



САНКТ-ПЕТЕРБУРГСКИЙ  
ГОСУДАРСТВЕННЫЙ  
ЭЛЕКТРОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ «ЛЭТИ»  
им. В.И. Ульянова (Ленина)

ИНСТИТУТ ФИЗИОЛОГИИ  
им. И.П. Павлова  
РОССИЙСКОЙ  
АКАДЕМИИ НАУК



**IEEE**

РОССИЙСКАЯ  
СЕВЕРО-ЗАПАДНАЯ СЕКЦИЯ  
МЕЖДУНАРОДНОГО  
ИНСТИТУТА ИНЖЕНЕРОВ  
ЭЛЕКТРОТЕХНИКИ И  
ЭЛЕКТРОНИКИ IEEE

**NEURONT.ETU.RU**

# II МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ ПО НЕЙРОННЫМ СЕТЯМ И НЕЙРОТЕХНОЛОГИЯМ

*16 июня 2021*

**NeuroNT'21**

## СБОРНИК ДОКЛАДОВ



Санкт-Петербург, 2021

**II МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ  
ПО НЕЙРОННЫМ СЕТЯМ  
И НЕЙРОТЕХНОЛОГИЯМ  
(NEURONT'2021)**

**16 июня 2021 года**

*Сборник докладов*

Санкт-Петербург  
2021

УДК 004.032.26

ISBN 978-5-7629-2878-6

II Международная конференция по нейронным сетям и нейротехнологиям (NeuroNT'2021):  
сб. докладов. СПб. 2021. – 66 с.

## СОДЕРЖАНИЕ

### I. КОНЦЕПЦИИ ПОСТРОЕНИЯ ГИБРИДНОГО ИНТЕЛЛЕКТА

<i>Средства программно-аппаратной реализации нейро-нечетких систем</i> Морозов С.М., Куприянов М.С. ....	3
---	---

### II. ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ИХ ПРИЛОЖЕНИЯ

<i>Самонастройка ПИД-регулятора с помощью нейронной сети для нелинейной системы управления экзоскелетом</i> Белов М.П., Чьонг Д.Д., Фам Ван Туан .....	7
<i>Анализ весовых коэффициентов нейронной сети прямого распространения как метод оценки модели</i> Жангиров Т.Р., Крвчик А.Б., Зуев Д.В., Лисс А.А., Экало А.В., Григорьева Н.Ю. ....	11
<i>XAI Evaluation: Evaluating Black-Box Model Explanations for Prediction</i> Zhang Yuyi, Xu Feiran, Zou Jingying, Petrosian O.L., Krinkin K.V. ....	15
<i>Сравнение методов кластеризации данных в газотранспортной системе</i> Петрова А.К., Синица А.М. ....	19
<i>Использование OpenMP для оптимизации процесса обучения модели в алгоритмах машинного обучения</i> Mohammed Omar T., Heidari Moeid S., Пазников А.А. ....	23
<i>Интеллектуальное управление информационной безопасностью киберфизических систем</i> Шичкина Ю.А., Фаткиева Р.Р. ....	29
<i>Исследование результатов тестирования собак-поводырей и информации об их происхождении</i> Бекенева Я.А. ....	33
<i>Нейросетевой подход к преодолению априорной неопределённости при оптимальном планировании действий интеллектуальных информационных агентов для мягких архитектур сервис-ориентированных систем</i> Птицына Л.К., Эль Сабаяр Шевченко Н.Н., Белов М.П., Птицын А.В. ....	36

### III. УПРАВЛЕНИЕ ДАННЫМИ И ОРГАНИЗАЦИЯ ВЫЧИСЛЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

<i>Подход к трансляции базы данных с MySQL в Cassandra</i> Ха В.М., Шичкина Ю.А. ....	40
<i>Проектирование структуры нейросети для прогнозирования заболеваемости злокачественными новообразованиями</i> Жилкина М.В., Шулунова И.Р., Федоров А.Э. ....	44
<i>Элементная база узлов импульсных нейронных сетей</i> Буренева О.И. ....	47
<i>Адаптивная нейро-нечеткая система (ANFIS) информационного взаимодействия в сетях промышленного интернета вещей с учётом выравнивания нагрузки между узлами</i> Мукешимана К., Куприянов М.С. ....	51
<i>Протокол взаимодействия агентов, решающих задачу распределения анализа данных</i> Гонсалес Перес Юлейси, Холод И.И. ....	55
<i>Обработка данных в туманных вычислительных средах для машинного обучения</i> Субботин А.Н. ....	59
<i>Изометрический векторный нейроанализатор</i> Сафьянников Н.М., Буренева О.И., Чепасов А.Е., Алексанян З.А. ....	63

# Средства программно-аппаратной реализации нейро-нечетких систем

С. М. Морозов

Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)  
frostsergei01@gmail.com

М. С. Куприянов

Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)  
mikhail.kupriyanov@gmail.com

**Аннотация.** В работе представлен краткий обзор аппаратных средств построения систем, основанных на нейро-нечетких механизмах, и их программных аналогов. Эти средства позволяют выполнить аппаратную реализацию нечеткой логики и нейронных сетей для того, чтобы на их основе создавать нейро-нечеткие вычислительные механизмы в рамках встраиваемых систем. Представленные специализированные средства дают возможность создавать эффективные встраиваемые системы гибридного искусственного интеллекта.

**Ключевые слова:** нейро-нечеткие вычисления, нечеткие процессоры, нейронные процессоры, микроконтроллеры, встраиваемые системы

## I. ВВЕДЕНИЕ

Нейро-нечеткие системы – это системы гибридного искусственного интеллекта, основанные на концепциях нейронных сетей и нечеткой логики, которые широко используются для построения систем искусственного интеллекта.

Как правило, нейро-нечеткие вычислительные механизмы реализуются в рамках программных вычислительных систем, которые обрабатывают большие потоки информации. Однако иногда необходимо осуществить аппаратную или программно-аппаратную реализацию этих механизмов.

Существующие подходы по разработке нейро-нечетких систем могут быть программно выполнены на практически любой аппаратной базе, однако это не всегда является эффективным. Нейронные сети требуют достаточно высоких аппаратных характеристик системы для эффективного функционирования. В частности, нейронные сети требуют больших объемов памяти для хранения.

Для решения проблем, связанных с построением компонентов нейро-нечетких систем, существуют аппаратные решения: процессоры с нечеткой логикой и нейропроцессоры. Эти компоненты являются наиболее подходящими средствами для реализации нейро-нечетких систем, однако они очень слабо распространены. Производители процессоров без аппаратной поддержки этих концепций предлагают программные решения, которые упрощают программное исполнение нечеткой логики и нейронных сетей. Производители могут

реализовывать эти решения как модули для существующих программных продуктов. Такие решения работают с готовыми моделями, что обеспечивает максимальную эффективность при программной разработке составляющих нейро-нечетких систем.

## II. НЕЧЕТКИЕ ПРОЦЕССОРЫ

Идею использования нечеткой логики в области электроники начали развивать группы исследователей в Америке и Японии. Это позволило упростить техническое описание различных проблем и, соответственно, разработать более эффективное решение.

Нечеткая логика дает возможность перейти от классической двузначной логики к многозначной, в которой значения равны не только 0 и 1, но и могут принимать значения, находящиеся между ними. С аппаратной точки зрения нечеткая логика позволяет осуществлять переход между цифровыми сигналами и аналоговыми.

На основе этой идеи в 1986 году были разработаны первые процессоры, которые имели поддержку команд нечеткой логики. Архитектура нечеткого компьютера представлена на рис. 1. [1]

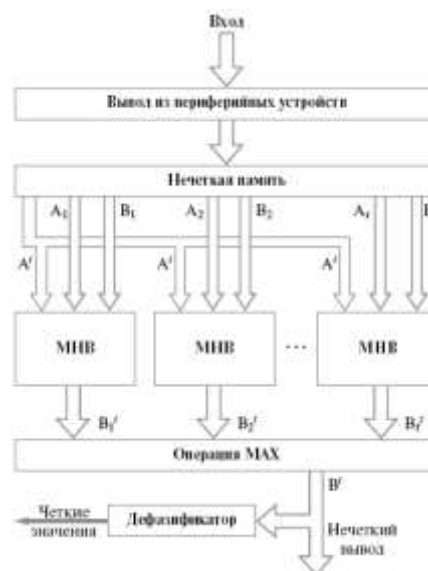


Рис. 1. Архитектура нечеткого компьютера

В 90-х был выпущен первый кристалл, сделанный по аналого-цифровой технологии. Этот кристалл переводил аналоговые сигналы в нечеткий вид, обрабатывал их логическими правилами и преобразовывал в аналоговый сигнал. Этот процессор – первая аппаратная реализация нечеткой логики без программного обеспечения. Применение таких процессоров позволяет создавать системы, которые проще, чем системы с использованием процессоров на классической логике. Нечеткие системы при этом будут быстрее.

Каскадирование процессоров на нечеткой логике является одним из вариантов реализации нейропроцессоров. Иногда такие каскады называют нейро-нечеткими процессорами [1].

Производство нечетких вычислительных узлов является достаточно сложной задачей, поэтому эти процессоры не получили широкого распространения. Однако производители микроконтроллеров и процессоров встраивали элементы нечеткой логики в свои микросхемы. Одним из самых первых нечетких процессоров является 68HC12 фирмы Motorola [2]. Другие известные представители – SAE 81C99 от Siemens [3] и FP-3000 от Ormon [4]. Производитель STMicroelectronics, известный за микроконтроллеры семейств STM32 производил микроконтроллеры ST52 Dualogic [5] и ST62 [6]. Для работы с ними существуют специализированные среды разработки, например, FuzzyStudio 3.0 [7] или fuzzyTECH [8]. В 2014 году была представлена программа FuzzyStudio, которая генерирует код на языке Java [9]. Предыдущие программы генерировали код на языке C.

Многие из этих процессоров являются устаревшими, однако производители предлагают программные решения применения нечеткой логики. Одним из таких производителей является фирма Texas Instruments, предлагающая пример программной реализации нечеткой логики на микроконтроллере MSP430 [10].

### III. НЕЙРОПРОЦЕССОРЫ

Нейропроцессоры являются специализированными микросхемами, реализующими аппаратное ускорение ряда алгоритмов искусственного интеллекта. В частности, такие процессоры выполняют аппаратное ускорение нейронных сетей.

Главным отличием нейронного процессора от традиционного является особенность архитектуры. В классическом процессоре существуют четко выраженные блоки, каждый из которых выполняет свою функцию. Нейронные процессоры имеют большое количество одинаковых ячеек (нейронов), каждая из которых имеет свою память и вычислительные механизмы. В результате получается многоядерная система, способная ускорять выполнение большого ряда задач за счет использования параллельной вычислительной архитектуры: распознавание изображений, фильтрация и т. д. [11].

Существуют простые и доступные нейропроцессоры, на которых можно рассмотреть процесс аппаратных нейровычислений. Например, процессоры CIMK [12] и

NM500 [13] от фирмы NeuroMem. Они требуют наличия управляющей микросхемы, которая обеспечит их данными. Связь осуществляется по 2-направленной 26-битной шине. Существуют и нейронные процессоры серии QuarkSE/Curie от фирмы Intel [14]. Их сравнение и основные функции представлены в таблице.

ТАБЛИЦА I СРАВНЕНИЕ НЕЙРОПРОЦЕССОРОВ

Характеристики			
Параметры	QuarkSE/Curie	CIMK	NM500
Производитель	Intel	NeuroMem	NeuroMem
Число нейронов	128	1024	576
Память нейрона	128 байт	256 байт	
Категории	15 бит		
Дистанция	16 бит		
К-ближайших соседей	7 бит		
Радиально-базисных функций	Есть		
Расчет дистанции LSUP	Есть		
Расчет дистанции L1 (Manhattan)	Есть		
Каскадирование	Нет	Есть	
Частота / частота при каскадировании	32/- МГц	27/16 МГц	37/20 МГц
Технология	22 нм	130 нм	110 нм

Представленные нейропроцессоры работают в двух режимах: обучение и распознавание. Рассмотрим процесс обучения на примере NM500. Каждый нейрон имеет собственную программируемую память: основную 256 байт (шаблон), контекст (8 бит), категорию (16 бит). Категория используется для классификации объектов, а контекст для разделения нейронов на подсети. Кроме того, у каждого нейрона есть собственный 24-битный идентификатор NID (Neuron ID). В обычном процессоре программа записывается перед началом работы, в то время как во многих нейропроцессорах (не во всех) программирование может осуществляться прямо в процессе работы и называется обучением [11].

Некоторые нейропроцессоры поддерживают возможность каскадирования. Это необходимо на случай нехватки процессором имеющихся у него ресурсов. Каскад процессоров позволяет наращивать вычислительную мощность системы.

Такие процессоры могут применяться для аппаратного ускорения задач искусственного интеллекта, которые связаны с использованием нейронных сетей. Применение большого числа обучаемых процессоров позволяет создавать обучаемые системы, однако такое решение является достаточно дорогим.

### IV. ПРОГРАММНЫЕ АНАЛОГИ

Представленные нечеткие процессоры и нейропроцессоры позволяют создавать эффективные вычислительные системы. Однако у этих процессоров есть недостатки. Процессоры нечеткой логики на данный

момент слабо распространены, а нейропроцессоры являются достаточно дорогими.

Производители микроконтроллеров осознают преимущества использования нечеткой логики и нейронных сетей, поэтому предлагают программистам подходы и средства по эффективной программной реализации этих концепций на своих микроконтроллерах. Такой способ не позволяет получить существенное увеличение производительности как при использовании аппаратных средств, но дает возможность реализации указанных концепций.

Как уже было упомянуто, фирма Texas Instruments предлагает подход по использованию нечеткой логики в микроконтроллерах серии MSP430. Их пример описывает нечеткое управление мотором [10]. В документации приведено описание всех необходимых процедур нечеткой логики (фаззификация, вывод, дефаззификация) и формулы, которые нужно запрограммировать.

Реализовать нейронные сети сложнее, чем нечеткую логику. Можно выполнить существующие алгоритмы в процессоре или микроконтроллере, однако это является очень сложной задачей в условиях ограниченного объема памяти микросхем. Также процесс обучения может занимать очень много времени.

Для решения этих проблем существуют специальные программные средства. Одним из таких средств является модуль STM32CubeAI для генератора кода STM32CubeMX. Этот модуль позволяет использовать предварительно обученные модели в программе микроконтроллеров STM32, осуществляя перенос этой модели в память микросхемы. Модели могут быть обучены с помощью самых известных библиотек по созданию нейронных сетей: TensorFlow, Keras, и т. д. [15]

Рассмотрим перенос модели в микроконтроллер. Создадим и обучим её для решения задачи XOR. Модель, разработанная на языке Python с помощью библиотеки Keras, занимает мало места, что позволяет оценить эффективность её переноса в микросхему. Предусмотрена возможность её сжатия для более эффективного использования памяти. Без сжатия модель занимает 36 байт Flash памяти и 24 байта оперативной памяти. Следует отметить, что приложение не только позволяет легко сохранить сеть в памяти микроконтроллера, но и подключить функции для того, чтобы эта сеть заработала. Однако при этом подключаются все библиотеки, необходимые для корректной работы, что также занимает память микроконтроллера. Это стоит учитывать при разработке приложения. Несмотря на то, что сама нейронная сеть небольшая (36 весовых коэффициентов, каждый из которых занимает один байт), подключение библиотеки нейронных сетей вместе со стандартными библиотеками приведет к тому, что программа будет занимать 82 килобайта. Из этого можно сделать вывод, что подключение такого механизма требует достаточно большого объема памяти микроконтроллера. Например, низкобюджетные микросхемы серии STM32F303R8 не смогут сохранить всю сеть, потому что 64 килобайт встроенной памяти не хватит для подключения

необходимых библиотек. Также модуль предлагает проверить модель на компьютере или микроконтроллере. На рис. 2 представлен анализ модели для решения задачи XOR.

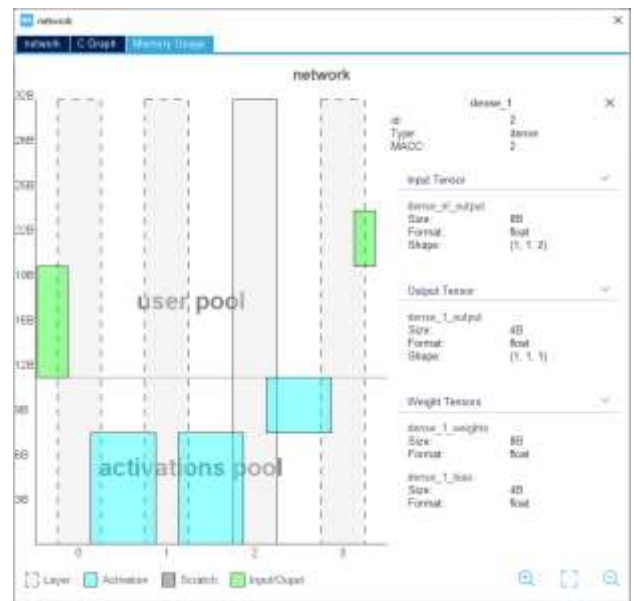


Рис. 2. Анализ модели нейронной сети

Совмещение этих подходов позволяет создать статический нейро-нечеткий вычислительный механизм в рамках задач по управлению. При этом можно использовать один микроконтроллер как для реализации нечеткой логики, так и для нейронной сети, а можно выполнять эти задачи на двух разных микросхемах.

Также возможно комбинирование подходов. Например, реализовать нечеткие вычисления программным способом, а нейронные сети выполнить на отдельном процессоре.

## V. ЗАКЛЮЧЕНИЕ

Представленные в работе средства позволяют осуществить реализацию нейро-нечетких выполнение во встраиваемых системах. Использование процессоров с нечеткой логикой и нейропроцессоров дает возможность существенно ускорить выполнение задач, связанных с машинным обучением.

Было рассмотрено использование программных средств, предназначенных для применения концепций нечеткой логики и нейронных сетей. Такой подход позволяет использовать эти концепции во встраиваемых системах. С помощью этого подхода можно создать модель искусственного интеллекта и встроить её в разрабатываемое устройство.

Рассмотренные способы построения нейро-нечетких вычислительных механизмов, а также их комбинации, дают возможность создавать встраиваемые системы гибридного искусственного интеллекта.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Богданов А.В., Корхов В.В., Мареев В.В., Станкова Е.Н., Архитектуры и топологии многопроцессорных вычислительных систем. Москва: Интернет-Университет Информационных Технологий, 2004. 176 с.
- [2] Motorola: "CPU12 Reference Manual: M68HC12 & HCS12 Microcontrollers." Rev. 3, 2002.
- [3] Шувьуты AG, Semiconductor Group: "SAE 81C99", Data Sheet, 1994.
- [4] Ormon Corp.: "FP-3000 Digital Fuzzy Processor", Data Sheet, 1990.
- [5] STMicroelectronics: "ST52 TM DualLogic A New Family of MCU With Fuzzy Capabilities", 2000.
- [6] STMicroelectronics: "ST62 MCU Family: Products and Development Tools Selector", 2000
- [7] STMicroelectronics: "FuzzyStudio. User Manual, 1<sup>st</sup> edition", May 1998.
- [8] FuzzyTECH [Электронный ресурс], URL: <https://fuzzytech.com/> (дата обращения: 13.03.2021)
- [9] Marcelo de Souza, Fernando Santos, Adolfo R. de Soto, Adilson Vahldick, "FuzzyStudio: A Web Tool for Modeling and Simulation of Fuzzy Systems" // Brazilian Conference on Intelligent Systems, December 2014.
- [10] Texas Instruments: "Fuzzy Logic Motor Control With MSP430 MCUs", Application Report, 2018
- [11] Терраэлектроника: "Кратко о том, как работают нейропроцессоры на примере NM500 от NeuroMem", 2018 [Электронный ресурс]. URL: <https://spb.terraelectronica.ru/news/5549> (дата обращения: 13.03.2021)
- [12] NeuroMem: "CM1K. Hardware User's Manual", Data Sheet, 2012
- [13] NeuroMem: "NM500 Preliminary", Data Sheet, 2017
- [14] Intel: "Intel Quark SE microcontroller" Data Sheet, 2015.
- [15] STMicroelectronics: "Artificial Intelligence (AI) software expansion for STM32Cube" Data Brief, 2021.

# Самонастройка ПИД-регулятора с помощью нейронной сети для нелинейной системы управления экзоскелетом

М. П. Белов<sup>1</sup>, Д. Д. Чыонг<sup>2</sup>

Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)

<sup>1</sup>milesa58@mail.ru, <sup>2</sup>dinh dangtruong@gmail.com

Фам Ван Туан

Вьетнамский государственный технический  
университет им. Лэ Куи Дона

tuanhvhq@gmail.com

**Аннотация.** В статье предлагается применение метода аналоговых нейронных сетей (НС) с радиальной базисной функцией (РБФ) и самонастройкой коэффициентов пропорционально-интегрального дифференциального (ПИД)-регулятора для нелинейной системы управления нижних конечностей экзоскелетона. Реализация точного и качественного управления нелинейными системами, включающими неопределенности параметров и внешние возмущения, возможно с помощью аналогового нейросетевого контроллера, обладающего возможностью непрерывного обучения и адаптации. НС в ПИД-регуляторе позволяет исправлять возникающие ошибки из-за неопределенностей и изменений параметров при движении нижних конечностей экзоскелетона. Результаты работы предложенного алгоритма управления продемонстрированы посредством имитационного моделирования в среде Matlab/Simulink, подтверждающие его эффективность.

**Ключевые слова:** нейронная сеть, экзоскелетон, самонастройка, система управления, нелинейная система

## I. ВВЕДЕНИЕ

В настоящее время исследование и разработка современных экзоскелетов является актуальной задачей, которой занимаются ученые всего мира. Структура экзоскелета представляет собой систему сложных электромеханических связей, в которых невозможно полностью измерить все состояния системы. Другими словами, невозможно построить точную динамическую модель со всеми нелинейными компонентами, учитывающую динамические изменения, связанные с движением носителя. Одним из решений этих проблем является применение адаптивных регуляторов, которые широко исследованы в [1, 2, 3]. ПИД-регулятор широко применяется в различных системах управления благодаря своей простой структуре и реализации. Одной из самых больших проблем при его проектировании является нахождение подходящего значения коэффициента усиления. Применение ПИД регулятора для нелинейных систем и систем, имеющих изменяющиеся параметры в процессе работы, не эффективно.

НС являются эффективным инструментом для управления нелинейными системами. Они обладают преимуществами такими как адаптивное обучение, отказоустойчивость и обобщение. Фиерро и Льюис [4]

разработали регулятор на основе искусственной нейронной сети сочетающий метод управления скоростью с обратной связью и регулированием крутящего момента многослойной нейронной сетью в прямом канале. Но структура регулятора и алгоритм обучения НС очень сложны и требуют больших вычислительных ресурсов. В этой работе представлена возможность применения ПИД-регулятора и НС, способной самообучаться и адаптироваться в нелинейной системе управления. Однако в работе [5] линейный ПИД-регулятор, настраиваемый с использованием НС, но он обычно может ухудшить эффективность управления, если в управляемом процессе присутствует сильная нелинейность. Поэтому в данной статье предлагается нелинейный ПИД-регулятор, настраиваемый с использованием НС, который может улучшить эффективность управления нелинейными системами.

## II. ПОСТРОЕНИЕ СТРУКТУРЫ НЕЙРОАДАПТИВНОГО ПИД-РЕГУЛЯТОРА

### A. Схема структуры систем управления

Структура предложенного алгоритма управления с нейронными сетями на основе ПИД-регуляторов показана на рис. 1, где  $e_p = q_d - q$  – ошибка следящего управления траектории суставов.

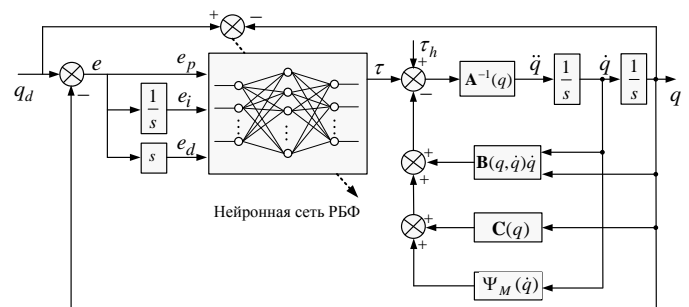


Рис. 1. Структура нейроадаптивного ПИД-регулятора для экзоскелетона

Этот алгоритм управления имеет простую структуру и метод постоянной автонастройки регулятора НС, небольшое время вычислений.



Экзоскелет нижних конечностей является системой с множеством степеней свободы и нелинейностями высокого порядка, что затрудняет оценить точность его динамики. Математическая модель системы нижних конечностей экзоскелетона-человека (НКЭЧ) с нелинейными механическими характеристиками можно записать в виде:

$$\mathbf{A}(q)\ddot{q} + \mathbf{B}(q, \dot{q})\dot{q} + \mathbf{C}(q) + \Psi_M(\dot{q}) = \tau + \tau_h \quad (1)$$

где  $\tau = [\tau_1 \quad \tau_2]^T$ ,  $\tau_h = [\tau_{h1} \quad \tau_{h2}]^T$ ;

$$\mathbf{A}(q) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \mathbf{B}(q, \dot{q}) = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \mathbf{C}(q) = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix},$$

где  $q \in \mathbb{R}^n$  – обобщенная координата позиции звеньев;  $\mathbf{A}(q)$  – матрица инерции, являющаяся положительно определенной;  $\mathbf{B}(q, \dot{q})$  – матрица центробежных сил и силы Кориолиса;  $\mathbf{C}(q)$  – вектор моментов силы тяжести;  $\tau_h$  – вектор момента ноги человека (считается внешним возмущением);  $\tau$  – вектор управляющего воздействия. Функция трения  $\Psi_M(\dot{q})$  представлена Кулоновским трением и вязким трением.

### В. Нейронная сеть РБФ

Сеть РБФ является нейронной сетью с прямой связью, которая обучается с использованием контролируемой техники обучения. Радиальные функции представляют собой особый класс, характерной особенностью которых является монотонно убывающий или увеличивающийся отклик от центральной точки [6, 7]. Сети РБФ способны аппроксимировать любое разумное отображение непрерывных функций с удовлетворительным уровнем точности [8, 9]. Обычная РБФ состоит из трех слоев: входного, скрытого и выходного. Связи между слоями умножаются на веса соответствующих размеров. Структура нейронной сети РБФ представлена на рис. 2, где

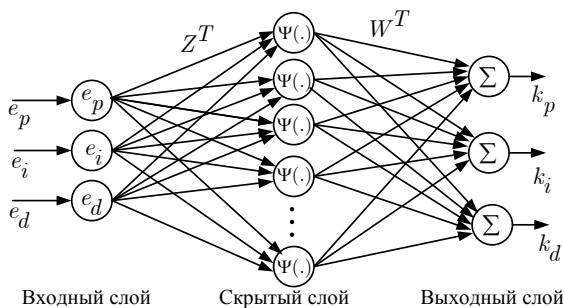


Рис. 2. Структура нейронной сети РБФ

входной слой нейронной сети  $\mathbf{x} = [e_p \quad e_i \quad e_d]^T$ ; скрытый слой: 10 нейронов; выходной слой

$\mathbf{Y} = [y_1 \quad y_2 \quad y_3]^T = [k_p \quad k_i \quad k_d]^T$ . Каждый вывод скрытого слоя может представлять следующим видом:

$$\Psi_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{2b_i^2}\right), \quad (i = 1, 2, \dots, 10)$$

где  $c_i$  – центральное положение функции Гаусса в нейронах скрытого слоя;  $b_i$  – ширина функции Гаусса в скрытом слое нейронов;  $\Psi(x) = [\Psi_1(x), \Psi_2(x), \dots, \Psi_{10}(x)]^T$  – выходной вектор нейронов скрытого слоя.

Уравнение выходного слоя можно записать как:

$$y_j = \sum_{i=1}^{10} W_{ij} \Psi_i(x), \quad (i = 1, 2, \dots, 10; j = 1, 2, 3)$$

где  $W_{ij}$  – веса между нейронами скрытого и выходного слоев.

Сетевая структура с трехслойным нейроном показана на рис. 3.

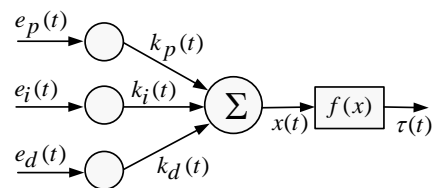


Рис. 3. Схема трехслойного нейрона на основе ПИД-регулятора

где  $\tau(t)$  – выход нейронной сети;  $x$  – вход сигмоидной функции  $f(\cdot)$ , имеющая нелинейную зависимость в виде функции:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (2)$$

### III. СИСТЕМА УПРАВЛЕНИЯ НЕЙРОАДАПТИВНОГО ПИД-РЕГУЛЯТОРА

Прямое управление аналоговой нейронной сетью РБФ на основе ПИД-регулятора применяется для слеящего управления траектории суставов экзоскелетона. Выход регулятора  $\tau(t)$  можно получить из следующего уравнения:

$$\tau(t) = f(x) \quad (3)$$

Входной сигнал сигмоидной функции в выходном слое определяется следующим образом:

$$x(t) = k_p(t)e_p(t) + k_i(t)e_i(t) + k_d(t)e_d(t) \quad (4)$$

где  $e_p(t) = q_d(t) - q(t)$ ,  $e_i(t) = \int_0^t e_p(t) dt$ ,  $e_d(t) = \frac{de_p(t)}{dt}$

НС обучаются с помощью обычного алгоритма обратного распространения для минимизации системной ошибки между выходными значениями углового

положения экзоскелетона и желаемым, определяемой из следующего уравнения:

$$E(t) = \frac{1}{2}(e_p(t))^2 \quad (5)$$

Из дискретного алгоритма с использованием метода наискорейшего спуска [5], [9] можно получить следующие уравнения (6).

$$\begin{cases} k_p(t) = k_p(0) - \eta_p \int_0^t \frac{\partial E(t)}{\partial k_p} dt, \\ k_i(t) = k_i(0) - \eta_i \int_0^t \frac{\partial E(t)}{\partial k_i} dt, \\ k_d(t) = k_d(0) - \eta_d \int_0^t \frac{\partial E(t)}{\partial k_d} dt. \end{cases} \quad (6)$$

где  $\eta_p, \eta_i, \eta_d$  – скорость обучения, определяющая скорость сходимости. Преобразовав уравнение (5), получим:

$$\begin{cases} \frac{\partial E(t)}{\partial k_p} = \frac{\partial E(t)}{\partial q} \frac{\partial q}{\partial \tau} \frac{\partial \tau(t)}{\partial x} \frac{\partial x}{\partial k_p} = -e_p(t) \frac{\partial q}{\partial \tau} f'(x(t))e_p(t), \\ \frac{\partial E(t)}{\partial k_i} = \frac{\partial E(t)}{\partial q} \frac{\partial q}{\partial \tau} \frac{\partial \tau(t)}{\partial x} \frac{\partial x}{\partial k_i} = -e_p(t) \frac{\partial q}{\partial \tau} f'(x(t))e_i(t), \\ \frac{\partial E(t)}{\partial k_d} = \frac{\partial E(t)}{\partial q} \frac{\partial q}{\partial \tau} \frac{\partial \tau(t)}{\partial x} \frac{\partial x}{\partial k_d} = -e_p(t) \frac{\partial q}{\partial \tau} f'(x(t))e_d(t). \end{cases} \quad (7)$$

Запишем производную уравнения (2) как:

$$f'(x) = \frac{3e^{-2x}}{(1+e^{-2x})^2} \quad (8)$$

Для удобства, как в работе Ямада и Ябуга [10], имеем:

$$\frac{\partial q(k)}{\partial \tau} = 1 \quad (9)$$

Подставив уравнения (8), (9) в уравнения (7) и затем в уравнение (6), получим:

$$\begin{cases} k_p(t) = k_p(0) + \eta_p \int_0^t e_p(t)e_p(t) \frac{3e^{-2x}}{(1+e^{-2x})^2} dt, \\ k_i(t) = k_i(0) + \eta_i \int_0^t e_p(t)e_i(t) \frac{3e^{-2x}}{(1+e^{-2x})^2} dt, \\ k_d(t) = k_d(0) + \eta_d \int_0^t e_p(t)e_d(t) \frac{3e^{-2x}}{(1+e^{-2x})^2} dt. \end{cases} \quad (10)$$

Проиллюстрируем эффективность предлагаемого алгоритма НС в ПИД-регуляторе посредством моделирования системы управления движением экзоскелетона.

#### IV. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Для моделирования системы управления примем: заданная траектория угла тазобедренного сустава  $q_{d1} = \sin(0.5t) + \sin(0.75t) + \cos(t)$ , рад и коленного сустава  $q_{d2} = 0.5 \sin(t)$ , рад. Начальные коэффициенты усиления ПИД-регулятора:  $k_p = \mathbf{diag}\{500, 200\}$ ;  $k_i = \mathbf{diag}\{60, 20\}$   $k_d = \mathbf{diag}\{30, 30\}$ . Использован метода обучения НС Байесовской регуляризации (Bayesian Regularization) (*trainbr*) обратного распространения с весами следующим:

b1 = [-0.49950367689623154099; 0.024977842154682083214; 0.35074128346375410548; -0.76156068934439691276; -0.079942077742934825046; 0.93674243078055230427; 0.0089281464566229318258; -1.1498044985131152806; -0.014466071094719871321; -0.010521772603769135798];

W1 = [-0.178849699671383644 0.0044604676959411181797 0.17231285619157155065; 0.065118367254124578936 0.12371480786058125512 0.026030875702891278362; 0.17867613562056222642 0.0032600399207136070839 -0.13709048048862976232; 0.21228874175274772829 0.0065332501360735673393 0.28502935587001670958; -0.18329430994026979507 -0.0029863939843420552953 -0.22422027779071587728; -0.22358419843172658292 0.0012204622694466852988 0.19568878492659907975; -0.060887217578337822499 0.12261676881041208564 0.016497631517890350844; -0.22095020233071546722 -0.0070232496025983068288 -0.29241418332370999744; -0.043190259276334588834 0.10734357819554612989 -0.0076347851671893802311; -0.039338988634882693374 -0.11440115825020892582 -0.014317538911954126243];

b2 = [-0.17921821884589267393; -0.024057544656470676581; -0.53233152658194071716];

W2 = [-1.5988517765338208232 -0.16840665059226397893 1.1960249915047604929 1.0095810070432040817 -1.0531083665633920177 -1.7673597189962970688 0.42097239470496095048 -1.0823575235012232731 -0.71049679067506477104 -0.33948567499553228277; -0.13827241270920290206 2.349495962654474468 0.045094895309028992791 -0.18378951488454403629 0.15468703980230497086 -0.17869340949409298624 2.5165615198939312869 0.13077189831272648113 1.5960112565758028946 -2.0562832340260865926; 0.93259017897433738042 0.33184275736528612288 -1.2280272095753033135 1.1505931988509703334 -1.2583041515052268977 1.338470017362430875 0.28976658143182559835 -1.2498501187930755041 -0.33416988500927807815 0.45765032062742821983];

На рис. 4, 6 представлены результаты НС с самонастройкой коэффициентов ПИД-регулятора; на рис. 5, 7 – отработка движения суставов по заданным

траекториям с маленькой ошибкой при воздействии управляющего воздействия на экзоскелетон (рис. 8).

