Количество нечетких классов C_m , к которым принадлежит объект $x_i \in X$ также ограничено. Таким образом, элементы A_{lt} и C_m определяют алфавит продукционной модели.

Формально матрица антецедентов A задаётся упорядоченным набором $\langle A_{11}, ..., A_{lt} \rangle$, где каждая вектор-строка матрицы A представляет собой множество условий продукционного правила. Вектор консеквентов \vec{c} состоит из классов C_m и имеет размерность равную количеству правил, задает соответствие между продукционным правилом в матрице A и нечетким классом C_m .

Множество продукционных правил Π можно представить в виде: $\Pi_p = \langle a_p, c_p \rangle$, где a_p – вектор-строка матрицы A . Терм-множество $A_l=(A_{l1}, A_{l2}, A_{l3},..., A_{lt})$ – упорядочено относительно семантического порядка и границ термов. Для построения матрицы A предлагается использовать декартово произведение:

$$A_1 \times A_2 \times A_3 \times ... \times A_l = A^{*l}.$$

После чего, данное множество A^{*l} , можно представить в виде следующей матрицы A :

$$A = \begin{pmatrix} A_{11} & A_{21} & A_{31} & ... & A_{l1} \\ A_{12} & A_{22} & A_{32} & ... & A_{l2} \\ A_{13} & A_{23} & A_{33} & ... & A_{l3} \\ ... & ... & ... & ... & ... \\ A_{1t} & A_{2t} & A_{3t} & ... & A_{lt} \end{pmatrix}. \quad (6)$$

Количество правил P определяется из свойств декартового произведения и равно произведению количества элементов множеств-сомножителей:

$$P = \prod_{l=1}^L |A_l| = \prod_{l=1}^L T_l. \quad (7)$$

Формирование вектора консеквентов осуществляется при помощи процедуры обучения базы знаний. Каждый объект из обучающей выборки X^{Train} проходит процедуру фаззификации $fuzz(x_i^{Train})$, после чего вектор $\vec{c}=(c_1, c_2, ..., c_p)$ определяется следующим образом:

$$c_p = C_m : \max \left(\text{card} \left(fuzz(x_i^{Train}) \in a_p, C_m \right) \right), \quad (8)$$

где $card()$ – функция мощности множества.

Вектор \vec{c} должен содержать в себе все классы C_m , в противном случае на вход подана неполная или неверная обучающая выборка и необходимо переобучить систему на другом наборе данных.

5. 3. Анализ корректности базы знаний

База знаний должна удовлетворять единым формальным требованиям корректности, а именно, быть полной, минимальной, связной и непротиворечивой [3].

Под полнотой следует понимать, что любому входящему сочетанию значений термов входных лингвистических переменных соответствует определенное правило в базе правил. Минимальная база правил – это база из которой нельзя удалить ни одного из продукционных правил, не нарушив тем самым её полноту. База знаний является непротиворечивой (согласованной), если она не содержит несовместные правила – правила с одинаковыми лингвистическими условиями, но различными заключениями. Минимальность и непротиворечивость формируемой базы знаний обеспечивается свойствами декартового произведения. За связность отвечает вектор \vec{c} построенный в процессе обучения на основе обучающей выборки (8). Каждая строка матрицы (4) отличаются только одним из l подусловий, так как в начале формирования матрицы A терм-множество $A_l = (A_{l1}, A_{l2}, A_{l3}, \dots, A_{li})$ упорядочено относительно семантического порядка и границ термов.

Для автоматического доказательства полноты базы правил применяется логика Хоара [9]. Определим тройку Хоара: $\{Q\} S \{R\}$, где Q – предикат предусловия, S – набор команд, R – предикат постусловия. Под набором команд S будем понимать множество продукционных правил $\Pi_p = \langle a_p, c_p \rangle$:

$$S: \langle A, \vec{C} \rangle.$$

Предусловием в данном предикате описываются все возможные начальные значения, которые могут принимать лингвистические переменные, для замыкания множества принадлежности входных значений:

$$Q: \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right) \wedge \bigvee_{m=1}^M \text{result} = C_m.$$

Предикат $Q=T$ – истина, так как $\bigvee_{l=1}^L k_l$ принадлежит хотя бы одному A_{li} .

Предикат постусловия описывает ожидаемый результат выполнения S :

$$R: \bigvee_{p=1}^P \left(\left(\bigwedge_{l=1}^L k_l \in a_{pl} \right) \wedge \text{result} = c_p \right).$$

Если предусловие Q выполняется, команда S делает верным постусловие R [10].

