

Проблемы построения моделей машинного обучения для проведения практик студентам

А. Н. Субботин¹, Н. А. Жукова^{1,2}

¹Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

²Санкт-Петербургский институт информатики и автоматизации Российской академии наук
alesu1543@gmail.com, nazhukova@mail.ru

Аннотация. В статье рассмотрены проблемы, возникающие в процессе обучения студентов по курсам, предполагающие разработку моделей машинного обучения. Проанализирована предметная область. В качестве основных проблем определены низкая скорость построения моделей и их проверки. Предложен способ решения, основанный на использовании туманных и облачных вычислительных сред. Проведенные экспериментальные исследования показали, что время построения моделей снизилось в 22,17 раз, скорость работы преподавателя повысилась на 47 %. Представленный способ повышения скорости построения моделей может быть применен при проведении лабораторных работ, при руководстве выпускными квалификационными работами. Также способ применим для построения моделей машинного обучения при решении практических задач в области производства, городского хозяйства, в сфере предоставления транспортных услуг и пр.

Ключевые слова: машинное обучение; туманная и облачная вычислительная среда; образовательные процессы

I. ВВЕДЕНИЕ

В последнее время с переходом на дистанционное обучение появились новые проблемы, требующие решения. Проблемы, связанные с усвоением знаний, можно определить с трех позиций [1, 2]: 1) студента; 2) преподавателя; 3) учебного заведения. При подготовке специалистов, магистров и бакалавров в процессе преподавания необходимо обеспечить:

- быстроту (высокую скорость) работы преподавателя;
- точность проверки самостоятельных (лабораторных или практических) работ;
- объективность выставления оценок;
- удовлетворенность процессом обучения студентов;
- доступность учебной литературы как в электронном виде, так и в печатном виде в библиотеке;
- помощь преподавателя во время учебного процесса (практики), например, разбор примеров и пр.);
- согласованность изучаемых предметов в одном семестре.

С целью повышения объективности выставления оценки и удовлетворённости от учебного процесса студентов и самих преподавателей, была принята Единая Шкала Европейских Компетенций (или Кредитов) [1], которая распространена на территории стран Европейского Союза (ЕС), в частности, в Германии. Суть ее заключается в том, что итоговая оценка по предмету выставляется на основе результатов выполнения небольших заданий за определенный срок (зачетов, которые принято называть кредитами в Германии). Такой подход является более трудоемким для преподавателя, но больше нравится студентам, поскольку является достаточно прозрачным и объективным. Можно представить таблицу Excel в строках которой размещаются фамилии студентов, а в столбцах – номера лабораторных работ. В последнем столбце – оценка. Программы для работы с такими таблицами были разработаны в нескольких университетах ФРГ, впоследствии они были заменены веб приложением. Однако, сейчас появились аналоги Microsoft Excel от Google: Tables, которые позволяют одновременное редактирование и просмотр таблиц с правами доступа, что сильно упрощает использование шкалы оценок ECTS. Руководство (свод правил) для применения ECTS постоянно менялось на протяжении 20 лет и текущая версия доступна на сайте Европейской Комиссии [1].

Проблемы обучения можно разделить на две большие группы [3, 4]:

- 1) общепрофессиональные;
- 2) узкоспециализированные.

Узкоспециализированными проблемами обучения студентов можно считать проблемы, возникающие при построении моделей машинного обучения, поскольку некоторые модели могут строиться от суток до нескольких дней. В научных школах и университетах, обучающих студентов по курсу искусственного интеллекта: Massachusetts Institute of Technology MIT (<https://web.mit.edu>), Московский государственный университет МГУ (<https://www.msu.ru>), Санкт-Петербургский государственный университет СПбГУ (<https://spbu.ru>), Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» (<https://etu.ru>), Московский физико-технический институт МФТИ (<https://mipt.ru>) предусматривается выполнение практических, лабораторных работ, а также другие самостоятельные виды деятельности студентов [5, 6, 7]. Обучение предполагает решение практических задач, например, по классификации растений, прогнозированию экономических событий,

распознаванию образов, созданию чат-ботов и переводчиков с иностранных языков [8, 9].