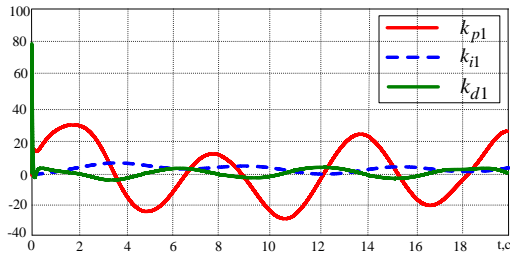


Рис. 4. НС с самонастройкой ПИД-регулятора тазобедренного сустава

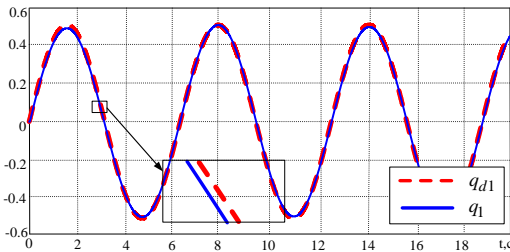


Рис. 5. Отработка движения по заданной траектории тазобедренного сустава

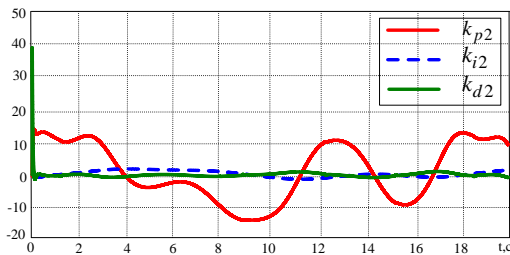


Рис. 6. НС с самонастройкой ПИД регулятора коленного сустава

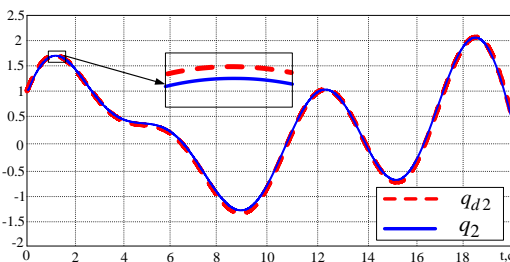


Рис. 7. Отработка движения траектории коленного сустава

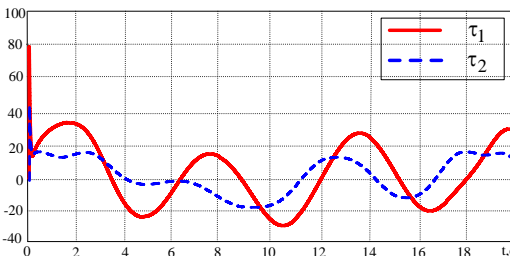


Рис. 8. Управляющее воздействие на экзоскелетон

## V. ЗАКЛЮЧЕНИЕ

Представленный нейронный регулятор позволяет осуществлять адаптивное управление, оптимизация параметров выполняется с помощью алгоритма обратного распространения НС. Он устойчив к возмущениям, включая неструктурированную, немоделируемую динамику. Аналоговая НС может осуществлять управление в реальном времени благодаря возможности быстрого онлайн-обучения с непрерывным алгоритмом, используя только ввод и вывод объекта для адаптации параметров управления, непрерывно настраивая их. Результаты показывают, что предложенный нелинейный ПИД-регулятор с использованием НС является одним из эффективных для применения в управлении экзоскелетом, удобного для человека.

## СПИСОК ЛИТЕРАТУРЫ

- [1] S. L. Dai, C. Wang, M. Wang, Dynamic Learning From Adaptive Neural Network Control of a Class of Nonaffine Nonlinear Systems // IEEE Trans. on Neural Network 25 (1) (2014) 111-123.
- [2] Z. Li, C. Y. Su, A. Xue, Development and Learning Control of a Human Limb With a Rehabilitation Exoskeleton // IEEE Transactions on Industrial Electronics 61 (7) (2014) 3776-3785
- [3] S. Balasubramaniana and J. Heb, Adaptive control of a wearable exoskeleton for upper-extremity neurorehabilitation // Applied Bionics and Biomechanics 9 (2012) 99-115.
- [4] R. Fierro, F.L. Lewis, Control of a nonholonomic mobile robot using neural networks, IEEE Trans. Neural Networks 9 (1998) 589-600
- [5] T.D.C. Thanh, K.K. Ahn, Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network // Mechatronics 16 (2006) 577-587.
- [6] Bianchini M, Frasconi P, Gori M. Learning without local minima in radial basis function networks // IEEE Trans Neural Networks 1995;6:749-56.
- [7] Park J, Sandberg IW. Universal approximation using radial basis functions network // Neural Comput 1991;3:246-57.
- [8] Broomhead DS, Lowe D. Multi-variable functional interpolation and adaptive networks // Complex Syst 1988;2:321-55.
- [9] J. Ye, Y.P. Zhao, Application of an analog compound orthogonal neural network in robot control, in: W. Shi, S.X. Yang (Eds.) // Proceedings of the International Conference on Sensing, Computing and Automation, Chongqing, China, Watam Press, Canada, 2006, pp. 455-458.
- [10] Yamada T, Yabuta T. Neural network controller using autotuning method for nonlinear functions // IEEE Trans Neural Networks 1992;3:595-601

# Анализ весовых коэффициентов нейронной сети прямого распространения как метод оценки модели

Т. Р. Жангиров<sup>1</sup>, А. Б. Кревчик<sup>2</sup>, Д. В. Зуев<sup>3</sup>,  
А. А. Лисс<sup>4</sup>, А. В. Экало<sup>5</sup>

Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)

<sup>1</sup>xxxxtestxxx@yandex.ru, <sup>2</sup>angelina151612@gmail.com,  
<sup>3</sup>danilzuev2144@gmail.com, <sup>4</sup>annaliss@mail.ru,  
<sup>5</sup>ekalo@nicetu.spb.ru

Н. Ю. Григорьева

Федеральный исследовательский центр  
Санкт-Петербурга  
renes3@mail.ru

**Аннотация.** Одним из основных параметров, влияющих на обучения нейронных сетей прямого распространения и итоговое качество модели, является инициализация весовых коэффициентов. В работе приведены различные методы анализа матриц весовых коэффициентов, позволяющие оценить качество начальной инициализации весовых коэффициентов. Рассматриваемые методы позволяют упростить подбор параметров начальной инициализации весовых коэффициентов и повысить эффективность обучения нейронных сетей. Все результаты приведены в рамках решения задачи регрессии.

**Ключевые слова:** нейронная сеть; инициализация весовых коэффициентов; машинное обучение

## I. ВВЕДЕНИЕ

Одним из свойств нейронных сетей является способность к обучению. Процесс обучения представляет собой последовательное изменение весовых коэффициентов с целью аппроксимации искомой нелинейной функции. Начальные веса обычно задаются нормальным распределением, однако существует ряд моделей, которые имеют трудности с обучением при такой инициализации. К ним относятся глубокие нейронные сети [1], нейронные сети с небольшим набором обучающих данных и другие. В первом случае проблема решается с использованием предобучением без учителя, однако это требует дополнительного времени, а также может привести к тому, что модель «застрянет» в локальном минимуме. В случае небольшого количества обучающих данных решением проблемы является обучение большого числа моделей с целью достижения требуемых значений метрик качества обучения. Глорот и Бенжио предложили свое распределение для инициализации весов [2], однако оно подходит только для сетей, функции активации которых линейны. Существуют исследования, показывающие, что VGGNet удастся обучить за одну оптимизацию благодаря правильно подобранной инициализации весовых коэффициентов. Всё это говорит о том, что инициализация весовых коэффициентов может быть существенна для обучения нейронной сети. В данной

работе представлены некоторые способы анализа весовых коэффициентов нейронной сети.

## II. ОПИСАНИЕ МОДЕЛЕЙ

Архитектура нейронной сети взята из [4]. Данная нейронная сеть была спроектирована для решения задачи регрессии. Сеть представлена на рис. 1, где  $x$  – входные признаки,  $\omega$  – весовые коэффициенты нейронов,  $s$  – состояние нейронов,  $y$  – выход сети,  $b$  – нейрон смещения). Количество нейронов на слоях следующее: входной слой – 10 нейронов, скрытые слои – 9 и 5 нейронов, соответственно, и 1 нейрон на выходном слое.

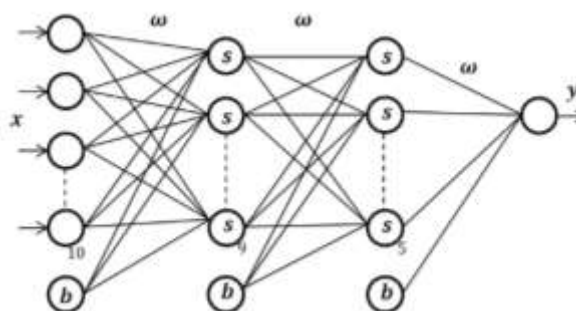


Рис. 1. Архитектура нейронной сети

Данные для обучения получены на трех научных станциях в южном и среднем коленах Кольского залива Баренцева моря [5–7] и представляют собой 10 признаков физико-химического состава среды. Данные разделены на две выборки: обучающую и тестовую. Наборы данных с первой и третьей станции представляют собой обучающую выборку, со второй – тестовую.

В качестве метрики для оценки качества обучения использовался коэффициент детерминации ( $R^2$ ), определяемый формулой:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

где  $SS_{res} = \sum_{i=1}^n (y_i - \hat{y})^2$ ,  $SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$ ,  
 $\bar{y} = \sum_{i=1}^n y_i / n$ ,  $y_i$  - фактическое значение величины,  $\hat{y}$  -  
предсказанное значение,  $n$  - количество наблюдений.

### III. МЕТОДЫ АНАЛИЗА ВЕСОВЫХ КОЭФФИЦИЕНТОВ

#### A. Анализ матрицы весовых коэффициентов

Анализировать матрицу весов можно непосредственно рассматривая значения всех весовых коэффициентов. Удобным для восприятия способом представления является тепловая карта. Она представляет собой матрицу, значения в которой представлены в виде цветов. На рис. 2 представлена матрица весовых коэффициентов между входным и скрытым слоем описанной выше нейронной сети, полученная на эпохе, где значение коэффициента детерминации на тестовой выборке было максимальным.

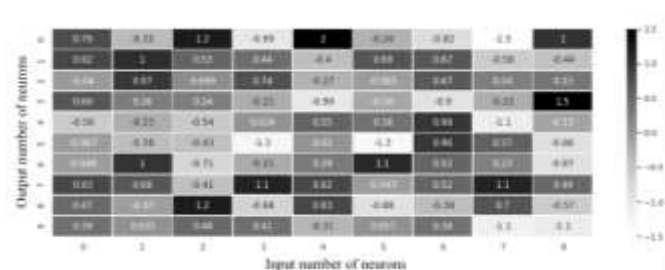


Рис. 2. Тепловая карта матрицы весовых коэффициентов

При анализе тепловой карты можно визуально оценить диапазон значений в матрице весовых коэффициентов, а также количество положительных или отрицательных значений. Например, если функция активации на данном слое ReLU, а отрицательных значений в матрице много, сеть будет давать неудовлетворительные результаты, так как большинство сигналов примут нулевое значение. Данный анализ помогает определить параметры для инициализации новой весовой матрицы. К примеру, если сеть дает неудовлетворительные результаты, и при этом в ней большое количество отрицательных весов, то можно изменить распределение (усечь, сдвинуть среднее и т. д.) таким образом, чтобы положительных значений в матрице весовых коэффициентов стало больше, а затем оценить полученные результаты.

#### B. Анализ гистограммы распределения весовых коэффициентов

Данный метод заключается в построении гистограммы распределения значений матрицы весовых коэффициентов. На рис. 3 представлен пример построения гистограммы для матрицы весовых коэффициентов.

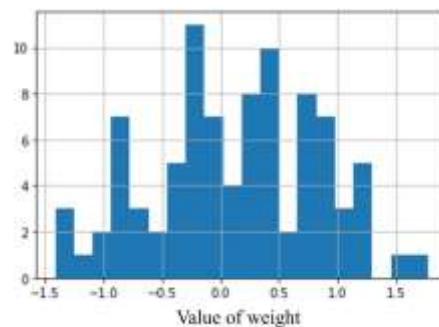


Рис. 3. Гистограмма матрицы весовых коэффициентов

С помощью данных матриц можно сравнить начальное и конечное распределение весов и визуально проанализировать изменение распределения: куда сдвинулось, изменение ширины и вида. Также это позволяет визуально проконтролировать правильность заданного изначально распределения. В некоторых функциях инициализации, использующих распределение случайной величины, существует проблема – при генерации матриц небольшого размера распределение полученных величин не соответствует заданному. При обучении сетей большим количеством обучающих данных весовые коэффициенты смогут подстроиться под нужное распределение, однако в некоторых научных областях, например, в биологии, собрать большое количество данных довольно затруднительно, поэтому требуется корректная генерация значений весов.

#### C. Анализ статистических характеристик

Имея некоторое количество моделей, можно составить таблицу со статистическими характеристиками весовых коэффициентов для каждой. К таким характеристикам можно отнести среднее значение, среднеквадратичное отклонение, минимум и максимум, медиана, мода и другие. При анализе такой таблицы, может оказаться, что для всех моделей с высоким коэффициентом детерминации на тестовой выборке характерно увеличение среднеквадратичного отклонения в одной из матриц весовых коэффициентов, то есть распределение стремится расширяться, и т. д.

#### D. Анализ начальных и конечных параметров распределения весов

Еще одним способом анализа матриц весовых коэффициентов является рассмотрение изменения параметров распределения относительно начального. Для такого анализа обучают некоторое количество моделей, а затем отбираются модели, схожие по оценкам качества обучения (рис. 4).

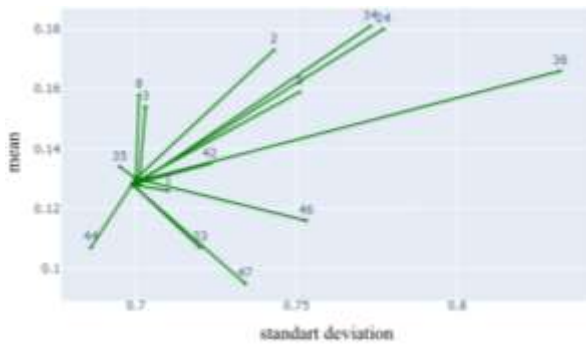


Рис. 4. Изменение статистических характеристик матрицы весовых коэффициентов после обучения

На рис. 3 изображены модели с близкими коэффициентами детерминации на обучающей и тестовой выборках. Общая точка на графике представляет собой начальное распределение, а треугольники на концах линий – конечные. На таком графике также можно отследить тенденции в изменении распределения весов и сделать вывод о том, как следует изменить распределение при инициализации весов, для получения требуемого результата.

### Е. Линейная интерполяция

Метод линейной интерполяции позволяет оценить значения функции между двумя известными значениями. Данный метод позволяет предсказать промежуточные значения в некотором диапазоне, которые вычисляются по формуле:

$$W(\alpha) = (1 - \alpha)W + \alpha W'$$

где  $W$  и  $W'$  - весовые коэффициенты рассматриваемых моделей. После получения взвешенного среднего  $W(\alpha)$  значений весовых коэффициентов двух моделей вычисляется значение функции потерь и строится ее график (рис. 5). Данный метод анализа весовых коэффициентов использовался в статье [8]. Линейная интерполяция позволяет оценить устойчивость минимума, которая была достигнута моделью, к изменению весовых коэффициентов, а также может быть использована для поиска потенциальных минимумов. На рис. 5–6 представлены график функции  $f(\alpha) = L(W(\alpha))$ , где  $L$  - функция потерь.

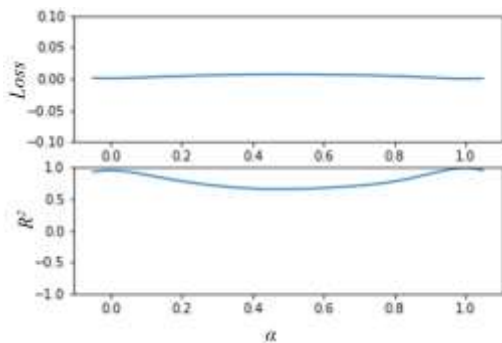


Рис. 5. График функции потерь и коэффициента детерминации для моделей, устойчивых к изменениям весовых коэффициентов

На рис. 5 представлен график функции ошибки и коэффициента детерминации для взвешенного среднего между двумя моделями. Первая модель находится в точке  $\alpha = 0$ , вторая – в точке  $\alpha = 1$ . Параметр изменяется с шагом 0.001. Можно сказать, что обе модели достаточно устойчивы к изменениям весовых коэффициентов, следовательно, можно предположить, что они находятся в одном локальном минимуме. По результатам, представленным на рис. 6, можно сказать, что малейшее изменение весовых коэффициентов модели, которая находится в точке  $\alpha = 0$ , приводит к тому, что значение ошибки резко возрастает. Из этого можно сделать вывод, что минимум неустойчив.

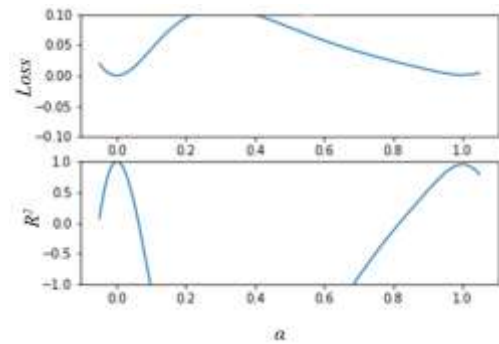


Рис. 6. График функции потерь и коэффициента детерминации для моделей, неустойчивых к изменениям весовых коэффициентов

### Ф. Построение контурной карты

Следующий метод позволяет построить двумерное пространство в базисах  $\delta$  и  $\eta$  для некоторой модели. Для построения контуров рассчитывается функция двух параметров:

$$f(\alpha, \beta) = L(\Theta^* + \alpha \cdot \delta + \beta \cdot \eta)$$

где  $\alpha$  и  $\beta$  координат модели в пространстве базисов  $\delta$  и  $\eta$ ,  $L$  – функция потерь. Пример контурной карты представлен на рис. 7.

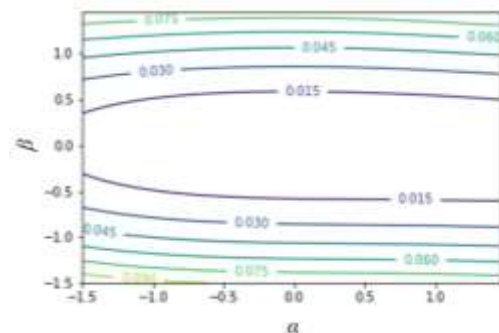


Рис. 7. Контурная карта в рамках базисов  $\delta$  и  $\eta$  для области вокруг рассматриваемой модели

На данном рисунке рассматриваемая модель находится в точке с координатами  $(0,0)$ . Расстояние между контурами невелико, что говорит о том, что функция потерь имеет крутой подъем. Подробнее данный алгоритм описан в статье [8].

#### IV. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЯ

Анализ проводился на наборе из пятидесяти обученных моделей нейронных сетей. Параметры обучения, к которым относятся функции активации, количество слоев и нейронов, и обучающие выборки, оставались неизменными, менялись только значения весов, сгенерированные по одному закону распределения. В результате обучения были получены модели как с высокими, так и с низкими коэффициентами детерминации на обучающей и тестовой выборках. Таким образом, можно сделать вывод, что результат зависит от инициализации.

Поскольку результат зависит от инициализации, можно выбрать ее на основе результатов описанных выше методов. Для этого производится кластеризация моделей по коэффициенту детерминации на обучающей и тестовой выборках (рис. 8). Далее выбирается кластер с наилучшими показателями качества обучения и исследуется. Для некоторых задач это кластер с самыми высокими значениями метрики качества обучения, для других – с примерно одинаковыми значениями. Исследуемый кластер выбирается в зависимости от задачи и ожидаемых результатов.

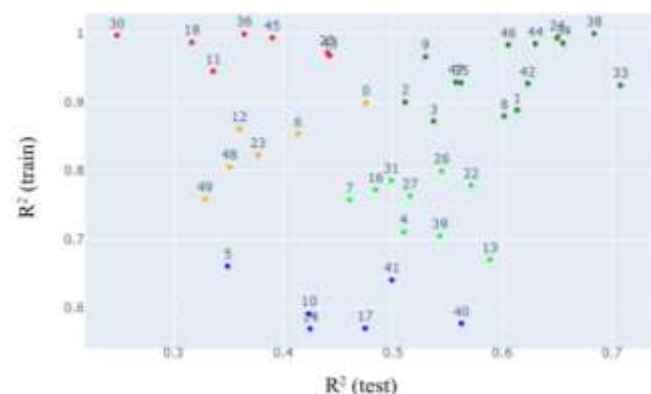


Рис. 8. Кластеризация набора моделей методом k-средних

Интересующий кластер определяется по статистически характеристикам. Далее по результатам анализа выбираются новые параметры распределений для матриц весовых коэффициентов, и получается новый набор моделей с улучшенной статистикой обучения. Для анализа используются методы, указанные выше.

#### V. ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены различные методы анализа матриц весовых коэффициентов нейронной сети прямого распространения. Приведенные методы позволяют упростить подбор параметров инициализации весовых коэффициентов и оценить устойчивость обученных моделей к слабым изменениям весов. Методы были проверены на наборе из 50 нейронных сетей.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Simonyan K. and Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.
- [2] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks // In International Conference on Artificial Intelligence and Statistics, pages 249–256, 2010.
- [3] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification // In International Conference on Computer Vision (ICCV), 2015.
- [4] Grigoryeva N.Y., Zhangiroy T.R., Liss A.A. Specific features of feed-forward neural networks application in classification and regression problems in algology // Journal of Physics: Conference Series. IOP Publishing, 2020. T. 1703. № 1. С. 012032.
- [5] Мирошниченко Е.С. Бактериоценозы эпилитона литорали южного и среднего колен Кольского залива Баренцева моря [Текст]: дис. .... канд. биол. наук / Мурман. гос. тех. университет. Мурманск, 2016. 176 с.
- [6] Бокова Е.Н. Количественное распределение бактерий в Баренцевом и Карском моря // Тр. ВНИРО. 1937. Т.4. №1. С. 373–377.
- [7] Макаревич П.П. Первичная продукция Баренцева моря // Вестник Мурманского государственного технического университета. 2012. Т. 15. № 4. С. 786-793.
- [8] Li H. et al. Visualizing the loss landscape of neural nets //arXiv preprint arXiv:1712.09913. 2017.

# XAI Evaluation: Evaluating Black-Box Model Explanations for Prediction

Yuyi Zhang

*Faculty of Applied Mathematics and Control Processes  
Saint-Petersburg State University  
Saint Petersburg, Russia  
Lesliezhang0825@gmail.com*

Feiran Xu

*Faculty of Applied Mathematics and Control Processes  
Saint-Petersburg State University  
Saint Petersburg, Russia  
xfr97120@gmail.com*

Jingying Zou

*Saint-Petersburg State University;  
General Development Department  
Huawei Technologies Co. Ltd.  
Saint Petersburg, Russia  
jesco4482@gmail.com*

Ovanes L. Petrosian

*Saint-Petersburg Electrotechnical University "LETI";  
Saint-Petersburg State University;  
General Development Department  
Huawei Technologies Co. Ltd.  
Saint Petersburg, Russia  
petrosian.ovanes@yandex.ru*

Kirill V. Krinkin

*Saint-Petersburg Electrotechnical University "LETI"  
Saint Petersburg, Russia*

**Abstract**—The results of evaluating explanations of the black-box model for prediction are presented. The XAI evaluation is realized through the different principles and characteristics between black-box model explanations and XAI labels. In the field of high-dimensional prediction, the black-box model represented by neural network and ensemble models can predict complex data sets more accurately than traditional linear regression and white-box models such as the decision tree model. However, an unexplainable characteristic not only hinders developers from debugging but also causes users mistrust. In the XAI field dedicated to “opening” the black box model, effective evaluation methods are still being developed. Within the established XAI evaluation framework (MDMC) in this paper, explanation methods for the prediction can be effectively tested, and the identified explanation method with relatively higher quality can improve the accuracy, transparency, and reliability of prediction.

**Keywords**—XAI evaluation, black-box model explanations, neural network, ensemble models

## I. INTRODUCTION

The XAI evaluation framework has been paid more and more attention under the condition that the XAI methods have been widely accepted. At present, the interpretability of the model is obviously focused on the field of ML and DL, especially when their models are used in medical, financial, military and other fields that require high transparency and trust, the interpretability of the model is even necessary. However, black-box models such as neural networks [1] and ensemble models [2] are not inherently interpretable. Therefore, the XAI method is absolutely needed to promote the more efficient application of black-box models with higher accuracy in real life. The XAI evaluation framework can select XAI methods that are more suitable for the model, which further improves the credibility of the black-box model.

The black-box models used for prediction mainly include neural network models and ensemble models. Among them,

the ANN [3] model is used as a representative of the neural network model, LightGBM [4] and random forest [5] are used as the representative of the ensemble model, LightGBM is a model based on the boosting algorithm in the ensemble model, and the random forest is based on the bagging algorithm. Both boosting and bagging [6] are composed of many simple tree models. If there is no connection between the tree models, and the final output result through “voting” is the bagging algorithm. If there is a strong connection between these tree models, a boosting algorithm is formed, that is, the output of the previous tree model will affect the next tree model. In addition, linear regression and white-box model-decision trees are also used in prediction tasks for comparison.

SHAP [7] and LIME [8] are used as the method to explain the model. On the one hand, both belong to the feature attribution explanation. The same working logic ensures that they are comparable and the validity of the evaluation framework we propose. On the other hand, considering the wide application of them in tabular data, they are representative. At the same time, random disturbances are used as the baseline for comparison, thus objectively showing the effectiveness of the XAI method. In order to establish the final XAI evaluation framework, in this paper, the main work includes:

- Comparison of accuracy measures of black box models (ANN, LightGBM and random forest) with linear regression and decision trees;
- Use XAI method to explain the black-box model;
- Create a new XAI evaluation framework, and demonstrate the principles of the framework through visualization technology, and finally apply the framework in the form of an evaluation matrix to select the most suitable XAI method.



## II. BLACK-BOX MODEL PREDICTIONS AND EXPLANATIONS

### A. Dataset Description

The data set used for prediction – Boston housing from “sklearn”, the original data comes from the U.S. Census Bureau. The data set includes 506 rows of instances, 13 features, and 1 predicted variable. The data set information can be viewed at the link:

<https://www.kaggle.com/jamieleeche/boston-housing-dataset>

### B. Black-box Model Predictions

Neural network and ensemble models are typical representatives of black-box model. Among them, artificial neural networks (ANN) [3] and ensemble models such as Random Forest (RF) [5] and LightGBM [4] have excellent performance for prediction.

ANN is an adaptive nonlinear dynamic system composed of a large number of simple basic units (neurons) interconnected by activation functions. The structure and function of each neuron are relatively simple, however, system behaviors produced by the combination of neurons are extremely complicated and difficult to explain. ANN is divided into three layers: input, hidden, and output. The ANN architecture we built is shown in Table 1.

TABLE I. ANN ARCHITECTURE

Layer (type)	Output shape	Activation
Dense	(None, 128)	ReLU
Dense	(None, 64)	ReLU
Dense (output)	(None, 1)	-

GBDT[9] is an extremely important model in the field of machine learning. Its main idea is to use weak classifiers (decision trees) to iteratively train to obtain the optimal model. This model has the advantages of excellent training effect and avoiding overfitting. LightGBM [4] (Light Gradient Boosting Machine) is a framework that implements the GBDT algorithm, which has the advantages of faster training speed, lower memory consumption, and better accuracy.

In a random forest, multiple decision trees are built, and they are merged together to obtain more accurate and stable predictions. The random forest classifier uses all the decision tree classifiers and the hyperparameters of the bagging classifier to control the overall structure. It should be noted that regression also has a random forest regressor.

In order to be able to visually see the advantages of the black-box model in prediction, the traditional linear regression (LR) and the classic white-box prediction model – decision tree (DT) are used as a control group to predict the Boston housing data set together with ANN, LightGBM and RF. The quality statistics of the prediction results are shown in Table 2.

TABLE II. THE QUALITY STATISTICS

Model	Metrics		
	$R^2$	MSE	MAE
LR	0.64856	28.40585	3.69136
DT	0.74361	20.72322	3.05065
ANN	<b>0.79477</b>	<b>16.58769</b>	<b>2.57977</b>
LGBM	<b>0.80417</b>	<b>15.82852</b>	<b>2.53292</b>
RF	<b>0.81888</b>	<b>14.63918</b>	<b>2.35044</b>

Metrics shows that ANN, LightGBM and RF have almost absolute advantages in prediction tasks. However, compared with linear regression and decision trees, the prediction process of ANN, LightGBM and RF is incomprehensible by humans due to their internal complex structure. As a result, black-box models with higher accuracy cannot completely replace traditional models and white-box models with lower accuracy in many fields. Therefore, the development of effective methods to explain the black-box model is crucial.

### C. The Meaning and Methods of XAI

Before starting to use the XAI method, the target task of model interpretation needs to be clarified, that is, the realization of the target of model interpretability. The interpretability of the model is by no means the full presentation of the internal working process of the black-box model, which is almost impossible to achieve and meaningless. The meaning of model interpretability is to allow the model to be safely applied to the corresponding field. In other words, model interpretability is also a tool for business guidance and case interpretation. Therefore, the meaning of model interpretability can be defined as:

- Business guidance: the identification of the most important features in the model to guide the business.
- Explain the case: Explain which feature is good or bad for a specific case.

SHapley Additive exPlanation (SHAP) can interpret complex machine learning models. Although it comes from game theory, it only uses this idea as a carrier.

- SHapley: Represents calculating the Shapley Value of each feature variable in each sample.
- Additive: Represents that the shapley value corresponding to the feature variable is additive for each sample.
- exPlanation: Represents the interpretation of a single sample, that is, how each feature variable affects the predicted value of the model.

The main idea of LIME is to use interpretability models (such as linear models, decision trees) to locally approximate the prediction of the target black box model. This method does not go deep into the model. By slightly perturbing the input, it detects what happens to the output of the black box model. According to this change, an interpretability model is trained at the original input point.

### D. XAI: Explanation for Black-box Model Predictions

In order to clearly show the XAI method's explanation of the black-box model prediction, take SHAP as an example to explain the prediction results locally and globally. SHAP's global explanation of LightGBM prediction is shown in Fig. 1. The 13 features are arranged in order according to the degree of influence on the prediction result. Taking the Boston housing data set as an example, according to the results of the global explanation, the government can more clearly understand the reasons that affect real house prices, which is helpful for policy macro-control.

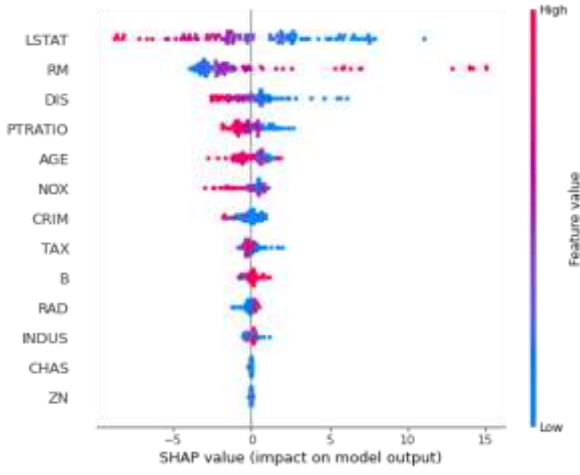


Fig. 1. SHAP global explanation of LightGBM prediction

SHAP local explanation of LightGBM prediction is shown in Fig. 2. Local explanation focuses on each instance, and outputs features that have an impact on each instance. According to local explanations, buyers can more accurately measure the target house price and make reasonable decisions.



Fig. 2. SHAP local explanation of LightGBM prediction

In the actual application of XAI, different XAI methods will output different explanations. On the one hand, this situation is due to the different principles of various XAIs. On the other hand, because XAI explanation of the predictive model is specific, in the case of the same data set but different black-box models, different explanation results will be output. In order to solve this problem, the evaluation of the XAI method is necessary for the correct explanation of the predictive model.

### III. XAI EVALUATION FRAMEWORK FOR BLACK-BOX MODEL EXPLANATIONS

The establishment of the XAI evaluation framework is based on a premise: deleting or changing the large contribution (calculated by the XAI method) features in the data set will result in a significant decrease in the accuracy of the model's prediction. Therefore, based on this premise, the degree of change in the metrics ( $R^2$ , MSE, MAE) can be used as the core of the evaluation framework.

#### A. XAI Evaluation Framework: Mean Degree of Metrics Change (MDMC)

In Perturb the original data according to the output of the XAI method, and input the modified data into the prediction model to obtain a new metrics ( $R_2^*$ ,  $MSE^*$ ,  $MAE^*$ ), then the degree of change of the metrics (D) can be defined as:

$$D = f(M - M^*) \quad (1)$$

$M$  is the original metric, and  $M^*$  is the changed metric.

Combining the degree of change of all metrics, the final evaluation framework (MDMC) can be defined as:

$$\begin{aligned} MDMC &= \frac{1}{n} \sum_{i=1}^n D = \frac{1}{n} \sum_{i=1}^n f(M - M^*) \\ &= \frac{1}{n} \sum_{i=1}^n [(R_2^0 - R_2^i) + (MSE_i - MSE_0) + (MAE_i - MAE_0)] \end{aligned} \quad (2)$$

It should be noted that the values of MSE and MAE in Equation (2) need to be used after normalization. In theory, the larger the value of MDMC, it means that the black-box prediction model has made significant changes to the data set, thus proving the effectiveness of XAI.

#### B. XAI Evaluation Framework: Visualization

In order to understand the MDMC framework intuitively, take the ANN model as an example to evaluate the XAI methods. The changes in the metrics are shown in Fig. 3. It is worth noting that if the stability of the ANN model itself is lacking, then randomly deleting features will also cause a decrease in the accuracy of the model's prediction. In order to eliminate this possibility, in the evaluation process, the data set of "randomly deleted features" was used as a comparison of the XAI methods to prove the stability of the ANN model we constructed.

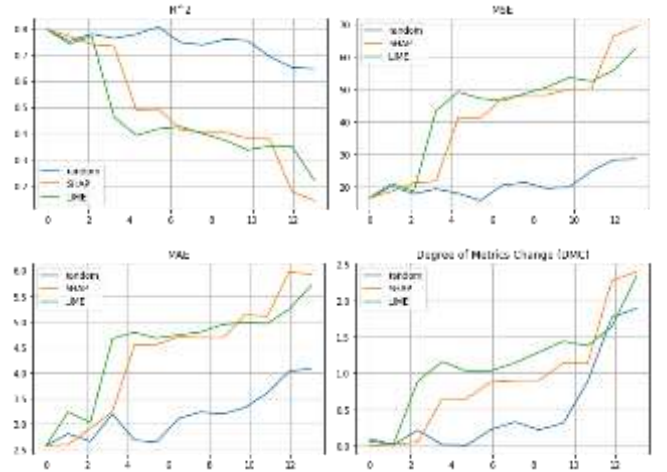


Fig. 3. Visualization of the changes in the metrics

As shown in Fig. 3, compared to randomly deleting features, the perturbation based on the results of the XAI methods significantly reduces the accuracy of the ANN prediction model.

From the metrics of view, when interpreting the ANN model, the performance of SHAP is more stable, and the trend of change is consistent; on the other hand, the performance of LIME is relatively unstable, however, from the early stage, the performance of LIME is significantly better than that of SHAP. These results mean that the LIME method is more sensitive to high-contribution features. Moreover, according to the sum of the degree of change of the three metrics (DMC), it is shown that when the three metrics are combined, LIME performs better overall.

#### C. XAI Evaluation Framework: Evaluation Matrix

Visualization can intuitively evaluate the XAI method, but for those tasks with high-precision requirements, a quantitative evaluation matrix is necessary. The process of the XAI methods on LightGBM and Random Forest is similar to that of ANN and follows the same calculation formula. The values in Table 3 are calculated based on Equation (2). The larger the value, the more effective the XAI method is.

TABLE III. EVALUATION MATRIX

MDCD	ANN	LightGBM	RandomForest
Random	0.8333	0.8222	0.6242
SHAP	1.4154	<b>1.5004</b>	1.3286
LIME	<b>1.6506</b>	1.3995	<b>1.6069</b>

The experimental results show that in the black-box model, the perturbation based on the XAI methods is more effective than the random perturbation. On the other hand, within the XAI method, LIME performs better in the ANN model and the random forest model, and SHAP is more effective for the LightGBM model.

