

УДК 004.891.3
DOI: 10.15827/0236-235X.136.511-523

Дата подачи статьи: 15.09.21, после доработки: 16.09.21
2021. Т. 34, № 4. С. 511–523

Демонстратор программной платформы для совместного использования алгоритмов теории свидетельств и нейронных сетей в нечетких системах

*В.К. Иванов*¹, к.т.н., доцент, *mtivk@mail.ru*

*Б.В. Палюх*¹, д.т.н., профессор, зав. кафедрой, *pboris@sttu.tver.ru*

¹ *Тверской государственный технический университет,
кафедра информационных систем, г. Тверь, 170026, Россия*

Диагностика состояния сложного многостадийного технологического процесса предполагает совместную обработку первичных данных для получения вероятностных характеристик аномальных критических событий или инцидентов в условиях неопределенности. В статье представлен исследовательский демонстратор «Статус-4» – прототип программной платформы для совместного использования методов алгоритмов теории свидетельств и нейронных сетей в нечетких диагностических системах.

Цель разработки демонстратора – создание научно-технического задела для передачи готовых к внедрению решений на следующие этапы проекта. Демонстратор дает возможность показать основные функциональные компоненты платформы, оценить уровень их системной готовности, провести исследовательские испытания платформы, выполнить в различных режимах тестирование программных реализаций выбранных и теоретически подтвержденных методов, быстро проверить работоспособность и эффективность функционирования при различных значениях параметров и их сочетаниях. С помощью демонстратора могут быть показаны варианты совместного применения методов нейронных сетей и теории свидетельств в гибридной экспертной системе для диагностики технологического процесса, получены экспериментальные подтверждения эффективности совместного применения этих методов в части уменьшения уровня неопределенности и увеличения уровня доверия к данным при принятии решений.

В статье приводятся краткие сведения о функциональных возможностях демонстратора, включая описание технологического процесса и предположений о влиянии диагностических переменных на его работоспособность, загрузку описаний инцидентов в технологическую БД, формирование гипотез о причинах инцидентов, генерацию продукционных правил, адаптацию параметров алгоритмов оценки состояния технологического процесса с помощью нейронной сети и нечеткого вывода. Рассматриваются основные параметры хранилища данных и объектной модели, приводятся сведения о программной реализации и пользовательском интерфейсе, которые иллюстрируются примерами. Выделены особенности используемых технологий, позволяющие надеяться на эффективность их совместного использования в диагностических системах.

Использование демонстратора способствует минимизации ключевых рисков создания полнофункциональной программной платформы для диагностики и оценки состояния сложного многостадийного технологического процесса.

Ключевые слова: *демонстратор, диагностика, инцидент, неисправность, нейронная сеть, нечеткая система, теория свидетельств, технологическая цепь, технологический процесс.*

Авторы данной работы исследуют алгоритмы совместной обработки первичных данных о состоянии сложного многостадийного технологического процесса (ТП) для получения вероятностных характеристик аномальных критических событий или инцидентов, которые потенциально могут привести к отдельным сбоям или даже к аварийным ситуациям. Преследуемая цель – разработка соответствующей технологии обнаружения и прогнозирования инцидентов, включающей ПО и технологическую БД.

Методологическая основа для разработки такой технологии должна учитывать объективные условия неопределенности при получении и анализе данных от сенсоров технологического оборудования, из технических регламентов, а также от специалистов-экспертов. Эта неопределенность порождает проблему доверия к данным, используемым для принятия решений, которые могут иметь критический характер для непрерывных производств.

Одним из общих подходов для снижения неопределенности в используемых данных

и повышения степени доверия к ним может быть гибридизация методов интеллектуальной обработки нечеткой информации. При этом ожидается более высокая синергетическая эффективность по сравнению с автономным использованием отдельных методов. Существенным обстоятельством является непростая формализация задачи снижения неопределенности, связанная с пониманием адекватности используемых методов. Должна приниматься во внимание и трудоемкость требуемых вычислений. Учитывая эти факторы, а также необходимость сложной системной интеграции используемых методов, актуальным решением представляется разработка демонстратора технологии.

В общем случае под демонстратором технологий понимается модель для оценки технической или производственной осуществимости, возможности и полезности применения конкретной технологии, процесса, концепции разрабатываемого изделия или конечной системы. Демонстраторы технологий широко применяются в высокотехнологичных отраслях, например, в авиастроении [1].

В настоящей статье представлен исследовательский демонстратор (рабочее название «Статус-4»), который является прототипом программной платформы для совместного использования моделей и методов *теории свидетельств* (ТС) и *нейронных сетей* (НС) в нечетких системах.

Цель проектирования и разработки демонстратора «Статус-4» – создание научно-технического задела для передачи готовых к внедрению решений на следующие этапы проекта.

Демонстратор «Статус-4» дает возможность:

- показать состав, внутренние связи, основные функциональные компоненты платформы;
- оценить уровень системной готовности функциональных компонентов платформы;
- выполнить тестирование в различных режимах программных реализаций выбранных и теоретически подтвержденных методов;
- выполнить быструю экспериментальную проверку работоспособности специального ПО при различных значениях параметров и их сочетаниях в условиях функционирования, близких к эксплуатационным;
- показать варианты совместного применения НС и алгоритмов ТС в гибридной экс-

пертной системе для диагностики состояния ТП;

- получить экспериментальные подтверждения эффективности совместного применения указанных методов в части уменьшения уровня неопределенности и увеличения уровня доверия к данным, используемым для принятия решений;

- при необходимости предоставить результаты вычислений специалистам-экспертам для их верификации и корректировки режимов вычислений;

- минимизировать ключевые (технические, временные, финансовые) риски создания полнофункциональной программной платформы для диагностики и оценки состояния сложного многостадийного ТП.

Далее в статье приводятся краткие сведения об особенностях многостадийного ТП, функциях демонстратора, включая обработку описания ТП и предположений о влиянии *диагностических переменных* (ДП) на его работоспособность, загрузку описаний инцидентов в технологическую БД, формирование гипотез о причинах инцидентов, генерацию производционных правил, адаптацию параметров алгоритмов оценки состояния ТП с помощью НС.

Особенности многостадийного ТП

Многостадийные ТП (возможно, циклические) состоят из ряда последовательных производственных стадий, выполнение которых и, следовательно, достижение производственных целей обеспечиваются взаимосвязанными и взаимодействующими совокупностями оборудования – *технологическими цепями* (ТЦ).

На рисунке 1 показана схема многостадийного ТП, состоящего из N стадий или числа ТЦ. Основные параметры диагностики и оценки состояния ТП: x – вектор ДП, характеризующих состояние ТП либо на выходе из i -й ТЦ или отдельной единицы оборудования, либо на входе в $(i+1)$ -ю ТЦ; u – вектор переменных управления i -й ТЦ; r_i – частный критерий оптимальности ТП на i -й стадии ТП или в i -й ТЦ.