Данное утверждение, доказывается с использованием предиката слабейшего предусловия $WP(S, R)$ [11]:

$$Q \Rightarrow WP(S, R). \quad (9)$$

Предикат WP выполняет подстановку $\langle A, \bar{C} \rangle$ из S в R .

$$Q \Rightarrow R_S. \quad (10)$$

Для доказательства истинности предиката (10) воспользуемся методом резолюций:

$$Q \models R_S,$$

$$\begin{aligned} & \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right) \wedge \bigvee_{m=1}^M \text{result} = C_m \models \\ & = \bigvee_{p=1}^P \left(\left(\bigwedge_{l=1}^L k_l \in a_{pl} \right) \wedge \text{result} = c_p \right) \Big|_S, \end{aligned}$$

$$\left\{ \begin{aligned} & \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right), \bigvee_{m=1}^M \text{result} = C_m, \\ & \neg \bigvee_{p=1}^P \left(\left(\bigwedge_{l=1}^L k_l \in a_{pl} \right) \wedge \text{result} = c_p \right) \Big|_S \end{aligned} \right\},$$

$$\left\{ \begin{aligned} & \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right), \bigvee_{m=1}^M \text{result} = C_m, \\ & \neg \left(\bigvee_{p=1}^P \left(\bigwedge_{l=1}^L k_l \in a_{pl} \right) \Big|_S \wedge \bigvee_{p=1}^P (\text{result} = c_p) \Big|_S \right) \end{aligned} \right\},$$

$$\left\{ \begin{aligned} & \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right), \bigvee_{m=1}^M \text{result} = C_m, \\ & \bigwedge_{p=1}^P \left(\bigvee_{l=1}^L \neg (k_l \in a_{pl}) \right) \Big|_S \vee \neg \left(\bigvee_{p=1}^P \text{result} = c_p \right) \Big|_S \end{aligned} \right\}.$$

Так как c_p – принимают значения из C_m , после подстановки S , получаем:

$$\left\{ \begin{aligned} & \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right), \bigvee_{m=1}^M \text{result} = C_m, \\ & \bigwedge_{p=1}^P \left(\bigvee_{l=1}^L \neg (k_l \in a_{pl}) \right) \Big|_S \vee \neg \left(\bigvee_{m=1}^M \text{result} = C_m \right) \end{aligned} \right\}.$$

Далее правило резолюции применяется ко второму и третьему дизъюнктам:

$$\left\{ \begin{aligned} & \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right), \bigwedge_{p=1}^P \left(\bigvee_{l=1}^L \neg (k_l \in a_{pl}) \right) \Big|_S \end{aligned} \right\}.$$

Учитывая (7), получим:

$$\left\{ \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right), \bigwedge_{p=1}^{\prod_{l=1}^L T_l} \left(\bigvee_{l=1}^L \neg (k_l \in a_{pl}) \right) \right\}_S.$$

Применим закон дистрибутивности ко второму дизъюнкту и осуществим подстановку S :

$$\left\{ \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right), \bigvee_{l=1}^L \left(\bigwedge_{t=1}^{T_l} \neg (k_l \in A_{lt}) \right) \right\},$$

$$\left\{ \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right), \neg \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} (k_l \in A_{lt}) \right) \right\}.$$

В результате применения правила резолюций к первому и второму дизъюнктам получим: $\{_ \}$. Приведенное доказательство показывает полноту постулов на основании предикат WP .

Для автоматической проверки базы правил будем использовать программное обеспечение Simplify [12], в котором реализован метод резолюций для доказательства истинности заданных предикатов на основании логики первого порядка. Формальное исчисление допускает высказывания относительно переменных, фиксированных функций и предикатов, что в свою очередь расширяет логику высказываний. Simplify никогда не образует бесконечный цикл в процессе доказательства и считает, что предикаты, доказательство истинности которых невозможно, являются ложными [13].

Сформулируем тройку Хоара:

$$Q: \left(\bigwedge_{l=1}^L \bigvee_{t=1}^{T_l} k_l \in A_{lt} \right) \wedge \bigvee_{m=1}^M \text{result} = C_m; \quad S: \langle A, \bar{C} \rangle;$$

$$R: \bigvee_{p=1}^P \left(\left(\bigwedge_{l=1}^L k_l \in a_{pl} \right) \wedge \text{result} = c_p \right).$$

Далее, для доказательства правильности сформулируем предикаты с использованием директив языка Simplify и проведём автоматическое доказательство.

Аксиома Simplify для первого конъюнкта предусловия Q , будет иметь вид:

$$\left(BG_PUSH \left(\left(FORALL (k_i \ ai1 \ ai2 \ ai3 \ ... \ ait), \right) \right. \right. \\ \left. \left. \left(OR (EQ \ k_i \ ai1) (EQ \ k_i \ ai2) \right) \right) \right. \\ \left. \left. \left((EQ \ k_i \ ai3) \dots (EQ \ k_i \ ait) \right) \right) \right).$$

Аксиому для второго конъюнкта предусловия Q представим в виде:

$$BG_PUSH \left(\left(\begin{array}{l} \text{FORALL}(\text{result } c1 \ c2 \ c3 \ \dots \ cm), \\ \text{OR} \ (EQ \ \text{result } c1) \\ (EQ \ \text{result } c2) \\ (EQ \ \text{result } c3) \dots (EQ \ \text{result } cm) \end{array} \right) \right).$$

После чего необходимо перевести Q и R на язык Simplify.

С использованием введённых аксиом и предикатов Q, S, R – организовано автоматическое доказательство правильности базы знаний.