#### IV. CONCLUSION

After comparing with the accuracy of traditional linear regression and white-box models, black box models such as neural networks and ensemble models including boosting algorithms and bagging algorithms have absolute advantages in prediction, however, the unexplainable nature makes the black-box model an obstacle in the process of practical application. The XAI methods can obviously alleviate this problem. By identifying features with relatively high contributions, users can understand the black box model more clearly when using the black box model to predict, thereby increasing trust. However, because many XAI methods have different principles and characteristics, different XAI methods output different results for the same black-box model. Therefore, the establishment of an evaluation framework for XAI methods is necessary. The XAI methods are evaluated through the established XAI

evaluation framework – MDMC. The results show that LIME is more suitable for ANN model and random forest model based on bagging algorithm, and SHAP is more suitable for LightGBM based on boosting algorithm.

#### REFERENCES

- [1] Aggarwal C.C. Neural networks and deep learning. Springer, 2018, 10: 978-3.
- [2] Xiao Z, Wang Y, Fu K, et al. Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. ISPRS International Journal of Geo-Information, 2017, 6(2): 57.
- [3] Agatonovic-Kustrin S, Beresford R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research Journal of pharmaceutical and biomedical analysis, 2000, 22(5): 717-727.
- [4] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 2017, 30: 3146-3154.
- [5] Breiman L. Random forests. Machine learning, 2001, 45(1): 5-32.
- [6] Wen L, Hughes M. Coastal wetland mapping using ensemble learning algorithms: A comparative study of bagging, boosting and stacking techniques. Remote Sensing, 2020, 12(10): 1683.
- [7] Lundberg S, Lee S I. A unified approach to interpreting model predictions. arXiv preprint arXiv:1705.07874, 2017.
- [8] Ribeiro M T, Singh S, Guestrin C. "Why should i trust you?" Explaining the predictions of any classifier[C]. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016: 1135-1144.
- [9] Friedman J H. Greedy function approximation: a gradient boosting machine. Annals of statistics, 2001: 1189-1232.

# Сравнение методов кластеризации данных в газотранспортной системе

А. К. Петрова

*Кафедра инновационного менеджмента  
Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)  
ak72p@yandex.ru*

А. М. Сеница

*Кафедра автоматики и процессов управления  
Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)  
amsinitca@etu.ru*

**Аннотация.** В статье рассматриваются подходы к решению актуальной задачи повышения эффективности управления процессами в газотранспортной системе. Для выявления закономерностей и организации данных применяются алгоритмы кластеризации, как классические (к-средних и др.), так и интеллектуальные (самоорганизующиеся карты), применение этих методов к статистическим данным газотранспортной системы рассмотрено в данной статье.

**Ключевые слова:** эффективность, данные, нейронные сети, самоорганизующиеся карты, кластеризация, газотранспортная система

## I. ВВЕДЕНИЕ

Эффективность хозяйственной деятельности является основой конкурентоспособности предприятий. Для предприятий газотранспортной системы наиболее важными факторами эффективности являются баланс, или разница между поставляемым и потребляемым объемом газа [1], а также оценка загруженности и контроль безопасности использования трубопроводов [2].

Несмотря на большое количество методик контроля баланса газа и оценки эффективности газопроводов, остается актуальной задача их совершенствования, в том числе в части, касающейся обработки и анализа статистических данных по учету газа; выявление и классификация закономерностей и факторов, влияющих на процесс учета, разработка новых алгоритмов и решений.

К настоящему времени в газораспределительных организациях накопились большие объемы статистических данных о работе оборудования, балансе природного газа, утечках и других производственных ситуациях [3]. В связи с ростом объема обрабатываемой информации и размерности задач возникает необходимость обработки, в том числе кластеризации данных.

В статье осуществляется применение методов кластеризации к статистической информации по учету расхода газа с помощью классических методов и самоорганизующихся карт (СОК) Кохонена.

## II. МЕТОДЫ КЛАСТЕРИЗАЦИИ

Кластеризация – автоматическое разбиение элементов некоторого множества на группы в зависимости от их

схожести [4]. Цель кластеризации – поиск существующих структур. Характеристиками кластера можно назвать два признака:

1. Внутренняя однородность.
2. Внешняя изолированность.

Задачи кластерного анализа можно объединить в следующие группы [4]:

1. Разработка типологии или классификации.
2. Исследование схем группирования объектов.
3. Представление гипотез.
4. Проверка гипотез.

Существует большое число методов кластеризации и их модификаций, например, алгоритмы, основанные на разделении данных, иерархические алгоритмы, методы, основанные на концентрации объектов, модельные методы, и др. [4].

В данной статье рассматривается анализ данных по учету газа за определенный период времени, то есть необходимо решить задачу кластеризации временных рядов. Анализируя данные такого вида, можно исследовать: ряды, похожие во времени, где особые точки и интервалы возрастания точно или почти точно соответствуют друг другу во времени; ряды, похожие по форме, имеющие одинаковые характерные особенности; и ряды, похожие по структуре, с одинаковыми законами изменения [5].

## III. КЛАСТЕРИЗАЦИЯ ДАННЫХ ГАЗОТРАНСПОРТНОЙ СИСТЕМЫ

Необходимо провести кластеризацию накопленной за 7 месяцев статистической информации по объемному расходу газа  $Q, \text{ м}^3$  в стандартных условиях на 181 газораспределительной станции (ГРС), на предмет выявления однородных несхожих между собой групп и поиска закономерностей. Для этого предварительно осуществлена обработка полученного массива данных размерностью  $181 \times 212$ , удалены выбросы, произведена нормализация на диапазон допустимых значений. Параметры: объемный расход газа  $Q, \text{ м}^3$ , время  $t$ , дни. Получены результаты применения некоторых

классических подходов (K-means (рис. 1), Gmdistribution (рис. 2), Linkage (рис. 3)) к кластеризации. На указанных рисунках на оси абсцисс расход газа, ординат – время. Расчеты проведены в среде Matlab [6].

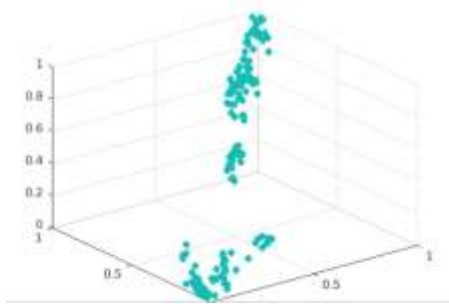


Рис. 1. Кластеризация методом Kmeans. Число кластеров 1

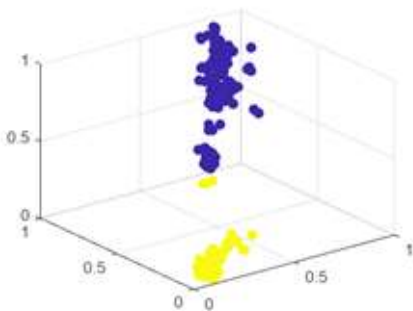


Рис. 2. Кластеризация методом Gmdistribution. Число кластеров 2

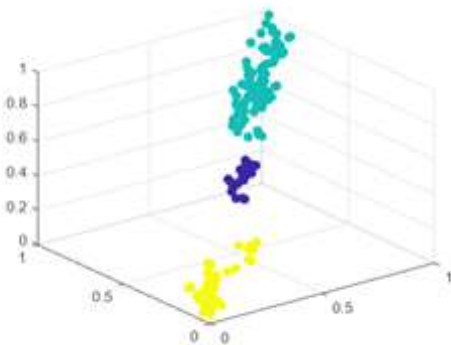


Рис. 3. Рис. 3. Кластеризация методом Linkage. Число кластеров 3

Проверка точности моделей на основе данных, не используемых в процессе настройки, показала, что некоторая часть новых данных не попала в выявленные кластеры.

Для проверки гипотезы о наличии в выборке трех кластеров были применены некоторые наиболее распространенные алгоритмы машинного обучения [7]. Результаты обучения показали, что разбиение данных на три кластера достаточно точно их характеризуют (рис. 4, на оси абсцисс расход газа, ординат – время).

С целью сравнения соответствия предсказаний и значений тестовой выборки построены матрицы спутанности (confusion matrix). Матрица идеального решения должна иметь диагональный вид. Матрица

полученного решения является диагональной, однако алгоритм SVM (Support Vector Machine, метод опорных Векторов, рис. 5), при котором происходит перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве [7, 13], показал более точный результат, чем алгоритм Ансамбль (Ensemble, рис. 6), который состоит из нескольких базовых алгоритмов машинного обучения [7].

**Визуальное представление кластеров**

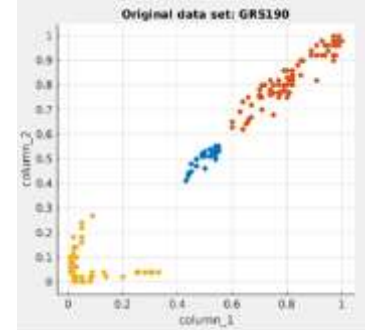


Рис. 4. Визуализация результатов кластеризации методом SVM

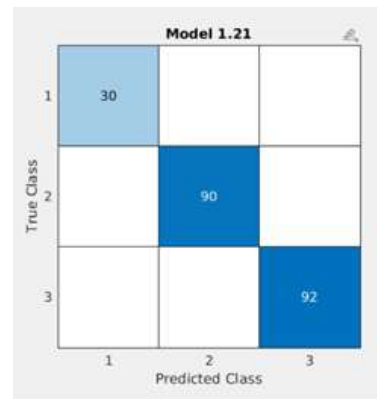


Рис. 5. Матрица спутанности (Confusion matrix) результатов обучения методом SVM

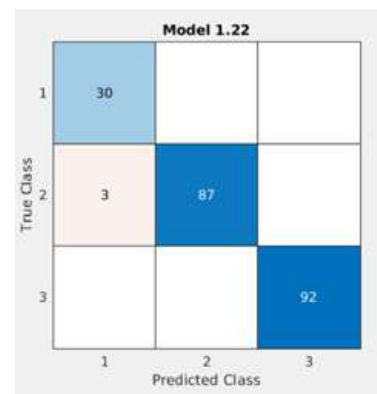


Рис. 6. Матрица спутанности (Confusion matrix) результатов обучения методом Ensemble

Несмотря на достаточно точные результаты, полученные с применением классических методов кластеризации и машинного обучения, поскольку задача поиска закономерностей в исследуемых

несбалансированных и нелинейных данных газотранспортных систем относительно нова, что затрудняло процесс принятия решений о количестве кластеров. было рассмотрено применение к ним двухслойной нейронной сети [8] и самоорганизующихся карт (СОК) Кохонена [9].

Для нейронной сети в качестве обучающей выборки были выбраны имеющиеся статистические данные в соответствии с тремя классами, полученными ранее при помощи классических методов кластеризации и подтверждёнными методами машинного обучения. Для обучения двухслойной нейросети было рассмотрено различное количество нейронов в скрытом слое (5, 10, 50). Минимальная ошибка получена при количестве нейронов в скрытом слое, равном 5.

В сети Кохонена обучающее множество состоит лишь из значений входных переменных, в процессе обучения нет сравнения выходов нейронов с эталонными значениями [9]. В процессе последовательной подачи на вход сети обучающих примеров определяется нейрон, у которого скалярное произведение весов и поданного на вход вектора минимально. Этот нейрон объявляется победителем и является центром при подстройке весов у соседних нейронов. Обучение при этом заключается не в минимизации ошибки, а в подстройке весов (внутренних параметров нейронной сети) для наибольшего совпадения с входными данными [10].

Процесс обучения сети Кохонена:

1. Инициализация весов.
2. Случайный выбор входной переменной.
3. Выбор нейрона-победителя.
4. Обновление весов.
5. Возврат к шагу 2 до момента достижения требуемой точности или минимизации ошибки [9].

Для обучения сети Кохонена в данной задаче было рассмотрено различное количество нейронов (2, 3, 5).

Результаты применения: ошибка обучения двухслойной нейросети с 5 нейронами в скрытом слое представлены на рис. 7.

На рис. 8 представлена «карта совпадений» («hits map»), получившаяся в результате применения сети Кохонена с двумя нейронами и показывающая, сколько раз каждый нейрон был «победителем» для каждого входного значения, что дает представление о количестве данных, которые собираются в каждом кластере и позволяет оценить его важность.

На рис. 9 отображено пространство входных векторов (в виде светлых точек) на каждый нейрон (темные точки), соседние нейроны соединены линиями. Результаты выявили возможность сгруппировать данные на 4 группы, что отличается от полученных ранее результатов классических методов.

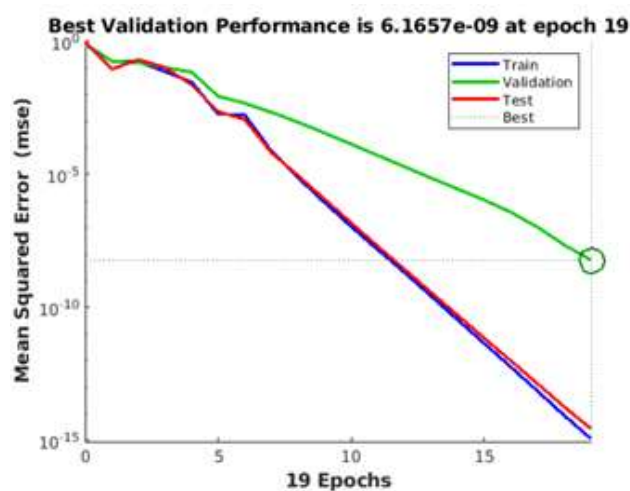


Рис. 7. Ошибка обучения двухслойной нейросети с 5 нейронами в скрытом слое

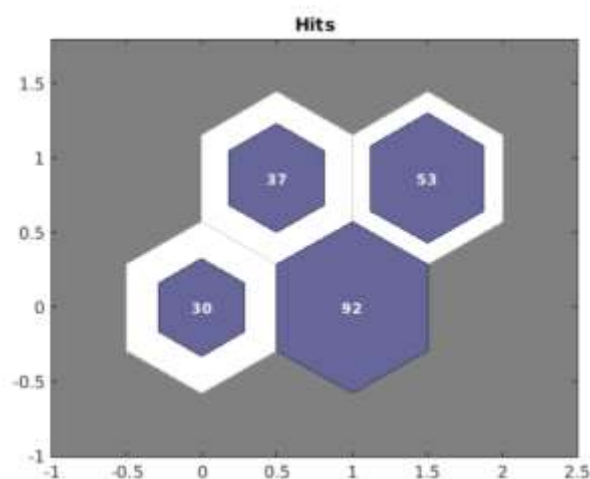


Рис. 8. Карта совпадений результатов обучения СОК

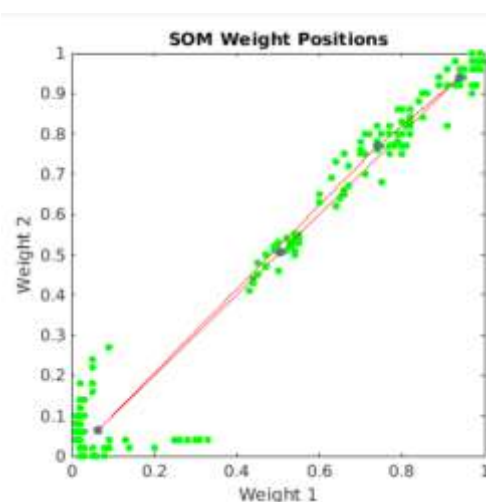


Рис. 9. Отображение входных векторов (светлые точки) на веса нейронов (темные точки)

Предварительное консультирование с экспертами газотранспортного предприятия позволило предположить целесообразность рассмотрения возможности сравнения эффективности разбиения данных как на 3, так и на 4 кластера. В частности, это позволит с большей точностью дифференцировать ГРС по степени их загруженности в том числе с учетом сезонности, например, если расход составляет 80–90 % от максимальной загрузки – ГРС оптимально загружена, а если 10–20 %, то имеет место недогрузка [11].

Дифференциация позволит принимать более взвешенные решения по использованию различных, соответствующих каждому уровню, алгоритмов выбора средств измерений, методик учета расхода и контроля баланса газа, оптимизировать эксплуатационные расходы [12].

#### IV. ЗАКЛЮЧЕНИЕ

В рамках работы проведено сравнение алгоритмов кластеризации для анализа данных газотранспортной системы. Полученные результаты показали эффективность кластеризации по помощи самоорганизующихся карт Кохонена в сравнении с другими методами. Данные исследования могут быть использованы для дальнейшего анализа и принятия решений, повышающих эффективность газотранспортной системы.

#### СПИСОК ЛИТЕРАТУРЫ

[1] Хворов Г.А., Козлов С.И., Аколова Г.С., Евстифеев А.А. Сокращение потерь природного газа при транспортировке по магистральным газопроводам ОАО «Газпром» // Газовая промышленность. 2013. № 12. С. 66–69.

- [2] Годовой отчет о деятельности федеральной службы по экологическому, технологическому и атомному надзору в 2016 году. Федеральная служба по экологическому, технологическому и атомному надзору: М. 2017.
- [3] Густов С.В., Воробьев Н.Ю., Пахомов С.Н., Панарин М.В. Технологии искусственного интеллекта для повышения эффективности эксплуатации сетей газо распределения и газопотребления // Газовая Промышленность Спецвыпуск № 3 | 789 | 2019.
- [4] Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к знаниям. М.: Издательство «Итер», 2013. 704 с.
- [5] Сообщество ИТ специалистов [Электронный ресурс] Режим доступа: // <https://habr.com/ru/post/334220/> (дата обращения 15.02.2021)
- [6] Мартынов Н.Н. Введение в MATLAB 6 / Н. Н. Мартынов М.: КУДИЦ-ОБРАЗ, 2002. 352 с.
- [7] Назаров А.В., Лоскутов А.А. Нейросетевые алгоритмы прогнозирования и оптимизации систем. СПб: Наука и Техника, 2003, 384 с.
- [8] Хайкин С. Нейронные сети. Полный курс. Вильямс, 2018, 1104 с.
- [9] Кохонен Т. Самоорганизующиеся карты. М.: Бином. Лаборатория знаний, 2008. 655 с.
- [10] Национальный открытый университет Интуит [Электронный ресурс] Режим доступа: // <https://intuit.ru/studies/courses/6/6/lecture/180?page=3> (дата обращения 15.02.2021)
- [11] СТО ГАЗПРОМ 2-3.5-051-2006 Нормы технологического проектирования магистральных газопроводов
- [12] СТО Газпром 097-2011 Автоматизация. Телемеханизация. Автоматизированные системы управления технологическими процессами добычи, транспортировки и подземного хранения газа. Основные положения.
- [13] Gang Xie, Shouyang Wang, Kin Keung Lai, Short-term forecasting of air passenger by using hybrid seasonal decomposition and least squares support vector regression approaches // J. Air Transp. Manag. 37 (2014) 20–26.

# Использование OpenMP для оптимизации процесса обучения модели в алгоритмах машинного обучения

Omar T. Mohammed<sup>1</sup>, Moeid S. Heidari<sup>2</sup>, А. А. Пазников<sup>3</sup>

*Кафедра информатики и инженерии  
Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)*

<sup>1</sup>omar.taha.mohammed@gmail.com, <sup>2</sup>moeidheidari@mail.ru, <sup>3</sup>apaznikov@gmail.com

**Аннотация.** Очевидно, что цифровой мир продолжает расширяться, в то время как создание решений для больших данных часто зависит от своевременного извлечения ценной информации из данных. Это будет по-прежнему усложняться из-за увеличения объема данных без соответствующего увеличения скорости. Следовательно, он может оптимизировать пространство для хранения, уменьшить объем обработки, необходимой для дальнейшего извлечения информации, а также сэкономить ввод-вывод и сетевую связь. В частности, в области искусственного интеллекта реализация алгоритмов машинного обучения для больших данных, таких как социальные сети и веб-графы, является сложной задачей. Несмотря на то, что многие виды исследований сосредоточены на том, чтобы сделать последовательные алгоритмы более масштабируемыми, тем не менее, время их выполнения остается чрезмерно большим. Становится необходимым разделить работу между несколькими потоками. В этой статье мы использовали возможности параллельных вычислений, в частности, реализацию OpenMP в контролируемых процессах машинного обучения, чтобы сократить время вычислений. Был проведен эксперимент по задаче двоичной классификации, который был обучен на наборе реалистичных данных, и он показал значительное сокращение времени вычислений по сравнению с классическими алгоритмами. В конце мы описали возможные направления дальнейших исследований, касающихся параллельной оптимизации расчета времени в контролируемых процессах обучения перцептронов.

**Ключевые слова:** параллельные вычисления, машинное обучение, нейронные сети, OpenMP

## I. ВСТУПЛЕНИЕ

Последние десять лет показали, что машинное обучение становится будущим для обработки данных, а технологии ЦП достигают своего пика. Это может быть подтверждено известным утверждением закона Мура, однако он уже не является точным, и, с другой стороны, квантовые вычисления все еще далеко не зависят от решения реальных проблем [1]. Следовательно, существует потребность найти способ, который сокращает время, необходимое для обучения и тестирования моделей машинного обучения. Мы считаем, что именно здесь нам может помочь параллельная обработка. Стоит отметить, что мас-

штабируемость моделей машинного обучения с данными по-прежнему представляет собой серьезную проблему, и поэтому в игру вступает множество областей распараллеливания, чтобы увеличить производительность этих моделей с точки зрения времени выполнения [2]. Когда разработка графического процессора продвинулась вперед, исследования стали зависеть от графического процессора для ускорения алгоритмов нейронной сети, поскольку скорость выполнения стала как минимум в 10 раз выше, чем за тот же период, по сравнению с центральным процессором CPU [3]–[5]. GPU содержит большую общую память, которая составляет до 24 гигабайт, и тысячи потоковых процессоров, подходящих для основной параллельной структуры моделей нейронных сетей, GPU действительно может ускорить выполнение [6], [7]. Однако, учитывая гетерогенную систему, алгоритм сопоставления с аппаратной системой не будет полностью поддерживать уровень аппаратных вычислений при работе с массивными данными размером в терабайт или петабайт. Решение заключалось в использовании параллельных вычислений для решения проблемы работы с большими данными, и модели нейронных сетей начали основываться на MapReduce для параллелизма [8], они также достигли лучших результатов, но затем с увеличением количества параметров в нейросетевые модели, обучение самой модели стало проблемой [9]. Следовательно, и в этой статье мы утверждаем, что оптимизируем алгоритмы машинного обучения, распределяя всю рабочую нагрузку между несколькими процессами параллельно, а не одним процессом.

Мы создали нейронную сеть прямого распространения и обучили ее на реальном наборе данных для решения проблемы двоичной классификации, сначала разделив исходный набор данных на мини-наборы данных, а затем назначив каждый отдельный поток для повышения производительности. После этого мы начали процесс обучения, выполнив все потоки параллельно, в конце мы агрегировали изученные веса или выходы в один выходной результат.

Остальная часть этого документа организована следующим образом: Остальная часть раздела 1 кратко рассказывает об OpenMP и используемом случае методов оптимизации. Раздел 2 посвящен предыдущим работам, которые были выполнены для повышения производительности процесса обучения в нейронных сетях. Раздел 3 представляет краткий обзор открытого многопроцессорного

Отчетное исследование финансировалось РФФИ в соответствии с исследовательскими проектами. № 19-07-00784, 18-57-34001.



OpenMP, как он работает и где он может вписаться в алгоритмы нейронной сети. В разделе 5 представлен предлагаемый нами подход к включению OpenMP в модели нейронных сетей. В разделе 6 мы обсуждаем структуру нашей нейронной сети и оптимизацию для последовательного кода, мы описываем наши эксперименты по оптимизации с применением стратегий распараллеливания библиотеки OpenMP. А в разделе 7 мы показываем полученные результаты. Наконец, в разделе 8 представлены некоторые выводы и будущие работы.

#### A. OpenMP

Кроме того, мы использовали знаменитую библиотеку OpenMP для распараллеливания сложных в вычислительном отношении частей кода с ее внутренними функциями прагмы. Таким образом, ключевое слово `pragma` сообщит компилятору, что данный код нуждается, чтобы использовать все доступные ядра на машине [10], и, кроме того, мы должны распечатать идентификатор каждого процесса. После исследования вычислительных «горячих точек» и, как следствие, мы реализуем способ распараллеливания на основе директив OpenMP в наиболее трудоемких «горячих точках», а также выполнили шаги оптимизации для ускорения процесса выполнения и достигли почти идеальной производительности [11], [12]. Однако мы пришли к выводу, что оптимизированная реализация алгоритма представляет собой сложную вычислительную задачу. Благодаря наличию разделяемой памяти в OpenMP, мы можем поделиться результатом каждого мини-пакета, выделенного для определенного блока обработки, с другими блоками обработки, работающими над другими мини-пакетами, чтобы знать полученные расчетные веса от каждого мини-пакета. Такой подход позволяет достичь коллективного взаимодействия между группой процессов.

#### B. Оптимизация

Что касается метода оптимизации, мы применили некоторые хорошо известные методы для повышения производительности и улучшения использования ловушек, поскольку нам нужно больше предварительной выборки данных в кеше, таких как разворачивание цикла, контроль размера детектора потока петли (LSD), устранение задержек LSD, перестройка выравнивания последовательности данных и т. д. Применение этих этапов оптимизации требует лучшего контроля над мини-пакетами, что вполне возможно в предлагаемом подходе. Каждый блок обработки отвечает за выделенный цикл и его процесс оптимизации, благодаря чему компромисс оптимизации одного процесса не повлияет на другие. Как показано на рис. 1, этап оптимизации реализован после статического анализа кода цикла (мини-пакетный цикл обучения) для достижения оптимальной скорости и производительности процесса обучения.

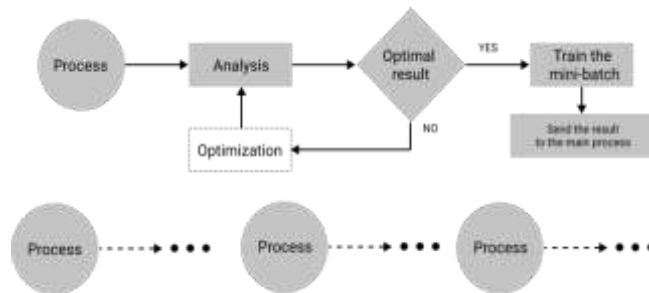


Рис. 1. Процесс оптимизации мини-партии

## II. РАБОТЫ БЛИЗКИЕ ПО ТЕМАТИКЕ

Некоторые из недавних работ по теме включают Nickolls et al. [13] и Che et al. [14] Кто также имел дело с параллельными подходами на базе ЦП, такими как MPI (интерфейс передачи сообщений), Pthreads или OpenMP. Авторы разработали CUDA (Compute Unified Device Architecture), в частности, для включения ядер потоковых процессоров, связанных с внешними динамическими разделами RAM. В отличие от наших БАРАН. В отличие от нашего подхода, их работа в основном использует параллелизм на базе GPU и CPU только на очень общем и высоком уровне, например, не обеспечивая надежного и подходящего графика оценки для конкретной цели.

Jang et al. [15] представил алгоритм обнаружения текста на основе MLP (Multi-Layer Perceptron) и реализовал его с использованием OpenMP и CUDA. Они попытались упростить использование графического процессора для начинающих программистов на графическом процессоре. Структура их подхода заключалась в том, что MLP состоял из одного входного слоя, одного или нескольких скрытых слоев и одного выходного слоя. Они указали на проблемы и препятствия в их реализации, представили пример кода с краткой оценочной частью. Тем не менее, авторы обнаружили улучшения, касающиеся времени вычисления их реализации, за счет использования методов распараллеливания. Напротив, наша работа была сосредоточена на обобщенном алгоритме обратного распространения ошибки и больше сосредоточена на разделе оценки более подробно.

Xavier et al. [16] Определяли параллельное обучение обратного распространения нейронной сети с использованием базовой подпрограммы линейной алгебры CUDA под названием CUBLAS. Они сравнили реализации CPU и CUDA для модификации скрытых нейронов и проиллюстрировали производительность GPU по сравнению с последовательным выполнением на традиционном CPU.

## III. РАСПАРАЛЛЕЛИВАНИЕ OPENMP

OpenMP расшифровывается как Open Multi-Processing, это открытая стандартная библиотека для программирования на многопроцессорных компьютерах, которая используется для определения параллелизма в языках C, C++ и Fortran. Основное преимущество использования OpenMP состоит в том, что нам нужны только минимальные изменения в исходном последовательном исходном коде, чтобы полученный исходный код можно было значительно улучшить. По этой причине мы используем OpenMP, в

частности, в нашем алгоритме, чтобы его можно было улучшить быстро и без особых сложностей в исходном коде. Основная функция OpenMP – это распределение тяжелых задач, и это может быть применено в процессе обучения нашей нейронной сети (рис. 2), как работает назначение стандартного последовательного кода, и (рис. 3) показано, как выполняются те же задачи параллельно используя OpenMP. Это важно, поскольку мы можем визуализировать и понять основное различие между тем, как задачи обрабатываются с помощью OpenMP, по сравнению с традиционным методом последовательного кодирования.

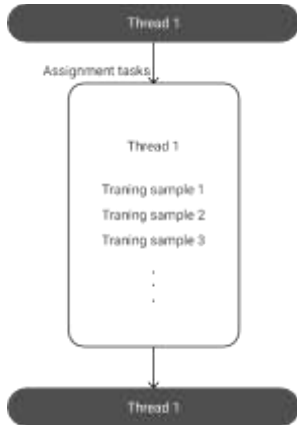


Рис. 2. Последовательность постановки задач тренировочного процесса

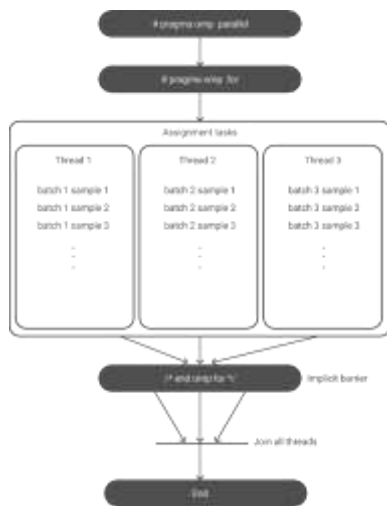


Рис. 3. Распределение задач по тренировочному процессу параллельно с использованием OpenMP

#### IV. НЕЙРОННАЯ СЕТЬ

##### A. Набор данных

Мы используем набор данных пациентов Cleveland Heart Disease. Набор данных содержит 13 переменных-предикторов. Цель состоит в том, чтобы предсказать результат от 0 до 4, где 0 означает отсутствие сердечных заболеваний, а 1–4 означает наличие сердечных заболеваний.

##### B. Модель

Мы используем обучение с учителем для решения этой конкретной задачи. Поскольку его известное контролируемое обучение работает таким образом, что в процессе обучения алгоритм получает пару, выборочные данные, в частности вектор и ожидаемое выходное значение в классе, к которому требуется классифицировать выборочные данные, пример которого перцептрон [17], [18]. Затем на основе выборок данных и их желаемых выходных значений модель пытается создать классификаторы, которые точно классифицируют образцы незнакомых данных.

В этой статье для простоты мы проектируем сеть с двумя входами (возраст и уровень холестерина) для входного слоя, 3 скрытых слоя по 8 нейронов в каждом и один выходной нейрон для выходного слоя. Целью нашей сети было спрогнозировать вероятность того, что пациент будет подвержен риску сердечных заболеваний, разделенный на четыре части: 0–0,25 означает низкий риск, 0,26–0,49 средний риск, 0,50–0,74 умеренный, высокий риск и 0,75–0,99 высокий риск (рис. 4 пример нейронной сети).

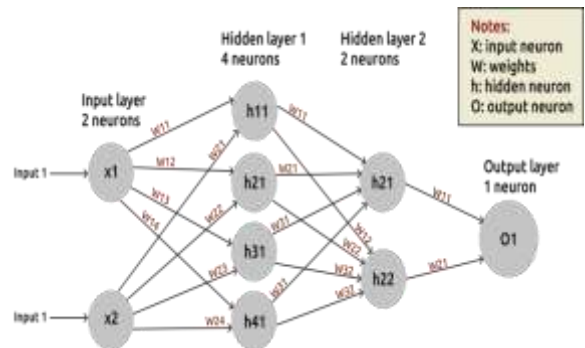


Рис. 4. Пример 4-х уровневой нейронной сети

#### V. ПРЕДЛАГАЕМЫЙ ПОДХОД

Наш подход заключается в использовании OpenMP для обеспечения многопоточного параллелизма в нейронной сети с прямой связью, мы распараллеливаем умножение матриц между слоями. Сначала мы вычисляем матрицу произведения, вычисляя скалярное произведение входного вектора с каждым столбцом в матрице весов. Мы также запускаем восемь потоков, в основном распределенных по количеству скрытых слоев, а затем используем OpenMP для сокращения суммирования по этим поэлементным продуктам (рис. 5). А для обратного распространения мы сначала вычисляем градиенты ошибок, а затем сохраняем дельты параллельно для каждого слоя. В конце концов, мы также обновляем весовые матрицы параллельно.

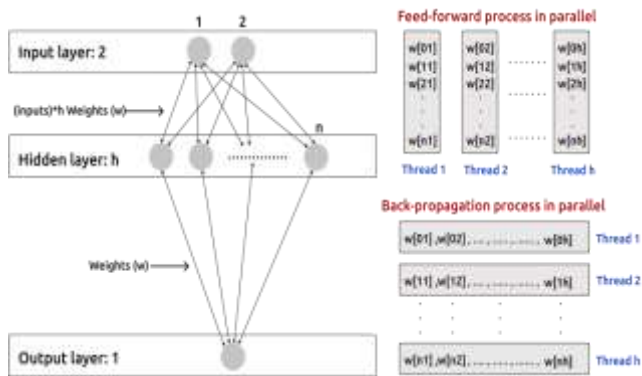


Рис. 5. Распределение потоков для процессов прямого и обратного распространения

Точнее, алгоритм 1 показывает, что в алгоритме нейронной сети самый внутренний цикл был распараллелен только потому, что внутри не было зависимостей данных.

**Данные:** подготовленный набор данных

**Результат:** изучить схему вычислений, сопоставить входные данные с целями.

инициализировать сеть нейронов; установить несколько слоев;

делать связи; установить альфа, эта;

сделать пересылку корма (); сделать обратное распространение ();

**#pragma omp parallel;**

**#pragma omp for;**

**while** набор данных дожда не в конце этого документа

trainingPass++;

**if** trainData.getNextInputs(inputVals) не равный topology[0] тогда

**break;**

**end**

делать переадресацию на входы для сети; получить результат;

делать обратное распространение против выходных целей; Обновить веса;

**end**

Алгоритм 1: Применение OpenMP к алгоритму распространения с прямой связью.

Как показано в алгоритме 1, мы сначала подготавливаем набор данных, который содержит различную информацию о пациенте (возраст, пол, уровень холестерина и т. Д.). Мы используем этот набор данных для решения задачи бинарной классификации, в которой нам нужно спрогнозировать вероятность того, что пациент будет подвержен риску сердечных заболеваний, разделенную на четыре части. Следующим шагом является инициализация нашей сети. Этап инициализации имеет решающее значение для максимальной производительности модели, поэтому после тщательного изучения мы считаем, что трехуровневой нейронной сети достаточно для нашей конкретной проблемы.

Как описано ранее, первый слой имеет 2 нейрона, скрытые слои – восемь, а третий слой – один нейрон, снабженные связями случайно инициализированных весов

и смещений. Затем мы установили и настроили параметры альфа и тета. После этого мы реализуем функцию прямой связи инициализированной сети, что означает, что данные перемещаются только в одном направлении – вперед – от входных нейронов через скрытые нейроны и к выходному нейрону. До этого момента в сети нет петель или циклов. Как мы знаем изнутри, функция прямой связи производит выходной результат, который представляет собой прогнозируемое значение, которое нам нужно сравнить с целевым значением, чтобы найти частоту ошибок, а затем попытаться уменьшить полученную частоту ошибок, выполнив обратную – разрастание всей сети.

Функция обратного распространения – это суть обучения любой нейронной сети. Он снижает частоту ошибок, получаемую путем точной настройки весов соединений нейронной сети с помощью метода, называемого цепным правилом, и имеет цикл, который показан на следующих этапах алгоритма. Проще говоря, после каждого прямого прохода по сети функция обратного распространения выполняет обратный проход при настройке параметров (весов и смещений).

