УДК 004.891.3
DOI: 10.15827/0236-235X.140.609-617
Дата подачи статьи: 20.09.22
2022. Т. 35. № 4. С. 609–617

Демонстратор программной платформы для настройки гиперпараметров нечеткой нейронной сети

В.К. Иванов [1], *к.т.н., доцент кафедры информационных систем, mtivk@mail.ru*
Б.В. Палюх [1], *д.т.н., профессор, зав. кафедрой информационных систем, pboris@tstu.tver.ru*

[1] *Тверской государственный технический университет, г. Тверь, 170026, Россия*

В статье приводится описание исследовательского демонстратора для экспериментальной проверки и оценки вариантов применения нечетких алгоритмов и нейронных сетей в экспертной системе для диагностики сложных многостадийных технологических процессов. Цель разработки демонстратора – создание научно-технического задела для передачи готовых к внедрению решений на следующие этапы проекта.

Демонстратор позволяет оценить уровень системной готовности разрабатываемых компонентов, провести исследовательские испытания, проверить работоспособность и эффективность функционирования программных реализаций при различных значениях параметров и их сочетаниях. Диагностика состояния сложного многостадийного технологического процесса предполагает совместную обработку первичных данных для получения вероятностных характеристик аномальных критических событий или инцидентов в условиях неопределенности.

Авторами предложен вариант использования нечеткой нейронной сети, обучение которой происходит данными, сгенерированными с помощью функций доверия. Подход дает возможность значительно ускорить вычисления и минимизировать ресурсную базу. В статье основное внимание уделяется описанию функций управления моделями нейронной сети и обучающими наборами данных, обучения нейронной сети и проверки его качества, диагностики технологического процесса в различных режимах. Подробно описаны настраиваемые гиперпараметры нейронной сети. Приведены примеры реализации диагностических процедур в различных режимах. Показано, что при функционировании программной диагностической системы в условиях, близких к реальным, могут быть проверены и экспериментально обоснованы исходные предположения, касающиеся сокращения времени обнаружения и прогнозирования инцидентов, и более точно определены множества технологических цепей, являющихся причинами инцидентов.

Ключевые слова: *демонстратор, диагностическая система, инцидент, многостадийный технологический процесс, нечеткая логика, нечеткая нейронная сеть, продукционное правило, теория свидетельств, технологическая цепь, функция доверия, функция принадлежности, ANFIS, TSK.*

Исследовательский демонстратор, рассмотренный в [1], является прототипом программной платформы для совместного использования моделей и методов теории свидетельств и *нейронных сетей* (НС) в нечетких системах. Цель создания демонстратора (рабочее название «Статус-4») – формирование научно-технического задела для передачи готовых к внедрению решений на следующие этапы проекта.

Демонстратор представляет собой программную модель для оценки осуществимости, возможности и полезности гибридизации методов и алгоритмов интеллектуальной обработки нечеткой информации, включая динамические экспертные системы. Одной из областей применения гибридных экспертных систем является совместная обработка первичных данных о состоянии сложного многостадийного *технологического процесса* (ТП) для получения вероятностных характеристик аномальных критических событий или инцидентов. Результатом исследований в этом направлении могла бы быть соответствующая технология обнаружения и прогнозирования инцидентов, которая учитывает объективные условия неопределенности при получении и анализе данных от сенсоров технологического оборудования, из технических регламентов, а также от специалистов-экспертов.

Демонстратор «Статус-4» позволяет показать варианты совместного применения НС и алгоритмов теории свидетельств в гибридной экспертной системе, получать экспериментальные подтверждения эффективности такого применения, представлять результаты специалистам-экспертам для их верификации и корректировки режимов и в итоге минимизировать ключевые риски создания полнофункциональ-

ной системы для диагностики и оценки состояния сложного многостадийного ТП.

В обзоре [2] подробно рассмотрены основные тренды совместного применения НС и алгоритмов теории свидетельств. Проведенный анализ позволил выделить следующие основные направления исследований по рассматриваемой тематике.

1. Использование теории свидетельств при подготовке данных для создания и настройки НС, например, в диагностических системах [3] и системах автономного управления транспортными средствами [4].

2. Использование НС при подготовке исходных данных для теории свидетельств. Эти технологии применяются, как правило, при разработке различных классификаторов [5, 6], а также в прогнозирующих системах [7].

3. Объединение результатов работы НС и других методов машинного обучения с помощью методов теории свидетельств. Применяется в системах принятия решений: мониторинг [8], медицинское диагностирование [9], диагностика оборудования [10] и др.

Подход, изложенный в настоящей статье, является вкладом авторов в развитие исследований по первому направлению из трех представленных выше.