II. ОПИСАНИЕ ПРОБЛЕМЫ

Проблемы обучения студентов по курсу искусственного интеллекта связаны с необходимостью затрачивать значительное время на построение моделей и их проверку. Требуется проверить разработанные студентами алгоритмы не только на отсутствие явных ошибок (необъявленных переменных, нарушение синтаксиса, неподключенных компонент и пр.), но и результат их выполнения. В некоторых случаях результатом работы алгоритма должна быть модель, которую нужно проверить на тестовом датасете. Данный процесс может занять сутки у преподавателя, если он не общается со студентом через Discord или другую платформу, где можно увидеть результат через стрим. Если представить себе более 20 студентов, каждый из которых сдает по одному выполненному заданию каждую неделю, то процесс их проверки может затянуться на месяцы. Вначале нужно построить модель по реализации студента. Если использовать слабый компьютер, то модель может строиться от 2 часов до 1 дня, а построение моделей, разрабатываемых в рамках дипломных работ, может потребовать от суток до 5 дней. Далее необходимо проверить полученную модель на соответствие требованиям по точности определения образов (или объектов, событий). Требуется найти решение, позволяющее преподавателю строить модели и проверять программы, разрабатываемые студентами при выполнении лабораторных работ или при написании выпускных квалификационных работ в приемлемый срок [10, 11].

III. СПОСОБ РЕШЕНИЯ

Применение туманных или облачных вычислений может значительно сократить время, затрачиваемое на построение моделей и ускорить проверку заданий. При проверке заданий может помочь автоматическая система проверки (бот), который будет получать скрипт для создания (обучения) модели, а на выходе выдавать сообщения в мессенджер Telegram для студента и преподавателя [12, 13, 14].

Можно упростить задачу: создать web-сайт (рис. 1), который после авторизации позволяет загрузить файлы и скрипт, который выполнится в облаке или туманной вычислительной среде и выдаст сообщение в виде таблицы: фамилия студента, № лабораторной работы, результат. Предполагается скрипт на языке Python, использующий библиотеку TensorFlow [3, 5, 7, 14].

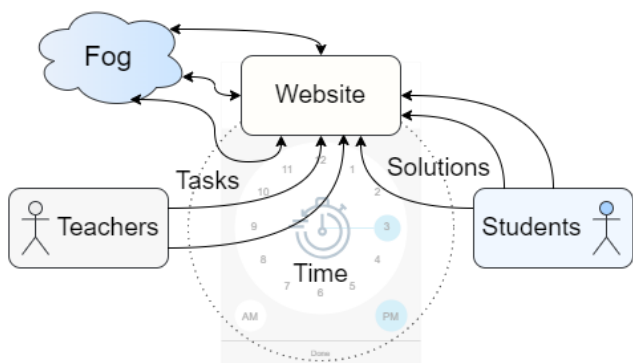


Рис. 1. Способ решения обозначенных проблем

Необходимо создание web-приложения, которое работает в браузере или в оболочке через компонент «Браузер» и может запускаться из файловой системы ПК или смартфона. В некоторых случаях, запуск браузера будет требовать большое количество ресурсов (оперативной памяти и процессорного времени), тогда возможно создание бинарного (нативного) приложения под определенную операционную систему: iOS, Android, Ubuntu, macOS или Windows для обычных компьютеров или встроенных технологических компьютеров (Advantech, Raspberry Pi, Banana PC, Milamdr и пр.). В любом случае, web-сайт должен работать быстро и загружать сам файлы студентов (архив Zip) в туманную вычислительную среду [10, 11, 15].

После обсуждения с коллегами-преподавателями, были определены следующие основные показатели эффективности (табл. 1) предлагаемого метода проверки заданий.

ТАБЛИЦА 1. ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ

No.	Предполагаемые значения эффективности	
	Показатель	Ожидаемое значение
1	Время построения моделей	Сокращение в 20 раз
2	Скорость проверки заданий	Увеличение на 45%

Предложенный способ обеспечивает значительное сокращение времени построения моделей благодаря использованию туманной вычислительной среды, а также достигается повышение скорости проверки заданий поскольку задания проверяются в определенное время, а не на протяжении всего дня.

IV. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

Приложение разработано на Lazarus IDE под macOS, поскольку на этом компьютере с RISC-чипом M1 обеспечивается быстрая работа. Главная экранная форма приложения для проверки заданий (рис. 2) содержит список студентов, отметки о выполнении и кнопки управления программой. Перед компиляцией «wTaskGet.app» под macOS Ventura 13.4.1, необходимо дополнительно установить Xcode 15 и отладчик LLDB (<https://lldb.lldb.org>) [11].