Особенностями многостадийных ТП являются:

- наличие собственных целей отдельных стадий, кроме общей цели функционирования;

- большое количество элементов ТЦ и параметров, характеризующих его функционирование;
- сложность поведения элементов ТЦ ввиду большого числа взаимосвязей между ними;
- наличие внешних неконтролируемых воздействий;
- неопределенность параметров, которая объясняется неполнотой знаний о природе всех происходящих процессов, большим количеством возмущающих и управляющих воздействий в реальных системах и сложностью понимания, моделирования и оценки их взаимовлияния.

Функции демонстратора

Функции демонстратора «Статус-4» и их взаимосвязи представлены на диаграмме сценариев выполнения UML (рис. 2). Взаимосвязи функциональных компонентов при работе демонстратора показаны на рисунке 3. Выделены две группы функций, выполняемых параллельно и независимо друг от друга: подготовка метаданных и загрузка описаний инцидентов, а также анализ и адаптация параметров состояния ТП.

Обработка описания ТП. Подфункция 2.1.1 «Загрузка описания ТП из файла в формате JSON в технологическую БД» обеспечи-

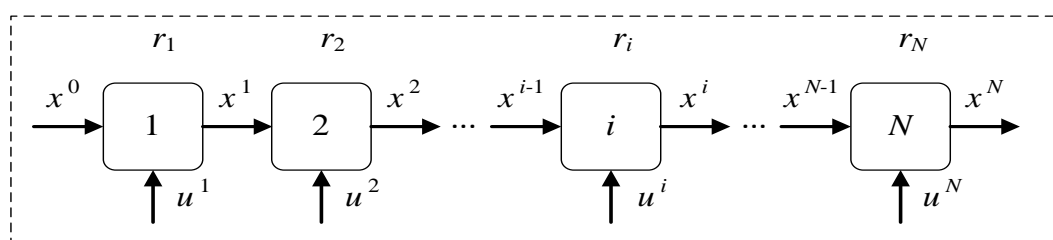


Рис. 1. Схема многостадийного ТП

Fig. 1. The multistage TP diagram

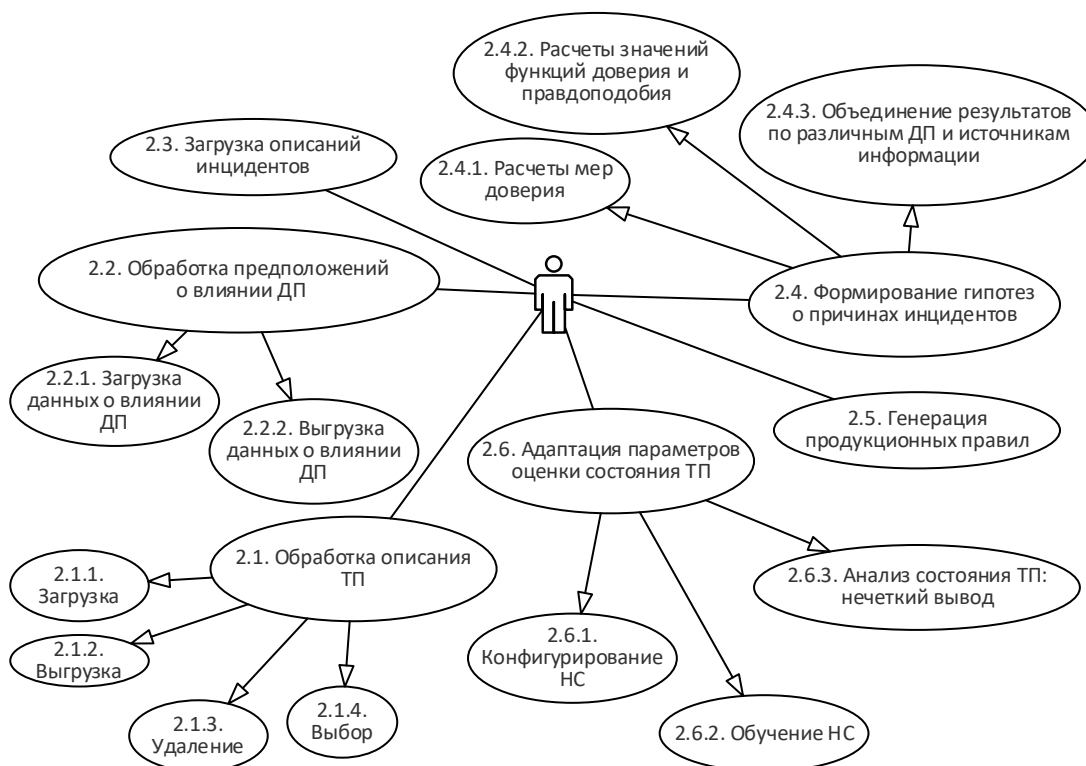


Рис. 2. Функциональность ПО демонстратора «Статус-4»