В работах [1, 11] были определены особенности многостадийного ТП, рассмотрена последовательность операций обработки диагностических данных, представлена формальная модель нечеткой системы диагностики многостадийного ТП, показано, как индикация инцидентов осуществляется соответствующими функциями принадлежности. Описаны основные функции демонстратора, включая обработку описания ТП и предположений о влиянии *диагностических переменных* (ДП) на его работоспособность, загрузку описаний инцидентов в технологическую БД, формирование гипотез о потенциальных причинах инцидентов с использованием функций доверия, автоматическую генерацию нечетких продукционных правил, что является предпосылкой к применению нечетких НС. Также выделены особенности используемых технологий и рассмотрены основные параметры хранилища данных и объектной модели. Как итог, обосновано использование сгенерированной базы продукционных правил для обучения НС ANFIS с архитектурой TSK, которая позволит оперативно вычислить оценку вероятности неисправности в *технологической цепи* (ТЦ) без ресурсоемких вычислений.

В настоящей статье представлены новые функции демонстратора «Статус-4», которые дают возможность оценить применимость и эффективность нечеткого вывода, реализованного с помощью НС.

Нечеткая система диагностики многостадийного ТП

Опишем кратко предлагаемый подход, основанный на интерпретации системы диагностики многостадийного ТП как нечеткой системы.

Предположим, что в процессе эксплуатации непрерывного ТП формируется нечеткое множество инцидентов $X' = \{x'_{nm}, \mu_I(x_{nm})\}$, где x'_{nm} – ненормативные интервальные значения *диагностических переменных* (ДП); $\mu_I(x_{nm})$ – функция принадлежности x'_{nm} множеству X'. Все ДП являются выходными технологическими переменными для диагностируемого оборудования рассматриваемой стадии ТП, которая состоит из оборудования, соответствующего технологической схеме его функционирования по технологическому регламенту в составе конкретной ТЦ; значения ДП поступают от сенсоров оборудования. Понятие *инцидента* соответствует аномальному критическому событию в какой-либо ТЦ. Функция $\mu_I(x_{nm})$ в общем случае может быть представлена как «перевернутая» нечеткая π-образная функция принадлежности, но возможны и другие варианты [12].

Непустое множество инцидентов X' есть следствие нарушения нормативного режима работы в одной или нескольких ТЦ. В таком случае имеем нечеткое множество $A = \{(c_n, \mu_{nm}(c_n))\}$, где c_n – ТЦ; $\mu_{nm}(c_n)$ – функция принадлежности ТЦ c_n множеству потенциально неисправных ТЦ A. Для определения $\mu_{nm}(c_n)$ предлагаем использовать ключевые понятия теории свидетельств [13] – функции доверия $Bel(A)$ и правдоподобия $Pl(A)$. Эти функции рассматриваются как нижняя и верхняя границы функции принадлежности $\mu_{nm}(c_n)$. Назовем интервал значений функции $\mu_{nm}(c_n) = [Bel(A), Pl(A)]$ *гипотезой о причине инцидента*. Множество таких гипотез может быть получено для всех возможных A. Подробности предложенного подхода описаны в [11].

Гипотезы о причинах инцидентов могут быть выражены с помощью продукционных правил вида: R_i : IF $\mu_I(x_{n1})$ AND $\mu_I(x_{n2})$ AND … AND $\mu_I(x_{nm})$ … THEN $\mu_{nm}(c_n)$, где каждое i-е правило относится к определенной n-й ТЦ;

функции принадлежности $\mu_l(x_{nm})$ выражают x_m – степень уверенности в том, что ненормативное значение m-й ДП является индикатором инцидента; функция принадлежности $\mu_{nm}(c_n)$ есть степень уверенности в том, что причина инцидента есть дефект в ТЦ c_{ni}. Значение $\mu_{nm}(c_n)$ гарантированно лежит внутри [$Bel(A)$, $Pl(A)$] и является достоверной оценкой соответствующей гипотезы c_n. Таким образом, очевиден следующий вывод: используя вычисленные значения функций доверия и правдоподобия, можно автоматически сгенерировать базу продукционных правил для всех гипотез о причинах инцидентов, которая обычно формируется специалистами предметной области.

Структура вышеприведенных правил аналогична структуре правил, используемой в нечеткой НС с архитектурой ANFIS в варианте Такаги–Сугено–Канга (TSK) [14]. Вывод: решение исходной задачи определения достоверной оценки вероятности дефекта ТЦ как причины диагностированного инцидента может быть найдено нечеткой НС ANFIS/TSK, обученной с помощью автоматически сгенерированной базы продукционных правил.