5. 4. Выбор метрик для оценки качества базы правил

Поскольку для решения задачи используется экспертная система с нечёткой базой правил, полученный результат классификации является приближенными, поэтому необходимо иметь возможность оценить качество полученного результата. Для оценки качества результата классификации, проведенного с использованием нечеткой базы правил, в работе используются следующие метрики:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}};$$

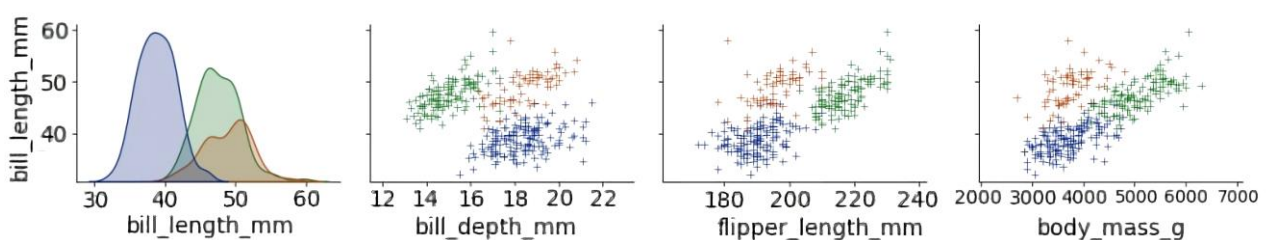
$$\text{precision} = \frac{TP}{TP+FP}; \text{recall} = \frac{TP}{TP+FN};$$

$$f1\text{-score} = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}},$$

где TP – True Positive; FP – False Positive; FN – False Negative; TN – True Negative.

5. 5. Проверка корректности базы правил для решения задачи классификации

Предложенный подход был применён для задачи классификации видовой популяции арктических пингвинов [14]. Набор данных содержит данные о 344 особях трех видов Chinstrap, Adelie, Gentoo. Признаки особей: `bill_length_mm` – длина клюва; `bill_depth_mm` – глубина клюва; `flipper_length_mm` – длина плавника; `body_mass_g` – масса тела. Для проведения классификации рис. 3, 4 были сформированы обучающая и тестовая выборки в размере 300/44 (обучающая/тестовая). На рис. 1 представлен результат программы для расчёта диаграммы предельного распределения.



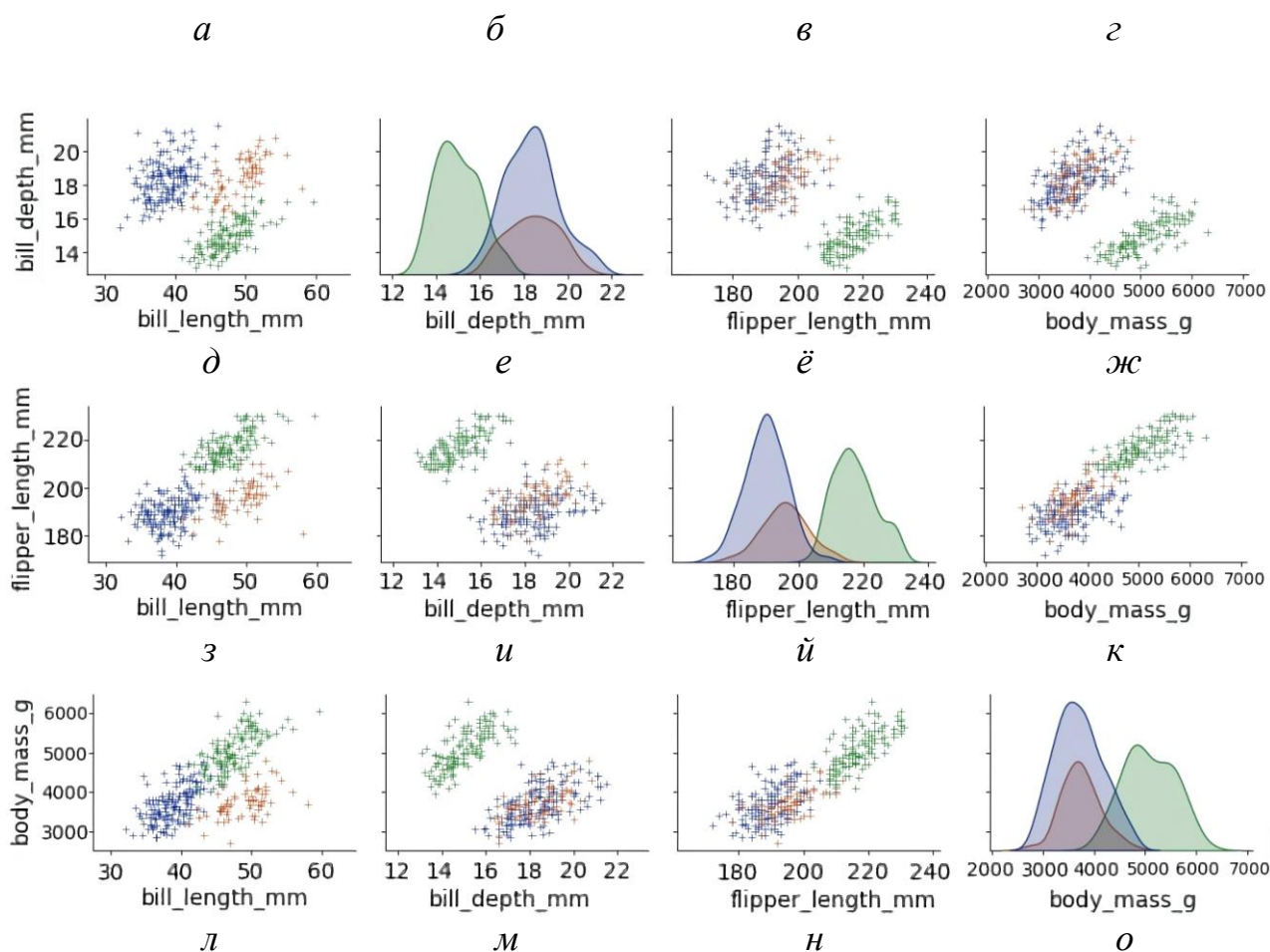


Рис. 1. Диаграмма предельного распределения классов: ● – Chinstrap, ● – Adelie, ● – Gentoo