Мы выполняем этот процесс, используя несколько потоков, помещая две прагмы OpenMP непосредственно перед описанным циклом для обработки параллелизма в цикле, что заметно ускоряет процесс выполнения.

## VI. ЭКСПЕРИМЕНТЫ

Для проведения экспериментов и из-за входных параметров мы тщательно анализируем наши образцы данных. Более практично запустить модель пару раз с параметрами из разных порядков, а затем сравнить результаты. Мы использовали два входных параметра, а затем запускали несколько раз с разными значениями (x1, x2 0.10, 1.0, 10, 100, 1000). Как мы знаем, время выполнения алгоритма нейронной сети зависит от типа данных, для типичных наборов данных оно не превышает 10 секунд.

Однако, поскольку мы запускаем нашу модель с различными вариациями параметров, мы заметили, что время выполнения резко увеличилось. Затем мы распараллеливаем вычисления в нашей модели. И для этого мы должны проанализировать, могут ли циклы в нашей модели выполняться параллельно, определяя, удовлетворяют ли они условиям Бернштейна или нет, когда есть вложенные циклы, только один цикл может быть распараллелен с OpenMP. Поскольку мы указываем перед этапом оптимизации нашего подхода, мы можем применить метод разветвления цикла, который полезен для использования LSD и получения наиболее распараллеливаемых циклов в нашем обучающем конвейере, наряду с использованием параллельных структур данных, поскольку нам нужно хранить наш набор данных в части разделяемой памяти, что неоспоримо, когда мы думаем о пакетной обработке.

Цель состоит в том, чтобы выбрать наиболее внешний контур, потому что таким образом достигается лучшая эффективность. Реализуем модель на языке C++. Чтобы иметь возможность визуализировать и сравнить эффективность и скорость нашего подхода, мы провели три экспе-

римента. Первый эксперимент обучал модель последовательно на всем наборе данных, а второй тренировал модель параллельно с использованием OpenMP, в то время как в последнем эксперименте мы разделили набор данных на мини-пакеты, а затем обучили модель параллельно с помощью OpenMP.

Все эксперименты проводились на компьютере с ОС Windows 10, Intel Core i7–8750HQ 2,20 ГГц, 12 ядрами и 16 ГБ ОЗУ.

## VII. РЕЗУЛЬТАТЫ

Полученные результаты наших экспериментов можно увидеть в таблице результатов, см. Таблицу I. Распараллеленный цикл выполнялся максимум с восемью потоками, так как он имел только восемь итераций на протяжении всего эксперимента. И (рис. 6) показывает сравнение между проведенными экспериментами относительно времени выполнения набора данных для от 1 до 8 потоков (р 1, ..., 8). Уменьшение времени выполнения наблюдалось за счет разделения набора данных на мини-пакеты и увеличения количества потоков. Чтобы дать более четкое представление о влиянии распараллеливания на время выполнения, мы представили два дополнительных параметра Ускорение (S) и Эффективность (E). Ускорение показывает, насколько наш параллельный алгоритм набирает скорость по сравнению с его последовательной версией. Здесь он вычисляется как соотношение между временем выполнения с одним потоком и временем выполнения с р потоками, в то время как эффективность – это параметр, указывающий на использование алгоритмом параллельных модулей.

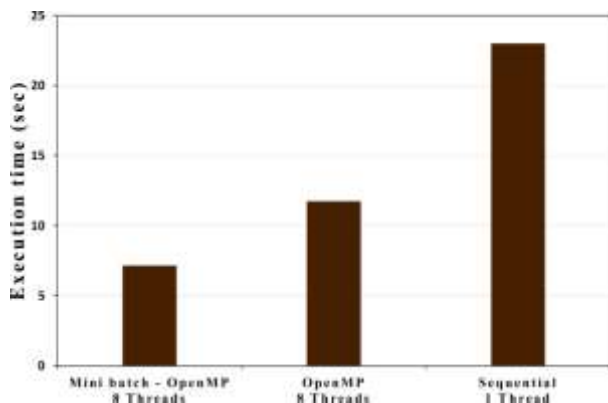


Рис. 6. Результаты эксперимента по времени выполнения (однопоточный или многопоточный).

ТАБЛИЦА I РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Мини-дозирование	OpenMP	Время исполнения сек	Точность
Нет	Нет	23.011	82.9 %
Нет	да	11.749	84.3 %
да	да	7.161	84.2 %

Стоит отметить, что абсолютная эффективность не была достигнута из-за случайного распределения нагрузки между потоками, зависящей от случайным образом перемешанных выборок данных и различного времени сходимости в соответствии с параметрами. Важно отметить, что для времени доступа к данным, чем больше потоков пыта-

ется получить доступ к данным, тем сложнее становится процесс по сравнению с последовательным алгоритмом.

Принимая во внимание все вышесказанное, для нашего набора данных результаты, достигнутые в экспериментах, можно рассматривать как положительные, что относительно уменьшило время выполнения процесса обучения.

## VIII. ЗАКЛЮЧЕНИЕ

В этой статье было доказано, что OpenMP можно эффективно использовать для улучшения приближения к качественным результатам контролируемого обучения. Эксперименты по бинарной классификации, проведенные на одном наборе практических данных, ясно показывают значительное сокращение времени вычислений по сравнению с традиционными алгоритмами машинного обучения. Возможны будущие исследования по включению интерфейса передачи сообщений MPI в нейронную сеть для систем с распределенной памятью. Хотя другие алгоритмы нейронных сетей могут быть распараллелены, и в этой области проводятся расширенные исследования. Применяя параллельную оптимизацию, можно достичь больших результатов и найти новые применения нейронных сетей для решения реальных проблем.

## СПИСОК ЛИТЕРАТУРЫ

- [1] E.A. Goncharenko, A.A. Paznikov, A.V. Tabakov, Evaluating the performance of atomic operations on modern multicore systems, *Journal of Physics: Conference Series* 1399 (2019) 033107. doi:10.1088/1742-6596/1399/3/033107.
- [2] K. Israel, *Computer Arithmetic Algorithms*, 2nd Edition, AK Peters, New York, NY, 2002.
- [3] K. Yu, Large-scale deep learning at baidu, in: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM'13*, Association for Computing Machinery, New York, NY, USA, 2013, p. 2211–2212. doi:10.1145/2505515.2514699.
- [4] A. Coates, B. Huval, T. Wang, D. J. Wu, A. Y. Ng, B. Catanzaro, Deep learning with cots hpc systems, in: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, JMLR.org, 2013, p. III–1337–III–1345.
- [5] O. Yadan, K. Adams, Y. Taigman, M. Ranzato, Multi-gpu training of convnets (2013). arXiv:1312.5853.
- [6] R. Uetz, S. Behnke, Large-scale object recognition with cuda-accelerated hierarchical neural networks, in: *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, Vol. 1, 2009, pp. 536–541. doi:10.1109/ICICISYS.2009.5357786.
- [7] D.C. Cireşan, U. Meier, J. Masci, L.M. Gambardella, J. Schmidhuber, Flexible, high performance convolutional neural networks for image classification, in: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two, IJCAI'11*, AAAI Press, 2011, p. 1237–1242.
- [8] Z. Liu, H. Li, G. Miao, Mapreduce based backpropagation neural network over large scale mobile data, in: *2010 Sixth International Conference on Natural Computation*, Vol. 4, 2010, pp. 1726–1730. doi:10.1109/ICNC.2010.5584323.
- [9] Q. Wang, J. Zhao, D. Gong, Y. Shen, M. Li, Y. Lei, Parallelizing convolutional neural networks for action event recognition in surveillance videos, *Int. J. Parallel Program.* 45 (4) (2017) 734–759. doi:10.1007/s10766-016-0451-4.
- [10] A. Avizienis, Signed-digit number representations for fast parallel arithmetic, *IRE Transactions on Electronic Computers* EC-10 (3) (1961) 389–400. doi:10.1109/TEC.1961.5219227.
- [11] O.T. Mohammed, M.S. Heidari, A.A. Paznikov, Mathematical computations based on a pre-trained ai model and graph traversal, in:

- 2020 9th Mediterranean Conference on Embedded Computing (MECO), 2020, pp. 1–4. doi:10.1109/MECO49872.2020.9134081.
- [12] O.T. Mohammed, M.S. Heidari, A.A. Paznikov, M. S. Kupriyanov, Towards optimization of big numbers computation through an ai pre-trained model and graph traversal, in: 2020 XXIII International Conference on Soft Computing and Measurements (SCM), 2020, pp. 153–156. doi:10.1109/SCM50615.2020.9198759.
- [13] J. Nickolls, I. Buck, M. Garland, K. Skadron, Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for?, *Queue* 6 (2) (2008) 40–53.
- [14] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, K. Skadron, A performance study of general-purpose applications on graphics processors using cuda, *Journal of Parallel and Distributed Computing* 68 (10) (2008) 1370–1380.
- [15] H. Jang, A. Park, K. Jung, Neural network implementation using cuda and openmp, in: Proceedings of DICTA '08, IEEE Computer Society, USA, 2008, p. 155–161. doi:10.1109/DICTA.2008.82.
- [16] X. Sierra-Canto, F. Madera-Ramirez, V. Uc-Cetina, Parallel training of a back-propagation neural network using cuda, *ICMLA '10*, IEEE, USA, 2010, p. 307–312. doi:10.1109/ICMLA.2010.52.
- [17] R. Sathya, A. Abraham, Comparison of supervised and unsupervised learning algorithms for pattern classification, *International Journal of Advanced Research in Artificial Intelligence* 2 (2) (2013). doi:10.14569/IJARAI.2013.020206.
- [18] I. Kholod, A. Rukavitsyn, A. Paznikov, S. Gorlatch, Parallelization of the self-organized maps algorithm for federated learning on distributed sources, *The Journal of Supercomputing* (Nov 2020). doi:10.1007/s11227-020-03509-2.

# Интеллектуальное управление информационной безопасностью киберфизических систем

Ю. А. Шичкина<sup>1</sup>, Р. Р. Фаткиева<sup>2</sup>

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

<sup>1</sup>strange.y@mail.ru, <sup>2</sup>rikki2@yandex.ru

**Аннотация.** Разработан метод управления информационной безопасностью с использованием структурно-параметрического синтеза моделей киберфизических систем на базе кусочно-линейных агрегатов. Метод позволяет осуществить реконфигурацию элементов сложных технических объектов с использованием оператора сопряжения.

**Ключевые слова:** киберфизические системы; кусочно-линейные агрегаты; информационная безопасность; структурно-параметрический синтез

## I. ВВЕДЕНИЕ

Автоматизация и гибкость систем управления информационной безопасностью киберфизических систем является ключевым шагом на пути к повышению интеграции оборудования, росту скорости передачи данных по сети и уменьшению затрат на обнаружения уязвимостей. Ущерб от нарушения функционирования киберфизических систем в результате деструктивных воздействий со стороны злоумышленников может исчисляться миллионами и миллиардами затрат на их устранение. Применение интеллектуальных алгоритмов управления для обеспечения заданного уровня безопасности позволяет осуществлять передачу и обработку данных с варьируемыми характеристиками. Для обеспечения данных требований необходимы соответствующие методы и технологии оперативного синтеза систем интеллектуального управления киберфизическими объектами. Одним из перспективных путей развития от теории к технологии интеллектуального управления является переход к структурно-параметрическому синтезу перестраиваемых моделей киберфизических систем. Это позволяет оперативно реагировать на изменения законов функционирования систем, своевременно предсказывать возможные события, а также обосновывать и реализовывать целесообразные мероприятия по обеспечения информационной безопасности и восстановлению работоспособности. Синтез систем управления рассматривался во многих работах [1–16] и др. Чаще всего при этом используются системы оптимального управления [9–10]. Весьма эффективным является применение структурно-функционального и логико-вероятностного подходов [11]. Применение мультиагентного подхода приводит к задаче декомпозиции [12] и оцениванию работоспособности на конечном элементе, что не всегда отражает общую картину нарушения информационной безопасности, поскольку каждая из них решается параллельно

соответствующим интеллектуальным агентом, и не всегда учитывает необходимые взаимосвязи. Математические модели структурно-функционального и структурно-параметрического синтеза обычно являются динамическими и представляют собою системы нелинейных дифференциальных уравнений весьма высоких порядков [13–14], что влияет на оперативность при принятии решения в условиях дефицита времени.

В настоящее время для обеспечения мониторинга, защиты и восстановления киберфизических систем от деструктивных воздействий используются математические модели оценивания протекающих в них процессов, а также влияющих на них применяемых моделей процессов мониторинга, защиты и восстановления. Однако оценка эффективности их функционирования осуществляется с затратой человеческих и материальных ресурсов. Системы мониторинга, управления и защиты киберфизических систем, функционирующие на основе таких моделей, имеют не высокий уровень адаптации к текущим изменениям. Известны также подходы математического и программного обеспечения подобных процессов на основе построения регрессионных моделей, частотно-временного анализа, статистические модели, методы формализации процессов в виде марковских и полумарковских моделей, активно развиваемые в последние годы методы нейросетевого моделирования и другие. К основному недостатку известных подходов следует отнести очень высокую сложность задач синтеза на большом числе условий, свойственных реальным процессам.

## II. ПОСТРОЕНИЕ СИСТЕМЫ УПРАВЛЕНИЯ КИБЕРФИЗИЧЕСКИМИ УСТРОЙСТВАМИ

Оперативное построение адекватных текущим ситуациям интеллектуальных систем управления информационной безопасностью киберфизических систем (СУИБКУ) позволяет обеспечить гибкость и снизить сложность решения задач прогнозирования их поведения, предоставить временной ресурс на выработку и реализацию целесообразных мероприятий защиты, повысить эффективность их мероприятий. Построение таких систем возможно за счет связывания наблюдаемых событий в пространстве и времени с управляющими воздействиями для формирования структурно-параметрической системы защиты киберфизических устройств. В этом случае для анализа поведения системы и связывания наблюдаемых деструктивных воздействий событий в пространстве и времени целесообразно

использовать известные методы системного анализа, с учетом свойственных им факторов. Это влечет за собой необходимость разработки систем мониторинга и формирования критериев эффективности поведения, а также оценку степени защиты с использованием общей и частных математических задач оптимизации процессов для обеспечения защиты киберфизических устройств. Подобный подход при формировании СУИБКУ, позволяет системе оставаться в устойчивом состоянии. Тогда СУИКУ можно описать в виде множества, состоящего из перестраиваемых конечных автоматов, связанных в единую систему, а сам автомат, в виде:

$$A = \langle X, \delta, \lambda, C, Y \rangle,$$

где  $X$  – конечный набор входных сигналов;  $\delta$  – конечный набор передаточных функций (множество связей между входом и выходом);  $\lambda$  – функция выходов;  $C$  – конечный набор внутренних состояний;  $Y$  – конечный набор выходных сигналов.

Для формирования управляющего воздействия в процессе обеспечения безопасности основной функцией СУИБКУ является оценка потенциальной неустойчивости системы и выработка соответствующей реакции на нее. Это возможно за счет воздействия на изменение параметров передаточных функций, выходов и набора состояний СУИБКУ, участвующих в управлении. Рассмотрим, какие элементы системы могут быть изменены системой управления для компенсации и предотвращения перехода системы в неустойчивое состояние. Такими элементами являются:  $(X, C, Y)$ , входное воздействие, внутренне состояние и выходное воздействие. Очевидно, что в зависимости от параметра, на который будет оказано воздействие, можно классифицировать систему управления как по выходу, входу, так и передаточной функции [15]. Тогда типовой процесс, который происходит в результате функционирования киберфизической системы, имеет начально входное воздействие  $x_0$ , состояние  $c_0$  и начальное выходное воздействие  $y_0$ , зная функцию переходов  $\delta$ , и функцию выходов  $\lambda$ , можем определить  $c_1 = \delta(c_0, x_1)$  и  $y_1 = \lambda(c_0, x_1)$ , с повторением цикла.

Рассмотрим более подробно функцию переходов  $\delta$ , зафиксируем какое – либо состояние  $c_n$  в системе. Тогда очевидно, что данная функция будет реализовывать отображение подмножеств  $X$  в подмножество  $C$   $\delta_n: x_n \rightarrow c_n, c_n \in C, x_n \in X$ . Назовём тогда  $X$  множеством безопасных входных воздействий для состояния  $c_n$ , а множество  $X \setminus x_n$  – множеством запрещенных воздействий для  $c_n$ . Очевидно, что аналогичные рассуждения можно провести для функции выходов  $\lambda$ , что позволит сформировать похожие множества для выходных воздействий.

Введем функцию  $U: X \times C \rightarrow \{1, 0, -1\}$ , и зафиксируем ее относительно состояния  $c_n \in C$ . Тогда функция  $U_{c_n}: X \times C \rightarrow \{1, 0, -1\}$  имеет такой вид:

$$\forall c \in C, \forall x \in X u_{c_n} = \begin{cases} 1, \delta(c_n, x) = c \\ 0, \delta(c_n, x) \neq c \\ -1, \delta(c, x) = c_n \end{cases}.$$

Введем упорядоченное множество пар  $E = ((S \times X) \times (S \times X))$ , а так же множество вершин  $V = (S \times X)$ .

Тогда граф  $G = (V, E)$  будет описывать все состояния и переходы системы. Для простоты понимания функцию  $U$  можно формализовать как свернутую матрицу смежности графа  $G$ .

Назовём любой путь в графе  $G$  событием и заметим, что последовательность вершин графа описывают историю состояний системы при ее функционировании с определенным входным и переходным воздействием, в общем виде позволяя осуществить мониторинг системы.

Если допустить, что формальное построение графа не зависит от воздействия  $x$ , (то есть набор вершин, в которые известен путь из текущей, зависит исключительно от функции переходов  $\delta$ , и воздействия  $x$  и описывает конкретную вершину из уже известного набора), то это позволяет, не выполняя само воздействие, оценить внутреннее состояние системы при попытке его выполнения.

Таким образом система управления, построенная на основе графа  $G$  позволяет сформировать два набора воздействия (не принадлежащие множеству разрешенных, и принадлежащее к нему), что дает возможность безопасно перейти из начального внутреннего состояния системы в конечное. Предположим, что в результате функционирования системы было, уставлено, что определенная последовательность состояний приводит к неустойчивому функционированию системы. Покажем, что такой вариант возможен. Пусть у системы 3 состояния и 5 воздействий, причем функционально разрешен переход в любое состояние при соответствующем воздействии, например, при воздействии 2 система переходит в состояние 2 и так далее (рис. 1). Предположим, что событие 1-2-3 приводит систему в неустойчивое состояние. Очевидно, что блокировка переходов (удаление ребра из графа) 1-2 или 2-3 ведет к существенному усечению функционала системы и плохо адаптируется к такого рода проблемам. Возникает необходимость введения в СУИБКУ систему оценивания и «перестроения», цель которых провести качественную оценку схожести событий и в случае выявления несоответствия эталонным событиям, например, аномалии. Это позволяет осуществить оценку событий, которые ведут к неустойчивости системы и сформировать набор управляющих воздействий «перестраивающих» систему для возврата в устойчивом состоянии. То есть сформировать набор таких управляющих воздействий, которые позволяют выбрать нужное решение об изменении выходного, входного воздействия или внутреннего состояния системы. В этих условиях СУИБКУ должна решать классическую задачу теории принятия решения в виде построения целевой функции и вычисления ее экстремумов.

### III. МЕТОД СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА КИБЕРФИЗИЧЕСКИХ УСТРОЙСТВ С ИСПОЛЬЗОВАНИЕМ АППАРАТА КУСОЧНО-ЛИНЕЙНЫХ АГРЕГАТОВ

При формировании СУИБКУ с возможностью «перестроения» основное внимание необходимо уделять определению основных операционных процессов и управляющих воздействий, а также адаптацию их под динамически изменяемые условия, что требует создания спецификаций для каждой из операций управления. Определение управляющих воздействий включает в себя создание спецификации технологического процесса приема, передачи, обработки информации как на каждом киберфизическом устройстве, так и системы в целом, а также набора структурно-параметрических устройств собственно выполняющих эту обработку. Основной проблемой при таком подходе является нахождение синтезирующей функции, которая в свою очередь, является многомерной за счет включения множества переменных элементов конечных устройств.

Это позволит осуществить наблюдение за изменением пространственно-временной адаптации к изменяющимся условиям внешней среды путем переконфигурирования внутренних режимов работы или связей между элементами системы. Частным случаем решения задачи реконфигурации является применение математического аппарата теории кусочно-линейных агрегатов (КЛА) [16]: с возможностью изменения функций выходов и переходов, поскольку входной сигнал, в широком смысле, возможно, представить какой управляющий.

Рассмотрим функционирование киберфизического устройства, описанного в логике КЛА с возможностью реализации перестроения внутреннего состояния за счет структурно-параметрического синтеза конечного набора передаточных функций  $\delta$ , например с использованием матричной передаточной функции. В простейшем случае функция переходов  $F$  зависит набора входных сигналов  $X$ , полученных с других конечных элементов, и внутреннего  $S$  состояния КЛА. Операционный блок конечного набора переходов  $\delta$  с перестраиваемой структурой можно представить в виде  $\delta = \langle \delta_F, \delta_C, \delta_\lambda \rangle$ , где  $\delta_F$  - функция формирования переходов внутренних состояний;  $\delta_C$  - функция отображения состояний в комплекс входных воздействий;  $\delta_\lambda$  - комплекс управляющих воздействий для перехода из входного в выходное состояние. Поскольку сами переходы могут изменяться под воздействием изменений входного сигнала, то операционный блок переходов  $\delta$  в общем виде может быть изменен при помощи функции отображения состояний переходов:

$$\delta_\lambda: \delta_C \rightarrow \delta_F, \quad (1)$$

а функция выходов  $\lambda$  при изменении  $\delta_\lambda$ .

В простейшем случае состояние КЛА зависит от внешнего сигнала  $X$ , поступившего в момент времени  $t$ , и начального состояния  $S$  киберфизического устройства  $S$ . При функционировании устройства в штатном режиме функция отображения состояний переходов  $\delta$  под управляющим воздействием генерирует функцию переходов  $\delta_F$ . Используя функцию переходов,

определяющий порядок переходов из одного внутреннего состояния в другое, можно записать функцию выходных состояний в виде  $Y(\lambda) = F\{X, \delta_F, \delta_C, \delta_\lambda\}$ . Однако при нарушении функционирования из-за отказа устройства или вследствие формирования деструктивных воздействий, возникает необходимость перестройки КЛА за счет реконфигурации переходов внутри элементов, либо введения обратной управляющей связи, что обеспечивает дополнительную надежность при функционировании.

Тогда структуру управляющего элемента можно также описать в виде последовательности переходов состояний КЛА при формировании управляющих воздействий.

В этом случае функцию переходов  $F$  можно представить в виде синтеза множества структурных модулей, отвечающих за ту или иную функцию перехода  $F = (F_{U_1}, F_{U_2}, \dots, F_{U_k})$ , сформированного за счет переходов из начального состояния в конечное состояние за конечное число шагов, под воздействием  $k$  - управляющих воздействий  $U = (U_1, U_2, \dots, U_k)$  для достижения киберфизическим устройством заданных показателей качества функционирования. Это позволяет перейти к рассмотрению класса КЛА с перестраиваемой структурой.

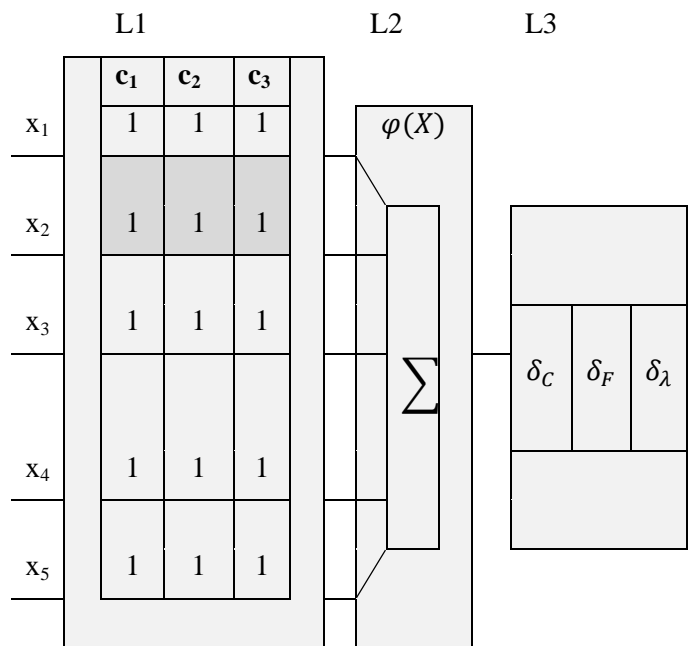


Рис. 1. Схема реализации кусочно-линейного агрегата с перестраиваемой структурой

Если процесс функционирования КЛА формализуется на  $t$ -й момент времени и задается в виде:

$$Y(\lambda)_t = \{C_\delta: C_X \rightarrow C_Y, S_\delta: S_X \rightarrow S_Y, \delta: \delta_C \rightarrow \delta_F\}, \quad (2)$$

где  $Y(\lambda)_t$  - состояние выхода системы на  $t$ -й момент времени относящееся к множеству допустимых состояний выхода;  $C_\delta$  - внутреннее состояние системы на  $t$ -й момент времени относящееся к множеству допустимых внутренних состояний, предшествующих моменту  $t$ ;  $S_\delta$  - структурная схема элемента на  $t$ -й момент времени



относящееся к множеству допустимых структурных схем;  $\delta_C$  – функция переходов из множества допустимых функций относительно  $t-1$  момента времени. Вид функции определяется внутренним состоянием объекта и состоянием его входов на  $t$  момент времени.

То модель КЛА с перестраиваемой структурой от момента  $t$  к моменту  $t+1$  можно записать в виде:

$$Y(\lambda)_{t+1}: Y(\lambda)_t \{C_\delta: C_X \rightarrow C_Y, S_\delta: S_X \rightarrow S_Y, \delta: \delta_C \rightarrow \delta_F\} \rightarrow Y(\lambda)_t \{C_\delta: C_X \rightarrow C_Y, S_\delta: S_X \rightarrow S_Y, \delta: \delta_C \rightarrow U_X \rightarrow \delta_F\} \quad (3)$$

В этом случае для каждого комплекса, состоящего из набора (2) с заданными показателями функционирования необходимо осуществить синтез внутреннего состояния объекта, в зависимости от состояния и набора входных и управляющих воздействий.

Наибольшие трудности при формировании КЛА с перестраиваемой структурой возникают при формировании функции отображения состояний переходов из  $X$  в  $Y$ , что связано это с необходимостью учета уровней при формировании киберфизической системы, завязанном на конечном множестве устройств, входящих в систему. Такое представление позволяет рассмотреть киберфизическую систему как иерархическую систему  $L$  – уровневую систему  $S = (S^1, S^2, \dots, S^l)$ , содержащую несколько подсистем с киберфизическими устройствами. В этом случае входное воздействие можно классифицировать как входное воздействие, поступившее из внутренней среды подсистемы, или как входное воздействие, поступившее из внешней окружающей среды. Такое разделение позволяет сформировать не только координацию управления по уровням киберфизической системы, но и сформировать каскадную систему СУИБКУ, позволяющую осуществлять управления по внутреннему и/или внешнему контурам системы.

Однако здесь могут возникнуть трудности управления, в случае, если у киберфизического устройства отсутствует возможность осуществить динамическое перестроение и/или возникают трудности сопряжения между двумя соседними элементами. В этом случае целесообразно необходимо ввести внешний элемент сопряжения ( $\varphi(X)$ , рис. 1), осуществляющий перестроение функций «вход-выход» двух соседних элементов с учетом передаточных функций и управляющих воздействий.

#### IV. ЗАКЛЮЧЕНИЕ

Дальнейшее исследование в области построения систем управления киберфизическими устройствами целесообразно осуществлять в направлении поиска методов сбалансированного управления для достижения оптимального состояния безопасности при начальной неопределенности и изменяющихся условиях функционирования киберфизических устройств. Это позволит обосновать методы, применяемые для реконфигурации внутренних состояний элементов в условиях переменной структурно-функциональной

организации систем. А также обосновать необходимость поиска методов структурно-параметрического синтеза моделей киберфизических систем, при условии деструктивных воздействий внешней среды.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] W. A. Apaza-Perez, A. Girard, C. Combastel, A. Zolghadri. Symbolic Observer-Based Controller for Uncertain Nonlinear Systems / IEEE Control Systems Letters, vol. 5, №. 4, 2021, pp. 1297-1302. DOI: 10.1109/LCSYS.2020.3034274.
- [2] S. Tseng, J. Anderson. Deployment Architectures for Cyber-Physical Control Systems / American Control Conference (ACC), Denver, CO, USA, 2020, pp. 5287-5294, DOI: 10.23919/ACC45564.2020.9147953.
- [3] Saurav Kumar Ghosh, Jaffer Sheriff R C, Vibhor Jain, Soumyajit Dey. Reliable and Secure Design-Space-Exploration for Cyber-Physical Systems / ACM Trans. Embed. Comput. Syst. 19, 3, 2020, 29 pages. DOI:https://doi.org/10.1145/3387927
- [4] L. Niu, B. Ramasubramanian, A. Clark, L. Bushnell, R. Poovendran. Control Synthesis for Cyber-Physical Systems to Satisfy Metric Interval Temporal Logic Objectives under Timing and Actuator Attacks / ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPs), Sydney, NSW, Australia, 2020, pp. 162-173, DOI: 10.1109/ICCPs48487.2020.00023.
- [5] I. Saglam, E. A. Gol. Cause Mining and Controller Synthesis with STL / IEEE 58th Conference on Decision and Control (CDC), Nice, France, 2019, pp. 4589-4594, DOI: 10.1109/CDC40024.2019.9029894.
- [6] Kudriashov, N. S., Markov, S. A., & Potekhin, V. V. Adaptive control system synthesis methods for complex manufacturing objects. / Paper presented at the Annals of DAAAM and Proceedings of the International DAAAM Symposium., 30(1), 2019, pp 493-499. DOI:10.2507/30th.daaam.proceedings.066
- [7] Y. Wang, N. Matni, J. C. Doyle. Separable and Localized System-Level Synthesis for Large-Scale Systems / IEEE Transactions on Automatic Control, vol. 63, №. 12, 2018, pp. 4234-4249. DOI: 10.1109/TAC.2018.2819246.
- [8] Rebaï, S. B., Voos, H., Sajadi Alamdari, S. A. A contribution to cyber-physical systems security: An event-based attack-tolerant control approach /IFAC-PapersOnLine, 51(24), 2018, pp 957-962. DOI:10.1016/j.ifacol.2018.09.690
- [9] Y. Chen, S. Kar, J. M. F. Moura. Resilient Distributed Estimation Through Adversary Detection / IEEE Transactions on Signal Processing, vol. 66, № 9, 2018, pp. 2455-2469. DOI: 10.1109/TSP.2018.2813330.
- [10] Tang, X., & Dong, Y. Expectation maximization based sparse identification of cyberphysical system. /International Journal of Robust and Nonlinear Control, 2020. DOI:10.1002/rnc.5325
- [11] Рябинин И.А. Логико-вероятностный анализ проблем надежности и безопасности // Palmarium Academic Publishing, Saarbrücken, Deutschland. 2012. С. 263
- [12] L. Burbano, L.F. Cómbita, N. Quijano and S. Rueda. Dynamic Data Integration for Resilience to Sensor Attacks in Multi-Agent Systems / IEEE Access, vol. 9, pp. 31236-31245, 2021, DOI: 10.1109/ACCESS.2021.3059560.
- [13] L. Pan, Z. Gu, Y. Ren, C. Liu, Z. Wang. An Anomaly Detection Method for System Logs Using Venn-Abers Predictors / IEEE Fifth International Conference on Data Science in Cyberspace (DSC), Hong Kong, China, 2020, pp. 362-368. DOI: 10.1109/DSC50466.2020.00063.
- [14] Kasenov A.A., Kustov E.F., Magazev A.A., & Tsyrlunik V.F. A markov model for optimization of information security remedies / Journal of Physics: Conference Series, 2020, 1441(1), pp 49-53. DOI:10.1088/1742-6596/1441/1/012043
- [15] Бусленко Н.П. Моделирование сложных систем. М.: Наука, 1978. 400 с.
- [16] Жмудь В.А. Адаптивные системы автоматического управления с единственным основным контуром // Автоматика и программная инженерия. 2014. № 2 (8). С. 106-122.

# Исследование результатов тестирования собак-поводырей и информации об их происхождении

Я. А. Бекенева

Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)  
yana.barc@mail.ru

**Аннотация.** Изучение характеристик, наследуемых потомками от родителей и более дальних предков представляет собой важную задачу селекции. В данной работе рассматривается набор данных, представляющих собой результаты тестирований, проведенных в центре подготовки собак-поводырей. Предлагается разработка искусственной нейронной сети для нахождения наиболее/наименее важных признаков поводырей. Представлены результаты исследования применения искусственной нейронной сети для оценки пригодности собак к выполнению работы поводыря. Определено, от каких родителей по отдельности и как их пар родителей наибольшая/наименьшая вероятность получить собак-поводырей, используя различные операции над матрицами.

**Ключевые слова:** анализ происхождения; наследование характеристик; родословная; собаки-поводыри; тестирование собак

## I. ВВЕДЕНИЕ

Существует множество задач, связанных с отбором желаемых характеристик объектов, улучшением параметров различных процессов. Одной из таких задач является селекция, где из поколения в поколения желаемые качества должны передаваться потомкам. При селекции большое внимание уделяется требуемым характеристикам, и отбор ведется, прежде всего, по выраженности этих характеристик. При планировании племенного разведения важно учитывать особенности выполняемых задач, так как для выполнения разных задач могут требоваться совершенно разные характеристики.

При разведении селекционеры изучают, в каких линиях наиболее вероятно получение особей с ярко выраженными желаемыми характеристиками, а в каких линиях могут передаваться нежелательные качества [1]. В настоящее время существуют различные базы, где содержатся разного рода сведения об отдельных группах животных. Например, в кинологических центрах существуют базы, где хранится информация о проведенных тестированиях. Чем более подробная информация представлена в таких базах, тем больше выводов может быть сделано о наследуемых качествах. Зачастую базы данных могут содержать информацию о большом количестве животных, изучение и анализ может занять много времени. В этой работе предлагается использование нейронной сети для выявления появления определенных качеств у потомков разных производителей.

Сначала нейронные сети обучаются на тренировочных данных, затем применяются для решения поставленной задачи. Благодаря обучению с учителем достигается высокая точность выхода сети. Причем в большинстве случаев результат работы сети рассчитывается быстрее и точнее, чем, если бы эту работу выполнял человек. Операции над матрицами являются одними из самых необходимых при разработке многих программных продуктов. Без них невозможно представить ни одной современной программы. Матрицы играют ключевую роль и при разработке нейронных сетей, ведь практически все операции там происходят в матричной форме.

В качестве примера рассмотрена база данных о результатах тестирования собак, претендующих на зачисление в школу собак-поводырей. База содержит сведения о родителях тестируемых собак, поэтому задачей этой работы является определение производителей, от которых получение потенциального поводыря наиболее вероятно. В данной работе анализ происхождения тестируемых собак будет практически полностью состоять из матричных операций, в то время как поиск важнейших для поводырей характеристик будет проходить с использованием нейронной сети.

## II. ОПИСАНИЕ ПОДХОДА

Для разработки программы используется бесплатное IDE PyCharm 2019.3.1 [2]. Программа состоит из 4-х модулей.

Модуль чтения входных данных использует библиотеку pandas [3] для чтения и нормализации из excel-таблицы данных, используемых для обучения и тестирования нейронной сети. Также он считывает данные о родителях для дальнейшего анализа того, от каких родителей по отдельности и каких пар родителей наибольшая/наименьшая вероятность получить собак-поводырей. Для чтения файлов применяется функция `read`, куда передаются названия файлов. После чтения, функция приводит считанные данные типа DataFrame к типу ndarray, нормализует полученные массивы с помощью библиотеки numpy [4] и возвращает подготовленные тренировочные/тестовые данные и данные о родословной.