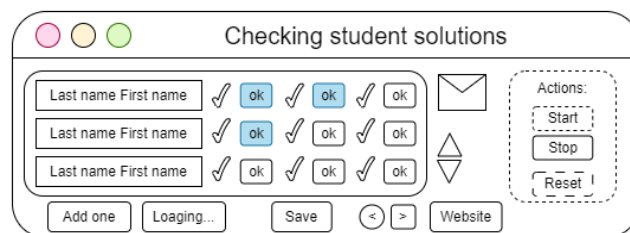


Рис. 2. Внешний вид приложения для проверки заданий

Программа подключается к web-серверу по адресу: <http://rss2fido.sf.net/wTasks.dart>, скачивает и анализирует страницу. Анализ элементов выполняется с периодичностью 15 секунд, результат выдается в программе под macOS из структуры DOM одной HTML-страницы. Страница «wTask.dart» разработана на языке Dart от Google. Для работы web-страницы нужно активировать «dart_frog_cli» и установить «dart pub install mysql», подключить пакет «import 'package: mysql1 /mysql1.dart';», после чего запустить web-сайт: «dart wTasks.dart». Подключаться к MariaDB необходимо через объект: «final conn = await MySqlConnection.connect()». Запуск web-сервера должен

выполняться через создание объекта: «var server = await HttpServer.bind(IPAddress.anyIPv6, 498);» [11].

V. ПРИМЕР ИСПОЛЬЗОВАНИЯ

Авторами исследования разработано приложение для проверки заданий (рис. 3), которые были выданы преподавателями, и решения которых были загружены студентами через web-сайт. Слева представлен список с фамилиями и именами студентов, который может быть загружен (по кнопке «Load!»), могут быть добавлены новые студенты (по кнопке «Add ...»). Преподаватель наблюдает за прогрессом: если напротив фамилии студента появилась галочка, то задание было проверено автоматизированной системой (скрипт студента был загружен на туманный вычислительный сервер, запущен и получен требуемый результат на выходе). После этого преподаватель подтверждает успешно выполненное задание (по кнопке «Up!») напротив галочки. Результаты проверки могут быть сохранены (по кнопке «Save») или отменены из статусной строки (рис. 3, сверху).

Допустимо листать задания и просматривать фамилии студентов, нажимая кнопки в виде угловых скобок. Допустима остановка получения информации (парсинга сайта на HTML5 в формате DOM) по кнопкам в группе управления из поля «Actions»: Start, Stop, Reset.

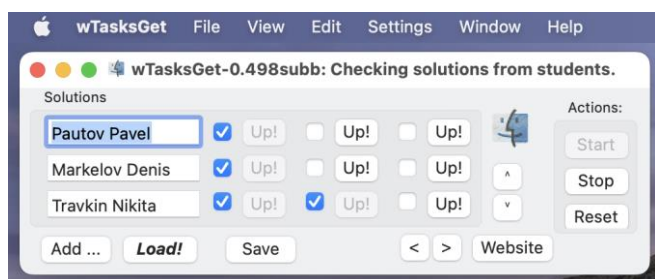


Рис. 3. Внешний вид главной формы для проверки заданий

После разработки программного обеспечения (программы для проверки заданий для преподавателя под macOS и web-сайта для загрузки работ студентами на проверку), проведены эксперименты. Основным показателем эффективности предлагаемого метода проверки заданий, является время построения моделей. Для сравнения было выбрано 3 конфигурации:

1) NoteBook ASUS E510, Full HD (1920 x 1080), TN + film, Intel Celeron N4020, ядра: 2 x 1.1 ГГц, RAM 4 ГБ, eMMC 128 ГБ, Intel UHD Graphics 600.

2) MacBook Air, WQXGA (2560 x 1600), IPS, Apple M1, ядра: 4 + 4, RAM 16 ГБ, SSD 256 ГБ, Apple M1 7-core.

3) Fog Server Cisco UCS C480 ML M5, Intel Xeon Scalable CPUs (2), 24 DDR4 DIMM slots: 16, 32, 64, and 128 GB; up to 2933 MHz, Dual M.2 SATA SSD.

Туманный сервер был взят в аренду на сайте VK Cloud (<https://mcs.mail.ru>), но хватило пробного периода (1 неделя). Технические конфигурации 1 и 2 для эксперимента-1 были предоставлены соавтором статьи. В результате первого эксперимента выяснилось, что время построения моделей сократилось в 22,1746 раз относительно первой конфигурации и туманного сервера (табл. 2). Разница между NoteBook на основе IBM-PC и MacBook составила 10,8 раз.