На рис. 2 приведен результат проверки базы знаний на полноту при помощи предиката слабейшего постусловия и программы Simplify.

```

...
(OR
(EQ k_L a_L1) (EQ k_L a_L2) (EQ k_L a_L3)
(EQ k_L a_L4) (EQ k_L a_L5) (EQ k_L a_L6)
(EQ k_L a_L7) (EQ k_L a_L8) (EQ k_L a_L9)
...
(EQ k_L a_Lt)
)
(OR
(EQ result c1)
(EQ result c2)
(EQ result c3)
...
(EQ result cm)
)
;R_5
(OR
(AND (EQ k_1 a_11)(EQ k_2 a_21)(EQ k_3 a_31)...(EQ k_L a_Lt) (EQ result c1))
(AND (EQ k_1 a_12)(EQ k_2 a_21)(EQ k_3 a_31)...(EQ k_L a_Lt) (EQ result c1))
(AND (EQ k_1 a_11)(EQ k_2 a_22)(EQ k_3 a_31)...(EQ k_L a_Lt) (EQ result c1))
...
(AND (EQ k_1 a_1t)(EQ k_2 a_2t)(EQ k_3 a_3t)...(EQ k_L a_Lt) (EQ result cm))
)
)
1: Valid.

```

Рис. 2. Результат проверки полноты, «Valid» – истина

На рис. 3, 4 показаны результаты классификации разработанной программы до и после обучения нечеткой базы знаний.

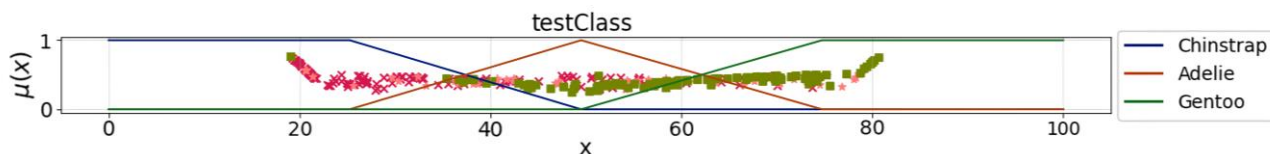


Рис. 3. До обучения

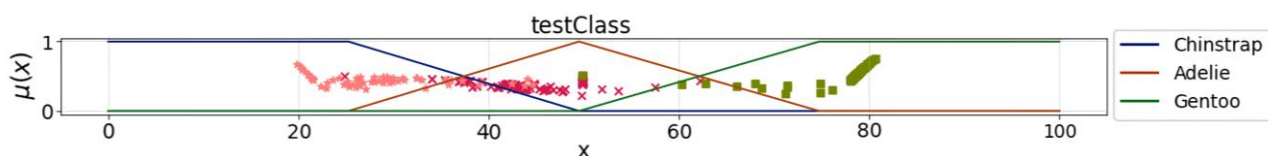


Рис. 4. После обучения

Рассмотрена зависимость результата классификации, от размера обучающей выборки до/после обучения. Для этого были сформированы обучающая и тестовая выборки в размере 200/144 (обучающая/тестовая). На рис. 5, а представлена матрица противоречий до обучения, на рис. 5, б после обучения системы. В табл. 1, 2 приведены значения метрик качества соответственно до и после обучения. Матрицы противоречий рис. 5, 6 построены разработанной программой на языке Python.