В модуле построения нейронной сети содержится программная реализация нейронной сети. Для ее обучения и тестирования исходные данные были заранее разбиты на выборки (тренировочную и тестирующую). Для

корректной работы нейронной сети необходима нормализация входных данных, чтобы разница в их значениях не оказывала сильного влияния на вычисления.

Модуль тестирования выполняет тесты нейронной сети для поиска оптимального значения скорости обучения и размера мини-пакетов, а также рассчитывает кривую обучения для тренировочной и тестовой выборок. С помощью библиотеки matplotlib [5] модуль строит графики зависимости точности нейронной сети от скорости обучения, зависимости точности нейронной сети от размера мини-пакетов, рассчитанной кривая обучения.

В модуле анализа важности входных характеристик выполняется поиск самых важных характеристик для отбора собак-поводырей, путем составления тепловой карты степени влияния входных характеристик на скрытый слой сети. Данный модуль не содержит своих методов. Он использует модуль построения нейронной сети для тренировки сети и библиотеки seaborn [6] и matplotlib для составления и вывода тепловой карты.

В модуле анализа родителей выполняется поиск родителей, от которых наибольшая/наименьшая вероятность получить поводыря, путем составления графиков зависимости между родителями/парами родителей от количества в потомстве собак годных/негодных к работе поводыря. Данный модуль библиотеку matplotlib для составления и вывода графиков.

### III. ЭКСПЕРИМЕНТЫ

Для проведения экспериментов была использована таблица, содержащая результаты тестирования тысячи собак для определения пригодности к обучению и дальнейшей работе в качестве поводыря. При тестировании оценивалось 50 характеристик поведения собак и выставлялся статус, является ли собака пригодной для несения службы поводыря, а также маркировался признак, по которому собака была признана непригодной.

Поскольку нейронная сеть может работать только с числовыми данными, то для ее корректной работы необходимо заменить текстовые значения в исходных данных на числовые. В частности, были назначены числовые эквиваленты статусам собак по итогам тестирования, а вместо кличек собак использовались идентификационные номера в десятичном формате.

Для поиска родителей, от которых наибольшая/наименьшая вероятность получить поводыря необходимо из исходных данных взять столбцы с номерами родителей и результирующими переменными.

В результате анализа особей из множества матерей была выявлено, что наибольшей и наименьшей вероятностью получить поводыря обладает мать с номером 975. Такой результат показывает, что большее влияние на потомство от этой матери оказывает отец. Анализ показал, что для матери 975 наибольшим количеством поводырей обладает пара 975-860, а наименьшим – 975-1233. Однако, для пары 975-860 количество поводырей/не поводырей одинаково, из чего можно сделать вывод, для наибольшей вероятности

получить поводыря для матери 975 надо взять отца 860, но вероятность появления поводыря для этой пары будет 50/50.

В результате анализа особей из множества отцов было выявлено, что наибольшей вероятностью получить поводыря обладает отец с номером 860, а наименьшей отцы с номерами 1188 и 1221. Причем для отца 860 наилучшей парой для получения потомства будет мать с номером 758, причем эта пара обладает наибольшей, по сравнению со всеми другими парами, вероятностью получить в потомстве поводыря. Для отца 1188 наилучшей парой будет мать 1232, а для 1221 – 1073. Причем для 1221 пары с матерью 1073 и 1141 дают одинаковое соотношение поводырь/ не поводырь.

При анализе комбинаций родительской пары можно отметить, что, как уже было сказано наибольшим потенциалом получить поводыря обладает пара 860-758, однако вероятность его не получить не является самой низкой. Наименьшей вероятностью не получить поводыря обладают пары 627-489, 654-680, 654-866. Но пары 627-489 и 654-680 сложно проанализировать объективно, так как по ним есть всего одно наблюдение (оно дало поводыря), из-за чего их результаты не будут рассматриваться в выводах. Худшей парой для получения поводыря является пара 654-957 (из 16 наблюдений не получено ни одного поводыря).

Для анализа важности характеристик собак построим тепловую карту весов между входным и скрытым слоями. Чем больше значение веса, тем большее влияние на работу нейронной сети оказывает соответствующая ему входная характеристика собаки-поводыря (рисунок).

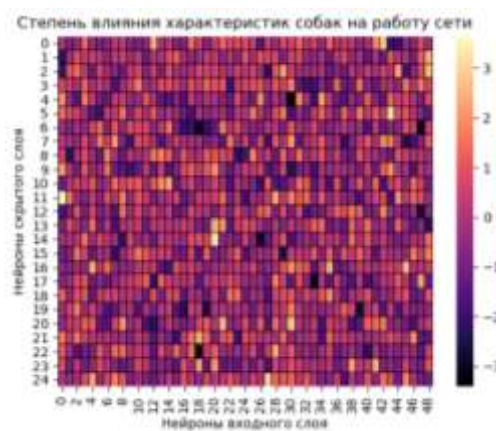


Рис. 1. Тепловая карта влияния входных характеристик на скрытый слой

По тепловой карте можно определить степень важности того или иного признака. На ней значение, находящееся по координатам (input\_layer, hidden\_layer) показывает влияние входного нейрона на соответствующий нейрон скрытого слоя (чем светлее цвет, тем больше оказываемое влияние). Соответственно, самый светлый столбец показывает самую важную характеристику, а самый темный – наоборот. Для удобства значения наиболее/наименее важной величины выводятся в консоль в процессе работы программы. В данном случае,

по диаграмме можно определить, что наиболее светлый цвет у столбца 20, а наиболее темный – у 37. Это значит, что в данном случае наиболее важной характеристикой собаки-поводыря является суета/беспокойство пока хозяин бездействует, а наименее значимой – незначительные отклонения от заданного пути по время работы.

Эта диаграмма также может менять значения из-за того, что начальные веса и смещения нейронной сети инициализируются случайным образом. Но в результате множества запусков наиболее важными характеристиками оказались нюх, суета/беспокойство пока хозяин бездействует или боязнь лифтов, а наименее важными являются рассеянность собаки, способность защитить хозяина.

Причем других наиболее/наименее значимых признаков нейронная сеть не выдала ни разу. Это объясняется тем, что среднее значение весов этих характеристик каждый раз отличается на небольшое значение, следовательно, при определенном изменении начальных весов можно получить один из этих результатов. Причем получить другие невозможно, так как начальные веса инициализируются в определенном диапазоне, и каждый раз отличаются на небольшие значения.

#### IV. ЗАКЛЮЧЕНИЕ

Была разработана программа для поиска родителей с наибольшей/наименьшей вероятностью получения в потомстве поводыря и обучения/тестирования нейронной сети для поиска наиболее/наименее важных характеристик для поводырей. Для нейронной сети были проведены тесты для поиска оптимальных параметров. Средняя точность разработанной нейронной сети составляет 86–88 %.

Несмотря на достаточно высокую точность результатов нейронной сети, она обладает и недостатками. Одним из них является неоднозначность. Из-за того, что начальные веса и смещения инициализируются случайными

значениями на выходе часто можно получать разные значения при одинаковых входных данных. Также недостатком является достаточно большое время обучения и тестирования.

С помощью одного из модулей программы было определено, от каких родителей по отдельности и как ких пар родителей наибольшая/наименьшая вероятность получить собак-поводырей. С помощью нейронной сети были выделены наиболее/наименее важные признаки собак для их пригодности к работе поводыря. Наиболее важными характеристиками оказались нюх, суета/беспокойство пока хозяин бездействует или боязнь лифтов, а наименее важными являются рассеянность собаки, способность защитить хозяина и незначительные отклонения от заданного пути по время работы.

Дальнейшие исследования будут направлены на изучения глубоких нейронных сетей с 4 и больше скрытыми слоями, а также на улучшение архитектуры сети. Результаты этих исследований могут помочь добиться точности результатов 95–97 %, что значительно повысит шанс заметить потенциального поводыря во время разведения.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Werner C.R., Gaynor R.C., Gorjanc G., Hickey J.M., Kox T., Abbadi A., ... & Stahl A. How Population Structure Impacts Genomic Selection Accuracy in Cross-Validation: Implications for Practical Breeding // *Frontiers in plant science*. 2020, vol. 11, p. 2028.
- [2] PyCharm 2019.3.1, URL: <https://blog.jetbrains.com/pycharm/2019/12/pycharm-2019-3-1/> (last accessed 18.03.2021).
- [3] Python Data Analysis Library, URL: <https://pandas.pydata.org/>, (last accessed 18.03.2021).
- [4] NumPy, URL: <https://numpy.org/>, (last accessed 18.03.2021).
- [5] Matplotlib: Python plotting — Matplotlib 3.3.4 documentation, URL: <https://matplotlib.org/>, (last accessed 18.03.2021).
- [6] seaborn: statistical data visualization, URL: <https://seaborn.pydata.org/>, (last accessed 18.03.2021).

# Нейросетевой подход к преодолению априорной неопределённости при оптимальном планировании действий интеллектуальных информационных агентов для мягких архитектур сервис-ориентированных систем

Л. К. Птицына<sup>1</sup>, Н. Н. Эль Сабаяр Шевченко<sup>2</sup>  
Санкт-Петербургский государственный университет  
телекоммуникаций им. проф. М.А. Бонч-Бруевича  
<sup>1</sup>ptitsina\_lk@inbox.ru, <sup>2</sup>nzs.vus@gmail.com

М. П. Белов  
Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)  
milesa58@mail.ru

А. В. Птицын  
Национальный исследовательский университет ИТМО  
pticin@inbox.ru

**Аннотация.** Обоснована необходимость развития мягких архитектур сервис-ориентированных систем. Выделены ключевые свойства мягких архитектур сервис-ориентированных систем. Обладание ключевыми свойствами связано с преодолением априорной неопределённости с помощью нейросетевых моделей. Описана функциональность мягких архитектур сервис-ориентированных систем с интеллектуальными информационными агентами. Показана открытость жизненного цикла мягких сервис-ориентированных систем по отношению к методологии проектирования планировщиков действий. Рассмотрена связь методологии с решением задачи выбора алгоритма планирования действий. Выявлены причины априорной неопределённости в знании относительно аналитической зависимости эффективности планирования от условий решаемой задачи и характеристик предметной области. Предложен нейросетевой подход к преодолению априорной неопределённости в знании при оптимальном планировании действий интеллектуальных информационных агентов для мягких архитектур сервис-ориентированных систем. Раскрыто содержание сквозной цепочки этапов реализации предложенного подхода.

**Ключевые слова:** неопределённость в знании, планирование, алгоритм, сервис-ориентированная система, мягкое конфигурирование, эффективность, параметрическое пространство, нейросетевая модель, оценивание

## I. АКТУАЛЬНОСТЬ

В условиях динамического изменения рынков труда, спроса на товары и услуги, окружающей среды и быстрого расширения научно-технологических достижений возрастает востребованность мягких архитектур сервис-ориентированных систем [1]. Представленные динамичные

изменения характеризуются широким разнообразием априорных неопределённостей.

Свойства подобных архитектур ассоциируются со способностями к оперативному и эффективному реагированию в контексте происходящих изменений при выполнении основного предназначения в условиях априорной неопределённости [2]. При концептуальном проектировании мягких архитектур сервис-ориентированных систем предусматривается использование нейросетевых моделей для преодоления априорной неопределённости.

Одно из перспективных направлений в развитии мягких архитектур ориентируется на применение интеллектуальных информационных агентов в процессе комплексирования сервисов. В этом случае состав и способы интеграции сервисов в мягкой архитектуре определяются посредством планирования действий по комплексированию сервисов [3, 4]. Согласно этому принципу в жизненный цикл мягких сервис-ориентированных систем помимо модельно-аналитического интеллекта [5, 6] требуется включить методологию проектирования планировщиков действий интеллектуальных информационных агентов.

Жизненный цикл мягких сервис-ориентированных систем с планируемой агентом конфигурацией расширяется формализациями жизненного цикла интеллектуальных информационных агентов. Для расширения потребуется решить задачу выбора алгоритма планирования действий. В жизненном цикле интеллектуальных информационных агентов выделяется множество алгоритмов планирования действий. Сравнительный анализ качества и эффективности алгоритмов планирования действий проводится по

результатам имитационных экспериментов и обобщения практического опыта применения интеллектуальных информационных агентов. В результате сравнительного анализа устанавливается, что ни один из известных алгоритмов не обладает неоспоримыми преимуществами по отношению к различным мерам их качества и эффективности. В связи с этим становится необходимым решить задачу выбора наилучшего алгоритма. С позиции исследования операций наилучший выбор обеспечивается при решении задачи оптимизации.

В настоящее время вопросы аналитического определения зависимости эффективности планирования действий интеллектуальных информационных агентов от характеристик окружающей среды и параметров задачи планирования при любом алгоритме остаются открытыми. Подобные обстоятельства предопределяют объективную необходимость преодоления этой неопределённости для опорного базиса алгоритмов в целях заполнения знаниями жизненного цикла мягких архитектур интеллектуальных сервис-ориентированных систем.

## II. ОСНОВНОЕ СОДЕРЖАНИЕ

В предлагаемом нейросетевом подходе к преодолению априорной неопределённости при оптимальном планировании действий интеллектуальных информационных агентов для мягких архитектур сервис-ориентированных систем выделяются следующие этапы:

1. Параметризация предметной области для планирования конфигурации интеллектуальной сервис-ориентированной системы.
2. Определение критерия эффективности планирования конфигурации интеллектуальной сервис-ориентированной системы.
3. Отбор входной информации для нейронной сети, преодолевающей априорную неопределённость относительно зависимости эффективности планирования действий по интеграции сервисов в мягкой архитектуре сервис-ориентированной системы от условий решаемой задачи и характеристик предметной области.
4. Выбор нейросетевой модели для преодоления априорной неопределённости относительно зависимости эффективности планирования действий по интеграции сервисов в мягкой архитектуре сервис-ориентированной системы от условий решаемой задачи и характеристик предметной области.
5. Выбор метода обучения нейросетевой модели.
6. Оценка критерия эффективности планирования конфигурации интеллектуальной сервис-ориентированной системы.
7. Выбор оптимального алгоритма планирования действий интеллектуального информационного агента для мягкой архитектуры сервис-ориентированных систем.

На первом этапе предлагаемого подхода при параметризации предметной области считаются известными множество операторов предметной области и множество условий предметной области.

Планировщик действий интеллектуального информационного агента в мягкой сервис-ориентированной системе реализуется в соответствии с условиями решаемой задачи, характеристиками предметной области и выбираемым критерием оптимальности.

Качество планирования действий по интеграции сервисов в мягкой архитектуре находится в непосредственной зависимости от характеристик окружающей среды  $\mathbf{I}_E$  и параметров задачи планирования  $\mathbf{I}_D$ .

Характеристики окружающей среды определяются в результате обработки множества операторов предметной области и множества условий предметной области. В задачах планирования действий интеллектуальных информационных агентах выделяются следующие характеристики:

- $N_{op}$  – общее количество доступных действий;
- $N_c$  – общее количество условий;
- $N_{pre}$  – среднее количество предусловий одного действия;
- $N_{add}$  – среднее количество добавляемых условий для одного действия;
- $N_{del}$  – среднее количество удаляемых условий для одного действия;
- $n_+$  – среднее количество действий, добавляющих условие;
- $n_-$  – среднее количество действий, удаляющих условие;
- $H_+$  – количество высокочастотных условий с большими  $n_+$ ;
- ;
- $H_-$  – количество высокочастотных условий с большими  $n_-$ ;
- ;
- $h_+$  – плотность высокочастотных условий с большими  $n_+$ ;
- $h_-$  – плотность высокочастотных условий с большими  $n_-$ .

По результатам обработки формируется множество:

$$\mathbf{I}_E = \{N_{op}, N_c, N_{pre}, N_{add}, N_{del}, n_+, n_-, H_+, H_-, h_+, h_-\}.$$

Задача планирования характеризуется  $N_g$  общим количеством условий в целевом множестве целевого состояния. В таком случае  $\mathbf{I}_D = \{N_g\}$ .

На втором этапе предлагаемого подхода вводится понятие оптимальности алгоритма планирования действий по интеграции сервисов в мягкой архитектуре сервис-ориентированной системы, его возможные трактовки и соответствующие им формальные критерии.

На втором этапе различаются два рода задач в зависимости от способа оценки значения критерия эффективности.

При решении задачи первого рода на втором этапе оценивается непосредственно  $\xi_v$  значение информативной составляющей показателя эффективности  $v$ .

При решении задачи второго рода на втором этапе оценивается  $\xi_v'$  выполняемость условия  $\varepsilon$ -оптимальности

алгоритма планирования относительно информативной составляющей показателя эффективности  $v$ .

$$\xi'_v = \begin{cases} 1, & v \leq 1 + \varepsilon; \\ 0, & v > 1 + \varepsilon. \end{cases}$$

При решении задачи и первого рода, и второго рода предусматривается использование различных представлений исходной информации для нейросетевой модели. Вариативы в представлениях базируются на определении параметрического пространства описания алгоритмического обеспечения планировщиков действий интеллектуальных алгоритмов планирования действий по интеграции сервисов в мягкой архитектуре сервис-ориентированной системы.

Формирование вариаций в определении критерия эффективности осуществляется на основе теории исследования операций и различении типа задач планирования. В этом процессе выделяются автономное планирование, оперативное планирование и распределенное планирование.

При автономном планировании определение плана реализуется независимо от состояния текущего процесса функционирования интеллектуального агента и сервис-ориентированной системы.

При оперативном планировании время определения плана ограничивается из-за функциональных особенностей агента и сервис-ориентированной системы и постоянных изменений в среде или условиях задачи.

При распределенном планировании обеспечивается возможность параллельного поиска по нескольким ветвям дерева решений.

На третьем этапе предлагаемого подхода выполняется отбор входной информации для нейронной сети при преодолении априорной неопределённости относительно зависимости эффективности планирования действий по интеграции сервисов в мягкой архитектуре сервис-ориентированной системы от условий решаемой задачи и характеристик предметной области. Отбор осуществляется посредством статистической проверки гипотез.

Рассматриваются две альтернативные гипотезы: 1) множество случайных величин  $z$  и случайная величина  $\xi_v$  зависимы, 2) множество случайных величин  $z$  и случайная величина  $\xi_v$  независимы.

При определении множества случайных величин позиционируются два варианта. Первый вариант ориентируется на использование  $I_A$  объединения характеристик окружающей среды  $I_E$  и параметров задачи планирования  $I_L$ . Второй вариант формируется в итеративном процессе выявления  $I_V^*$  возможного усечения полного множества. При усечении отбрасываются те характеристики, которые не оказывают существенного влияния на эффективность планирования.

На четвертом этапе при выборе нейросетевой модели для преодоления априорной неопределённости относительно зависимости эффективности планирования

действий по интеграции сервисов в мягкой архитектуре сервис-ориентированной системы от условий решаемой задачи и характеристик предметной области позиционируется опорный вариант.

Опорный вариант ориентируется на простейшую нейросетевую модель  $R_N(x)$  – модель сети прямого распространения с одним скрытым слоем и единственным выходным элементом. Во входном слое выполняется преобработка исходной информации. При определении исходной информации рассматриваются два варианта. При первом варианте в качестве исходной информации представляются характеристики окружающей среды  $I_E$  и параметры задачи планирования действий по интеграции сервисов  $I_L$  в полном объёме. При втором варианте в качестве исходной информации используется усечённый набор характеристик окружающей среды  $I_E$  и параметров задачи планирования действий по интеграции сервисов  $I_L$ . Усечённый набор формируется в результате реализации второго этапа. Количество узлов во входном слое варьируется размерности соответствующих наборов.

В скрытый слой включаются узлы с задаваемой функцией активации  $A(x)$ . В качестве функции активации предпочтение отдается гиперболическому тангенсу:

$$A(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

При определении размера скрытого слоя  $N_{hid}$  вводятся формальные правила. В составе формальных правил предусматривается и применение двух простейших вариантов определений:

При первом варианте размер скрытого слоя  $N_{hid}$  определяется на основании размеров  $N_{in}$  входного и выходного  $N_{out}$  слоев

$$N_{hid} = N_{in} \text{ или } N_{hid} = [0.5(N_{in} + N_{out})],$$

где  $[X]$  – операция определения ближайшего к  $X$  целого числа.

При втором варианте размер скрытого слоя  $N_{hid}$  находится через преобразование  $\Pi_L$  мощности обучающего множества

$$N_{hid} = [0.1 \Pi_L].$$

В выходной слой вводится один узел, обеспечивающий постпроцессинг результатов в зависимости от характера решаемой задачи.

На пятом этапе предлагаемого подхода при настройке параметров нейросетевого описания отдается предпочтение процедуре обучения нейросетевой модели на основе минимизации функции ошибки с использованием обобщенного метода оптимизации с применением квази-ньютоновского алгоритма. Процесс обучения заканчивается в том случае, когда относительное изменение функции ошибки не превосходит устанавливаемый уровень на протяжении определённого числа последовательных итераций.

На шестом этапе предлагаемого подхода оценивание критерия эффективности планирования конфигурации интеллектуальной сервис-ориентированной системы выполняется посредством нейросетевого моделирования. При этом оценивание проводится для решения представленной выше задачи первого и второго рода.

На седьмом этапе оптимальность алгоритма планирования действий интеллектуального информационного агента по интеграции сервисов в мягкой архитектуре сервис-ориентированных систем определяется с помощью целевого функционала. Целевой функционал образуется суммой взвешенных относительных значений показателей качества планирования действий. Взвешивание проводится путём умножения на весовые коэффициенты, отражающие важность каждого показателя в конкретном типе задачи планирования действий.

### III. ЗАКЛЮЧЕНИЕ

Научная новизна в планировании действий интеллектуальных информационных агентов для мягких архитектур сервис-ориентированных систем на основе нейросетевого подхода состоит в следующем:

- определена сквозная цепочка этапов преодоления априорной неопределённости относительно знания зависимости эффективности планирования действий интеллектуальных информационных агентов по конфигурированию сервис-ориентированных систем от характеристик окружающей среды и параметров задачи планирования;
- формализован процесс формирования исходной информации для нейросетевой модели для оценивания критерия эффективности планирования действий интеллектуальных информационных агентов по конфигурированию сервис-ориентированных систем в зависимости от характеристик окружающей среды и параметров задачи планирования;
- установлена сквозная связь процедур проверки статистических гипотез, нейросетевого моделирования и оптимального выбора алгоритма планирования действий интеллектуальных информационных агентов по конфигурированию сервис-ориентированной системы с учётом характеристик окружающей среды и параметров задачи планирования;
- обеспечено мягкое планирование конфигурации интеллектуальных сервис-ориентированных систем за счёт оптимизации планировщиков интеллектуальных информационных агентов.

Практическая значимость нейросетевого подхода к преодолению априорной неопределённости при оптимальном планировании действий интеллектуальных информационных агентов для мягких архитектур сервис-ориентированных систем предопределяется следующими обстоятельствами:

- предоставлена возможность снижения сложности нейросетевого описания поведения критерия эффективности планирования действий интеллектуальных информационных агентов по конфигурированию сервис-ориентированных систем с учётом характеристик окружающей среды и параметров задачи планирования;
- обеспечена возможность альтернативного оценивания эффективности планирования действий интеллектуальных информационных агентов по конфигурированию сервис-ориентированных систем в зависимости от характеристик окружающей среды и параметров задачи планирования на основе нейросетевого моделирования.

### СПИСОК ЛИТЕРАТУРЫ

- [1] Птицына Л.К., Кондратьев Д.А., Эльсабаяр Шевченко Н.Н. Интеллектуальные профили сервис-ориентированных архитектур // Труды учебных заведений связи. 2016. Т.2, № 2. С. 72-77.
- [2] Птицына Л.К., Кондратьев Д.А., Эльсабаяр Шевченко Н.Н. Концептуальные модели интеллектуализации сервис-ориентированных архитектур // Актуальные проблемы инфотелекоммуникаций в науке и образовании. V Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 3 т.; Т.2 / под ред. С.В. Бачевского, сост. А.Г. Владыко, Е.А. Аникевич, Л.М. Минаков. СПб.: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2016. С. 108-113.
- [3] Птицына Л.К., Кондратьев Д.А., Эльсабаяр Шевченко Н.Н. Выбор алгоритма планирования для интеллектуальных сервис-ориентированных систем // Актуальные проблемы инфотелекоммуникаций в науке и образовании. VI Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4 т. / Под ред. С.В. Бачевского, сост. А.Г. Владыко, Е.А. Аникевич. СПб.: СПбГУТ, 2017. Т. 3. С. 277-282.
- [4] Ptiitsyna L.K., Shevchenko N.E.S., Belov M.P., Ptiitsyn A.V. Planning Architecture of Service-oriented Systems under Uncertainty // Proceedings of 2020 23rd International Conference on Soft Computing and Measurements, SCM 2020, 2020, с. 101-104, 9198763
- [5] Птицына Л.К. Методология генерации модельно-аналитического интеллекта сервис-ориентированных систем с гарантиями качества // Актуальные проблемы инфотелекоммуникаций в науке и образовании. VI Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4 т. / Под ред. С.В. Бачевского, сост. А.Г. Владыко, Е.А. Аникевич. СПб.: СПбГУТ, 2017. Т. 3. С. 351-354.
- [6] Эльсабаяр Шевченко Н.Н. Генерация модельно-аналитического интеллекта сервис-ориентированных систем // Труды учебных заведений связи. 2020. Т. 6. № 1. С. 109-116.



# Подход к трансляции базы данных с MySQL в Cassandra

В. М. Ха<sup>1</sup>, Ю. А. Шичкина<sup>2</sup>

*Кафедра вычислительных техник*

*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)*

<sup>1</sup>muon.ha@mail.ru, <sup>2</sup>strange.y@mail.ru

**Аннотация.** В этой статье рассматривается подход к проблеме трансляции данных с базы данных MySQL в базу данных Cassandra. Подход заключается в построении модели на основе графа для описания общей схемы базы данных для реляционной схемы и схемы NoSQL. Описываются правила выбора схемы семейства столбцов на основе графа. Представлен метод определения структуры базы данных Cassandra с учетом выбранных схем семейств столбцов. В конце статьи приводятся результаты тестирования подхода к трансляции базы данных MySQL в формат Cassandra, которые показали его эффективность и возможность применения на практике.

**Ключевые слова:** MySQL, Cassandra, база данных, схема базы данных, преобразование схемы данных

## I. ВВЕДЕНИЕ

Сбор, консолидация и хранение больших объемов информации быстро становятся необходимостью среди разработчиков программного обеспечения, а также в научно-исследовательской деятельности в области баз данных. До сих пор реляционная база данных представляла собой наиболее часто используемый вид хранения данных. Однако реляционные базы данных (РБД) имеют фиксированную схему и позволяют хранить и обрабатывать структуры данных только в виде наборов или отношений [1]. РБД обеспечивает только вертикальную масштабируемость при более высокой стоимости аппаратных средств, но без горизонтальной масштабируемости. Вертикальная масштабируемость позволяет только вертикальный рост в структурах данных, добавляя только новые записи во время выполнения. Горизонтальная масштабируемость позволяет добавлять поля во время выполнения. Она необходима в современных программных приложениях для обработки высокоскоростных гетерогенных данных.

В настоящее время, РБД приходится сталкиваться с различными проблемами на уровне разработки приложений и оперативном уровне. На уровне разработки приложений разработчики системы нуждаются в высокой скорости кодирования для обработки большого числа пользователей; тем не менее, эта возможность недоступна в реляционных базах данных.

NoSQL был сформирован, чтобы преодолеть недостатки реляционной базы данных. NoSQL был введен в 1998 году Карло, и это название было дано его решению для реляционной базы данных из-за того, что не

использовался язык структурированных запросов (Structured Query Language – SQL) [2]. Идея NoSQL была пересмотрена в 2009 году и стала конкурентом РБД. Теперь он стал основой крупных предприятий, таких как Google, Twitter, Facebook, Amazon и т. д. из-за его особенностей, таких как высокая доступность (когда данные автоматически распределены равномерно по всему кластеру, без единого хоста), высокая производительность, горизонтальная масштабируемость, а также поддержка различных моделей данных и запросов.

Многие организации обращаются к базе данных NoSQL и используют ее для хранения и управления данными, а реляционная база данных существующих приложений должна быть перенесена в базу данных NoSQL. Однако схемы данных полностью отличаются между этими двумя видами баз данных. Кроме того, операция «join» не поддерживается в базе данных NoSQL, и требуется отдельно несколько операций для объединения. Это вызывает плохую производительность.

Для трансляции базы данных важно знать схему реляционной базы данных и схему базы данных NoSQL и способ, с помощью которого будет сделано импортирование данных из реляционной схемы на NoSQL. Кроме того, очень важно обеспечить высокую эффективность чтения данных после преобразования схемы. Хотя есть различные реляционные базы данных, такие как Oracle, SQL Server, MySQL, все они имеют одинаковую реляционную схему. Однако каждая база данных NoSQL имеет собственную схему данных [3]. Существует четыре основных категории баз данных NoSQL: (1) Ключ – значение, (2) Семейство столбцов, (3), Документно-ориентированная, и (4) Графовая. Перенос данных из SQL в NoSQL является актуальной и трудной задачей, над которой трудятся многие исследователи в области баз данных. Все решения проблемы ускорения обработки данных можно разделить на основные группы:

- денормализация: авторы в [4] предлагают основу для систематической денормализации базы данных как способ улучшения производительности данных и снижения времени обработки без потери целостности данных. В [5] авторы реструктурировали данные веб-приложения в независимые службы передачи данных, а улучшение пропускной способности достигли по сравнению с обычным подходом «ведущий –

ведомый» путем добавления репликации и масштабирования отдельных фрагментов данных;

- трансляция данных: многие инструменты были разработаны для трансляции данных из реляционной базы данных в базу данных NoSQL, например: Apache Sqoop [6] и Datax [7];
- преобразование схемы данных [8], [9].

Таким образом, существующие исследования в основном сосредоточены на улучшении производительности чтения или трансляции данных в базу данных NoSQL для получения высокой доступности и масштабируемости, но не предоставляют эффективное решение для устранения проблемы системных зависимостей между таблицами в реляционной базе данных или трансформации отношений между таблицами в базе данных NoSQL. Запрос должен выполнять несколько таблиц для получения данных, которые изначально должны присоединиться к более чем одной таблице в реляционной базе данных, что приводит к низкой эффективности запроса.

## II. ТРАНСЛЯЦИЯ ДАННЫХ С MYSQL В CASSANDRA

В этой главе представлен графовый алгоритм для определения схемы базы данных на Cassandra. Прежде всего, мы предоставляем графовую модель для описания общей схемы базы данных, включая реляционную схему и схему NoSQL. Через эту модель мы представляем процедуру трансляции схемы, последовательность операций расширения, которые преобразуют оригинальную схему в окончательную схему.

### A. Графовая модель для реляционной базы данных

Основные определения:

DB – реляционная база данных;

$T = \{t_1, t_2, \dots, t_i \mid 1 \leq i \leq n\}$  – T множество таблиц РБД,  $t_i$  – таблица  $i$  и  $n$  – количество таблиц РБД.

$V_i$  – 1 вершина графа, соответствует одну таблицу в базе данных;

$V = \{V_1, V_2, \dots, V_i \mid 1 \leq i \leq n\}$ : множество вершин графа, соответствует всех таблиц РБД;

$e = \langle V_i, V_j \rangle$ : 1 рёбра между вершинами  $V_i, V_j$  соответствует связь между таблицами  $t_i$  и  $t_j$ ;

$E = \{\langle V_i, V_j \rangle \mid 1 \leq i, j \leq n, i \neq j\}$ : Множество рёбер графа соответствует множество связей между таблицами РБД.

$G = \langle V, E \rangle$  – Графовая модель реляционной базы данных.

### B. Правила расширения для отношений

#### a) Простое расширение

Простое расширение является просто процедурой миграции из реляционной таблицы в NoSQL, в данном случае отношение относится только к одному отношению

или к никакому. Правило простого расширения показано на рис. 1:

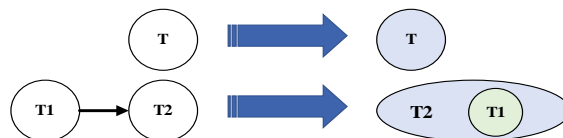


Рис. 1. Простое расширение

#### b) Горизонтальное расширение

Определить  $n$  как глубину отношения зависимости.

- Если  $n=1$  в этом случае в реляционной схеме есть только 2 отношения, то вернуться к случаям простого расширения.
- Если  $n>1$  в случае в реляционной схеме есть более двух отношений и их связи построены горизонтально, то процесс расширения выполняется следующим образом:

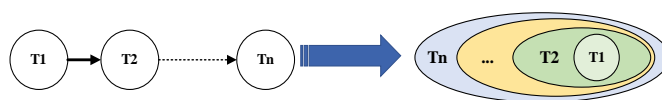


Рис. 2. Горизонтальное расширение

#### c) Вертикальное расширение

Пусть  $n$  – число отношений, которые прямо ссылаются к одному отношению.

- Если  $n=1$  вернуться к случаям простого расширения.
- Если  $n>1$ , то есть отношения S одновременно ссылаются к  $T_1, T_2, \dots, T_n$ , тогда процесс расширения выполняется следующим образом:

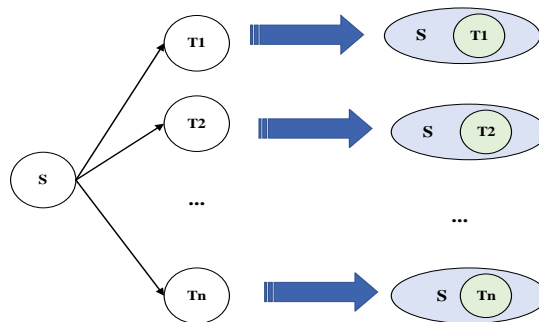


Рис. 3. Вертикальное расширение

### C. Алгоритм определения модели данных на Cassandra

Путь реляционной схемы базы данных представлен в ориентированной графовой модели, которая показана в разделе A). Тогда конечными точками являются вершины с нулевой выходной степенью, начальные точки – это вершины с нулевой входной степенью. На этом этапе, чтобы найти модель хранения данных на Cassandra на основе теории графов, из правил, изложенных в предыдущем разделе, алгоритм должен найти все пути от

конечных точек до начальных точек графа. Количество найденных путей равно количеству таблиц в Cassandra.