Полные формальные описания модели диагностики ТП, процедур формирования нечетких множеств X' и A, преобразования продукционных правил, обоснования использования НС ANFIS/TSK приведены в [1, 11]. Демонстратор «Статус-4» реализует основные алгоритмы обсуждаемой нечеткой системы диагностики многостадийного ТП.

Основные функции демонстратора и хранилище данных

Приведем краткое описание укрупненных функций демонстратора «Статус-4».

1. *Управление описаниями ТП*. Выбор ТП для дальнейшей работы, загрузка описания ТП из файла JASON, выгрузка описания ТП из БД в файл JASON, удаление описания ТП из БД.

2. *Управление инцидентами*. Загрузка данных о влиянии ДП на формирование гипотез о дефектах, выгрузка этих данных из БД в файл JSON, загрузка описаний дефектов или инцидентов.

3. *Формирование гипотез о возникновении дефектов*. Расчеты вероятностей возникновения дефектов в ТЦ методами теории Демпстера–Шафера.

4. *Генерация продукционных правил*. Генерация базы нечетких продукционных правил из объединенных данных об инцидентах при исполнении ТП.

5. *Управление диагностическими моделями*. Создание моделей нечеткой НС ANFIS/TSK для диагностики ТП, сохранение и удаление их из БД, выбор модели из БД.

6. *Управление обучающими наборами данных*. Создание обучающих наборов данных, их сохранение и удаление из БД, выбор набора данных для обучения НС.

7. *Обучение НС*. Выбор модели НС, обучающего набора данных, алгоритма обучения и обучение НС – подбор значений в параметрических слоях.

8. *Диагностика состояния ТП*. Демонстрация функций диагностики состояния ТП с помощью нечеткой НС ANFIS/TSK.

На рисунке (см. http://www.swsys.ru/uploaded/image/2022-4/2022-4-dop/3.jpg) показан внешний вид страницы главного меню демонстратора «Статус-4».

Модель «сущность–связь» хранилища данных демонстратора «Статус-4» представлена на рисунке 1. На основе этой модели спроектирована и создана технологическая БД, предназначенная для хранения необходимых исходных данных и результатов расчетов в соответствии с алгоритмами демонстратора.

Описание основных функций

Опишем классы программных объектов демонстратора, относящиеся непосредственно к функциям управления моделями НС и обучающими наборами данных, собственно обучения и диагностики ТП. Остальные классы описаны в [1], информация о сторонних программных библиотеках приведена в [15–17].

Демонстратор реализован как web-приложение. Языковая среда разработки – Python 3.7.

Свободная для запуска версия демонстратора доступна по адресу в Интернете http://ivkconsulting.ru/status_4.

Управление диагностическими моделями. Функция обеспечивает создание моделей нечеткой НС для диагностики ТП, сохранение их в БД и удаление оттуда, выбор модели из БД. Работа с моделями НС реализуется классом models. Программные модули класса реализованы на базе фреймворка tensorflow.

Основные атрибуты класса models (параметры модели): имя, текстовое описание, число входных переменных (регрессоров), число правил, число эпох при обучении, тип функции потерь (Huber, MSE, MAE), итоговое значение

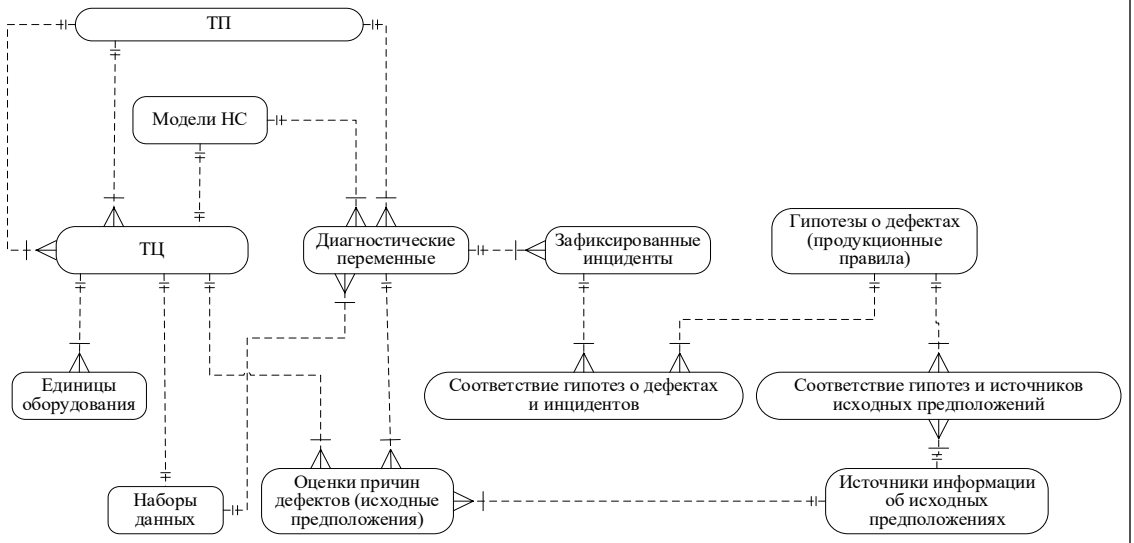

Рис. 1. Модель «сущность–связь» хранилища данных демонстратора

Fig. 1. An entity–relationship model of the demonstrator data warehouse

функции потерь после обучения, способ оптимизации (ADAM, RMSProp, метод градиентного спуска).