Fig. 2. The functionality of the Status-4 software demonstrator

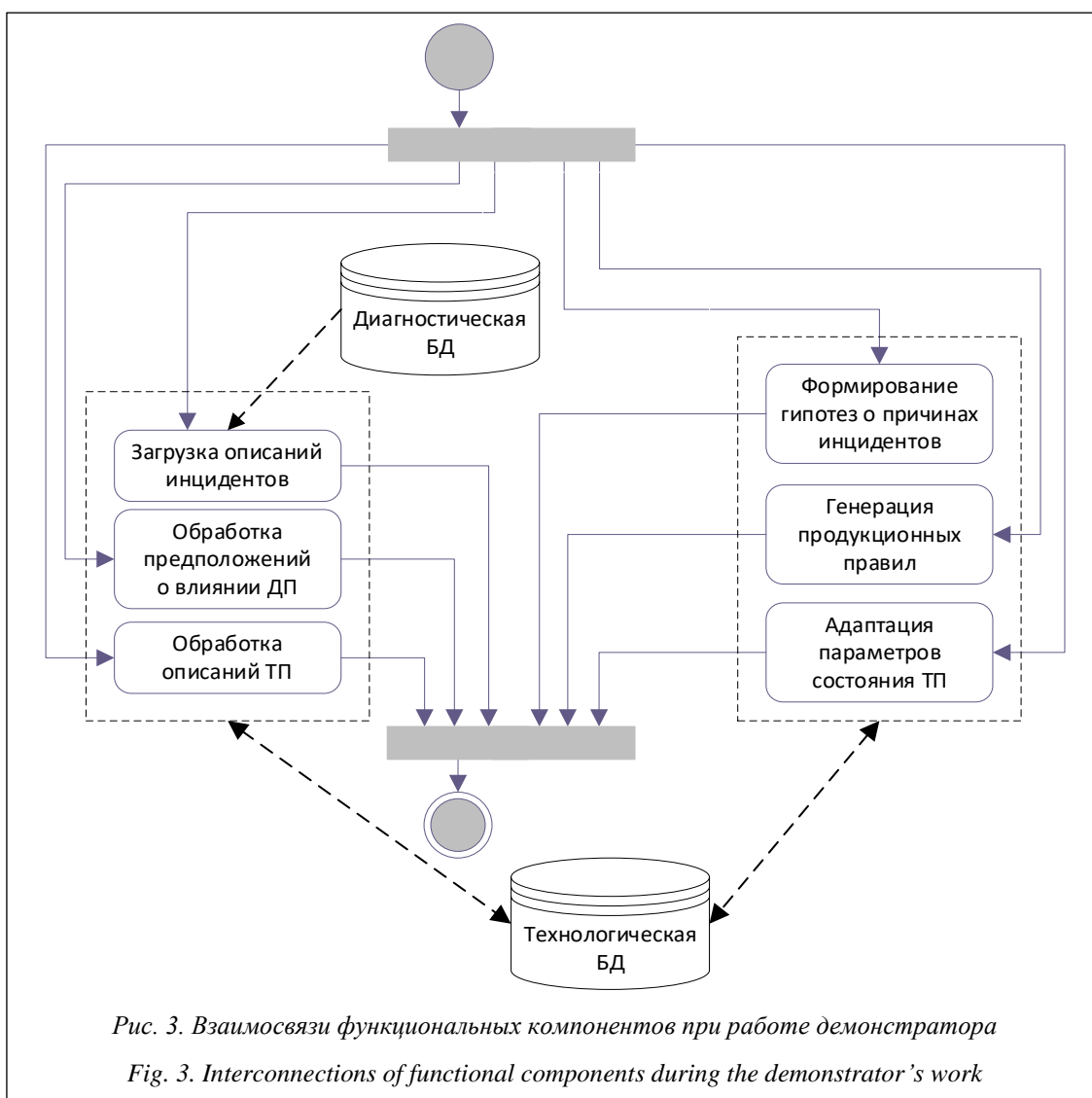


Рис. 3. Взаимосвязи функциональных компонентов при работе демонстратора

Fig. 3. Interconnections of functional components during the demonstrator's work

вает наполнение БД и легкую модификацию существующих данных.

Описание ТП содержит:

- описание ДП (идентификаторы, наименование, единица измерения, нижняя и верхняя границы нормативных значений);
- описание ТЦ оборудования (идентификаторы, наименование, состав (описание единиц оборудования));
- описание единицы оборудования (идентификаторы, наименование).

Пример описания ТП в формате JSON:

```
{
  "id": 2155, "code": "SAK", "name":
  "Производство слабой азотной кислоты",
  "diagvars": [
    {
      "id": 3641, "code": "Z1", "name":
      "Концентрация САК",
      "measure": "%", "lovalue": 50.0,
      "hivalue": 55.0, {"id": 3642, "code":
      "Z2", "name": "Расход САК", "measure":
      "мм3час, ш/час", "lovalue": null,
```

```
      "hivalue": 21.0}, ...
    ],
    "chainofunits": [
      {
        "id": 3165, "code": "c1", "name":
        "Газовая связь подачи воздуха на
        технологию и собственные нужды
        ГТТ",
        "units": [
          {
            "id": 6353, "code":
            "Ф1016", "name": ""},
            {
              "id": 6354, "code":
              "101м", "name": ""},
              {
                "id": 6355,
                "code": "M101т",
                "name": ""},
                {
                  "id": 6356,
                  "code": "M1016",
                  "name": ""},
                  {
                    "id": 6357,
                    "code": "газопровод",
                    "name": ""},
                    {
                      "id": 6358,
                      "code": "X201",
                      "name": ""}
                ]
              },
              {
                "id": 3173, "code": "c9", "name":
                "Гидравлическая связь движения
                обратной воды",
                "units": [
                  {
                    "id": 6385, "code":
                    "насос", "name": ""},
                    {
                      "id": 6386, "code":
                      "трубопровод",
                      "name": ""}
                ]
              },
              ...
            ]
          }
        ]
      }
    ]
  }
}
```

Подфункция 2.1.2 «Выгрузка описания ТП из технологической БД в файл в формате JSON» реализует процедуру, обратную выполняемой подфункцией 2.1.1.

Подфункции 2.1.3 «Удаление описания ТП из технологической БД» и 2.1.4 «Выбор и загрузка описания ТП из технологической БД для дальнейшей работы» реализуют стандартные операции доступа к технологической БД.

Обработка предположений о влиянии ДП на работоспособность ТП. Данная функция реализована исходя из вероятного наличия нескольких источников предположений о влиянии ДП на работоспособность ТП. Такими источниками считаются:

- экспертные оценки;
- статистические данные о функционировании ТП;
- технические регламенты для ТП.

Процедура экспертной оценки заключается в следующем. Эксперту предъявляется технологическая ситуация, соответствующая нарушению по одной ДП, а эксперт оценивает числом в диапазоне $[0, 100]$ степень подозрения о наличии источника отказа в каждой из диагностируемых ТЦ.

Статистические данные о выполнении ТП должны быть представлены эмпирическим распределением отказов в ТЦ, вызванных отклонением ДП от нормативных значений. Для получения базового распределения вероятностей отказов и последующей его загрузки в БД в демонстраторе использованы следующие подходы: неточная модель Дирихле [2], максимизация уверенности решением задачи линейного программирования, включая частные случаи аналитического решения [3, 4], байесовское распределение.

Анализ технических регламентов может позволить получить дополнительную информацию о выполнении установленных нормативно-технической документацией требований по обнаружению дефекта или неисправности, измерению параметров дефекта и оценке влияния дефектов на состояние объекта.

Подфункция 2.2.1 «Загрузка данных о степени влияния ДП на работоспособность ТП из файла JSON в технологическую БД» обеспечивает (как и подфункция 2.1.1) наполнение БД и легкую модификацию существующих данных. Данные о степени влияния ДП на работоспособность ТП включают (по каждому источнику информации):

- идентификатор ДП;
- идентификатор ТЦ оборудования;
- оценку степени уверенности (%).