Конкретный алгоритм выглядит следующим образом:

**Входные данные:** Схема базы данных на MySQL

**Выходные данные:** Множество путей

```
start_vercites ← {};
end_vercites ← {};
S ← {};
```

1. Получить структуру схемы базы данных из MySQL;
  2. Преобразовать структуры схемы в ориентированную графовую модель  $G = \langle V, E \rangle$ ;
  3. Вычислить входные степени  $deg^+(g)$  и выходные степени  $deg^-(g)$  вершины;
  4. For  $g$  в  $G$
  5.     If  $deg^+(g) = 0$  do
  6.          $end\_vercites.add(g)$ ;
  7.     If  $deg^-(g) = 0$  do
  8.          $start\_vercites.add(g)$ ;
  9.     For end в  $end\_vercites$  :
  10.         Найти все пути  $S'$  с end до start ;
  11.          $S.add(S')$
- Вывод S;

### III. ЭКСПЕРИМЕНТ

База данных, используемая для тестирования, получена из системы управления вызовами и сообщениями телекоммуникационного бизнеса. Она содержит 5 таблиц и 27 атрибутов. Схема базы данных представлена на рис. 4:



Рис. 4. Схема тестовой базы данных на MySQL

Данная схема может быть переписана таким образом:

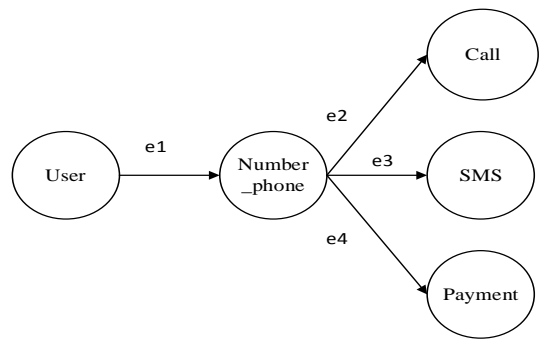


Рис. 5. Схема тестовой базы данных в виде графа

Из данной схемы:

```
T = {User, Number_phone, Call, SMS, Payment}
V = {V_User, V_Number_phone, V_Call, V_SMS, V_Payment}
E = { < V_User, V_Number_phone >, < V_Number_phone, V_Call >,
      < V_Number_phone, V_SMS >, < V_Number_phone, V_Payment > }
G = < {V_User, V_Number_phone, V_Call, V_SMS, V_Payment}
      , { < V_User, V_Number_phone >, < V_Number_phone, V_Call >,
          < V_Number_phone, V_SMS >, < V_Number_phone, V_Payment > } >
```

Применение графового алгоритма, найденное множество путей:

$$S1 = \{V_{Call}, V_{Number\_phone}, V_{User}\}$$

$$S2 = \{V_{Payment}, V_{Number\_phone}, V_{User}\}$$

$$S3 = \{V_{SMS}, V_{Number\_phone}, V_{User}\}$$

Следует структура данных, сохраняющихся на Cassandra:

$$Call\_view = \{Call \cup Number\_phone \cup User\}$$

$$Payment\_view = \{Payment \cup Number\_phone \cup User\}$$

$$SMS\_view = \{SMS \cup Number\_phone \cup User\}$$

Физическая схема базы данных на Cassandra показана на рис. 6:



Рис. 6. Физическая схема базы данных на Cassandra

Тестирование производительности выполнения запроса проводилось на 3-х объемах данных:

**Объем 1:** User(10.000), Number\_phone(10.000), Payment(25.000), SMS(43.000), Call(32.000);

**Объем 2:** User(50.000), Number\_phone(50.000), Payment(65.000), SMS(81.000), Call(32.000);

**Объем 3:** User(100.000), Number\_phone(100.000), Payment(123.000), SMS(230.000), Call(320.000).

Количество запросов выполнения – 5 запросов. Для обеспечения объективности в процессе проверки каждый запрос выполняется 25 раз, а время вычислялось в среднем. Время измерялось в миллисекундах. Результаты выполнения запросов приведены в таблице.

ТАБЛИЦА I РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЗАПРОСОВ

Запрос	Объем 1		Объем 2		Объем 3	
	MySQL	Cassandra	MySQL	Cassandra	MySQL	Cassandra
Q1	70.64	65.83	87	84.69	183.56	176.87
Q2	120.4	117.29	147	131.66	720.6	699.7
Q3	4.56	3.87	163.6	145.83	273.76	286.58
Q4	129.57	121.5	84.96	74.3	722.5	711.55
Q5	3.5	3.15	161.6	148.33	274.88	262.77

График результата выполнения запросов показан на рис. 7:



Рис. 7. График результата для третьей объемы

Из результата выполнения запроса видно, что разница между временем выполнения запросов Cassandra и MySQL незначительна. Точность результатов может быть гарантирована, так как запросы, выполненные на Cassandra и MySQL, были одинаковыми.

#### IV. ЗАКЛЮЧЕНИЕ

Таким образом, в данной статье представлен подход трансляции данных из MySQL в Cassandra. Результаты тестирования показали, что разработанный процесс достиг базовой эффективности. Наши результаты могут быть использованы для решения задачи консолидации различных типов баз данных. В следующих исследованиях авторы рассмотрят проблему оптимизации структуры базы данных Cassandra.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] E. Codd, "Relational Completeness of Data Base Sublanguages," in R. Rustin, ed., Data Base Systems, Englewood Cliffs, NJ: Prentice-Hall, 1972.
- [2] Scherzinger S, Klettke M, Störl U. Managing Schema Evolution in NoSQL Data Stores[J]. arXiv preprint arXiv:1308.0514, 2013.
- [3] X. Li, Z. Ma, and H. Chen, "QODM : A Query-Oriented Data Modeling Approach for NoSQL Databases," pp. 338–345, 2014.
- [4] Yma P. A Framework for Systematic Database Denormalization[J]. Global Journal of Computer Science and Technology, 2009, 9(4).
- [5] Wei Z, Dejun J, Pierre G, et al. Service-oriented data denormalization for scalable web applications[C]//Proceedings of the 17th international conference on World Wide Web. ACM, 2008: 267-276.
- [6] Apache Software Foundation (2021). Available at: <http://sqoop.apache.org/> (accessed 2 March 2021).
- [7] DataX Wiki (2021). Available at: <http://code.taobao.org/p/datax/wiki/index/> (accessed 2 March 2021).
- [8] Chung W C, Lin H P, Chen S C, et al. JackHare: a framework for SQL to NoSQL translation using MapReduce[J]. Automated Software Engineering, 2013: 1-20.
- [9] Li C. Transforming relational database into HBase: A case study[C]//Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on. IEEE, 2010: 683-687.

# Проектирование структуры нейросети для прогнозирования заболеваемости злокачественными новообразованиями

М. В. Жилкина<sup>1</sup>, И. Р. Шулунова<sup>2</sup>  
ФГБОУ ВО «Восточно-Сибирский государственный  
университет технологий и управления»  
<sup>1</sup>zhilkina.1993@bk.ru, <sup>2</sup>rubinovna@mail.ru

А. Э. Федоров  
ГБУЗ «Бурятский республиканский клинический  
онкологический диспансер»  
Alexfedorerx@gmail.com

**Аннотация.** Основными причинами смертности в Российской Федерации являются сердечно-сосудистые заболевания, внешние причины и злокачественные новообразования. Исследование и разработка методов, подходов и инструментария для прогнозирования заболеваемости и смертности от злокачественных новообразований являются актуальными, так как позволяют разрабатывать корректирующие мероприятия по профилактике и предупреждению развития онкологических заболеваний.

В статье рассматриваются подходы построения нейросетевых моделей распространения опасных инфекционных заболеваний животных, прогнозирования инфекционной заболеваемости и возрастных коэффициентов смертности. Предложена структура нейросети для прогнозирования заболеваемости злокачественными новообразованиями.

**Ключевые слова:** искусственный интеллект; нейронная сеть; прогнозирование; онкологические заболевания; Республика Бурятия

## I. ВВЕДЕНИЕ

Нейросети являются популярным средством анализа данных, работы с изображениями и видеоинформацией, а также при решении задач прогнозирования и являются хорошим дополнением классическим методам анализа.

Смертность от злокачественных новообразований (ЗНО) в Российской Федерации занимает второе место и прогнозные расчеты заболеваемости ЗНО в целом, а также по отдельным органам, имеют важное значение для формирования профилактических мер и принятия управленческих решений. Федеральный проект «Борьба с онкологическими заболеваниями» направлен на снижение смертности от ЗНО [1], таким образом, прогнозирование заболеваемости и смертности от онкозаболеваний является актуальным.

В работе рассмотрены несколько подходов построения нейронных сетей прогнозирования распространения инфекционных заболеваний и заболеваемости ими, а также предложена структура нейросети для прогнозирования заболеваемости злокачественными новообразованиями.

## II. ОСНОВНАЯ ЧАСТЬ

Исследование применения нейросетевого моделирования для прогнозирования временных рядов направлено на определение сложных скрытых связей между входными и выходными параметрами.

Алгоритм проектирования нейросети:

- выделение факторов, оказывающих существенное влияние на моделируемый процесс;
- ранжирование показателей, чтобы каждому можно было поставить в соответствие определенное число;
- выделение выборки для обучения нейросети;
- обучение модели, корректировка структуры сети;
- проверка адекватности, полученной нейросетевой модели [4].

В работе Балбаровой Д.Г. и др. исследована возможность применения нейросетевого моделирования для анализа эпидемиологического уровня инфекционного заболевания свиней в Республике Бурятия [2].

Входные параметры модели, в которой:

- пространственные (площадь территории районов и площади сельскохозяйственных угодий и пастбищ);
- природные (количество дней в году с ветром более 15м/с, сумма среднемесячных температур; среднегодовое количество осадков);
- предметные (поголовье свиней и его плотность).

Расчеты модели показали, что повышенный уровень инфекционного заболевания свиней в Республике Бурятия в районах с большим числом дней в году с ветром более 15 м/с, с малой площадью территории и большим поголовьем.

В [8] авторы рассмотрели 12 территориальных образований Приволжского Федерального округа и проанализировали уровень заболеваемости инфекционными патологиями, в качестве исходных

данных использованы помесечные значения за период 2010–2017 гг.

Этапы прогнозирования в [8]:

- декомпозиция уровня заболеваемости (тренд сезонной и случайной компонент);
- раздельное моделирование тренда и сезонной компоненты;
- итоговое моделирование уровня заболеваемости (сложения моделей тренда и сезонной компоненты).

Показана эффективность применения смешанной техники прогнозирования.

Горлищев В.П. и Михальский А.И. в своем исследовании предложили новый подход к прогнозированию смертности с помощью нейронной сети. Для построения модели использован метод кросс-валидации с учетом временной структуры данных [3].

Структура нейронной сети в [3]:

- входной слой состоит из 106 нейронов (логарифм смертности по полу и в определенной стране в одногодичных возрастных группах от 0 до 105 лет);
- скрытый слой из 300 нейронов;
- выходной слой из 106 нейронов (одногодичные возрастные группы диапазона от 0 до 105 лет для последующего года).

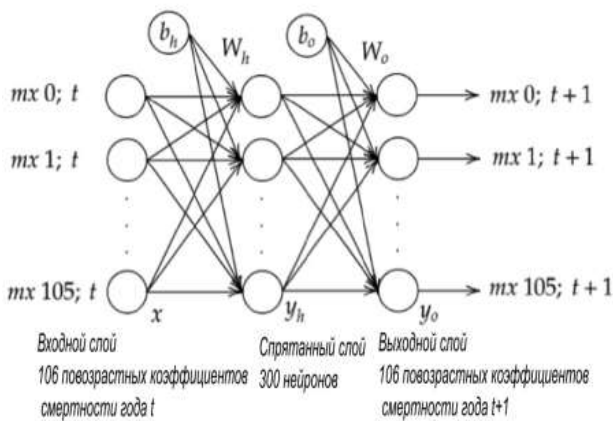


Рис. 1. Архитектура нейронной сети для прогнозирования смертности

Для обучения и оценки модели использованы данные Западной Германии, Испании, Италии, в этих странах зафиксирована низкая смертность при высокой численности населения в период с 1950 по 2016 год. Обучение нейронной сети проведено на основе данных 1950–2011 гг., разделение по полу, году и региону не производилось. Исследование проведено путем построения 50 нейронных сетей, для обучения порядок данных разных стран и периодов выбирался случайным образом.

В итоге применение алгоритма нейронной сети позволило получить плавный и реалистичный прогноз, с минимальным отличием между прогнозируемыми реальными данными.

Для прогнозирования заболеваемости ЗНО предложена трехслойная структура нейронной сети: входной слой – 8 нейронов, скрытый слой, выходной слой – 10 нейронов (рис. 2).

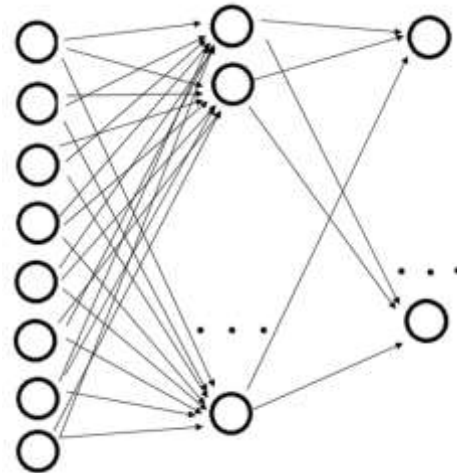


Рис. 2. Структура нейронной сети для прогнозирования заболеваемости онкозаболеваниями

Факторы входного слоя:

- Онконаследственность (0/1)
- Курение (0/1)
- Возраст (0..100)
- Пол (0/1)
- Ожирение (0/1/2)
- Сопутствующие заболевания
  - Болезни легких (0)
  - Болезни крови (1)
  - Болезни нервной системы (2)
  - ...
- Акушерский анамнез (у женщин)
  - Количество беременностей (0/1/2/...)
  - Количество родов (0/1/2/...)
  - Количество аборт (0/1/2/...)
  - Кормление грудью (0/1)
- Работа
  - Нормированный день (0)
  - Сменная работа (1)
  - Вахта (2)
- Образование

- Неполное среднее/среднее (0)
- Среднее-специальное/Начальное специальное (1)
- Высшее/Неполное высшее (2)
- Район Бурятии (и города Улан-Удэ)
  - Районы (0-21)
  - Левый берег (22)
  - Сотые квартала (23)
  - Восточный (24)
  - Энергетик(25)
  - Центр (26)
  - Элеватор (27)
  - Квартала (28).

В качестве входных факторов, отслеживающих географическое положение, рассматриваются районы Республики Бурятия и город Улан-Удэ, что позволит учесть влияние загрязнения воздуха на заболеваемость ЗНО в определенных районах.

Входные факторы определены из условия доступности статистической информации. Так, например, в истории болезни и диспансеризации фиксируется информация о росте и весе, что позволяет определить наличие ожирения, также эти документы содержат информацию о текущих заболеваниях.

Выходной слой содержит информацию о вероятности заболевания онкологией по определенным органам в следующем формате:

- Заболевание легких 74,5 %
- Заболевание головы 1,7 %
- ...

Для обучения нейросети используется выборка данных заболевших (ЗНО зарегистрировано впервые на 1 и 2 стадиях) в Республике Бурятия за период 2015–2020 гг.

### III. ЗАКЛЮЧЕНИЕ

Результаты нейросетевого моделирования заболеваемости ЗНО позволят получить прогноз, который поможет скорректировать профилактические мероприятия, такие как формирования культуры здорового образа жизни, контроль за выбросами канцерогенных веществ в окружающую среду на определенных территориях. Так же данные прогноза могут быть использованы для формирования расчета бюджета на высокотехнологичную медицинскую помощь и закуп лекарственных препаратов. Исследование нейросетевого моделирования планируется расширить построением нейросети для прогнозирования смертности от злокачественных новообразований

### СПИСОК ЛИТЕРАТУРЫ

- [1] Указ Президента Российской Федерации от 7 мая 2018 г. № 204 "О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года"
- [2] Балбарова Д.Г., Олзоева С.И., Садуев Н.Б. Нейросетевое моделирование в задачах прогнозирования распространения опасных инфекционных заболеваний животных // Вестник Бурятского государственного университета. Математика, информатика, 2015 г. С. 27-30.
- [3] Горлищев В.П., Михальский А.И. Применение нейронной сети в задаче прогнозирования возрастных коэффициентов смертности // Труды 13-й Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2020, Москва). 2020г. С. 1774-1778.
- [4] Зенкова Н.А. Нейросетевое моделирование в психологических и социальных исследованиях // Вестн. ТГУ. Сер. Естеств. и технич. науки. Тамбов, 2005. Т. 10. Вып. 1. С. 112-114.
- [5] Нагорных Д.Ю. Прогнозирование временных рядов с помощью гибридной нейронной сети // Труды XVII Международной конференции DICR-2019, Новосибирск. 2019г. С.135-138.
- [6] Осовский С. Нейронные сети для обработки информации. Москва: Финансы и статистика, 2002. 344 с.
- [7] Сергеев, А.П. С32 Введение в нейросетевое моделирование : учеб. пособие / А.П. Сергеев, Д.А. Тарасов; под общ. ред. А.П. Сергеева. Екатеринбург: Изд-во Урал. ун-та, 2017. 128 с.
- [8] Филатова Е.Н., Солнцев Л.А. Смешанная техника прогнозирования инфекционной заболеваемости с применением методов декомпозиции временного ряда и SARIMA // Журнал микробиологии, эпидемиологии и иммунобиологии. 2019 г. С.76-83.

# Элементная база узлов импульсных нейронных сетей

О. И. Буренева

Санкт-Петербургский государственный электротехнический  
университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

oibureneva@etu.ru

**Аннотация.** Важным вопросом при моделировании процессов обработки информации в биологических системах является кодирование информации нервными импульсами. Формой представления информации, наиболее близкой к биологической, являются импульсные потоки. Для построения элементов и узлов импульсной нейронной сети необходимо реализовать базисный набор элементов, способных выполнять различные арифметические и логические преобразования в импульсной форме. Разработанный набор элементов может найти применение при разработке нейрокомпьютеров, и технических систем на основе нейронных сетей, используемых для распознавания образов, анализа и обработки изображений и других приложений в сфере искусственного интеллекта.

**Ключевые слова:** цифровые импульсные преобразователи; FPGA; аппаратная реализация ИНС; импульсный нейрон; импульсный умножитель

## I. ВВЕДЕНИЕ

В последнее время наблюдается активное развитие в области аппаратной реализации искусственных нейронных сетей, поскольку использование программных подходов сталкивается с проблемой нехватки вычислительных мощностей современных компьютерных средств. Мощностной потенциал человеческого мозга составляет приблизительно 15Вт, а по оценкам максимально достижимые вычислительные способности мозга оцениваются в диапазоне от  $10^{16}$  до  $10^{28}$  FLOPS (FLOPS – количество операций с плавающей запятой в секунду), что превосходит возможности суперкомпьютеров [1].

Возникает задача разработки нового класса схем, которые могут имитировать поведение нейронов и характеризуются высокой активностью, простыми базовыми элементами, использованием распределенной памяти и параллельной организацией вычислений. Целью создания таких схем является не точное воспроизведение процессов работы мозга, а имитация некоторых особенностей его работы при использовании совсем иных вычислительных элементов.

На сегодняшний день создано большое количество тематических моделей искусственного нейрона: нейрон

---

Работа выполнена в СПбГЭТУ «ЛЭТИ» при финансовой поддержке Министерства науки и высшего образования Российской Федерации в рамках Соглашения № 075-11-2019-053 от 20. 11 2019 г. (на основании постановления Правительства Российской Федерации от 9 апреля 2010 г. № 218) по теме: «Создание отечественного высокотехнологичного производства систем безопасности автотранспорта на основе блока управления и интеллектуальных датчиков, включающих миллиметровые радары диапазона 76 – 77 ГГц»

МакКаллока-Питса, Ван-Дер Поля, Ходжкина-Хаксли, Маркина, Моррис-Лекара, Хиндмарша-Розе, Ижкевича, Хубера-Брауна и другие, а также их различные модификации [2–5]. Бинарные модели нейронов имеют следующие недостатки:

- сигналами биологического нейрона являются импульсные потоки, а не бинарные кодовые комбинации, как предусмотрено в модели;
- в биологическом нейроне не используются весовые градации входных сигналов и, соответственно, не требуется хранение весовых коэффициентов;
- нейронные сети на основе нейронов МакКаллока–Питса и его аналогов при большом количестве входных сигналов могут терять устойчивость, чего не случается в реальной биологической системе.

Альтернативой бинарным нейронным моделями являются модели импульсных нейронов [6–8].

Импульсная форма передачи информации является наиболее эффективной для описания модели нейрона. Анализ процессов управления движением, например, мышечной ткани показывает, что механизм управления сокращением мускульной ткани реализуется путем передачи информации между управляющим звеном и объектом с помощью импульсов, которые, как правило, имеют прямоугольную форму и малую длительность.

При передаче информации с использованием импульсных сигналов в качестве носителя информации могут выступать разные характеристики. В большинстве случаев носителем является средняя частота импульсов в потоке, однако такое представление мало информативно: различные по передаваемой последовательности импульсов потоки могут характеризоваться одинаковой средней частотой. Для расширения возможностей представления информации с помощью импульсной формы используются различные варианты модуляции или иные способы передачи информации с использованием времени [9, 10]:

- фазовая модуляция – информация о сигнале задается определенным положением импульсов относительно общей опорной временной точки;
- широтно-импульсная модуляция (ШИМ) – информация о сигнале задается отношением длительности импульса к его периоду;



- интервальные параметры – информация связывается с расстоянием между импульсами;
- резонансные характеристики – информация о сигнале задается плотной последовательностью импульсов, которая приводит к резонансу в узле сети, при этом одиночные импульсы не вносят никакого вклада в передаваемую информацию.

В рассмотренных вариантах импульсной передачи информации амплитуда импульсов одинакова и принимает значение условной единицы.

Целью исследования является разработка элементного базиса для обработки импульсных потоков, которые могут использоваться при построении моделей узлов (нейронов, синапсов) импульсных нейронных сетей.

## II. ОСНОВНЫЕ ЭЛЕМЕНТЫ ИМПУЛЬСНЫХ НЕЙРОННЫХ СЕТЕЙ

Для описания устройств в нейроинформатике выработан специальный базис [11], представленный набором элементарных устройств: сумматоры, синапсы, нейроны и другие. Эти элементы объединяются в сети, предназначенные для решения конкретных задач. При программной или аппаратной реализациях описание работы этих устройств переводятся на соответствующие языки программирования или на языки описания аппаратуры [12].

Обобщенная модель нейрона состоит из сумматора, нелинейного преобразователя и точки ветвления, соединенных, как показано на рис. 1.

Одним из основных элементов базиса является адаптивный сумматор. Он вычисляет скалярное произведение вектора входного сигнала  $X\{x_0, x_1, \dots, x_{n-1}\}$  на вектор параметров  $A\{a_0, a_1, \dots, a_{n-1}\}$ . При работе с импульсными представлениями данных сумматор может быть реализован в двух вариантах, отличающихся форматом представления параметров  $A$ .

В первом варианте функция импульсного умножения в адаптивном сумматоре реализуется двоичным множителем частоты. Он выполняет умножение двоичного кода  $N$  и частоты  $f$  и строится на основе счетчика и триггерной линейки, которая обеспечивает выявление позиции в выходном коде счетчика, где произошел переход из состояния «0» в состояние «1» при появлении очередного импульса на входе устройства.

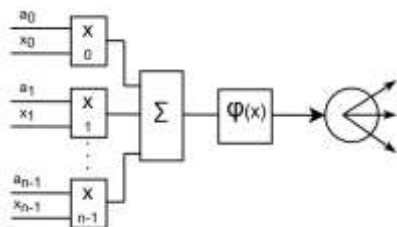


Рис. 1. Структурная схема унифицированной модели нейрона

Триггер, подключенный к выходу счетчика, в котором выявлен такой переход, формирует кратковременный единичный импульс. Импульсные потоки с выходов всех триггеров маскируются входным кодом  $N$ , и на выходе получается поток, характеризующийся средней частотой

$$f_y = f \frac{N}{2^m},$$

где  $m$  – разрядность устройства.

Адаптивный сумматор строится на основе сборки из двоичных умножителей частоты, как показано на рис. 2,а. На входы  $f$  каждого умножителя поступают импульсные потоки  $x$  с частотой  $f$ , на входе  $N$  зафиксированный соответствующий элемент параметра  $A$ . Средняя частота  $f_i$  импульсного потока на выходе умножителя определяется выражением

$$f_i = \frac{a_i x_i}{2^m}.$$

Во втором варианте функция импульсного умножения в адаптивном сумматоре реализуется конъюнкцией, как показано на рис. 2,б. Для этого параметры  $A$  должны быть представлены в форме широтно-импульсно-модулированного (ШИМ) сигнала. Преобразование кода в ШИМ сигнал может выполняться на базе двоичного счетчика.

В обеих реализациях при обеспечении фазового сдвига между импульсными потоками  $f_i$  в качестве импульсного сумматора может использоваться элемент OR.

При решении разных задач возникает необходимость реализации линейной неоднородной функции. Для ее вычисления можно использовать адаптивный сумматор с дополнительным  $n+1$  каналом, в котором используется установленный параметр и постоянный единичный сигнал. В импульсной реализации в качестве такого импульсного единичного сигнала может использоваться частота квантования процессов  $f$ , которую для простоты расчетов удобно связывать с разрядностью устройства. В соответствии с этим положением период работы устройства  $T$  определится как  $T=2^m/f$ .

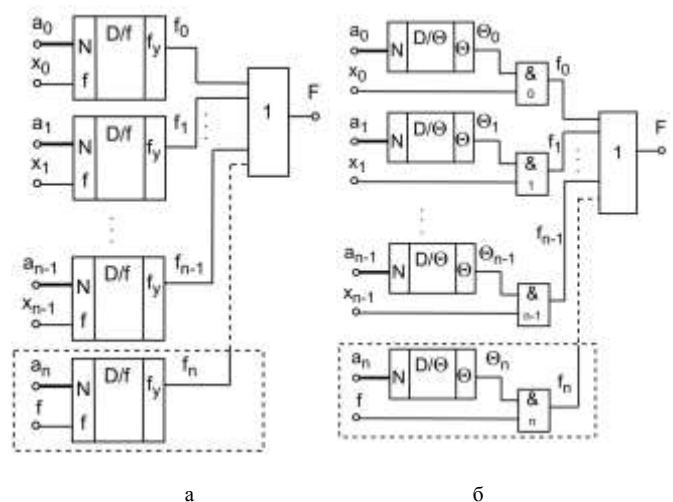


Рис. 2. Варианты реализации адаптивного импульсного сумматора

Элементы, обеспечивающие реализацию линейной неоднородной функции в адаптивном сумматоре, показаны на рис. 2 пунктиром.

Нелинейный преобразователь сигнала  $\varphi(x)$  преобразует скалярный входной сигнал  $x$  в соответствии с функцией из определенного набора, в качестве такого набора могут использоваться следующие:

$$\varphi_1(x_i) = \frac{x_i}{c+x_i}; \quad \varphi_2(x_i) = |x_i|$$

или

$$\varphi_1(x_i) = \text{th}(x_i); \quad \varphi_2(x_i) = |x_i|; \quad \varphi_3(x_i) = x_i^2.$$

Реализация перечисленных функций также может выполняться в потоковой импульсной форме. Как показано в [13, 14] и простые линейные преобразования, и аппроксимирующие полиномиальные функции могут быть вычислены в потоковом виде в базе простейших логических элементов, кодово-частотных преобразователей, реверсивных счетчиков и регистров.

Точка ветвления служит для рассылки одного сигнала по нескольким адресам, полученный скалярный входной сигнал  $x$  копируется на все выходы. Специальной реализации для такого элемента не требуется, эта функция реализуется путем организации соответствующих электрических соединений.

### III. АППАРАТНАЯ РЕАЛИЗАЦИЯ ЭЛЕМЕНТОВ ИМПУЛЬСНЫХ НЕЙРОННЫХ СЕТЕЙ

Рассмотренные базисные элементы ориентированы на реализацию в базе цифровых логических элементов и могут быть имплементированы в программируемые логические интегральные схемы (ПЛИС) [15], что позволяет получать высокопроизводительные решения для нейронных приложений. При проектировании импульсных нейроэлементов основными модулями являются типовые логические примитивы, счетчики и регистры [16], в этом же базе реализуем предложенные элементы. В качестве инструмента реализации выбраны языковые средства.

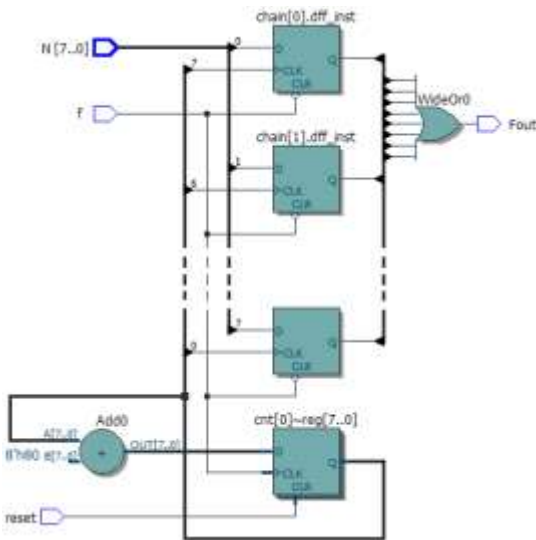


Рис. 3. RTL представление двоичного умножителя частоты

Универсальные HDL описания могут использоваться как элементы нижнего уровня при описании импульсных нейронных сетей, а также структур более высокого уровня [17]. Все описания выполнены параметризованными, где в качестве параметра указана разрядность устройства.

Фрагмент RTL схемы двоичного умножителя частоты, полученного в результате его синтеза средствами САПР Quartus II (13.1 sp. 1), показан на рис. 3. Для упрощения процесса моделирования при синтезе указывалась разрядность  $m=8$ . На схеме представлены: cnt[0]~reg[7..0] – внутренний счетчик умножителя, chain[i].dff\_inst – элементы триггерной линейки. Маскирование выполняется путем подключения разрядов кода N на информационные входы триггеров, причем старший разряд кода N подключается к младшему триггеру, а младший разряд кода – к старшему триггеру.

Синтез адаптивного сумматора выполнялся в четырехканальном варианте, его RTL схема для варианта на базе двоичного умножителя частоты показана на рис. 4. Результат моделирования адаптивного сумматора показан на рис. 5. Для проведения модельного эксперимента разработано тестовое окружение, состоящее из генераторов входных воздействии на базе потоков импульсов, отличающихся фазовым сдвигом ( $f_{ref1} - f_{ref4}$ ) и набора констант ( $N_{in1} - N_{in4}$ ). Путем умножения потоков формируются тестовые входные импульсные потоки ( $f_1 - f_4$ ). Каждый из каналов принимает соответствующий поток  $f_i$  и умножает его на  $a_i$ , заданный в описании сумматора как внутренний параметр.

Благодаря сдвигу фаз в опорных последовательностях для суммирования можно использовать элемент OR. Формула, описывающая среднее значение частоты выходного импульсного потока ( $F_{out}$ ), выглядит следующим образом:

$$F_{out} = \sum_{i=0}^{i=n} f_{ref_i} \frac{N_i a_i}{2^m}.$$

Рассчитанное значение совпало с результатом, полученным в результате моделирования.

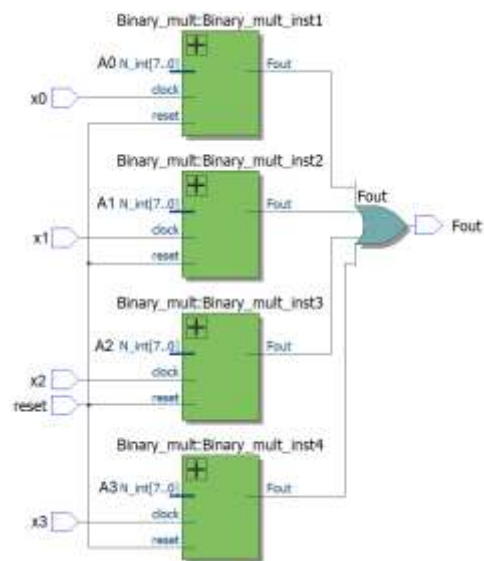


Рис. 4. RTL представление адаптивного сумматора

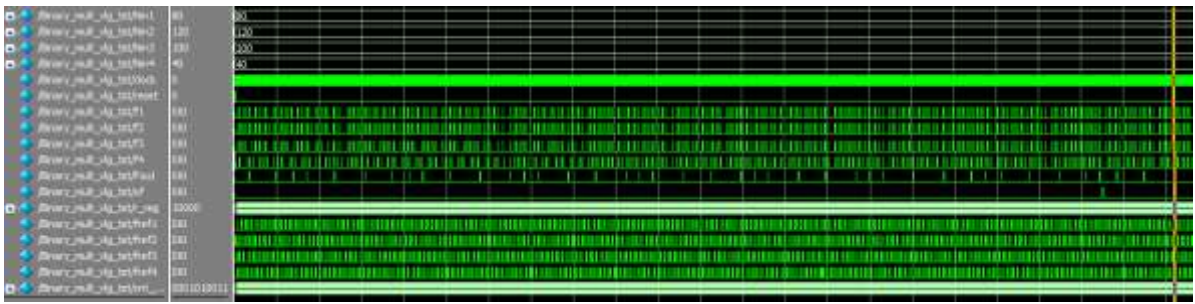


Рис. 5. Результат моделирования адаптивного сумматора

#### IV. ЗАКЛЮЧЕНИЕ

Рассмотренные в докладе элементы представляют собой базис, являющийся достаточным для создания импульсных нейронов. Предложенные элементы обеспечивают формирование у ИНС следующих свойств:

- возможность работы с динамическими процессами (распознавание речи и динамических изображений);
- реализация предвидения: не обязательно обладать полной информацией об объекте, по первой производной характеристики переходного процесса с некоторой точностью можно оценить результат;
- помехоустойчивость: вклад одиночного импульса в потоке в результат минимален, и его потеря не сильно скажется на полученном результате;
- минимальные аппаратные затраты: специальные вычислители не требуются, сложные вычислительные операции выполняются в базисе логических примитивов, регистров и счетчиков;
- высокая скорость работы за счет возможностей распараллеливания.

Развитие работы состоит в интеграции разработанных адаптивных сумматоров и нелинейных преобразователей для создания полнофункциональных импульсных нейронов. Полученные результаты могут найти применение при обработке сигналов и изображений, поступающих непрерывно от различных источников (видеокамер или радаров).

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Spiking brain models: Computation, memory and communication constraints for custom hardware implementation / A. Lansner, A. Hemani, N. Farahini // 19th Asia and South Pacific Design Automation Conf. (ASP-DAC), Singapore, 2014, С. 556-562, doi: 10.1109/ASPDAC.2014.6742950.
- [2] Skocik M. J., Long L. N. On the Capabilities and Computational Costs of Neuron Models // IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 8, pp. 1474-1483, doi: 10.1109/TNNLS.2013.2294016.
- [3] The McCulloch-Pitts model / S. Hayman // IJCNN'99. International Joint Conf. on Neural Networks. Proceedings (Cat. No.99CH36339), Washington, DC, USA, 1999, pp. 4438-4439 vol.6.
- [4] Modelling of an Interactive Network of Hodgkin-Huxley Neurons / B. Dutta, R. Sunitha and A. Sreedevi // 2020 International Conference on Communication and Signal Processing (ICCS), Chennai, India, 2020, pp. 1302-1306, doi: 10.1109/ICCS48568.2020.9182079.
- [5] Izhikevich E. M. Simple model of spiking neurons // IEEE Transactions on Neural Networks, vol. 14, no. 6, pp. 1569-1572, Nov. 2003, doi: 10.1109/TNN.2003.820440.
- [6] Brette R. Philosophy of the Spike: Rate-Based vs. Spike-Based Theories of the Brain // Front. Syst. Neurosci. – 2015. DOI: 10.3389/fnsys.2015.00151
- [7] Classification of different neuron behavior by designing spiking neuron model / A. Kumar, S. Kansal, M. Hanmandlu // 2013 IEEE International Conf. ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), Tirunelveli, India, 2013, pp. 25-30, doi: 10.1109/ICE-CCN.2013.6528592.
- [8] Leigh A. J., Mirhassani M., Muscedere R. An Efficient Spiking Neuron Hardware System Based on the Hardware-Oriented Modified Izhikevich Neuron (HOMIN) Model // IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 12, pp. 3377-3381, Dec. 2020, doi: 10.1109/TCSII.2020.2984932.
- [9] Pulse width modulation (PWM) signals using spiking neuronal networks / M. Jalilian, M. Nouri, A. Ahmadi, N. Kandalaf // 2017 IEEE International Conf. on Signal and Image Processing Applications (ICSIPA), Kuching, Malaysia, 2017, pp. 180-184, doi: 10.1109/ICSIPA.2017.8120602.
- [10] Improved winner-take-all circuit for neural network based on frequency-modulated signals / H. Hikawa // 2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS), Monte Carlo, Monaco, 2016, pp. 85-88, doi: 10.1109/ICECS.2016.7841138.
- [11] Горбань А. Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сиб. журн. вычисл. матем., 1998, том 1, № 1, 11–24.
- [12] Design and implementation of Izhikevich, Hodgkin and Huxley spiking neuron models and their comparison / J. Kumar, J. Kumar, S. Murali, R. Bhakthavatchalu // 2016 International Conf. on Advanced Communication Control and Computing Technologies (ICACCCT), Ramanathapuram, India, 2016, pp. 111-116, doi: 10.1109/ICACCCT.2016.7831611.
- [13] Bit-Stream Functional Converters for Decentralized Sensor Systems / N. Safyannikov, O. Bureneva // 9th Mediterranean Conf. on Embedded Computing (MECO), Budva, Montenegro, 2020, pp. 1-4, doi: 10.1109/MECO49872.2020.9134176.
- [14] Assurance of Fault-Tolerance in Bit-Stream Computing Converters / A. I. Gulin, N. M. Safyannikov, O. I. Bureneva, A. Y. Kaydanovich // 2018 IEEE East-West Design & Test Symposium (EWDTS), Kazan, 2018, pp. 1-4, doi: 10.1109/EWDTS.2018.8524812.
- [15] Weinstein R. K., Reid M. S., Lee R. H. Methodology and Design Flow for Assisted Neural-Model Implementations in FPGAs // IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 15, no. 1, pp. 83-93, March 2007, doi: 10.1109/TNSRE.2007.891379.
- [16] Comparative Analysis of Digital STDP Learning Circuits Designed Using Counter and Shift Register / J. Sim, S. Joo, S. Jung // 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), JeJu, Korea (South), 2019, pp. 1-4, doi: 10.1109/ITC-CSCC.2019.8793424.
- [17] Electronic model of human brain using Verilog / S. Khatri, S. Tiwari, N. Z. Rizvi // 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 2016, pp. 4250-4254, doi: 10.1109/ICEEOT.2016.775552

# Адаптивная нейро-нечеткая система (ANFIS) информационного взаимодействия в сетях промышленного интернета вещей с учётом выравнивания нагрузки между узлами

К. Мукешимана

Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)  
mucam2@mail.ru

М. С. Куприянов

Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)  
mskupriyanov@etu.ru

**Аннотация.** В настоящее время промышленный Интернет вещей (IIoT) превратился в многообещающую парадигму будущей сети, которая намеревается соединить различные виды интеллектуальных объектов. Несмотря на то, что проблема организации взаимодействия устройств IIoT уже привлекла большое внимание многих исследователей, текущие исследования маршрутизации в IIoT не могут эффективно решить проблему обмена данными самоадаптирующимся и самоорганизованным образом, потому что количество подключенных устройств совершенно большое. В этой статье представляется адаптивный алгоритм нейро-нечеткой кластеризации для равномерного распределения нагрузки между взаимодействующими узлами. Мы синтезировали нечеткую логику и нейронную сеть, чтобы уравновесить выбор оптимального количества головок кластера и равномерное распределение нагрузки между датчиками. Проводится сравнение с другими методами балансировки нагрузки в подобных беспроводных сенсорных сетях.