Основные операции класса models: создание модели, сохранение ее в файл и БД, загрузка для обработки, удаление из БД, отображение параметров, обучение, создание инфраструктуры для tensorflow, формирование списка доступных моделей.

Управление обучающими наборами данных. Функция обеспечивает создание обучающих наборов данных, сохранение их в БД и удаление оттуда, выбор набора данных для обучения НС. Работа с обучающими наборами данных реализуется классом dataset.

Основные атрибуты класса dataset: имя, текстовое описание, список ДП, размер блока значений каждой ДП для формирования их допустимых сочетаний, идентификатор ТЦ, тип комбинирования данных из разных источников (нормализованная или ненормализованная конъюнктивная комбинация, дизъюнктивная комбинация), признак генерации набора данных только для инцидентов.

Основные операции класса dataset: генерация данных (значений входных и выходной переменной), конвертирование данных в формат для обучения НС (массив NumPy), сохранение набора данных в файловой системе и его описания в БД, удаление набора данных и его описания, загрузка набора данных и его описания для обработки, формирование списка доступных наборов данных, отображение параметров.

Обучение НС. Функция выполняет настройку параметров НС. Используются созданная модель НС и сгенерированный обучающий набор.

В процессе обучения НС настраиваются параметры d_1, d_2 и d_3 гауссовской функции принадлежности $\mu_I(x_{nm}) = d_1 \exp\{-[(x_{nm} - d_2)/2d_3^2]\}$. Предполагая нормальное распределение x_{nm}, имеем в этом случае: $d_1 = 1/\sigma\sqrt{2\pi}$, $d_2 = \mu$, $d_3 = \sigma$, где σ – среднеквадратичное отклонение; μ – математическое ожидание. Таким образом фактически настраиваются σ и μ. С точки зрения архитектуры ANFIS/TSK настройка этих параметров происходит в параметрическом слое 1 [14].

В параметрическом слое 3 архитектуры ANFIS/TSK весовые коэффициенты p_{i0} и p_{ij} линейных функций $y_i(x)$, определяющих значимость каждого правила [1], не настраиваются, так как предполагается, что все правила равнозначны.

Таким образом, при наличии G правил и N входных ДП число настраиваемых параметров в процессе обучения НС равно $3GM$.

При обучении НС по заданной модели используются набор данных для обучения, сгенерированный для конкретных ТЦ и множества ДП, и набор данных для проверки качества обучения, сгенерированный для тех же ТЦ и множества ДП.

Исходные гиперпараметры для поиска вариантов их сочетаний, обеспечивающих при-

емлемое качество обучения: число продукционных правил, скорость обучения, число эпох при обучении, тип функции потерь, способ оптимизации.

Результаты обучения при работе демонстратора протоколируются, визуализируются и сохраняются в БД. Динамика изменения потерь при обучении НС (пример) иллюстрируется графиком на рисунке 2. На рисунке 3 представлен пример результата проверки качества НС, при этом использован сгенерированный обучающий набор данных. Графики на рисунке 4 представляют функции принадлежности, выполняющие фаззификацию исходных данных для каждого правила и настроенные при обучении НС. Все представленные графики автоматически формируются демонстратором.