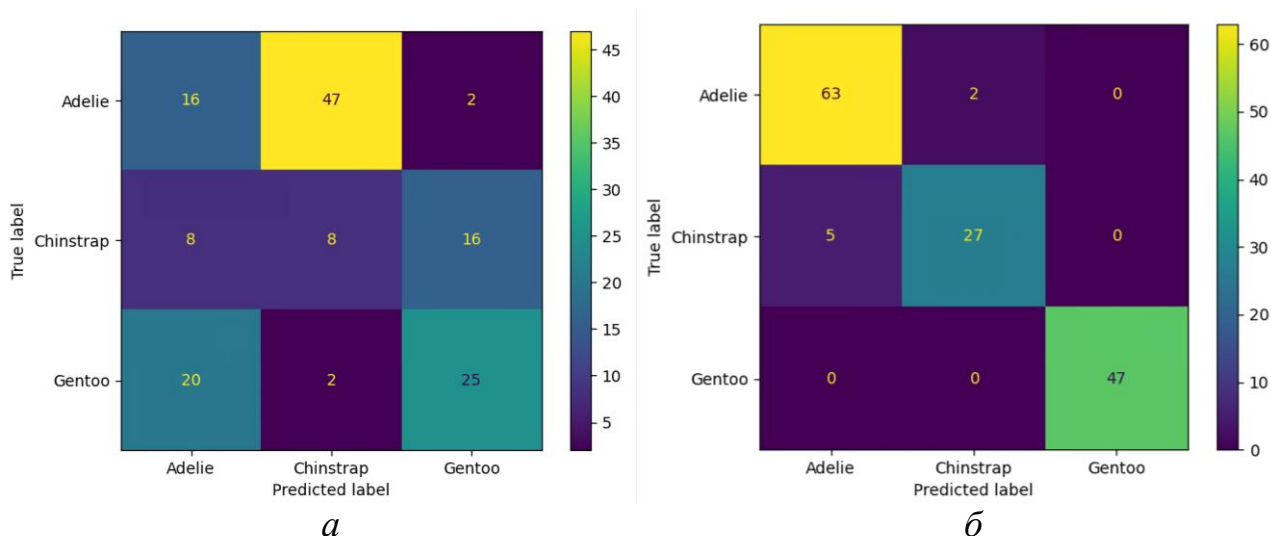


Рис. 5. Матрицы противоречий – 200/144 (обучающая/тестовая): а – до обучения; б – после обучения

Таблица 1

До обучения: 200/144 (обучающая/тестовая)

Вид	precision	recall	f1-score	support
Adelie	0.36	0.25	0.29	65
Chinstrap	0.14	0.25	0.18	32
Gentoo	0.58	0.53	0.56	47

avg accuracy	0.34
--------------	------

Таблица 2

После обучения: 200/144 (обучающая / тестовая)

Вид	precision	recall	f1-score	support
Adelie	0.93	0.97	0.95	65
Chinstrap	0.93	0.84	0.89	32
Gentoo	1.00	1.00	1.00	47
avg accuracy	0.90			

Для анализа влияния процесса обучения на результат классификации были изменены размеры выборок 250/94 (обучающая/тестовая). На рис. 6, *а* представлена матрица противоречий до обучения, на рис. 6, *б* после обучения системы. В табл. 3, 4 приведены значения метрик качества соответственно до и после обучения.

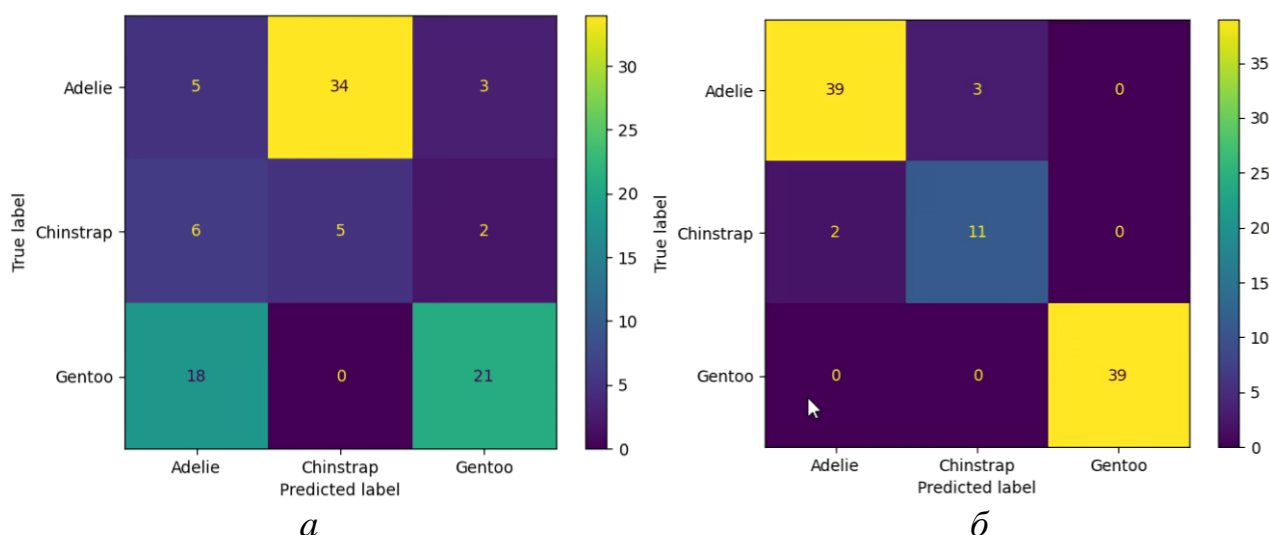


Рис. 6. Матрицы противоречий – 250/94 (обучающая/тестовая): *а* – до обучения; *б* – после обучения

Таблица 3

До обучения: 250/94 (обучающая / тестовая)

Вид	precision	recall	f1-score	support
Adelie	0.17	0.12	0.14	42
Chinstrap	0.13	0.38	0.19	13
Gentoo	0.81	0.54	0.65	39
avg accuracy	0.33			

Таблица 4

После обучения: 250/94 (обучающая / тестовая)

Вид	precision	recall	f1-score	support
Adelie	0.95	0.93	0.94	42

Chinstrap	0.79	0.85	0.81	13
Gentoo	1.00	1.00	1.00	39
avg accuracy	0.93			

Рассмотрена зависимость результата классификации, от размера обучающей выборки до/после обучения. Использовано выборки 200/144 и 250/94 (обучающая/тестовая).