**Ключевые слова:** Интернет вещей; нечёткая логика; нейронная сеть; беспроводная сенсорная сеть; балансировка нагрузки; информационное взаимодействие

## I. ВВЕДЕНИЕ

В настоящее время промышленный Интернет вещей (IIoT) превратился в многообещающую парадигму будущей сети, которая намеревается соединить различные виды интеллектуальных объектов, включая домашнее оборудование, приводные устройства, встроенные и внешние датчики, программное обеспечение, исполнительные механизмы и т. д., между которыми устанавливается сетевое подключение. Несмотря на то, что, проблема организации взаимодействия устройств IIoT уже привлекла большое внимание многих исследователей, текущие исследования маршрутизации в IIoT не могут эффективно решить проблему обмена данными самоадаптирующимся и самоорганизованным образом, потому что количество подключенных устройств совершенно большое [1]. В этой статье предлагается новый механизм маршрутизации, основанный на использовании теории нечёткой логики и нейронной сети.

Маршрутизация пакетов в сетях интернета вещей от источника к получателю стала критической, и для определения наилучших усилий по обслуживанию в сети используются дополнительные механизмы управления трафиком. Эти подходы позволяют уменьшить задержку канала передачи и скорость потери пакетов, а также выбирать оптимального маршрута для передачи данных [2].

Метод, основанный на нейро-нечеткой логике, является новым подходом для интеллектуальной маршрутизации. Предложенный алгоритм выбирает оптимальный маршрут пакета на основе качества обслуживания (quality of service QoS) и скорости передачи пакетов.

Для улучшения качества обмена данными в сети используются различные методы маршрутизации. Все сенсорные устройства группируются в кластеры, в каждом кластере должен быть назначен головной узел, который взаимодействует с базовой станцией. В данной работе разработаем новый адаптивный нейро-нечеткий алгоритм (АННА) кластеризации, комбинируя теорию нечёткой логики с искусственной нейронной сетью.

## II. МЕТОДОЛОГИЯ ИССЛЕДОВАНИЯ

Нейро-нечеткая модель, используемая в этой работе для разработки интеллектуального алгоритма выравнивания нагрузки между узлами IIoT сети, основана на адаптивной системе нейро-нечеткого вывода (Adaptive Neuro-fuzzy Inference System ANFIS) с использованием системы нечеткого вывода Такаги-Сугено-Канга (TSK), которая была предложена в 1900-х годах [3].

Этот метод во многом похож на метод Мамдани. Первые две части процесса нечеткого вывода, Фазификация входных переменных и применения нечеткого оператора, одинаковы. Основное различие между ними состоит в том, что выходные функции принадлежности TSK являются либо линейными, либо постоянными. Типичное правило в нечеткой модели Сугено имеет вид [4]:

$$\text{Если } vx_1 = x \text{ и } vx_2 = y, \text{ то } vvx_1 = ax + by + c \quad (1)$$

На этапе оптимизации нечетких правил, параметры исходного нечеткого множества уточняются с помощью нейросетевых методов. Здесь правила, полученные на этапе генерации правил, используются для построения нейронной сети с пятью слоями. Каждый слой построенной системы определяет определенный шаг в механизме нейро-нечеткого вывода. Этот механизм вывода указывается на рис. 1.

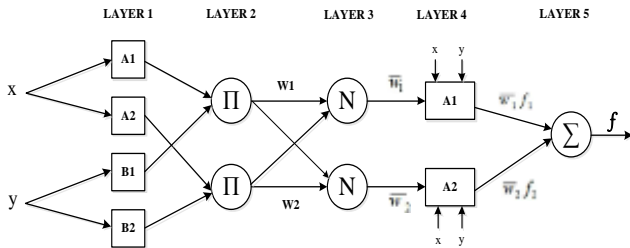


Рис. 1. Структура нейро-нечеткой сети ANFIS

Слой 1 представляет входные переменные, которые представляют текущее состояние системы. Нейроны в этом слое передают эти входные переменные непосредственно своим соответствующим модулям членства в следующем слое. Выходы данного слоя представляют собой значения функции принадлежности (ФП) при определенных значениях входов. Каждый узел является адаптивным с ФП  $\mu_{Ai}(x)$ , где  $x$  – вход  $i$ -го узла,  $i = 1, \dots, n$ ;  $Ai$  – ассоциированная с данным узлом лингвистическая нечеткая переменная. Обычно применяются гауссовы функции принадлежности вида:

$$A_{ij} = \exp\left[\frac{1}{2}\left(\frac{x_i - a_{ij}}{b_{ij}}\right)^2\right] \quad (2)$$

$$B_{ij} = \exp\left[\frac{1}{2}\left(\frac{x_2 - a_{ij}}{b_{ij}}\right)^2\right] \quad (3)$$

где  $a_{ij}$  и  $b_{ij}$  – множество параметров, которые необходимо настраивать в процессе обучения и, как следствие, проявляет различные формы функций принадлежности.

Узлы второго слоя называются ещё узлами правил, и выходные данные каждого нейрона представляют вес ( $W_i$ ), которой вычисляется как произведение входов по формуле:

$$W_i = \mu_{A_i}(x) * \mu_{B_i}(y), \forall i = 1, 2 \quad (4)$$

Выходы нейронов данного слоя представляют уровень активации правил.

Узлы третьего слоя являются фиксированными узлами, как и узлы второго слоя. Значения на выход второго слоя становятся входами третьего слоя, который вычисляет нормализованные уровни активизации правил ( $\bar{W}_i$ ) по формуле:

$$\bar{W} = \frac{W_i}{\sum_{i=1}^n W_i} \quad (5)$$

Узлы четвертого слоя являются адаптивными узлами. Выходные данные каждого узла являются произведением нормализованного веса из предыдущего слоя и полинома первого порядка, как показано в уравнении 6, где  $p_i$ ,  $q_i$  и  $r_i$  – проектные параметры узла, используемые для обработки второй части правила нечеткого логического вывода.

$$\bar{w}_i f_i = \bar{w}_i (p_i(x) + q_i(y) + r_i), \forall i = 1, 2 \quad (6)$$

Узел фиксированного слоя 5 вычисляет сумму всех аргументов и на выходе получаем результаты нечеткого логического вывода

$$\text{Результат} = \sum_{i=1}^n \bar{w}_i f_i = \frac{\sum_{i=1}^n w_i f_i}{\sum_{i=1}^n w_i} = \sum_{i=1}^n \frac{A_i(x) * B_i(y)}{\sum_{i=1}^n A_i(x) * B_i(y)} \quad (7)$$

#### А. Адаптивная система нечеткого логического вывода

В этом разделе представлена адаптивная система нечеткого логического вывода для получения информации об узле сети, который может стать головным узлом кластера. Выходные данные системы нечеткого логического вывода передаются в качестве набора входных данных для следующей предлагаемой адаптивной системы нейро-нечеткого вывода.

Для разработки системы вывода нечеткой логики использован алгоритм Мамдани с тремя лингвистическими переменными, 3 входных и 1 выходная. Входные переменные: остаточная энергия (Energy), плотность активных узлов сети (Density) и расстояние от узла до сетевого шлюза (Dgtw), который играет роль базовой станции (БС). На выходе системы получаем вероятность, с которой узел может назначаться головным в кластере (HProb). Структура представлена на рис. 2.

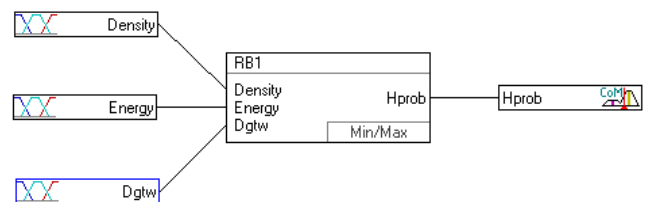


Рис. 2. Графический интерфейс редактора в среде fuzzyTECH

Входные и выходные лингвистических переменных с их термами представлены в табл. 1.

Графики функций принадлежности входных и выходных переменных представлены на рис. 3 (3.а. – функция Energy, 3.б. – функция density, 3.в. – функция Dgtw и 3.г. – функция HProb)

Входные переменные			Выходная переменная
Остаточная энергия (Energy)	Расстояние до БС (Dgw)	Плотность узлов (Density)	Вероятность (HProb)
Низкая (LOW)	Маленькое (SMALL)	Неплотно (LOOSE)	Сильно слабая (VVSmall)
Средняя (AVERAGE)	Среднее (AVERAGE)	Среднее (MEDIUM)	Очень слабая (VSmall)
Высокая (HIGH)	Большое (LONG)	Плотно (DENSE)	Слабая (Small)
-	-	-	Средняя (Medium)
-	-	-	Большая (High)
-	-	-	Очень большая (VHigh)
-	-	-	Сильно большая (VVHigh)

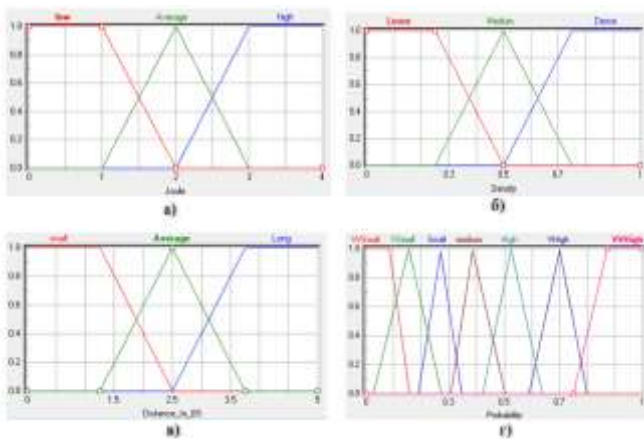


Рис. 3. Графики функций принадлежности входных и выходной переменных

Общее количество правил рассчитывается по формуле 8.

$$N = \mu^f \text{Inputs} \quad (8)$$

где  $N$  – количество нечетких правил,  $\mu^f$  – количество функций принадлежности на вход и  $\text{Inputs}$  – количество входов. В данной работе разработана модель с тремя функциями принадлежности и тремя выходными переменными. Используя выражение (8) мы получаем количество нечетких правил  $N = 3^3 = 27$  правил (табл. 2)

Разработанная адаптивная нечеткая логическая система вывода АНЛС используется для получения набора обучающих данных для настройки и обучения нечеткой нейронной сети ANFIS. Выходные данные АНЛС являются входами для ANFIS [7].

Правило	ЕСЛИ			ТО
	Energy	Dgw	Density	HProb
П1.	low	small	loose	vvsmall
П2.	low	small	medium	vsmall
П3.	low	small	dense	small
П4.	low	average	loose	vvsmall
П5.	low	average	medium	vvsmall
П6.	low	average	dense	vsmall
П7.	low	long	loose	vsmall
П8.	low	long	medium	vvsmall
П9.	low	long	dense	vvsmall
П10.	average	small	loose	small
П11.	average	small	medium	medium
П12.	average	small	dense	high
П13.	average	average	loose	small
П14.	average	average	medium	small
П15.	average	average	dense	medium
П16.	average	long	loose	vvsmall
П17.	average	long	medium	vsmall
П18.	average	long	dense	vsmall
П19.	high	small	loose	high
П20.	high	small	medium	vhigh
П21.	high	small	dense	vvhigh
П22.	high	average	loose	medium
П23.	high	average	medium	high
П24.	high	average	dense	vhigh
П25.	high	long	loose	high
П26.	high	long	medium	vhigh
П27.	high	long	dense	vhigh

### В. Адаптивная нейро-нечеткая система выводов

Нечеткая логика и искусственные нейронные сети являются лучшими инструментами для разработки и создания интеллектуальных систем искусственного интеллекта благодаря их обобщенности и нелинейным свойствам [5]. В ходе исследования разработана гибридная система, состоящая из двух различных методов мягкой вычислений: нечеткая логика и искусственная нейронная сеть. Система называется гибридной системой адаптивной нейро-нечеткой системой вывода (ANFIS) [6].

Для регулировки веса система ANFIS использует трапециевидную функцию принадлежности. Эти функции принадлежности использовались с правилами вывода продукта на уровне фаззификации. Предложенная ANFIS модель с тремя входами и одним выходом показана на рис. 4. У каждой входной переменной три функции принадлежности, следуя модели типа Такаги–Сугено, содержащей  $N$  логических правил (в нашей работе  $N=27$ ) и выглядят следующим образом:

Правило 1 : If Energy is Low AND Dgw is Small AND Density is Loose,  
Then  $F(x_1) = P_1e + Q_1d + R_1o + S_1$

Правило 2: If Energy is Low AND Dgw is Small AND Density is Medium,  
Then  $F(x_2) = P_2e + Q_2d + R_2o + S_2$

Правило 3 : If Energy is Low AND Dgw is Small AND Density is Dense,  
Then  $F(x_3) = P_3e + Q_3d + R_3o + S_3$

.....

Правило 25: If Energy is High AND Dgw is Long AND Density is Loose,  
Then  $F(x_{25}) = P_{25}e + Q_{25}d + R_{25}o + S_{25}$

Правило 26: If Energy is High AND Dgw is Long AND Density is Medium,  
Then  $F(x_{26}) = P_{26}e + Q_{26}d + R_{26}o + S_{26}$

Правило 27: If Energy is Low AND Dgw is Small AND Density is Dense,  
Then  $F(x_{27}) = P_{27}e + Q_{27}d + R_{27}o + S_{27}$

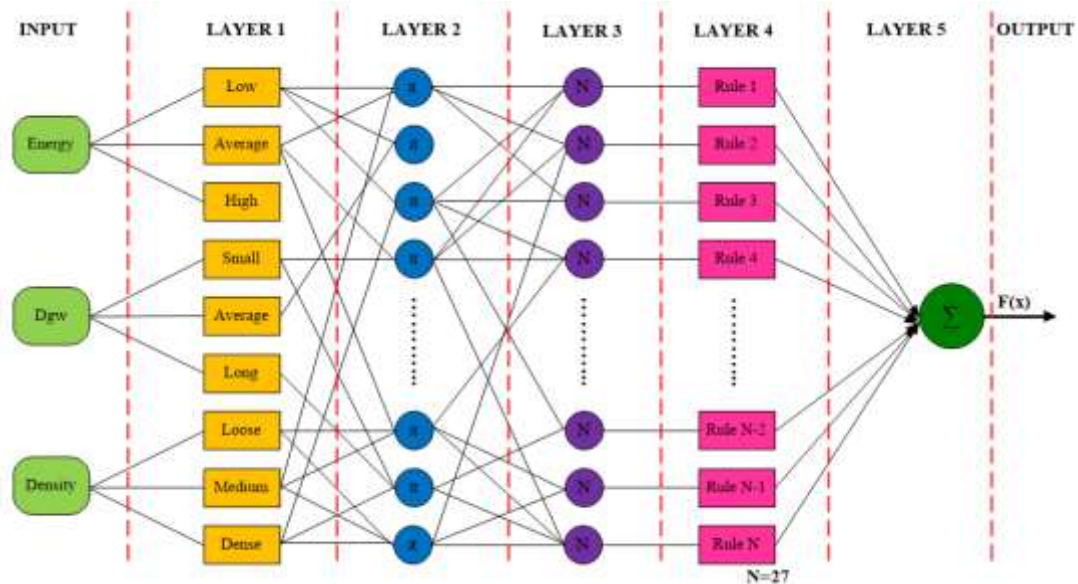


Рис. 4. Адаптивной нейро-нечеткой системе вывода (ANFIS) с 3 входами, 27 правилами и одним выходом

Предлагаемый алгоритм работает в четырех фазах: этап выбора, фаза формирования кластера, фаза агрегирования данных и фаза передачи данных. На первом этапе выбираются кластерные головки. На втором этапе кластеры образуются путем расчета их области на основе радиуса [8]. На последнем этапе кластерные головки передают агрегированные данные на базовую станцию как показано на рис. 5.

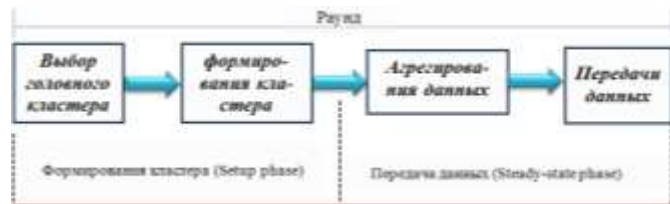


Рис. 5. Этапы работы алгоритма

### III. ЗАКЛЮЧЕНИЕ

В ходе исследования были синтезированы теория нечеткого логического вывода и нейронная сеть для правильного и грамотного выбора оптимального количества кластерных головок и улучшения качества взаимодействия сетевых устройств путём балансировки нагрузки между датчиками. Разработанный адаптивный нейро-нечеткий алгоритм (АННА) были сравнен с другими алгоритмами сетевой кластеризации. Для сравнения были выбраны:

- Алгоритм LEACH (Low-energy adaptive clustering hierarchy): самый первый алгоритм иерархической кластеризации в сенсорных сетях, работа, которого осуществляется в двух этапах: формирование кластеров и выбор головных узлов и передача данных с гарантией отсутствия коллизий.

- Алгоритм прямого взаимодействия DTM (Directed Transmission Method).

Принцип работы предлагаемого АННА алгоритма такой же, как у LEACH. Отличие заключается в том, что АННА учитывает не только остаточную энергию, но и расстояние между узлами и базовой станцией, а так же плотность активных узлов.

Результаты доказали, что предлагаемым алгоритмом сформулированы равномерно распределенные кластеры, а также снижено рассеивание энергии в сети, что непосредственно важно для улучшения жизненного цикла сети. Данное преимущество получилось путём применения две стратегии: искусственная нейронная сеть и нечёткая логика.

### СПИСОК ЛИТЕРАТУРЫ

- [1] Захаров М.В. Ситов А.А. Промышленный интернет вещей и его внедрение на производстве // Молодой ученый. 2020. № 20 (310). С. 36-38.
- [2] Жаркой С.А. Маршрутизация данных. Механизмы, критерии выбора маршрута и будущее // Молодой ученый. 2019. № 48 (286). С. 19-20.
- [3] Пегат А. Нечеткое моделирование и управление: пер. с англ. 2-е изд. М.: БИНОМ. Лаборатория знаний, 2017. 798 с.
- [4] Могилевич Е.О., Шведов А.С. Анализ динамики фондовых индексов с использованием нечетких моделей Такаги-Сугено // Экономический журнал ВШЭ. 2017 Т. 21 № 3 С. 434-450.
- [5] Смолин Д.В. Введение в искусственный интеллект: клнспект лекций. М.: ФИЗМАТЛИТ, 2004. 208 с.
- [6] Shihg, J. ANFIS: Adaptive-network based fuzzy inference System / J. Shihg, R. Jang // IEE Transactions on SMC. 1993. Vol. 23. P. 665-685.
- [7] Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры: Учеб. пособие для вузов 2-ое изд, перераб. и доп. М.: изд-во МГГУ им. Н.Э.Баумана 2004. 400 с.
- [8] Алексеев А.П. Сенсорная сеть с беспроводным интерфейсом на базе микроконтроллера фирмы «МИЛАНДР». 2016.

# Протокол взаимодействия агентов, решающих задачу распределения анализа данных

Юлейси Гонсалес Перес<sup>1</sup>, И. И. Холод<sup>2</sup>

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

<sup>1</sup>yuleisy2688@gmail.com, <sup>2</sup>iiholod@mail.ru

**Аннотация.** При выполнении анализа данных, размещенных на распределенных источниках, используются многоагентные технологии. Фундаментальным компонентом взаимодействия агентов в является их координация. Существуют разные стратегии выполнения анализа распределенных данных. Например, федеративное обучение может выполняться: последовательно, централизованно или децентрализованно. Координация агентов и используемый для взаимодействия протокол связи должны выполняться с учетом стратегии обучения. В этой статье описаны: протоколы связи, взаимодействие между агентами и типы агентов, используемые в каждой из стратегий федеративного обучения.

**Ключевые слова:** агент; федеративное обучение; коммуникация; многоагентные системы; протокол

## I. ВВЕДЕНИЕ

Анализ данных находящихся на распределенных узлах сталкивается с такими проблемами как: приватность данных, ограничение каналов связи, большие объемы информации и др. Для решения этих проблем была предложена концепция федеративного обучения (Federated Learning – FL) [1], [2], [3], которая выполняет обучение без передачи данных от источников. При этом выделяют различные топологии FL: централизованные и децентрализованные. В них возможны разные стратегии обучения: циклическое, федеративное и роевое.

Для реализации такого обучения хорошо подходят многоагентные системы (МАС). Коммуникация – один из важнейших компонентов МАС. Коммуникация очень важна, для достижения общих целей нескольких агентов. При этом состав агентов и характер координации сильно зависит от топологии сети и стратегии выполнения задачи, в том числе и для FL. МАС предоставляют возможности адаптации систем с учетом топологии и условий [4].

Для реализации такой адаптации состав агентов и протокол их взаимодействия должен поддерживать все топологии и стратегии обучения. В этой статье описаны важные элементы, которые следует учитывать при передаче и координации данных в распределенной системе FL.

## II. ТОПОЛОГИИ И СТРАТЕГИИ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ

При выполнении FL выделяют следующие этапы [5]:

1. сбор информации о среде (опционально, если пользователь не имеет информации о данных и узлах, на которых они размещаются) – для пользователя собирается информация о данных, узлах на которых они размещены и т. п.;
2. постановка задачи – пользователь указывает, какие данные должны быть проанализированы, каким алгоритмом, с какими параметрами и т. п.;
3. рассылка модели (опциональный если обучение осуществляется параллельно на разных узлах) – передача модели на каждый из узлов с данными;
4. обучение – на каждом из узлов на тех данных, на которых они размещаются;
5. агрегация (опционально если обучение осуществляется параллельно на разных узлах) объединение моделей, обученных на каждом узле.

Этапы 3–5 повторяются в процессе всего обучения.

Выделяют различные топологии и стратегии FL [5], [6] (рис. 1 и рис. 2):

- Централизованная схема (рис. 1а) включает центральный сервер, который координирует всех клиентов. Поскольку все участвующие узлы должны отправлять обновления на один сервер, этот сервер может стать узким местом системы.
- В децентрализованной схеме (рис. 1б) все клиенты координируют свои действия для получения глобальной модели.
- Иерархическая схема состоит из суб-федераций с централизованной и/или децентрализованной схемами (рис. 1в).



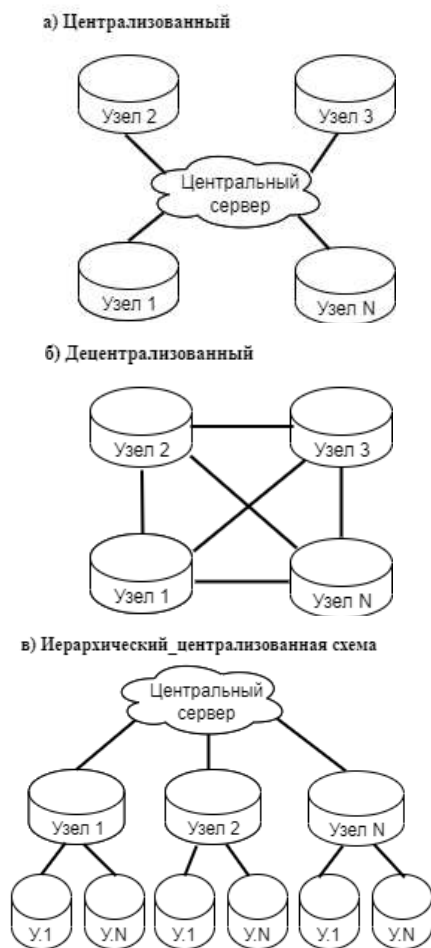


Рис. 1. Топологии FL [6]

Стратегия FL – это план обучения, который определяет, как модель обучается несколькими клиентами и как организована связь между ними и/или сервером:

- *Циклическая* (рис. 2а) обучает модель на каждом клиенте последовательно и передает следующему. Агрегирование модели не требуется;
- *Федеративная* (рис. 2б) обучает модель локально на каждом клиенте параллельно и агрегирует ее на сервере;
- *Роевая* (рис. 2в) обучает модель локально на каждом клиенте и распространяет ее по остальным. Агрегирование реализуется на каждом клиенте.

Разработанные в настоящее время системы FL не поддерживают все виды топологий и стратегий [7]. Основной причиной является жесткая структура систем и используемые протоколы взаимодействия между компонентами.

Решением этой проблемы может быть использование MAS для FL. MAS может использовать агентов для задач обучения, которые будут реализовывать разные стратегии. Используемый язык ACL [8] универсален для разных задач и может быть расширен для FL.

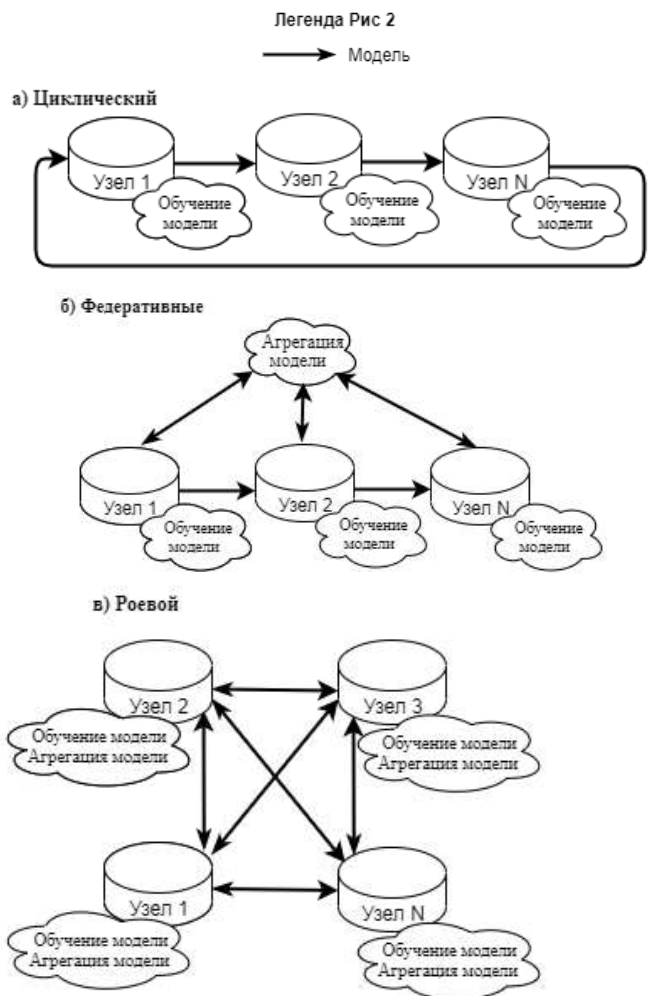


Рис. 2. Стратегии FL [6]

### III. ПРОТОКОЛ ВЗАИМОДЕЙСТВИЯ АГЕНТОВ

Исследование, проведенное авторами в [10], с учетом этапов выполнения FL позволило выявить следующие типы агентов в распределенной среде для анализа данных.

**Агент пользователя (User Agent – UA)** должен уметь:

1. получать и интерпретировать запросы пользователей;
2. передать запрос пользователя другим агентам;
3. возвращать результаты пользователю.

**Агент-посредник (Facilitator Agent – FA)** способствует распределению запросов между агентами, регистрирует агентов, содержит мета-знания о возможностях других агентов в системе.

**Агент задачи (Task Agent – TA)** отвечает за активацию и синхронизацию агентов, выполняющих обучение модели, планирует работу и отслеживает ход его выполнения.

**Агент ИАД (Data Mining Agent – DMA)** реализует алгоритм ИАД, принимает запрос от TA и запускает

алгоритм ИАД, используя значения, содержащиеся в запросе.

**Агент данных (Data Agent – DA)** отвечает за работу с данными и поддерживает информацию метаданных об источнике данных.

**Агент агрегации (Aggregate Agent – AA)**, собирает модели от агентов ИАД, агрегирует их и пересылает их обратно.

Для реализации различных стратегий FL агенты могут реализовывать различное поведение [10]:

- последовательная работа одного агента, когда он перемещается от источника к источнику, достраивая модель (рис. 3а и рис. 3б), реализует циклическую стратегию FL (рис. 2а).
- параллельная работа с мастером нескольких агентов, когда агенты извлекают локальные модели из данных и передают их агенту-мастеру для их объединения (рис. 3в) реализует федеративную стратегию (рис. 2б).
- независимая параллельная работа агентов, которые обмениваются моделями каждый с каждым и выполняют их агрегацию (рис. 3г) соответствуют роевой стратегии (рис. 2в).

Возможны два варианта последовательной стратегии:

- использовать одного мобильного агента, который будет перемещаться вместе с моделью. Он может объединять в себе функции всех агентов. В этом случае агенту не требуется взаимодействие с другими агентами (рис. 3а).
- использовать стационарных агентов ИАД на каждом из узлов, которые будут передавать модель по очереди друг другу. В этом случае агрегация моделей не требуется, а функцию агента пользователя будет играть первый агент (рис. 3б).

Последовательная стратегия применима к данным, обработка которых зависит от результатов, полученных на предыдущих источниках.

Параллельная стратегия применима, когда данные могут быть обработаны независимо друг от друга. На каждом из узлов должен быть установлен агент ИАД и данных, который выполняет обучение модели. В случае с централизованной агрегацией моделей, на сервере требуется агенты посредник, задачи и агрегации. В случае с независимой работой агентов, все типы агентов размещаются на узлах источниках.

Легенда рис. 3

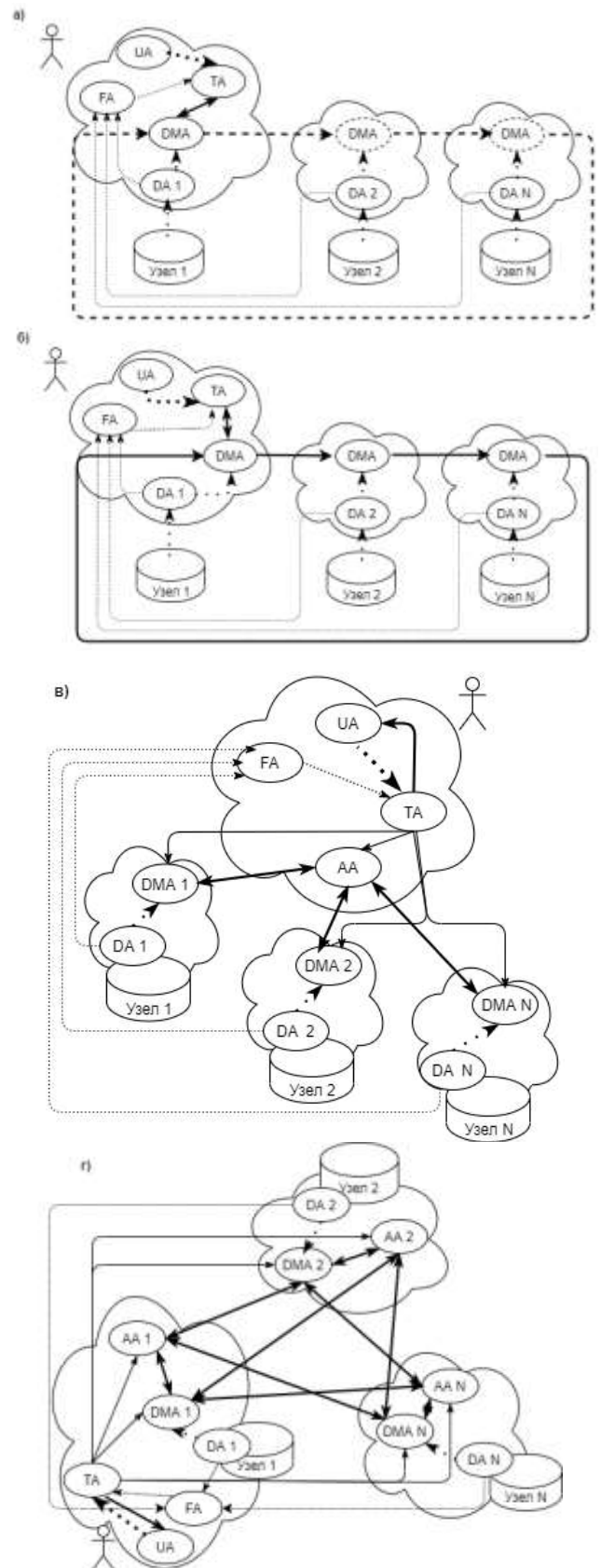


Рис. 3. Стратегии мета-обучения а), б) параллельный, в) последовательный г) роевой

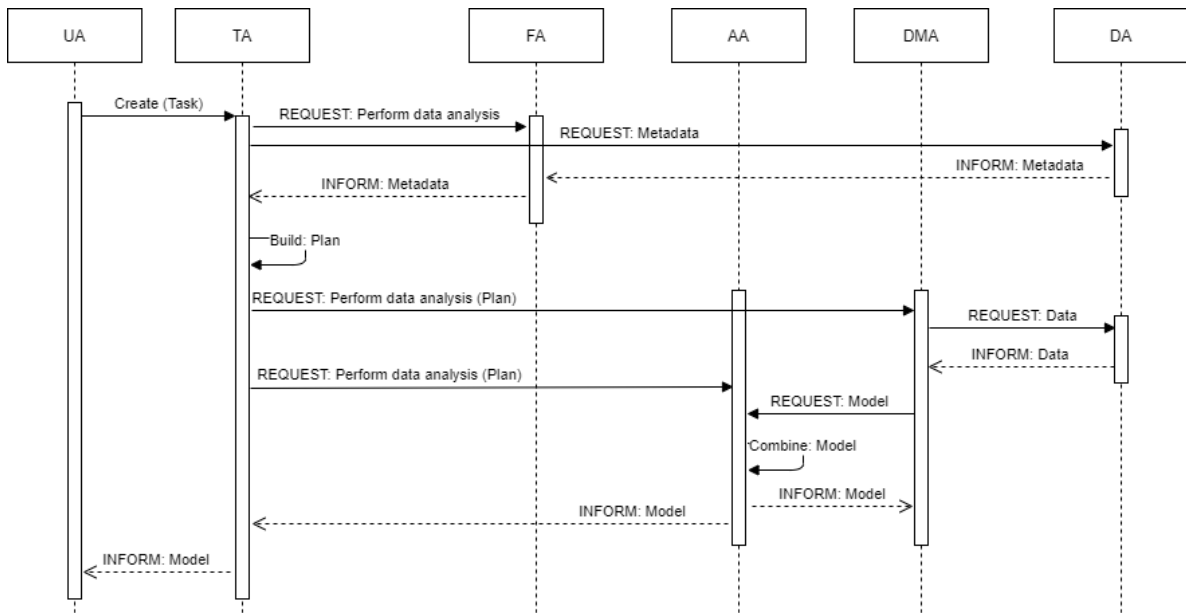


Рис. 4. Выполнить анализ данных

#### IV. ЗАКЛЮЧЕНИЕ

Проведенное исследование позволило ввести понятие FL и связать его стратегии и топологии с МАС. Был определен набор агентов, которые будут взаимодействовать в процессе коммуникации и координации в МАС. Модели взаимодействия выполнялись в МАС с учетом FL. Наконец, была составлена диаграмма последовательности, показывающая поток событий между каждым из агентов, идентифицированных в ходе расследования для выполнения модели анализа.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] McMahan B., Moore E., Ramage D., Hampson S., Aguera y Arcas B., Communication-Efficient Learning of Deep Networks from Decentralized Data, *Artificial Intelligence and Statistics*, April 2017, pp. 1273-1282.
- [2] Li L., Fan Y., Tse M., Lin K. Y., A review of applications in federated learning, *Elsevier*, 2020.
- [3] Li Q., Wen Z., Wu Z., Hu S., Wang N., Li Y., Liu X. et al., A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection, *arXiv*, 2021.
- [4] Chaimontree S., Atkinson K., Coenen F., A Multi-Agent Based Approach To Clustering: Harnessing The Power of Agents, *Springer*, 2012, pp. 1-2.
- [5] Kairouz P., McMahan B. H., Avent B. et al., Advances and Open Problems in Federated Learning, *arXiv*, 2019.
- [6] Rieke N., Hancox J., Li W., Milletari F., Roth H., Albarqouni S. et al, The Future of Digital Health with Federated Learning, *arXiv*, 2021.
- [7] Kholod I., Yanaki E., Fomichev D., Shalugin E., Novikova E., Filippov E., Nordlund M., Open-Source Federated Learning Frameworks for IoT: A Comparative Review and Analysis // *Review and Analysis. Sensors*, V. 21, no. 1, 2021.
- [8] FIPA. Foundation for Intelligent Physical Agents. (2021) Available at: [http://www.fipa.org/specs/fipa00061/SC00061G.html#\\_Toc26669700](http://www.fipa.org/specs/fipa00061/SC00061G.html#_Toc26669700) (accessed 5 February 2021)
- [9] Singh R., Singh A., Mukherjee S., A Critical Investigation of Agent Interaction Protocols in Multiagent Systems // *International Journal of Advancements in Technology*, V. 5, no. 2, 2014.
- [10] Gonzalez Perez Y., Kholod I., Analysis of Multiagent System for Data Analysis // 2020 XXIII International Conference on Soft Computing and Measurements (SCM). St. Petersburg, Russia, September 2020, pp. 218-221.