Таким образом, каждой i -й ДП ставится в соответствие нечеткое множество ТЦ C_i , в которых потенциально мог произойти сбой:

$$\forall P(Y_i) \in P: P(y_i) = 1 \rightarrow \{(c_j; \mu_i(c_j))\}, C_i \neq \emptyset, \quad (1)$$

где $Y = \{y_i\}$ – множество ДП; $N = |Y|$; $P = \{P(y_i)\}$ – множество признаков того, что инцидент произошел (значение логической переменной $P(y_i) = 1$) или не произошел ($P(y_i) = 0$); $C = \{c_1, \dots, c_m\}$ – множество диагностируемых ТЦ оборудования; $C_i \subseteq C$ – множество ТЦ, неисправности в которых могли бы вызвать нарушение ограничений по i -й ДП в случае инцидента; $\mu_i(c_j)$ – степень уверенности в принадлежности элемента $c_j \in [0, 1]$ множеству C_i .

Пример файла в формате JSON с данными для загрузки в БД:

```
{ "processcode": "SAK", "comment": "Степень влияния значений диагностических переменных на работоспособность технологических цепей.", "srcdata": {
  "experts": [
    { "diagvarcode": "Z1", "chainofunitscode": "c1", "evaluation": 10 },
    { "diagvarcode": "Z1", "chainofunitscode": "c2", "evaluation": 30 },
    { "diagvarcode": "Z1", "chainofunitscode": "c8", "evaluation": 40 },
    { "diagvarcode": "Z2", "chainofunitscode": "c8", "evaluation": 40 },
    { "diagvarcode": "Z2", "chainofunitscode": "c9", "evaluation": 20 },
    { "diagvarcode": "Z3", "chainofunitscode": "c1", "evaluation": 40 },
    { "diagvarcode": "Z3", "chainofunitscode": "c2", "evaluation": 40 }, ...
  ],
  "documentation": [
    { "diagvarcode": "Z2", "chainofunitscode": "c8", "evaluation": 50 },
    { "diagvarcode": "Z2", "chainofunitscode": "c9", "evaluation": 30 }, ...
  ],
  "statistics": [
    { "diagvarcode": "Z2", "chainofunitscode": "c8", "evaluation": 50 },
    { "diagvarcode": "Z2", "chainofunitscode": "c9", "evaluation": 30 }, ...
  ]
}
```

Подфункция 2.2.2 «Выгрузка данных о степени влияния ДП на работоспособность

ТП из технологической БД в файл JSON» реализует процедуру, обратную выполняемой подфункцией 2.2.1.

Загрузка описаний инцидентов в технологическую БД. Данная функция включает выполнение следующих шагов:

- локализация в диагностической БД ненормативных значений ДП с подозрением на инцидент (неисправность);

- вычисление степени уверенности в том, что ненормативное значение ДП есть инцидент (для минимизации случайных погрешностей и получения описания гарантированного инцидента);

- загрузка описаний выявленных инцидентов в технологическую БД для формирования гипотез о причинах инцидентов.

Таким образом, формируется нечеткое множество инцидентов $A = \{x_i\}$, элементами которого являются ненормативные значения i -й ДП с функцией принадлежности $\mu_A(x) \in [0, 1]$, называемой также индикаторной функцией. При этом значения ДП задаются интервалом $x_i = [\underline{x}_i, \bar{x}_i]$ и для каждой ДП задан диапазон нормальной работоспособности в виде интервала $d_i = [\underline{d}_i, \bar{d}_i]$.

В демонстраторе используются следующие варианты функции принадлежности $\mu_A(x)$, реализованные в [5]: сигмоидная, двойная сигмоидная, произведение сигмоидных, Гаусса, двойная Гаусса, обобщенная колоколообразная, s -образная, π -образная, z -образная, кусочно-линейная (γ -вида), треугольная (t -вида), линейная трапецевидная.

Особенности различных функций принадлежности в нечетких системах подробно описаны в [6].

Формирование гипотез о причинах инцидентов. Данная функция демонстратора реализована с использованием алгоритмов ТС или теории Демпстера–Шафера [7–9], которая является общей основой для рассуждений с неопределенностью и позволяет, объединив свидетельства из разных источников, прийти к более определенной степени уверенности в наличии того или иного события.

Предположим, что имеются множество объектов и множество некоторых возможных свойств этих объектов. В рассматриваемом случае объектами являются ТЦ оборудования (или отдельные единицы оборудования), которые могут быть причинами зафиксированных инцидентов в результате

какой-либо неисправности. Наличие неисправности в ТЦ считается *событием*. К отдельным событиям или к любым подмножествам событий (включая их полное множество) могут быть отнесены некоторые меры доверия к ним или *базовые вероятности* этих событий. Базовые вероятности событий рассчитываются с помощью функции масс.

Укрупненная блок-схема алгоритма вычислений представлена на рисунке 4. Некоторые пояснения даны далее.

1. Расчеты значений мер доверия (подфункция 2.4.1).

Расчеты базируются на результатах выполнения функции «Обработка предположений о влиянии ДП на работоспособность ТП». Пусть $C = \{c_1, \dots, c_m\}$ – конечное множество диагностируемых ТЦ оборудования; $A \subseteq C$ – свидетельство, которое в данном случае является подмножеством потенциально неисправных ТЦ оборудования; N_A – число A ; $S(C)$ – множество всех подмножеств C ; N – число предположений, наблюдений или измерений элемента $c \in C$, которое было получено при оценке неисправностей ТЦ в случае инцидента.

Частотная функция m , называемая базовой вероятностью, определяется как

$$m: S(C) \rightarrow [0, 1], \quad \sum_{i=1}^{N_A} m(A_i) = 1, \quad A_i \in S(C). \quad (2)$$

Множество значений m образует распределение базовых вероятностей $m(A)$. Если $m(A) > 0$, то A называется фокальным элементом, а пара $(A, m(A))$ телом свидетельства.

Если $m(\emptyset) = 0$, распределение $m(A)$ является нормализованным. Значение $m(\emptyset) > 0$ для ненормализованных распределений интерпретируется как мера доверия к ситуации $c \notin C$, то есть причина неисправности не в рассматриваемых ТЦ.

2. Расчеты значений функций доверия и правдоподобия (подфункция 2.4.2).