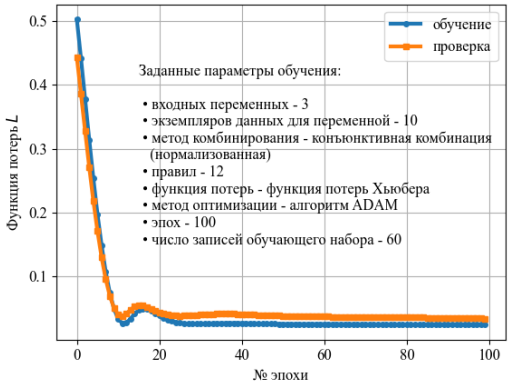

Рис. 2. Изменение функции потерь при обучении НС

Fig. 2. Changing the loss function during training of the neural network

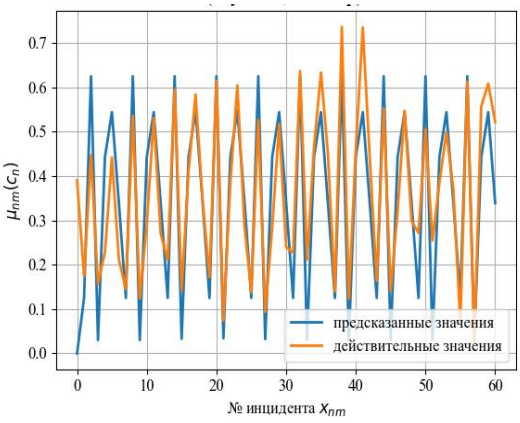

Рис. 3. Соответствие предсказанных и действительных значений $\mu_{nm}(c_n)$ (обучающий набор)

Fig. 3. A correspondence between predicted and actual values of $\mu_{nm}(c_n)$ (a training set)

Пример протокола обучения НС (фрагмент):

```
   12.09.2022 15:38:28 Модули TensorFlow подключены.
   12.09.2022 15:38:34 Обучение модели началось...
   Обучается модель m_z1_z2_z5_y (Ид: 207, Ид: 2159)
   Регрессоров: 3
   Правил: 12
   Скорость обучения: 0.01
   Эпох: 100
   Расчет потерь: Функция Хьюбера
   Оптимайзер: Алгоритм ADAM
   Файл: D:\ivk\clouds\Mail.Ru.Cloud\status_4_models\achives\ m_z1_z2_z5_y
   Модель уже была обучена: Да (итоговые потери при обучении: 0.00272688, при проверке: 0.00332527)
   Набор данных для обучения: ds_c1_cn_100_z1_z2_z5_y
   Набор данных для проверки: ds_c1_cn_100_z1_z2_z5_y_check
   Затраченное время: 1.406487 s
   Итоговое значение функции потерь (обучение): 0.002990392
   Итоговое значение функции потерь (проверка): 0.007112003
   12.09.2022 15:43:57 Обучение модели завершилось успешно.
   12.09.2022 15:43:58 Сохранение модели нейронной сети в базе данных завершилась успешно.
   12.09.2022 15:43:58 Существующая модель "m_z1_z2_z5_y" успешно заменена.
   12.09.2022 15:44:01 Сохранение модели нейронной сети в файловой системе завершилась успешно.
```

Диагностика состояния ТП. Функция демонстрирует работу процедур диагностики ТП с помощью нечеткой НС.

Реализованы следующие виды анализа, выполняемые после выбора диагностической модели НС.

1. Определение вероятности дефекта в ТЦ по заданным ненормативным значениям ДП: ввод интервальных значений ДП, входящих в состав модели; расчет степени уверенности в том, что заданный интервал значений ДП есть инцидент; определение вероятности дефекта в ТЦ с помощью НС.

2. Определение вероятности дефекта в ТЦ по данным потока значений ДП: получение потока значений ДП от сенсоров оборудования ТЦ; выявление инцидентов с расчетом степени уверенности в том, что инцидент произошел; определение вероятности дефекта в ТЦ с помощью НС.