6. Обсуждение результатов классификации до и после обучения в зависимости от размеров обучающей выборки

Анализ диаграммы предельного распределения (рис. 1) показал, что виды Adelie и Chinstrap достаточно связаны и их классификация может быть затруднительной. Особенно это заметно для признаков `body_mass_g` и `bill_depth_mm` – рис.1, *ж, м*.

На рис. 2 приведен результат проверки базы знаний на полноту при помощи предиката слабейшего постусловия. Результат «Valid» показывает истинность предиката $Q \Rightarrow WP(S, R)$, а, следовательно, полноту базы знаний.

Применяя полученную базу знаний до обучения на рис. 3 можно увидеть, что классы не определены и значения в результатах перемешаны. Отсутствует группирование объектов по классам. Но, после обучения на рис. 4 можно увидеть, что данные сгруппировались и каждый класс находится в границах своей функции принадлежности. Следует отметить, что на графике присутствуют одновременно все 344 объекта.

Далее было рассмотрено влияние размера обучающей выборки на результат классификации и качество экспертной системы. Для этого была сформирована обучающая и тестовая выборки в размере 200/144 (обучающая/тестовая). До обучения на рис. 5, *а* представлена матрица противоречий и анализ метрик в табл. 1. По результатам можно заметить, что точность достаточно низкая 34 %. После обучения по рис. 5, *б* и анализу метрик в табл. 2 можно заметить, что точность системы возросла до 90 %, а количество неверных ответов в матрице противоречий значительно уменьшилось.

Далее была сформирована обучающая и тестовая выборки в размере 250/94 (обучающая/тестовая). По рис. 6, *а* и табл. 3 также можно заметить низкую точность 33 %. Но, после обучения рис. 6, *б* точность возросла до 93 % как и остальные показатели в табл. 4.

Из анализа табл. 1–4 видно, что увеличение объема обучающей выборки, позволяет получить приемлемую точность классификации для выборок, а также настроить вектор консеквентов для решения задачи классификации.

Полученные результаты отличаются от работ [2–7] тем, что разработан метод автоматического построения системы правил с использованием декартового произведения и матричного представления антецедентов и консеквентов правил, с последующим автоматическим доказательством корректности системы правил на основании метода резолюций с применением логики Хоара и предиката слабого предусловия $Q \Rightarrow WP(S, R)$.

Разработанный подход позволит находить скрытые зависимости между

набором входных параметров и выходных данных модели, в автоматическом режиме создавать базу правил экспертной системы и доказывать ее корректность. Данный подход можно применять для решения практических задач мультиклассовой классификации, управления, принятия решений, кластеризации, глубинного анализа данных.

К ограничениям применения данного метода следует отнести то, что в случае классификации объектов, признаки которых сильно связаны, не достигается необходимая точность классификации.

Недостатком данного подхода является значительный объем базы правил. Для устранения указанной проблемы рекомендуется применять методы редукции базы знаний, что позволит применять предлагаемый подход для задач классификации объектов с большим числом признаков.

7. Выводы

1. В работе была построена продукционная модель представления знаний, объединяющая модели Мамдани и Такаги-Сугено-Канг и использующая функцию в качестве весового коэффициента для консеквента правила. Это позволило определить степень влияния каждого правила на результат вывода продукционной системы.

2. Отличительной особенностью построенной модели является автоматическое формирование базы правил на основе данных модели предметной области. Матрица antecedentes создается при помощи признаков объектов из некоторого конечного множества с использованием декартового произведения. Для формирования вектора консеквентов используется обучающая выборка. Благодаря этому обеспечивается возможность создавать базу правил и осуществлять настройку модели в автоматическом режиме.

3. Разработан подход для проверки корректности продукционных правил на основе метода резолюций с применением логики Хоара и предиката слабого предусловия $Q \Rightarrow WP(S, R)$. Применение такого подхода позволяет автоматизировать проверку корректности базы правил на этапе обучения.

4. Для оценки качества решения задачи классификации использованы метрики accuracy, precision, recall, f1-score. Высокие показатели оценок качества получены для обученной системы (accuracy=93 %, precision=0.91, recall=0.93, f1-score=0.92), в то время как до обучения качество модели было значительно ниже (accuracy=33 %, precision=0.37, recall=0.35, f1-score=0.33). До обучения точность модели была незначительной 33 %, после обучения точность модели достигает 93 %. Установлено, что увеличение объема обучающей выборки позволяет получить приемлемую точность классификации, а также настроить вектор консеквентов для решения задачи классификации. Также после обучения увеличиваются оценки качества модели: accuracy, precision, recall, f1-score.

5. Предложенный подход применен для решения задачи классификации. Проведена проверка корректности автоматически построенной нечеткой базы правил для решения задачи классификации. Проведена оценка качества полученной нечеткой базы правил, в результате которой точность модели

достигаєт 93 %. Виполнен анализ результатів вичислительного експеримента, установлено, що розмір навчаючої виборки впливає на результат класифікації і якість експертної системи.

Конфликт интересов

Автори декларують, що не мають конфлікту інтересів відносно даного дослідження, в тому числі фінансового, особистого характеру, авторства або іншого характеру, який міг би впливати на дослідження і його результати, представлені в даній статті.