Степень уверенности в наличии заданного свойства у объектов (в нашем случае – гипотезы о наличии неисправности в ТЦ) определяется *функцией доверия* $Bel(A)$, которая выражает поддержку свидетельствами только данной гипотезы. Степень правдоподобия наличия заданного свойства у объектов определяется *функцией правдоподобия* $Pl(A)$, которая выражает поддержку всеми свидетельствами данной гипотезы, включая те, которые

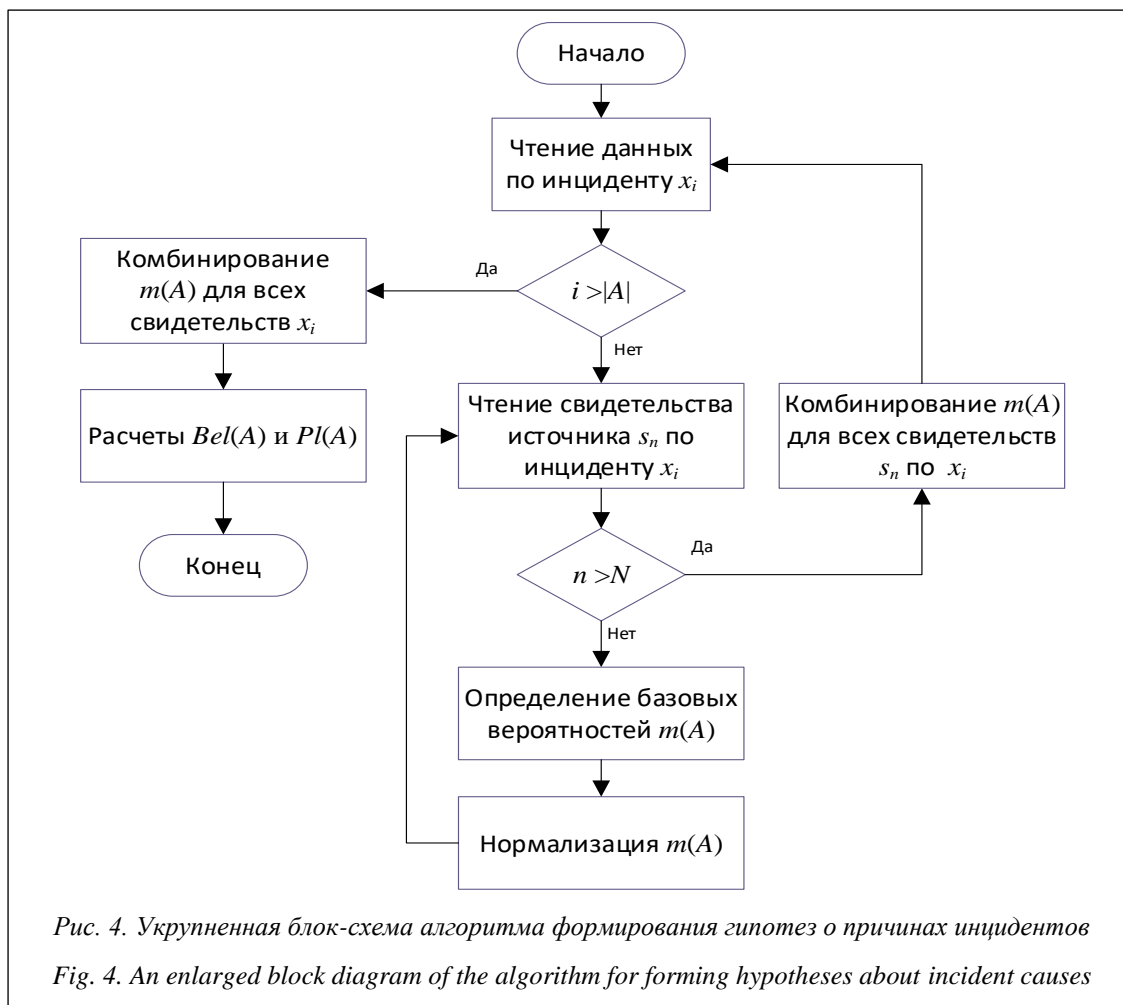


Рис. 4. Укрупненная блок-схема алгоритма формирования гипотез о причинах инцидентов
 Fig. 4. An enlarged block diagram of the algorithm for forming hypotheses about incident causes

поддерживают и другие гипотезы. Таким образом, предполагая существование некоторой истинной вероятности $p(A)$ наличия заданного свойства объектов, имеем:

$$Bel(A) \leq p(A) \leq Pl(A). \quad (3)$$

3. Объединение результатов вычислений (подфункция 2.4.3).

Подфункция реализована с использованием следующих правил комбинирования тел свидетельств о неисправностях, полученных от различных источников и по различным ДП:

- ненормализованное конъюнктивное правило комбинирования Демпстера;
- нормализованное конъюнктивное правило комбинирования Демпстера;
- дизъюнктивное правило комбинирования.

Все правила ассоциативны, поэтому они позволяют итеративно объединять любое число свидетельств.

Генерация продукционных правил. Данная функция заключается в создании базы не-

четких продукционных правил (далее правил) из объединенных данных об инцидентах, возникающих при выполнении ТП.

Каждое правило R_i имеет следующий вид:

$$R_i: \text{ЕСЛИ } x \text{ есть } A_i \text{ ТО } c \text{ есть } B_i, \quad (4)$$

где $i = 1, 2, \dots, G$ – номер правила; G – число правил в базе; $x = \{x_1, \dots, x_m\}$ – вектор значений входных ДП, $x \subseteq X$, X – область определения антецедента правила, $X = \mathbb{R}^m$; A_i – нечеткое множество ненормативных значений ДП с функцией принадлежности $\mu_A(x) \in [0, 1]$; B_i – нечеткое множество неисправных ТЦ, определенное на C функцией принадлежности $\mu_B(c) \in [Bel(c), Pl(c)]$; $c = \{c_1, \dots, c_n\}$ – вектор значений выходных лингвистических переменных, определяющих неисправные ТЦ, $c \subseteq C$, C – область определения консеквента правила (все потенциально неисправные ТЦ); $Bel(c)$ и $Pl(c)$ – функции доверия и правдоподобия соответственно, показывающие минимальную и максимальную вероятность неисправности c в ТЦ.

Пример правила:

ЕСЛИ

Показатель Концентрация САК (Z1) имеет ненормативное значение с вероятностью 0.755
И

Показатель Давление конверсии (X5) имеет ненормативное значение с вероятностью 0.5

ТО

Неисправность в технологической цепи Газовая связь подачи воздуха на технологию и собственные нужды ГТТ (с1) с вероятностью 0.453
ИЛИ

Неисправность в технологической цепи Газовая связь подачи аммиака в смеситель (с2) с вероятностью 0.547

Адаптация параметров оценки состояния ТП. Реализация подфункции 2.6.1 «Конфигурирование нейронной сети» основана на использовании архитектуры адаптивной системы нейро-нечеткого вывода ANFIS (adaptive network-based fuzzy inference system). Возможны три модификации в зависимости от применения алгоритмов нечеткого вывода: Такаги–Сугено (ANFIS), Такаги–Сугено–Канга (сеть TSK) и Ванга–Менделя (частный случай сети TSK) [10, 11].

В общем случае в НС выделены пять слоев.