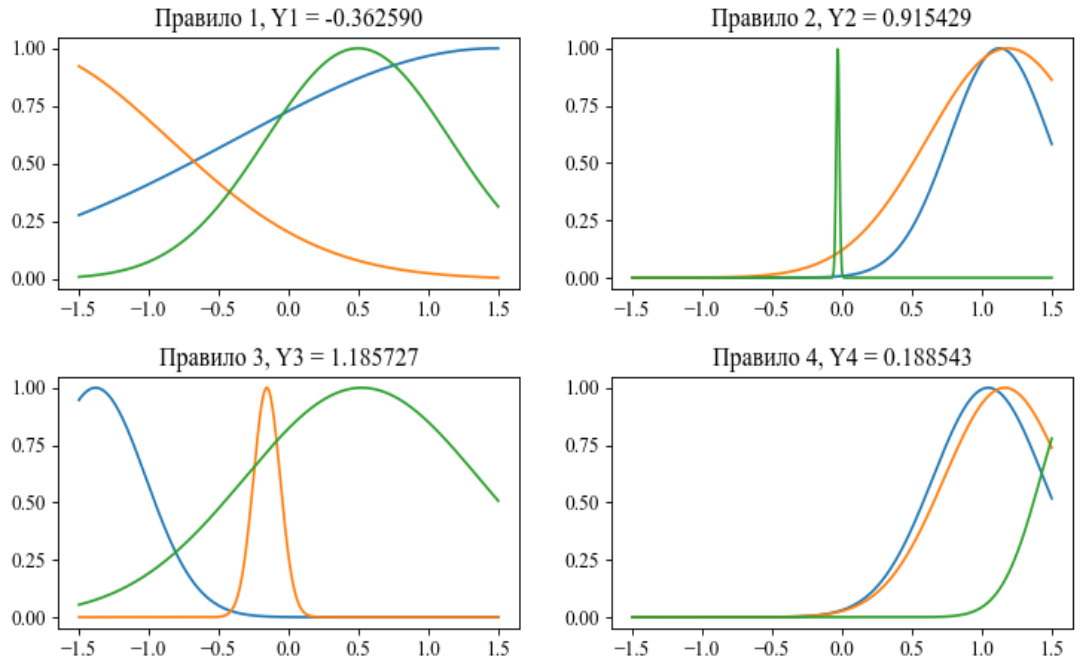

Рис. 4. Пример итогового вида функций принадлежности $\mu_l(x_{nm})$ для каждого из правил настроенной модели НС и каждой входной переменной x_{nm}: — $\mu_l(x_{n1})$; — $\mu_l(x_{n2})$; — $\mu_l(x_{n3})$

Fig. 4. An example of the final form of membership functions $\mu_l(x_{nm})$ for each rule of the tuned neural network model and each input variable x_{nm}: — $\mu_l(x_{n1})$; — $\mu_l(x_{n2})$; — $\mu_l(x_{n3})$

На рисунке 5 представлен пример проверки качества НС при расчете вероятностей дефекта в ТЦ по заданным значениям $\mu_l(x_{nm})$ (использован сгенерированный проверочный набор данных).

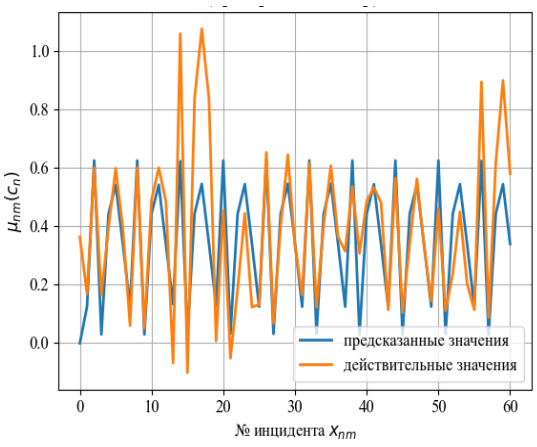

Рис. 5. Соответствие предсказанных и действительных значений $\mu_{nm}(c_n)$ (проверочный набор)

Fig. 5. A correspondence between predicted and actual values of $\mu_{nm}(c_n)$ (a test set)

Заключение

Разработанный исследовательский демонстратор в целом готов к использованию в экспериментальных исследованиях особенностей применения нечетких НС в диагностических системах. Решены задачи генерации обучающих данных и продукционных правил для настройки НС в модуле нечеткого вывода диагностической экспертной системы. Реализованы протоколирование, визуализация и сохранение метаданных, включая модели НС с различными конфигурациями гиперпараметров.

Очевидно, что наличие демонстратора дает возможность получать экспериментальные подтверждения эффективности совместного применения рассмотренных подходов, моделей и методов. При функционировании программной диагностической системы в условиях, близких к реальным, могут быть проверены и экспериментально обоснованы исходные предположения, касающиеся сокращения времени обнаружения и прогнозирования инцидентов при выполнении сложного многостадийного ТП и более точного опреде-

ления множества ТЦ, являющихся причинами инцидентов.

Разработанной функциональности достаточно для решения этих исследовательских задач. Отметим, что демонстратор имеет открытую модульную архитектуру, что позволяет встраивать новые модули для дополнительных функций и использовать отлаженные алгоритмы и настроенные модели в производственных программных комплексах. Таким образом будет достигаться поставленная цель создания научно-технического задела для передачи готовых решений на следующие этапы проекта с минимизацией рисков. В дальнейшем предполагается разработка имитационной модели многостадийного ТП с использованием обсуждаемого демонстратора как основы для моделирования системы диагностики такого процесса.

Работа выполнена при финансовой поддержке РФФИ, проект № 20-07-00199.

Литература

1. 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.

2. Иванов В.К., Палюх Б.В. Совместное использование моделей и методов нейронных сетей и теории свидетельств в нечетких системах управления и диагностики // Искусственный интеллект и принятие решений. 2021. № 4. С. 75–88. DOI: 10.14357/20718594210407.

3. Hoang D.T., Kang K.J. A bearing fault diagnosis method using transfer learning and Dempster–Shafer evidence theory. Proc. Int. Conf. AIRC, 2019, pp. 33–38. DOI: 10.1145/3388218.3388220.