Фінансування

Дослідження проводилося без фінансової підтримки.

Доступність даних

Рукопис не має пов'язаних даних

Литература

1. Zadeh L. A., Abbasov A. M., Yager R. R., Shahbazova S. N., Reformat M. Z. (2014). Recent developments and new directions in soft computing. Eds. Studfuz 317, Springer, 3-10. doi: <https://doi.org/10.1007/978-3-319-06323-2>
2. Терновой М. Ю., Штогрин Е. С. (2015). Формальная спецификация свойств баз нечетких знаний Мамдани на основе метаграфа. Вісник ХНУ імені В. Н. Каразіна. Серія: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління, Вип. 27, 157-171.
3. Кривуля Г. Ф., Шкиль А. С., Кучеренко Д. Е. (2013). Анализ корректности продукционных правил в системах нечеткого логического вывода с использованием квантовых моделей. АСУ и приборы автоматики: всеукр. межвед. науч.-техн. сб. – Х.: Изд-во ХНУРЕ, Вип. 165. 42-53
4. Darwiche A., Marquis P. (2002). A Knowledge Compilation Map. Journal of Artificial Intelligence 17. 229-264. doi: <https://doi.org/10.1613/jair.989>
5. Sugiuraj A., Riesenhuberj M., Kosekij Y. (1993). Comprehensibility Improvement of Tabular Knowledge Bases. AAAI-93 Proceedings, ISBN 978-0-262-51071-4, 716-721
6. Sugiuraj A., Kosekij Y. (1995). Comprehensibility Improvement of Tabular Knowledge Bases. Journal of the Japanese Society for Artificial Intelligence, Volume 10, Issue 4, 628-635. doi: https://doi.org/10.11517/jjsai.10.4_628
7. Кондратенко Ю. П., Козлов А. В. (2019). Генерація баз правил нечітких систем на основі модифікованих мурашиних алгоритмів. Міжнародний наук.-техн. журнал «Проблеми керування та інформатики», № 2, 59-79.
8. Желдак Т. А., Коряшкіна Л. С. (2020). Нечіткі множини в системах управління та прийняття рішень. навч. посіб. М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка» – Дніпро : НТУ «ДП», ISBN 978-960-350-726-2, 222- 227.
9. Hoare, C. A. R. (1969) An axiomatic basis for computer programming

Communications of the ACM, 12 (10), 576–580 and 583.
doi: <https://dl.acm.org/doi/10.1145/363235.363259>

10. Gries D. (1987). The Science of Programming (Monographs in Computer Science). Springer New York Inc, ISBN-13:978-0387964805, 108-114.

11. Dijkstra E. W. (1975) Guarded commands, nondeterminacy and formal derivation of programs. Commun. ACM, 18, ISSN 0001-0782, 453–457.
doi: <https://doi.org/10.1145/360933.360975>.

12. Simplify The Extended Static Checker software. Available at: <https://www.kindsoftware.com/products/opensource/escjava2>

13. Хижа А.Л., Высокопоясний И.Г. (2017). Автоматическая проверка семантической правильности решений задач по программированию. Питання прикладної математики і математичного моделювання. Випуск 17, ISSN 2074-5893, 234-246. doi: 10.15421/321727

14. Gorman K.B., Williams T.D., Fraser W.R. (2014). Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins (genus *Pygoscelis*). PLoS ONE 9(3):e90081, 1-14. doi: <https://doi.org/10.1371/journal.pone.0090081>

References

1. Zadeh L.A., Abbasov A.M., Yager R.R., Shahbazova S.N., Reformat M.Z. (2014). Recent developments and new directions in soft computing. Eds. Studfuz 317, Springer, 3-10. doi: <https://doi.org/10.1007/978-3-319-06323-2>

2. Ternovoy M. YU., Shtohryna E. S. (2015). Formal'naya spetsyfykatsyya svoystv baz nechetkykh znanyy Mamdany na osnove metahrafa. Visnyk KHNU imeni V. N. Karazina. Seriya: Matematychno modelyuvannya. Informatsiyi tekhnolohiyi. Avtomatyzovani systemy upravlinnya, Vyp. 27, 157-171.

3. Krivulya G.F., Shkil' A.S., Kucherenko D.Ye. (2013). Analiz korrektnosti produktsionnykh pravil v sistemakh nechetkogo logicheskogo vyvoda s ispol'zovaniyem kvantovykh modeley. ASU i pribory avtomatiki: vseukr. mezhved. nauch.-tekhn. sb. – KH.: Izd-vo KHNURE, Vyp. 165. 42-53

4. Darwiche A., Marquis P. (2002). A Knowledge Compilation Map. Journal of Artificial Intelligence 17. 229-264. doi: <https://doi.org/10.1613/jair.989>

5. Sugiuraj A., Riesenhuberj M., Kosekij Y. (1993). Comprehensibility Improvement of Tabular Knowledge Bases. AAAI-93 Proceedings, ISBN 978-0-262-51071-4, 716-721

6. Sugiuraj A., Kosekij Y. (1995). Comprehensibility Improvement of Tabular Knowledge Bases. Journal of the Japanese Society for Artificial Intelligence, Volume 10, Issue 4, 628-635. doi: https://doi.org/10.11517/jjsai.10.4_628

7. Kondratenko YU.P., Kozlov A.V. (2019). Heneratsiya baz pravyl nechitkykh system na osnovi modyfikovanykh murashynykh alhorytmiv. Mizhnarodnyy nauk.-tekhn. zhurnal «Problemy keruvannya ta informatyky», № 2, 59-79.