Слой 1. Выполняется раздельная фазификация каждой ДП вектора $x_j \in x$ для каждого правила R_i . В общем случае x_j имеют интервальные значения. В качестве базовой функции фазификации используется обобщенная колоколообразная (из класса гауссовых) функция

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - d_1}{d_3} \right|^{2d_2}}, \quad (5)$$

где d_1 – координата точки смещения центра симметрии кривой функции по оси X; d_2 – параметр, определяющий ширину верхней части кривой; d_3 – коэффициент пологости левой и правой частей кривой. Параметры d_1 , d_2 и d_3 подлежат адаптации в процессе обучения.

Слой 2. Выполняется агрегирование x_j , то есть определяется w_i – коэффициент принадлежности вектора x к i -му правилу в соответствии с формулой

$$w_i = \mu_A^i(x) = \prod_{j=1}^N \mu_A(x_j). \quad (6)$$

Слой 3. Генерируются функции вида

$$y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij}x_j \quad (7)$$

и вычисляется значение $w_i y_i(x)$. На этом слое подлежат адаптации коэффициент веса p_{ij} для правил с номерами $i = 1, \dots, M$ и ДП с номерами $j = 1, \dots, N$.

Слой 4. Имеет два нейрона-сумматора, один из которых рассчитывает сумму $f_1 = \sum_{i=1}^M w_i y_i(x)$, а второй вычисляет сумму $f_2 = \sum_{i=1}^M w_i$.

Слой 5. Выполняет дефазификацию результата, вычисляя значение $y(x) = f_1/f_2$.

Подфункция 2.6.2 «Обучение нейронной сети» выполняет настройку параметров НС, конфигурация которой обеспечивается выполнением подфункции 6.1. Из приведенного описания конфигурации следует, что НС содержит параметрические слои 1 и 3. В процессе обучения НС настраиваются параметры d_1 , d_2 и d_3 гауссовских функций принадлежности (слой 1) и параметры p_{i0} и p_{ij} функций $y_i(x)$ (слой 3).

При наличии M правил и N входных ДП число параметров слоя 1 равно $3NM$, слоя 2 – $M(N+1)$. Общее число настраиваемых параметров в процессе обучения НС составляет $M(4N+1)$.

Для обучения НС в демонстраторе используется гибридный алгоритм [10]. Обучающий набор данных, включающий продукционные правила и массив значений ДП, генерируется автоматически в ходе выполнения данной подфункции.

Подфункция 2.6.3 «Анализ состояния ТП: нечеткий вывод» реализует нечеткий вывод НС, обученной при выполнении подфункции 2.6.2. При этом демонстрируется порядок определения ТЦ, которые являются причиной инцидентов, детектируемых ненормативными значениями ДП.

Полная реализация функции 2.6 планируется в расширенной версии демонстратора «Статус-4».

Хранилище данных и объектная модель

Модель «сущность–связь» (ER-модель) информационного обеспечения «Статус-4» представлена на рисунке 5. Исходя из этой модели, спроектирована и создана технологическая БД, которая предназначена для хранения всех необходимых исходных данных и результатов расчетов в соответствии с алгоритмами демонстратора. Используемая СУБД – MySQL.

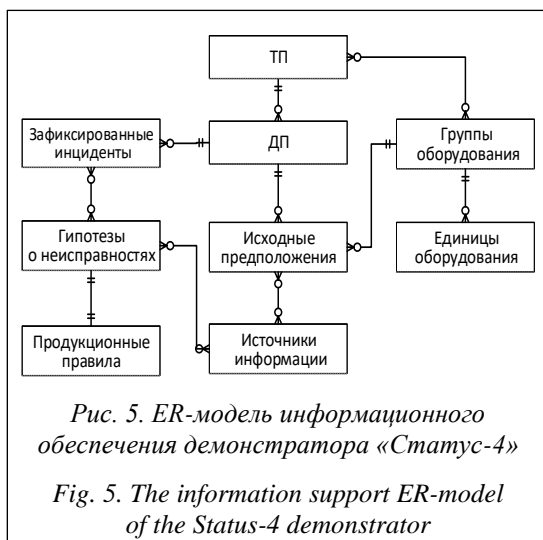


Рис. 5. ER-модель информационного обеспечения демонстратора «Статус-4»

Fig. 5. The information support ER-model of the Status-4 demonstrator

Основные классы объектной модели демонстратора «Статус-4» представлены в таблице 1, описания основных методов классов – в таблице 2. Перечень используемых библиотек сторонних разработчиков приведен в таблице 3. Отметим, что методы классов доступны пользователям как API демонстратора.

Программная реализация и пользовательский интерфейс

Демонстратор «Статус-4» реализован как web-приложение. В качестве языковой среды разработки выбран Python (версия 3.7). Выбор обоснован следующими факторами: компактный код за счет наличия предопределенных пакетов, использование готовых библиотек для реализации различных алгоритмов

машинного обучения, простой синтаксис языка, многоплатформенность, большое сообщество квалифицированных пользователей.

Также использован ряд известных библиотек, обеспечивающих расширенную функциональность (табл. 3).

Приведем фрагменты кода, иллюстрирующие применение объектов и методов демонстратора «Статус-4» для выполнения заданных функций:

```
# Открытие описания техпроцесса и выгрузка его в файл JSON
...
ps = s4process.process(log)
rc = ps.open(id)
rc = ps.unload(id)
...
# Загрузка инцидентов в технологическую БД из диагностической БД
...
ps = s4process.process(log)
rc = ps.open(id, False)
dg = s4diagnostics.diagnostics(log)
rc = dg.getdata(ps)
return dg.results
...
# Формирование гипотез о возникновении неисправностей
...
incidents = s4incidents.incidents(log, process)
rc = incidents.getselected(form)
dst = s4dst.s4dst(log)
dst.dst_calc(incidents)
...
# Генерация нечетких продукционных правил
...
incs = s4incidents.incidents(log, process)
if incs != None:
    incidents = incs
    rc, hypotheses = incs.gethypotheses()
    if len(hypotheses) > 0:
        for hypo in hypotheses:
```

Таблица 1

Основные классы объектной модели демонстратора «Статус-4»

Table 1

The main classes of the object model of the Status-4 demonstrator

Класс	Описание класса
process	Обеспечивает описание ТП и доступ к его элементам
dvcsu	Обеспечивает информацию о степени влияния значения ДП на работоспособность ТЦ
incidents	Обеспечивает хранение общей информации об инцидентах ТП и доступ к ней
incident	Обеспечивает хранение информации об инциденте (неисправности) и доступ к ней
hypothesis	Обеспечивает хранение формальных описаний гипотез о вероятных причинах неисправностей оборудования
rules	Обеспечивает генерацию и хранение продукционных правил, соответствующих гипотезам о вероятных причинах неисправностей и используемых для формирования БЗ экспертной системы
dst	Обеспечивает вычислительные процедуры в соответствии с алгоритмами ТС
sak	Обеспечивает получение информации об инцидентах (неисправностях) из диагностической БД
database	Обеспечивает операции доступа к БД
logging	Обеспечивает протоколирование работы демонстратора
options	Обеспечивает конфигурационные параметры демонстратора и доступ к ним
msg	Обеспечивает хранение и вывод сообщений демонстратора на заданном языке

```
for hypoid in hypoids.values():
    if str(hypo.id) == hypoid:
        rc = hypo.rules.generate()
        rc = hypo.rules.save()
        hypos.append(hypo)
        break
```