4. Itkina M., Driggs-Campbell K., Kochenderfer M.J. Dynamic environment prediction in urban scenes using recurrent representation learning. Proc. IEEE ITSC, 2019, pp. 2052–2059. DOI: 10.1109/ITSC.2019.8917271.

5. Denœux Th. Logistic regression, neural networks and Dempster–Shafer theory: A new perspective. Knowledge-Based Systems, 2019, vol. 176, pp. 54–67. DOI: 10.1016/j.knosys.2019.03.030.

6. Tong Z., Xu P., Denœux T. ConvNet and Dempster-Shafer theory for object recognition. Proc. SUM. Lecture Notes in Computer Science, 2019, vol. 11940, pp. 368–381.

7. Gao C., Wang F., Xu D. Gas outburst prediction based on the intelligent Dempster–Shafer evidence theory. Proc. IX ICMIC, 2017, pp. 897–901. DOI: 10.1109/ICMIC.2017.8321582.

8. Li L., Tang J., Liu Y. Partial discharge recognition in gas insulated switchgear based on multi-information fusion. IEEE Transactions on Dielectrics and Electrical Insulation, 2015, vol. 22, no. 2, pp. 1080–1087. DOI: 10.1109/TDEI.2015.7076809.

9. Saha Sh., Saha Sa., Bhattacharyya P.P. Classifier fusion for liver function test based Indian jaundice classification. Proc. Int. Conf. MAMI, 2015, pp. 1–6. DOI: 10.1109/MAMI.2015.7456588.

10. Chen Y., Yang Y., Li J., Zhang Y. Intelligent fault diagnosis technology based on hybrid algorithm. Proc. CCDC, 2016, pp. 3702–3706. DOI: 10.1109/CCDC.2016.7531627.

11. Ivanov V.K., Palyukh B.V., Sotnikov A.N. Generation of production rules with belief functions to train fuzzy neural network in diagnostic system. Lobachevskii J. of Mathematics, 2022, vol. 43, no. 10, pp. 141–150.

12. Халов Е.А. Систематический обзор четких одномерных функций принадлежности интеллектуальных систем // ИТиВС. 2009. № 3. С. 60–74.

13. Yager R., Liping L. Classic Works of the Dempster–Shafer Theory of Belief Functions. London, Springer Publ., 2010, 825 p.

14. Jang J.-S.R. ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics, 1993, vol. 3, no. 23, pp. 665−685. DOI: 10.1109/21.256541.

15. Reineking T. A Python library for performing calculations in the Dempster–Shafer theory of evidence. GitHub. URL: https://github.com/reineking/pyds (дата обращения: 12.09.2022).

16. Cuervo S. A Tensorflow implementation of the Adaptive Neuro-Based Fuzzy Inference System (ANFIS). URL: https://github.com/tiagoCuervo/TensorANFIS (дата обращения: 12.09.2022).

17. Gregor G. An implementation of Adaptive-Network-Based Fuzzy Inference System (ANFIS) based on Keras on top of Tensorflow 2.0. URL: https://github.com/gregorLen/AnfisTensorflow2.0 (дата обращения: 12.09.2022).

Software & Systems
DOI: 10.15827/0236-235X.140.609-617

Received 20.09.22
2022, vol. 35, no. 4, pp. 609–617

A software platform demonstrator for configuring ANFIS neural network hyperparameters in fuzzy systems

V.K. Ivanov [1], *Ph.D. (Engineering), Associate Professor of Information System Department, mtivk@mail.ru*
B.V. Palyukh [1], *Dr.Sc. (Engineering), Professor, Head of Information System Department, pboris@tstu.tver.ru*

[1] *Tver State Technical University, Tver, 170026, Russian Federation*

Abstract. This article describes the research demonstrator for experimental verification and evaluation of fuzzy algorithms and neural networks in an expert system for complex multi-stage technological processes. The demonstrator development purpose is to create a scientific and technical foundation for the ready-to-implement solutions transfer to the next project stages.

The demonstrator allows assessing the readiness level of the components being developed, conducting research tests, checking the operability and efficiency of the software implementations functioning proposed at various parameter values and their combinations. A complex multi-stage technological process state diagnostics involves the joint primary data processing to obtain probabilistic abnormal critical events or incidents characteristics under conditions of uncertainty.

The authors propose a way of using a fuzzy neural network, which is trained with data generated by belief functions. The approach makes it possible to significantly speed up calculations and to minimize the resource base. The article focuses on describing the neural network models and training datasets management, neural network training and quality control, the technological process diagnostics in various modes. The configurable hyper-parameters of the neural network are described in detail. There are examples of the diagnostic procedures implementation in various modes. It is shown that with the software diagnostic system functioning in conditions close to real, the initial assumptions concerning the time reduction for detecting and predicting incidents can be verified and experimentally substantiated. In addition, the technological chains sets that are the incidents causes can be more accurately determined.