8. Zheldak T. A., Koryashkina L. S. (2020). Nechitki mnozhyny v sistemakh upravlinnya ta pryynyattya rishen'. navch. posib. M-vo osvity i nauky Ukrainy, Nats. tekhn. un-t «Dniprov's'ka politekhnika» – Dnipro : NTU «DP», ISBN

978-960-350-726-2, 222- 227.

9. Hoare, C.A.R. (1969) An axiomatic basis for computer programming Communications of the ACM, 12 (10), 576–580 and 583.

doi: <https://dl.acm.org/doi/10.1145/363235.363259>

10. Gries D. (1987). The Science of Programming (Monographs in Computer Science). Springer New York Inc, ISBN-13:978-0387964805, 108-114.

11. Dijkstra E. W. (1975) Guarded commands, nondeterminacy and formal derivation of programs. Commun. ACM, 18, ISSN 0001-0782, 453–457.

doi: <https://doi.org/10.1145/360933.360975>.

12. Simplify The Extended Static Checker software. Available at: <https://www.kindsoftware.com/products/opensource/escjava2>

13. Khizha A.L., Vysokopoyasnyy I.G. (2017). Avtomaticheskaya proverka semanticheskoy pravil'nosti resheniy zadach po programmirovaniyu. Pitannya prikladnoï matematiki i matematichnogo modelyuvannya. Vipusk 17, ISSN 2074-5893, 234-246. doi: 10.15421/321727

14. Gorman K.B., Williams T.D., Fraser W.R. (2014). Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins (genus *Pygoscelis*). PLoS ONE 9(3):e90081, 1-14. doi: <https://doi.org/10.1371/journal.pone.0090081>

Англ.	Укр.
Natalia Guk Doctor Sciences in Physics and Mathematics, Professor, Head of the Department Department of Computer Technology Oles Honchar Dnipro National University D. Yavornitsky ave., 35, Dnipro, Ukraine, 49000 E-mail: natalyguk29@gmail.com Contact tel: (067) 565-60-71 ORCID: http://orcid.org/0000-0001-7937-1039 Number of articles in national databases – 60 Number of articles in international databases – 7 (Scopus) Scopus: https://www.scopus.com/authid/detail.uri?authorId=54791066900 h-index=1 ResearchGate: https://www.researchgate.net/profile/Natalia-Guk	Гук Наталія Анатоліївна Доктор фізико-математичних наук, професор, завідувачка кафедри Кафедра комп'ютерних технологій Дніпровський національний університет імені Олеся Гончара пр. Д. Яворницького, 35, м. Дніпро, Україна, 49000 E-mail: natalyguk29@gmail.com Контактний тел.: (067) 565-60-71 ORCID: http://orcid.org/0000-0001-7937-1039 Кількість статей в загальнодержавних базах даних – 60 Кількість статей в міжнародних базах даних – 7 (Scopus) Scopus: https://www.scopus.com/authid/detail.uri?authorId=54791066900 h-index=1 ResearchGate: https://www.researchgate.net/profile/Natalia-Guk

Danylo Yehoshkin Corresponding author Postgraduate student Department of Computer Technology Oles Honchar Dnipro National University D. Yavornitsky ave., 35, Dnipro, Ukraine, 49000 E-mail: KnightDanila@i.ua Contact tel.: (050) 481-75-56 ORCID: https://orcid.org/0000-0002-0937-4733 Number of articles in national databases – 2 Number of articles in international databases – 0	Єгошкін Данило Ігорович Аспірант Кафедра комп'ютерних технологій Дніпровський національний університет імені Олеся Гончара пр. Д. Яворницького, 35, м. Дніпро, Україна, 49000 E-mail: KnightDanila@i.ua Контактний тел.: (050) 481-75-56 ORCID: https://orcid.org/0000-0002-0937-4733 Кількість статей в загальнодержавних базах даних – 2 Кількість статей в міжнародних базах даних – 0
--	---

Єгошкін Данило Ігорович
Відділення Нової пошти №1 (вулиця Наукова, 1, Дослідне, Дніпропетровська
область, 52071)
Тел: (050) 481-75-56

Кісельова Олена Михайлівна
чл.-кор. НАН України,
професор, доктор фізико-математичних наук,
декан факультету прикладної математики
Дніпровського національного університету
імені Олеся Гончара
пр. Д.Яворницького, 35, корпус 3,
м. Дніпро, Україна, 49000
E-mail: leljka6@gmail.com
Контактний тел.: (050)-580-47-98
ORCID: <http://orcid.org/0000-0003-4303-1707>

Чумаченко Дмитро Ігорович
канд. техн. наук, доцент на кафедрі 304
Національного аерокосмічного університету
ім. М. Є. Жуковського «Харківський авіаційний інститут»
вул. Чкалова, 17, к. 135
м. Харків, Україна, 61070
E-mail: dichumachenko@gmail.com
Контактний тел.: +38 (057) 788-40-06
ORCID: <https://orcid.org/0000-0003-2623-3294>