Экранные формы демонстратора «Статус-4», иллюстрирующие исполнение некоторых функций, представлены на рисунках (см. [http://www.swsys.ru/uploaded/image/2021-](http://www.swsys.ru/uploaded/image/2021-4/2021-4-dop/1.jpg)

[4/2021-4-dop/1.jpg](http://www.swsys.ru/uploaded/image/2021-4/2021-4-dop/1.jpg), <http://www.swsys.ru/uploaded/image/2021-4/2021-4-dop/2.jpg>, <http://www.swsys.ru/uploaded/image/2021-4/2021-4-dop/3.jpg>, <http://www.swsys.ru/uploaded/image/2021-4/2021-4-dop/4.jpg>). Свободная для использования версия демонстратора доступна по адресу https://ivkconsulting.ru/status_4.

Таблица 2

Методы основных классов объектной модели демонстратора «Статус-4»

Table 2

The main class methods of the object model of the Status-4 demonstrator

Класс	Методы класса
process	Чтение описания ТП из технологической БД, его открытие (подготовка к использованию), закрытие, удаление из технологической БД. Загрузка описания ТП из файла JSON и его сохранение в технологическую БД. Выгрузка объекта «process» из технологической БД в файл JSON
dvscu	Загрузка данных о степени влияния аномальных значений ДП на работоспособность ТП из файла в формате JSON и его сохранение в технологическую БД. Выгрузка загруженных данных из технологической БД в файл в формате JSON
incidents	Подготовка описаний отобранных инцидентов к дальнейшей обработке. Сохранение описаний отобранных инцидентов в технологическую БД. Удаление описаний отобранных инцидентов из технологической БД. Чтение описаний инцидентов и сопутствующих данных из диагностической БД и сохранение их в технологической БД. Чтение и удаление описаний гипотез для всех инцидентов из технологической БД. Сохранение описаний гипотез для всех инцидентов в технологическую БД
incident	Сохранение описаний гипотез для инцидента в технологическую БД
hypothesis	Сохранение описания гипотезы в технологическую БД
dst	Вычислительные процедуры ТС: вычисление степени уверенности в инциденте, расчеты базового и нормализованного значений мер доверия, определение нечетких множеств неисправных ТЦ различных ДП и источников свидетельств о неисправностях. Вычисление значений функций доверия и правдоподобия. Объединение результатов вычислений по ДП и источникам свидетельств и вычисление значения веса конфликта
rules	Формирование продукционных правил для гипотез о вероятных причинах неисправностей. Сохранение описания правила в технологическую БД
sak	Чтение данных из диагностической БД и определение параметров инцидентов
database	Установление соединения с БД. Модификация, удаление, добавление, чтение данных БД. Сохранение описаний ТП, инцидентов, предположений об их причинах в БД
logging	Открытие и закрытие файла для журнализации процессов обработки данных, запись в данный файл сообщений о событиях

Таблица 3

Основные используемые библиотеки сторонних разработчиков

Table 3

The main used third-party libraries

Библиотека	Описание
pyds	Обеспечивает вычислительные процедуры ТС. Описание библиотеки класса см. в [12]
scikit-fuzzy	Обеспечивает вычислительные процедуры нечеткой логики. Описание см. в [5]
keras	Обеспечивает взаимодействие с искусственными НС. Описание библиотеки см. в [13]
scipy	Обеспечивает выполнение научных и инженерных расчетов
pyodbc	Обеспечивает доступ к БД
pandas	Обеспечивает обработку и анализ данных
numpy	Обеспечивает поддержку многомерных массивов и математических функций
pickle	Обеспечивает сериализацию и десериализацию объектов Python

Заключение

По результатам разработки исследовательского демонстратора «Статус-4» можно сделать вывод, что прототип программной платформы для совместного использования моделей и методов ТС и НС в нечетких системах в целом подготовлен к выполнению экспериментальных исследований. Разработанной функциональности достаточно для решения следующих исследовательских задач:

- моделирование и формализация процесса локализации ненормативных значений ДП с минимизацией случайных погрешностей и получением описания гарантированного инцидента;

- разработка экспериментально подтвержденной методики получения распределения базовых вероятностей неисправностей в ТС с объединением свидетельств о потенциальных сбоях из нескольких источников;

- выбор архитектуры и параметров нечеткой НС, оптимальных с точки зрения со-

отношения затрат на настройку и точности результатов диагностики;

- генерация обучающих данных и продукционных правил для настройки НС в модуле нечеткого вывода диагностической экспертной системы;

- обоснование вида функций принадлежности и значений их параметров для более точного определения множества ТЦ, являющихся причинами инцидентов.

Разработанный демонстратор дает возможность получать экспериментальные подтверждения эффективности совместного применения обсуждаемых подходов, моделей и методов обнаружения и прогнозирования инцидентов при выполнении сложного многостадийного ТП. Предполагается, что решение указанных задач будет способствовать достижению поставленной цели – создание научно-технического задела для передачи готовых решений на следующие этапы проекта с одновременным снижением рисков.

Исследовательская работа выполнена при финансовой поддержке РФФИ в рамках проекта № 20-07-00199.