# Обработка данных в туманных вычислительных средах для машинного обучения

А. Н. Субботин

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

<sup>1</sup>alesu1543@gmail.com

**Аннотация.** В данной статье рассмотрена обработка видеoinформации с применением облачных технологий и туманных сред. Затронуты особенности мобильных устройств и встроенных компьютеров. Рассмотрены процессы обработки данных для видеоаналитики. Рассказано о технологиях обработки данных, способах передачи и хранения. Поставлена задача сокращения задержек при обработке в облачных системах и предложено решение проблемы на примере фильтрации негативного и криминального контента на сайте. Разработано приложение для контроля и управления фильтрацией. Приведены конкретные примеры.

**Ключевые слова:** туманные вычислительные среды; обработка данных; машинное обучение; фильтрация негативного контента; способы передачи и хранения данных; преобработанные данные; сокращение задержек

## I. ВВЕДЕНИЕ

Каждая современная организация имеет в своем распоряжении сервер для хранения данных. Даже если это маленькая фирма, а не большой завод по производству кораблей, самолетов, поездов, занимающих большие производственные площади. В любом случае – используется сервер для общего использования данных. Это может быть банальная организационная работа, бухгалтерская отчетность, хранение документации, чертежей, организация закупок, либо видеоаналитика.

Многие фирмы применяют машинное обучение с целью распознавания образов. Применяются технологии машинного обучения для извлечения обобщенной информации из видеоизображения. Это последовательность кадров для последующего анализа. Применение данного подхода, а именно получение видеоаналитики нужно в тех случаях, когда организация применяет технологии электронного пропуска по биометрии, в целях безопасности или мониторинга важных объектов (ядерная электростанция, военная часть, магазин свободного доступа к продукции и т. д.).

Использование своего сервера требует дополнительных расходов, связанных с содержанием, электропитанием, обслуживанием, предоставлением специально оборудованного помещения, соответствующего всем стандартам. Проще и надежнее приобрести хранилище в облаке. Но такой подход накладывает ограничения на каналы связи, роутеры и маршрутизаторы, а также выдвигаются большие требования к Интернет-провайдеру, который должен обеспечить своевременный доступ к

облаку. В случае прекращения сетевого доступа, работы организации остановится, что нанесет большой ущерб акционерам организации.

На этот случай есть решение, а именно применение туманных сред [1, 3]. Такой термин был придуман главой корпорации Cisco в 2011 году. Многие пользователи заметили задержки при обращении к серверам Cisco, особенно с мобильных устройств, у которых трафик более всего неустойчив.

Такая проблема замечена и со встроенных компьютеров Raspberry Pi, Milandr и пр. по той причине, что вычислительные возможности сильно ограничены. Это отсутствие встроенного графического процессора, памяти в достаточном объеме и хорошего центрального процессора. Все эти технические ограничения приводят к замедлению систем, спроектированных на встроенных компьютерах. Для них тоже есть решение в виде туманных вычислительных сред [2, 5].

## II. ПОСТАНОВКА ЗАДАЧИ

Предотвратить задержки в обработке информации может помочь промежуточная среда между облаком и компьютером (будь то: смартфон, встроенный компьютер, ноутбук или планшет). Такой средой может быть сервер, группа серверов или сеть из мобильных устройств, распределяющая динамически трафик между собой. Все это и есть туманные вычислительные среды. Не важно, над, чем работает пользователь: бухгалтерская отчетность, создание нового фильма, анализ видеопотока с проходной завода. Для всех подобных задач может помочь применение туманных вычислительных сред [4, 6].

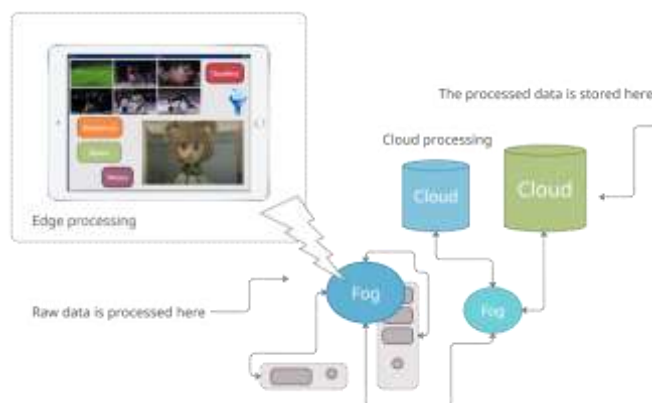


Рис. 1. Разделение процессов обработки информации относительно мобильных устройств (планшета)

Процессы обработки информации принято делить на три типа: на краю (Edge processing), в тумане (Fog processing) и облаке (Cloud processing). На рис. 1 изображены процессы обработки информации для мобильного устройства, но на самом деле их может быть большое количество, что вызывает большие задержки при обращении к облачным серверам [7, 9].

Задача ставится именно как видеоаналитика для мобильных устройств. На планшет поступает видеопоток, который потом анализируется и выдается информация для пользователя о наличии деструктивной информации. Предобработка данных, как показано на рисунке 1, происходит в туманной среде. Хранение информации и обработка средствами машинного обучения происходит в облаке с применением высокопроизводительных серверов, которым уступает любой домашний компьютер.

### III. ОПИСАНИЕ ПРОБЛЕМЫ

На видеохостинг поступает поток видеофайлов разной структуры и неизвестного до загрузки содержания. Современное законодательство предписывает строго наказывать не только тех, кто сам размещает криминальный контент, но и хостера, который предоставляет дисковое пространство для хранения видеофайлов в сети Интернет, а по факту – облако со своими настройками [8, 10, 11, 12].

#### A. Причины, по которым нецелесообразно хранить полные (необработанные) данные

- занимают в разы больше памяти в хранилище;
- не требуется полные данные (изображения высокого качества в очень большом, детальном разрешении);
- каналы связи (передачи данных) не позволяют передавать большой объем информации;
- целевая архитектура ИС не предусматривает обработку данных в большом объеме (встроенные компьютеры, смартфоны, стек Интернета Вещей).

#### B. Применение предобработанных данных

- повышает скорость передачи целевых данных в облако (или туман);
- снижает нагрузку на каналы связи и снижает задержки;
- повышает скорость обработки каждого кадра, потому что требуется меньше времени на передачу ответа из облака (или туманной среды).

Идеальным вариантом является применение предобработки данных, которые предобрабатываются в туманной вычислительной среде и отправляются в виде обработанных данных в облако для дальнейших действий (хранение в БД Oracle, MySQL, MSSQL; распознавание образов на Python с применением различных библиотек и моделей).

#### C. Форматы передачи предобработанных данных

- XML с догрузкой JSON;
- текстовые файлы;
- структурированные бинарные файлы с разделителем #0 (ноль-байт).

#### D. Способы передачи преобработанных данных

- зашифрованные каналы связи уровня стека IP;
- Web-технологии: PHP, JavaScript, JQuery через HTTP/HTTPS;
- VPN (Виртуальные частные сети).

#### E. Средства и инструменты разработки для передачи данных

- с применением программ на Java и сетевых библиотек (поддерживает XML-формат);
- с применением программ на Delphi в CodeGear (поддержка мобильных устройств и формата XML);
- с применением программ, разработанных на C# (в среде Visual Studio);
- с применением Qt и компилятора GCC.

#### F. Способы хранения предобработанных данных в хранилищах (data storages)

- в бинарном виде без файловой системы, с обращением через поток с адресом в памяти устройства;
- в бинарном виде с файловой системой NFS, FAT и т.д. (под Linux, Windows и др. операционные системы, поддерживающие файловые системы);
- в файлах в текстовом виде в самой файловой системе;
- в примитивных таблицах;
- в базах данных, поддерживающих стандарт SQL (без драйвера Telnet);
- в высокоорганизованных базах данных с поддержкой разных физических стандартов и оборудования (железа), таких, как: PostgreSQL, MS SQL, MySQL;
- в базах данных, поддерживающих свой синтаксис SQL (Oracle – PL/SQL и т.п.) с оптимизацией под свои хранилища (физическое оборудование);
- в специализированных БД под определенный сектор. Банковские системы (Oracle, PostgreSQL), под Web (MySQL, MariaDB и т.д.), под управление предприятием (MS SQL) и пр.

А так, это просто железка (рис. 2).



Рис. 2. Сервер производства компании Cisco для монтажа в стойку

#### G. Отличие processed data от других данных

- после обработки исходных данных, результат (processed data) становится меньше по объему занимаемой информации;
- Processed data незначительно уступает по действиям (возможностям) от raw data (исходных данных) для целевого назначения (поиск объектов на изображениях, определение голоса и т.д.);
- Processed data снабжены дополнительной служебной информацией (закрывающими тегами и параметрами, если это XML, например);
- хранятся в структурированном виде в data storages, но не обязательно, иногда и на обычном ПК, если объем небольшой;
- ограниченный спектр взаимодействия (только с определенными программами и алгоритмами; только с поддержкой определенных стандартов, которые были определены в ТЗ на этапе разработки программного обеспечения).

#### IV. ОПИСАНИЕ РЕШЕНИЯ

По причине нового законодательства планируется автоматическая фильтрация контента с помощью машинного обучения (искусственного интеллекта) и контроля всего процесса с помощью программы для планшета или смартфона.

Сама программа разработана в среде Visual Studio с подключением библиотек, обеспечивающих разработку приложений для мобильных платформ. Процесс компиляции так устроен, что в конечном итоге получается два приложения: под iOS и Android. Вся система фильтрации негативного содержания, а иногда и криминального так устроена, что взаимодействует с туманной средой [1], где происходит предобработки видеоинформации, а фактически её сжатие с потерями. Потом информация поступает в облачный сервер для последующей обработки.

Программа искусственного интеллекта в облаке разработана на языке Java с подключением сетевых библиотек и компьютерного зрения OpenCV [6, 11]. На сервер в облаке поступает информация о видео в сжатом виде, где средствами машинного обучения [12] происходит анализ переданной информации, делаются выводы о криминальном содержании, и выдается сообщение оператору на планшет о результатах проверки выбранного видеофайла.

#### V. ПРИМЕР ИСПОЛЬЗОВАНИЯ

Мы разработали приложение для планшета (рис. 3), с помощью которого оператор может контролировать закачиваемые видео на сайт.



Рис. 3. Разработанное мобильное приложение для контроля траффика

Для тестирования приложения мы использовали Яндекс.Видео, скачав оттуда несколько различных деструктивных фильмов и безобидной советской анимации. Все видеоматериалы были размещены на Google.Drive, откуда наша программа из облака проанализировала содержимое на предмет криминала.

Мой помощник Андрей открывал поочередно видео на планшете в нашей программе, потом происходил процесс кадрирования и отправка в туманную среду. Там происходит процесс потери информации (предобработка) и последующая отправка в формат XML на высокоскоростной облачный сервер. Там программа редактировала и запускала скрипты машинного обучения для проверки анализа полученной информации. Итоговый ответ отсылался в приложение на смартфоне.

ТАБЛИЦА I УМЕНЬШЕНИЕ ЗАДЕРЖЕК

Уменьшение времени задержек за 21 марта 2021 г.		
Время	Облако	С применением тумана
Суббота, 21:00-23:00	834 мс.	347 мс.
Воскресенье, 5:00-7:00	627 мс.	124 мс.
Воскресенье, 7:00-9:00	769 мс.	163 мс.
Воскресенье, 9:00-11:00	1032 мс.	527 мс.
Воскресенье, 11:00-13:00	1257 мс.	638 мс.

Теперь нужно проверить теорию на практике. Возьмем определенный период времени, а именно вечер субботы и утро воскресения 21 марта 2021 г. Как самые показательные в плане загрузки, потому что именно в это время домашние пользователи с криминальным контентом максимально активны в своих действиях. Именно в это время сильно загружены каналы связи и будут большие задержки. Нам удастся оценить насколько задержки стали меньше при обращении через туманную вычислительную среду.

Как можно наблюдать из таблицы, среднее время задержки значительно сократилось и составило всего 40 % (39,80969241) от первоначального значения. Это говорит о хороших результатах.

## VI. ЗАКЛЮЧЕНИЕ

В данном исследовании была рассмотрена обработка данных для мобильных устройств в туманных вычислительных средах. Обнаружена проблема с задержками и предложено решение. В ходе эксперимента удалось понизить время задержки, которое составило примерно 40 % от первоначального значения.

Мы получили хорошие результаты. Данный метод может быть использован в других областях.

## ВЫРАЖЕНИЕ БЛАГОДАРНОСТИ

Автор статьи благодарит СПИИРАН за научную поддержку и ГУП «Петербургский метрополитен» и лично Андрея за помощь в быстром проведении исследования и возможность создания статьи.

## СПИСОК ЛИТЕРАТУРЫ

- [1] Aazam M., Huh E.: Fog computing and smart gateway based communication for cloud of things // In: 2014 International Conference on Future Internet of Things and Cloud, August 2014, pp. 464–470 (2014)
- [2] Saecker M., Markl V.: Big Data Analytics on Modern Hardware Architectures: A Technology Survey, pp. 125–149. Springer, Berlin (2013)
- [3] Yi S., Hao Z., Qin Z., Li Q.: Fog computing: platform and applications // In: 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), November 2015, pp. 73–78 (2015)
- [4] Bonomi F., Milito R., Natarajan P., Zhu J.: Fog Computing: A Platform for Internet of Things and Analytics, pp. 169–186. Springer International Publishing, Cham (2014)
- [5] Sarkar S., Chatterjee S., Misra S.: Assessment of the suitability of fog computing in the context of internet of things // IEEE Trans. Cloud Comput. 6(1), 46–59 (2018)
- [6] Jaykrushna A., Patel P., Trivedi H., Bhatia J.: Linear regression assisted prediction based load balancer for cloud computing // In: 2018 IEEE Punecon, pp. 1–3. IEEE (2018)
- [7] Cristianini N., Shawe-Taylor J. An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods) / N. Cristianini, J. Shawe-Taylor. Cambridge University Press - University of London - United Kingdom (UK) – 2017.
- [8] Mahmud Redowan and Rajkumar Buyya. “Fog Computing: A Taxonomy, Survey and Future Directions.” ArXiv abs/1611.05539 (2018): n. pag. Fahad Fazal Elahi Guraya, Faouzi Alaya Cheikh. Neural networks based visual attention model for surveillance videos / Fahad Fazal Elahi Guraya, Faouzi Alaya Cheikh // Neurocomputing. 2015. Vol. 149. pp. 1348-1359.
- [9] G. Sreenu and M. A. Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis / G. Sreenu and M. A. Saleem Durai // Journal of Big Data. 2019. Vol. 6(48). pp. 3-27.
- [10] Huang C., Lu R., Choo K.-K.R.: Vehicular fog computing: architecture, use case, and security and forensic challenges. IEEE Commun. Mag. 55(11), 105–111 (2017)
- [11] Jaykrushna A., Patel P., Trivedi H., Bhatia J.: Linear regression assisted prediction based load balancer for cloud computing // In: 2018 IEEE Punecon, pp. 1–3. IEEE (2018)
- [12] Ali H. Sayed. Adaptation, Learning, and Optimization over Networks / Ali H. Sayed // Foundations and Trends® in Machine Learning. 2014. Vol. 7(4). pp. 74-91.

# Изометрический векторный нейроанализатор

Н. М. Сафьянников<sup>1</sup>, О. И. Буренева<sup>2</sup>,  
А. Е. Чепасов

*Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)*

<sup>1</sup>nmsafyannikov@etu.ru, <sup>2</sup>oibureneva@etu.ru

З. А. Алексанян

*Институт Мозга Человека им. Н.П. Бехтеревой РАН*  
aro@ihb.spb.ru

**Аннотация.** В сфере нейротехнологий важное значение занимают аппаратные системы для получения и анализа данных об особенностях функционирования ЦНС. Одно из направлений связано с развитием методов и аппаратных средств для оценки состояния ЦНС по двигательной активности. В предлагаемом докладе рассматривается изометрический векторный нейроанализатор, позволяющий оценить функциональное состояние центральной нервной системы человека на основании анализа произвольных движений конечностей, возникающих на фоне изометрических усилий. Анализатор может быть применен при проведении научных исследований, в неврологии, спортивной медицине, а также в других областях, где требуется оценка состояния ЦНС.

**Ключевые слова:** нейротехнологии; векторный нейроанализатор; изометрический метод; центральная нервная система; тремор

## I. ВВЕДЕНИЕ

В последнее время проводятся активные исследования в области нейротехнологий. Целью таких исследований является получение фундаментальных знаний в области функционирования мозга, мыслительной деятельности, психофизиологических функций; формирование методов оценки когнитивных возможностей человека и оценки различных аспектов состояния центральной нервной системы ЦНС [1]. Результаты исследований имеют прикладной характер: на их основе формируются методики медицинской диагностики, оценки профессиональных возможностей и потенциала человека. Полученные результаты актуальны и для развития технологий искусственного интеллекта, элементы которого являются цифровыми моделями биологических объектов.

Важное место в сфере нейротехнологий занимают аппаратные системы для получения и анализа данных об особенностях функционирования ЦНС, для проведения физиологических и клинических исследований ее структуры. Развиваются различные направления оценки состояния ЦНС.

Первую группу составляют методы на основе электрических измерений, имеющие специфические ограничения методического и технического характера (рентгеновская компьютерная томография, КТ, и радиоизотопные исследования) и продолжительное время обследования (магнито-резонансная томография, МРТ), спорные метрологические характеристики (электроэнцефалография и реоэнцефалография) [2]. Общеизвестен факт изменения электрических свойств нервной ткани в различных состояниях, с

другой стороны, использование этого наблюдения для оценки статуса головного мозга зачастую приводит к противоречивым результатам.

Вторую группу составляют нейротехнологии на основе оценки физиологических функций человека, связанных с анализом движения, базируются на различных инструментальных методах, таких как видеоанализ, стабилметрия, акселерометрия [3]. Эти методы являются косвенными и имеют метрологические недостатки.

Актуальна задача создания недорогой неинвазивной технологии, которая позволяла бы проводить исследование центральной нервной системы в тех ситуациях, когда перечисленные диагностические методы недоступны, противопоказаны или малоинформативны. Решение этой проблемы связано с развитием оригинальных подходов анализа моторных функций ЦНС на основе произвольных и произвольных движений.

## II. МЕТОД ОЦЕНКИ СОСТОЯНИЯ ЦНС НА ОСНОВЕ МОТОРНОГО ВЫХОДА

Для конечностей тела человека различают два основных вида движения: произвольные и произвольные [4].

Произвольное движение связано со стремлением к достижению цели. Произвольные движения выполняются рефлекторно. Показатели произвольных движений характеризуют состояние ЦНС.

В результате произвольного сокращения мышц возникают колебания, одним из проявлений такого состояния является тремор – это движения различных частей тела в виде дрожания, возникающего независимо от желания человека [5].

Изменения частоты, амплитуды и формы колебаний являются признаками нарушения периферических и центральных нейронных механизмов регуляции движений и могут быть использованы для оценки функций отдельных структур мозга [6].

Существует класс анализаторов по оценке состояния ЦНС, где используются акселерометры, прикрепляемые к частям тела человека, например [7, 8]. При этом регистрируются колебания подвижных звеньев тела, и выполняется соответствующий анализ. Однако регистрация движений связана со многими факторами и не позволяет объективно оценить состояние управляющих структур мозга [9]. Информация с анализаторов движения, затрудняет диффе-



ренциацию форм тремора, так как колебания звеньев тела в основном зависят от параметров инерционных масс.

В результате регистрируется абсолютная величина усилия, а произвольные и произвольные компоненты моторных команд не выделяются.

В анализаторах подобного типа не учитываются условия получения от датчиков соответствующих сигналов, кроме того, существует зависимость от расположения датчиков на теле человека, а также от их массы и размеров. Данный недостаток обычно компенсируют при обработке данных с датчиков при помощи алгоритмов, в том числе нейросетевых [10].

### III. МЕТОД ИЗМЕРЕНИЯ

Предлагается оригинальный метод оценки функционального состояния ЦНС человека. Этот метод основан на соприкосновении тела с опорным элементом и получении сигнала обратной связи. Сигнал представляет собой усредненную величину текущего изометрического усилия, которое прикладывается человеком к опорному элементу. В результате выделяется в реальном масштабе времени произвольная компонента этого усилия и определяется частота колебаний подвижных звеньев тела. В дальнейшем производится анализ этих колебаний. При этом произвольные компоненты изометрического усилия выделяются как разность между текущим значением усилия и его усредненной величиной. Далее по частоте, амплитуде и форме колебания определяется функциональное состояние нервной системы.

Однако при получении сигнала обратной связи и выделении произвольной компоненты изометрического усилия используются только произвольные усилия, прикладываемые человеком к опорному элементу. Это ограничивает функциональные возможности способа.

В следующем предлагаемом способе используются задаваемые усилия на сжатие по градациям от слабых к сильным. Здесь вводится управление от общего блока задания диапазона, воздействующего на блок сенсорных раздражителей, а также на элементы формирования градаций с пороговыми оценками диапазонов.

Задачей, решаемой предлагаемым изометрическим нейроанализатором, является более раннее выявление патологических состояний центральных структур нервной системы. Это обеспечивается учетом направления градаций усилий, то есть введением векторных подходов.

Проводится анализ произвольных управляющих воздействий с учетом диапазонов градаций усилий сжатия и растяжения. Техническим результатом является расширение функциональных возможностей, повышение точности диагностики. Оценка работы центральных структур системы управления движениями проводится с учетом активности, нисходящей по пирамидальному пути и экстрапирамидальным трактам. Активность формируется на различных уровнях центральной (моторной) системы ЦНС паттернами интегрированной активности в параметрах частота-амплитуда-форма колебаний.

Способ, используемый в изометрическом векторном нейроанализаторе, основан на приведении в соприкосновение тела с опорным элементом. Обеспечивается получение сигнала обратной связи в виде усредненной величины текущего изометрического усилия сжатия-растяжения.

### IV. СТРУКТУРА ИЗОМЕТРИЧЕСКОГО ВЕКТОРНОГО НЕЙРОАНАЛИЗАТОРА

Анализатор (рис. 1) содержит: измерительные блоки 1 по числу одновременно регистрируемых звеньев тела человека, каждый из которых содержит опорный элемент 2 сжатия, дифференциальный усилитель 3 интегрирующий усилитель 4 и дополнительный дифференциальный усилитель 5, многоканальный аналого-цифровой преобразователь 6 вычислитель 7 с подключенными средствами отображения информации 8, блок сенсорных раздражителей 9, внешний вход для ввода физиологических параметров 10, элемент формирования градаций 11 и пороговый элемент оценки диапазона 12, блок задания диапазона 13, тензодатчик сжатия-растяжения 14 и опорный элемент растяжения 15.

При работе с анализатором в процессе реализации данного способа пациент или испытуемый касается опорных элементов исследуемыми звеньями тела (пальцы, запястье, локоть) и совмещает (при зрительной обратной связи) метки на экране монитора, удерживая их вместе на требуемом уровне в течение заданного времени, необходимого для проведения достоверной статистической обработки данных. Возможны самые различные условия выполнения слежения за метками на экране, вплоть до отслеживания изменения их положения, что определяется задачами исследования.

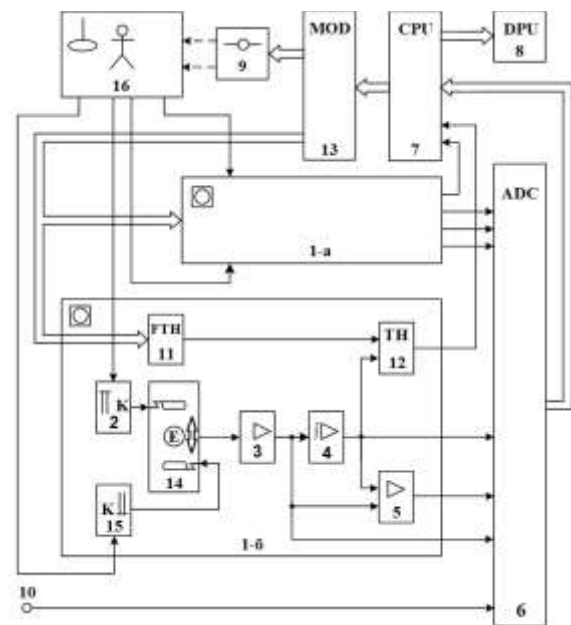


Рис. 1. Структура изометрического векторного нейроанализатора

Обследуемый пациент 16 воздействует пальцами рук на опорные элементы сжатия 2 и растяжения 15. Эти элементы находятся в соответствующих измерительных блоках 1-а и 1-б. По визуальным (в данном случае) сигналам обратной связи с блока сенсорных раздражителей 9 пациент

следит за уровнем собственного усилия. По инструкции он удерживает на заданном уровне метки на экране. Смещение меток пропорционально уровню формируемого в соответствии с градацией усилия.

Сигнал обратной связи с выхода интегрирующего усилителя 4 поступает на вход аналого-цифрового преобразователя 6. Также этот сигнал подается на вычислитель 7 и блок задания градаций 13. Он отображается в блоке сенсорных раздражителей 9, воздействуя на человека 16. Пациент прикладывает усилие к опорным элементам сжатия 2 и растяжения 15.

Количество измерительных блоков 1 соответствует числу одновременно регистрируемых звеньев тела. Выходы усилителей 3, 4, 5 каждого из измерительных блоков 1 подключены к соответствующим входам аналого-цифрового преобразователя 6. По другим входам 10 также могут быть введены другие физиологические параметры (например, электрокардиограмма). Прохождение сигналов осуществляется только по постоянному току. Это не вносит искажений в частотном диапазоне регистрируемого усилия.

Соответствующие метки, отдельно для левой и правой руки, различаются на экране монитора в блоке сенсорных раздражителей 9. При этом благодаря элементам формирования градаций 11 и оценки диапазона 12 в каждом измерительном блоке 1 вырабатывается сигнал. Он соответствует приложенному усилию по заданному диапазону в блоке 13.

Сглаженный сигнал с выхода интегрирующего усилителя 4 выполняет несколько функций.

Во-первых, он поступает на первый вход дополнительного дифференциального усилителя 5 как опорное напряжение. Относительно этого напряжения измеряются все отклонения удерживаемого по заданным градациям усилия. Эти усилия поступают с опорных элементов сжатия 2 и растяжения 15 на тензодатчик сжатия-растяжения 14. Тензодатчик выдает электрический сигнал через дифференциальный усилитель 3 и 5 на интегрирующий усилитель 4. На выходе усилителя 5 выделяется произвольный компонент управления. В параметрах отражено функциональное состояние структур центральной нервной системы.

При создании прибора использовали физиологический подход, основанный на особенностях иннервации моторной системы человека.

Так, движение пальцев иннервируется по пирамидальным путям, в то время как поддержание позы, связано с активностью экстрапирамидальной системы. В связи с этим использовано два подхода:

- в первом случае, регистрируется активность моторной системы при движении пальцев, для чего рука располагается на столе с опорой на предплечье и тыл кисти, а давление на датчик прибора оказывается только пальцами снизу;

- во втором случае, давление на датчики оказывается пальцами, свободно вытянутых рук, при этом, испытываемый сидит в, так называемой, «позе пианиста».

Каждый тест применяется в двух парадигмах с минимальным давлением (200–300 гр.) и с максимальным (1500–2000 гр.).

Подобный подход имеет особое значение для диагностики такого заболевания, как болезнь Паркинсона, при котором увеличение усилия приводит к уменьшению амплитуды тремора в отличие от других видов гиперкинеза, при которых увеличение усилия приводит к увеличению амплитуды тремора.

Анализатор не имеет регулировок и позволяет регистрировать усилие с точностью 5 г на 18-разрядном АЦП в диапазоне  $\pm 10$  кг с частотой квантования 320 Гц.

## V. ЗАКЛЮЧЕНИЕ

Изометрический векторный нейроанализатор: регистрирует удерживаемые в соответствии с градацией и направлением (сжатия, растяжения) усилия; выделяет соответствующий произвольный компонент управления в задаче слежения за величиной изометрического усилия.

Используются различные виды обратной связи – зрительной, слуховой, тактильной, смысловой. Это при предъявлении соответствующих сигналов открывает широкие возможности исследования нейронных механизмов управления движениями. Анализаторы также могут использоваться в процессе получения психофизиологических нагрузок в спортивной медицине.

Именно параметры изометрически регистрируемого усилия с учетом диапазонов градаций и направления (сжатия, растяжения) могут дать наиболее точную дифференцированную характеристику процессов центрального управления и охарактеризовать состояние организованных иерархически управляющих нейронных структур от спинального (сегментарного) уровня до коры больших полушарий мозга.

Кроме того, регистрация усилия и его векторного направления (сжатия, растяжения) в изометрическом режиме по градациям позволяет контролировать произвольный компонент управления и не требует наложения на обследуемого пациента каких-либо датчиков, что значительно упрощает процедуру регистрации и сводит до минимума (определяемого длительностью регистрации) время обследования, позволяя осуществлять массовую экспресс-диагностику населения.

Предполагается, что чувствительность метода в сочетании с оценкой процесса во времени и по градациям усилий с учетом направлений (сжатия, растяжения) позволит тестировать влияние на человека психофизиологических нагрузок и различных факторов среды, а также воздействие электрических и магнитных полей, широко применяемых в настоящее время не только в медицинских целях.

Предложенный анализатор позволяет охарактеризовать функциональное состояние центральной (моторной) нервной системы паттернами интегральной активности в параметрах частота-амплитуда в соответствии с отдельными уровнями её организации. Дополнительные показатели позволяют визуализировать активность, нисходящую по пирамидальному пути и экстрапирамидальным трактам, формирующуюся на различных уровнях моторной системы.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] IEEE Industry Connections (IEEE-IC) Standards Roadmap: Neurotechnologies for Brain-Machine Interfacing // IEEE Industry Connections (IEEE-IC) Standards Roadmap: Neurotechnologies for Brain-Machine Interfacing, 24 Feb. 2020, vol., no., pp. 1– 100.
- [2] Task based motion intention prediction with EEG signals / D. S. V. Bandara, J. Arata, K. Kiguchi // 2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Tokyo, 2016. С. 57– 60. DOI: 10.1109/IRIS.2016.8066066.
- [3] Real-time human motion analysis and IK-based human figure control / S. Yonemoto, D. Arita, R. Taniguchi // Austin, TX, USA. Proceedings Workshop on Human Motion, 2000. С. 149–154. DOI: 10.1109/HUMO.2000.897385.
- [4] Safyannikov N.M., Bureneva O.I., Aleksanyan Z.A. Non-invasive Method of Intelligent Sensory Control of Hands' Motor Functions for Bionic Systems // Procedia Computer Science. 2019. №150. С. 333–339. DOI: 10.1016/j.procs.2019.02.060.
- [5] Aleksanyan Z., Bureneva O., Safyannikov N. Tensometric Tremorography in High-precision Medical Diagnostic Systems // Medical Devices: Evidence and Research. 2018. №11. С. 321–330. DOI: 10.2147/MDER.S168831.
- [6] Pascoal-Faria P., Patrício M., Leal A., Caramelo F., Freire A., Januário C., Castelo-Branco M. B. Understanding Tremor in Rapid Upper Limb Movements Using 3D Accelerometers Data // Neuroscience and Medicine. 2014. №5, С. 205–213. DOI: 10.4236/nm.2014.55024.
- [7] Говорова Т.Г., Попова Т.Е., Таппахов А.А. Тремография в клинической практике // Нервно-мышечные Болезни. 2019. Т. 9. №4. С. 61–72. DOI: 10.17650/2222-8721-2019-9-4-61-72.
- [8] Multichannel Wireless sEMG Sensor Endowing a 0.13  $\mu\text{m}$  CMOS Mixed-Signal SoC / G. Gagnon-Turcotte, C.L. Fall, Q. Mascret, M. Biemann, L. Bouyer, B. Gosselin // 2018 IEEE Life Sciences Conference (LSC), Montreal, QC, Canada. 2018. С. 1–4. DOI: 10.1109/LSC.2018.8572118.
- [9] Hand Gesture Recognition Based on Surface Electromyogram Signal (sEMG) with Muscular Contraction Level and Real Time Implementation on An Artificial Prosthetic Wrist Using Artificial Neural Network (ANN) / A.F. Hasan, H.U. Masud, T. Anzar, P. Hasan // 2020 IEEE Region 10 Symposium (TENSYP), Dhaka, Bangladesh. 2020. С. 1648–1651. DOI:10.1109/TENSYP50017.2020.9231038.
- [10] Translating sEMG signals to continuous hand poses using recurrent neural networks / F. Quivira, T. Koike-Akino, Y. Wang D. Erdogmus // 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), Las Vegas, NV, USA. 2018. С. 166–169. DOI: 10.1109/BHI.2018.8333395.

***NEURONT.ETU.RU***

197376, Россия,  
Санкт-Петербург,  
ул. Профессора Попова,  
д.5, СПбГЭТУ «ЛЭТИ»  
Тел.: +7 (812) 346-46-37  
E-mail: [IRVC.eltech@mail.ru](mailto:IRVC.eltech@mail.ru)