Keywords: demonstrator, ANFIS, belief function, diagnostics, evidence theory, fuzzy logic, incident, membership function, multistage production process, neural network, process chain, production rule, TSK.

Acknowledgements. *The research has been financially supported by the Russian Foundation for Basic Research within the framework of project no. 20-07-00199.*

References

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

2. Ivanov V.K., Palyukh B.V. Joint use neural networks and evidence theory methods in control and diagnostic fuzzy systems. *Artificial Intelligence and Decision-Making*, 2021, no. 4, pp. 75–88. DOI: 10.14357/20718594210407 (in Russ.).

3. Hoang D.T., Kang K.J. A bearing fault diagnosis method using transfer learning and Dempster–Shafer evidence theory. *Proc. Int. Conf. AIRC*, 2019, pp. 33–38. DOI: 10.1145/3388218.3388220.

4. Itkina M., Driggs-Campbell K., Kochenderfer M.J. Dynamic environment prediction in urban scenes using recurrent representation learning. *Proc. IEEE ITSC*, 2019, pp. 2052–2059. DOI: 10.1109/ITSC.2019.8917271.

5. Denœux Th. Logistic regression, neural networks and Dempster–Shafer theory: A new perspective. *Knowledge-Based Systems*, 2019, vol. 176, pp. 54–67. DOI: 10.1016/j.knosys.2019.03.030.

6. Tong Z., Xu P., Denœux T. ConvNet and Dempster-Shafer theory for object recognition. *Proc. SUM. Lecture Notes in Computer Science*, 2019, vol. 11940, pp. 368–381.

7. Gao C., Wang F., Xu D. Gas outburst prediction based on the intelligent Dempster–Shafer evidence theory. *Proc. IX ICMIC*, 2017, pp. 897–901. DOI: 10.1109/ICMIC.2017.8321582.

8. Li L., Tang J., Liu Y. Partial discharge recognition in gas insulated switchgear based on multi-information fusion. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2015, vol. 22, no. 2, pp. 1080–1087. DOI: 10.1109/TDEI.2015.7076809.

9. Saha Sh., Saha Sa., Bhattacharyya P.P. Classifier fusion for liver function test based Indian jaundice classification. *Proc. Int. Conf. MAMI*, 2015, pp. 1–6. DOI: 10.1109/MAMI.2015.7456588.

10. Chen Y., Yang Y., Li J., Zhang Y. Intelligent fault diagnosis technology based on hybrid algorithm. *Proc. CCDC*, 2016, pp. 3702–3706. DOI: 10.1109/CCDC.2016.7531627.

11. Ivanov V.K., Palyukh B.V., Sotnikov A.N. Generation of production rules with belief functions to train fuzzy neural network in diagnostic system. *Lobachevskii J. of Mathematics*, 2022, vol. 43, no. 10, pp. 141–150.

12. Halov E.A. The regular review of crisp one-dimensional membership functions of intellectual systems. *JITCS*, 2009, no. 3, pp. 60–74 (in Russ.).

13. Yager R., Liping L. *Classic Works of the Dempster–Shafer Theory of Belief Functions*. London, Springer Publ., 2010, 825 p.

14. Jang J.-S.R. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 1993, vol. 3, no. 23, pp. 665−685. DOI: 10.1109/21.256541.

15. Reineking T. A Python library for performing calculations in the Dempster–Shafer theory of evidence. *GitHub*. Available at: https://github.com/reineking/pyds (accessed September 12, 2022).

16. Cuervo S. A Tensorflow implementation of the Adaptive Neuro-Based Fuzzy Inference System (ANFIS). *GitHub*. Available at: https://github.com/tiagoCuervo/TensorANFIS (accessed September 12, 2022).

17. Gregor G. An implementation of Adaptive-Network-Based Fuzzy Inference System (ANFIS) based on Keras on top of Tensorflow 2.0. *GitHub*. Available at: https://github.com/gregorLen/AnfisTensorflow2.0 (accessed September 12, 2022).

Для цитирования

Иванов В.К., Палюх Б.В. Демонстратор программной платформы для настройки гиперпараметров нечеткой нейронной сети // Программные продукты и системы. 2022. Т. 35. № 4. С. 609–617. DOI: 10.15827/0236-235X.140.609-617.

For citation

Ivanov V.K., Palyukh B.V. A software platform demonstrator for configuring ANFIS neural network hyperparameters in fuzzy systems. *Software & Systems*, 2022, vol. 35, no. 4, pp. 609–617 (in Russ.). DOI: 10.15827/0236-235X.140.609-617.