Литература

1. Сыпало К.И., Медведский А.Л., Бабичев О.В. и др. Создание демонстратора технологии авиастроения // Тр. МАИ. 2017. № 95. URL: http://trudymai.ru/upload/iblock/6fa/Sypalo_Medvedskiy_Babichev_Kazarinov_Kan_rus.pdf?lang=ru&issue=95 (дата обращения: 3.09.21).
2. Walley P. Inferences from multinomial data: Learning about a bag of marbles. J. of the Royal Statistical Society: Series B (Methodological), 1996, vol. 58, no. 1, pp. 3–34. DOI: 10.1111/j.2517-6161.1996.tb02065.x.
3. Denoeux T. Constructing belief functions from sample data using multinomial confidence regions. Int. J. of Approximate Reasoning, 2006, vol. 42, no. 3, pp. 228–252. DOI: 10.1016/j.ijar.2006.01.001.
4. Aregui A., Denoeux T. Constructing consonant belief functions from sample data using confidence sets of pignistic probabilities. Int. J. of Approximate Reasoning, 2008, vol. 49, no. 3, pp. 575–594. DOI: 10.1016/j.ijar.2008.06.002.
5. SciKit-Fuzzy 0.4.2. Fuzzy logic toolkit for SciPy. URL: <https://pypi.org/project/scikit-fuzzy/> (дата обращения: 01.09.2021).
6. Халов Е.А. Систематический обзор четких одномерных функций принадлежности интеллектуальных систем // Информационные технологии и вычислительные системы. 2009. № 3. С. 60–74.
7. Dempster A. A generalization of Bayesian inference. Royal Statistical Society: Series B (Methodological), 1968, vol. 30, no. 2, pp. 205–232. DOI: 10.1111/j.2517-6161.1968.tb00722.x.
8. Shafer G. A Mathematical Theory of Evidence. Princeton University Press Publ., 1976, 297 p.
9. Yager R., Liping L. Classic Works of the Dempster-Shafer Theory of Belief Functions. London, Springer Publ., 2010, 825 p.
10. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с.
11. Cai A., Quek C., Maskell D. Type-2 GA-TSK fuzzy neural network. Proc. IEEE CEC, 2007, pp. 1578–1585. DOI: 10.1109/CEC.2007.4424661.
12. GitHub. Reineking/Pyds. URL: <https://github.com/reineking/pyds> (дата обращения: 01.09.2021).
13. GitHub. Keras-team/Keras. URL: <https://github.com/keras-team/keras> (дата обращения: 01.09.2021).

A software platform demonstrator for joint use of evidence theory algorithms and neural networks in fuzzy systems

V.K. Ivanov¹, Ph.D. (Engineering), Associate Professor, mtivk@mail.ru

B.V. Palyukh¹, Dr.Sc. (Engineering), Professor, Head of Department, pboris@tstu.tver.ru

¹Tver State Technical University, Information System Department, Tver, 170026, Russian Federation

Abstract. The diagnostics of a complex multi-stage technical process involves the joint primary data processing to obtain probabilistic characteristics of abnormal critical events or incidents under uncertainty. The paper presents the research demonstrator Status-4 that is a software platform prototype for joint using the evidence theory and neural network methods in fuzzy diagnostic systems.

The purpose of the demonstrator development is to create a scientific and technical reserve for ready-to-implement solutions transfer to the next project stages. The demonstrator makes it possible to show the main platform functional components, assess their system readiness level, conduct the platform research tests, perform software implementations testing of the selected and theoretically confirmed methods in various modes, check the functioning operability and efficiency at various parameter values and their combinations quickly. The demonstrator shows the options for the joint application of neural network and evidence theory methods in a hybrid expert system for diagnostics process. In addition, these methods joint application effectiveness is experimentally confirmed in terms of reducing the uncertainty level and increasing the confidence in data level when making decisions. The demonstrator enables minimizing the key risks of creating a full-featured software platform for diagnosing and evaluating the complex multi-stage technologies state.

The paper provides brief information about the demonstrator functionality, including the technology description and the suppositions description about diagnostic variables influence on processing performance, loading incident descriptions into the technological database, forming hypotheses about the incidents causes, generating production rules, adapting the parameters of the technology state assessing algorithms using neural network and fuzzy inference. The paper considers the main data warehouse and object model parameters, provides the software implementation and user interface information and illustrates it by examples. It also highlights the used methods features, which allow us to hope for the effectiveness of their joint use in diagnostic systems.

Keywords: demonstrator, diagnostics, incident, malfunction, neural network, fuzzy system, evidence theory, technological chain, technology.

Acknowledgements. The research work was financially supported by the RFBR within the framework of project no. 20-07-00199.

References

1. Sypalo K.I., Medvedskiy A. L., Babichev O.V. et al. Engineering of aircraft demonstrator. *Trudy MAI*, 2017, no. 95. Available at: <http://trudymai.ru/published.php?ID=84545> (accessed September 03, 2021) (in Russ.).
2. Walley P. Inferences from multinomial data: Learning about a bag of marbles. *J. of the Royal Statistical Society: Series B (Methodological)*, 1996, vol. 58, no. 1, pp. 3–34. DOI: 10.1111/j.2517-6161.1996.tb02065.x.
3. Denoeux T. Constructing belief functions from sample data using multinomial confidence regions. *Int. J. of Approximate Reasoning*, 2006, vol. 42, no. 3, pp. 228–252. DOI: 10.1016/j.ijar.2006.01.001.
4. Aregui A., Denoeux T. Constructing consonant belief functions from sample data using confidence sets of pignistic probabilities. *Int. J. of Approximate Reasoning*, 2008, vol. 49, no. 3, pp. 575–594. DOI: 10.1016/j.ijar.2008.06.002.
5. *SciKit-Fuzzy 0.4.2. Fuzzy Logic Toolkit for SciPy*. Available at: <https://pypi.org/project/scikit-fuzzy/> (accessed September 01, 2021).
6. Halov E.A. Systematic review of clear one-dimensional membership functions of intelligent systems. *J. of Information Technologies and Computing Systems*, 2009, no. 3, pp. 60–74 (in Russ.).

7. Dempster A. A generalization of Bayesian inference. *Royal Statistical Society: Series B (Methodological)*, 1968, vol. 30, no. 2, pp. 205–232. DOI: 10.1111/j.2517-6161.1968.tb00722.x.
8. Shafer G. *A Mathematical Theory of Evidence*. Princeton University Press Publ., 1976, 297 p.
9. Yager R., Liping L. *Classic Works of the Dempster-Shafer Theory of Belief Functions*. London, Springer Publ., 2010, 825 p.
10. Osovskiy S. *Neural Networks for Information Processing*. Moscow, 2002, 344 p. (in Russ.).
11. Cai A., Quek C., Maskell D. Type-2 GA-TSK fuzzy neural network. *Proc. IEEE CEC*, 2007, pp. 1578–1585. DOI: 10.1109/CEC.2007.4424661.
12. *GitHub*. *Reineking/Pyds*. Available at: <https://github.com/reineking/pyds> (accessed September 01, 2021).
13. *GitHub*. *Keras-team/Keras*. Available at: <https://github.com/keras-team/keras> (accessed September 01, 2021).

Для цитирования

Иванов В.К., Палюх Б.В. Демонстратор программной платформы для совместного использования алгоритмов теории свидетельств и нейронных сетей в нечетких системах // Программные продукты и системы. 2021. Т. 34. № 4. С. 511–523. DOI: 10.15827/0236-235X.136.511-523.

For citation

Ivanov V.K., Palyukh B.V. A software platform demonstrator for joint use of evidence theory algorithms and neural networks in fuzzy systems. *Software & Systems*, 2021, vol. 34, no. 4, pp. 511–523 (in Russ.). DOI: 10.15827/0236-235X.136.511-523.