ТАБЛИЦА II. ВРЕМЯ ПОСТРОЕНИЯ МОДЕЛЕЙ

№	Уменьшение времени на построение моделей (в мин.)				
	Модель	Конф. 1	Конф. 2	Конф. 3	Сокращение (между 1 и 3)
1	Классификатор цветов	237	24,5	12,916	18,349
2	Предсказание экономических событий	345	31,2	15,98	21,589
3	Бинарная классификация отраженных сигналов радара	432	39,703	21,47	20,121
4	Распознавание рукописных символов	531	47,92	26,79	19,82
5	Распознавание объектов на фотографиях	879	78,209	32,91	26,709
6	Прогноз успеха фильмов по обзорам	546	52,21	23,8	22,941
7	Классификация обзоров фильмов	318	29,03	14,2	22,394
8	Генерация строк на основе текста для чат-бота	792	71,38	31,09	25,474
По среднему значению:					22,1746

Вспомогательным показателем эффективности является скорость работы преподавателя. Для проведения эксперимента-2 были взяты данные за 3 года. Каждое задание (табл. 3) повторяется по номеру из первого эксперимента (табл. 2), скорость проверки по годам является средним значением по 17 студентам, скорость указана в секундах, а увеличение – в процентах. Скорость проверки возросла на 47 % относительно среднего значения за три года.

ТАБЛИЦА III. ПРОВЕРКА ЗАДАНИЙ

№	Скорость проверки заданий (в сек.)				
	2021	2022	2023	Система	Увеличение скорости
1	452	439	421	237	38,799
2	679	637	587	241	55,037
3	792	781	774	329	50,822
4	847	831	812	401	44,517
5	1384	1275	1193	572	48,282
6	972	958	942	429	48,015
7	825	812	809	397	44,143
8	1734	1672	1591	782	45,872
По среднему значению:					46,935875

В результате 2-х проведенных экспериментов установлено, что система контроля и проверки заданий позволяет сократить время построения моделей в 22,1746 раз и повысить скорость проверки заданий на 47 %.

VI. ЗАКЛЮЧЕНИЕ

В представленном исследовании предложен метод повышения эффективности деятельности преподавателя при проверке заданий по курсу машинного обучения. Была проанализирована предметная область. Обозначены две основные проблемы, состоящие в низкой скорости проверки заданий и больших временных затратах, необходимых для построения

моделей. Разработано программное обеспечение для проверки решений студентов.

В ходе проведенных экспериментов установлено, что время построения моделей машинного обучения сократилось в 22,1746 раз, а скорость работы преподавателя повысилась на 47 %, что говорит об эффективности применения данного метода в образовательном процессе.

Предложенный метод также применим в других областях знаний: на производстве (в системах управления производственными процессами), в организационной деятельности фирм различного профиля и предприятий, где выпускается высокотехнологичная продукция: подводные лодки, корабли, краны, пассажирские составы в метро и пр.

БЛАГОДАРНОСТЬ

Авторы статьи выражают благодарность «ЛЭТИ» за возможность проведения практик по курсам машинного обучения и нейросетей; и СПИИРАН за научную поддержку и возможность доклада на конференции CTS-2023 в Санкт-Петербурге, Россия.

СПИСОК ЛИТЕРАТУРЫ

- [1] European Credit Transfer and Accumulation System (ECTS), European Education Area at European Commission with URL: <https://education.ec.europa.eu/node/1523> (Referenced date: 30 June 2023).
- [2] T. Zafarullah, A. Kanwal, "Job Satisfaction of Private Special Education Teachers About Teaching Profession: Perception of Special Education Teachers", 2022 PalArch's Journal of Archaeology of Egypt/ Egyptology, , vol. 19, no. 1, pp. 1240-1249, Mar. 2022.
- [3] N. Zhukova and A. Subbotin, "Using Applied Computing on Embedded Computers to Build Digital Twins in a Fog Computing Environment," 2023 12th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2023, pp. 1-6, doi: 10.1109/MECO58584.2023.10154931.
- [4] Ke Yuan, "Research on the Problems and Implementation Paths of Labour Education in Higher Education Institutions under the Background of "Five Education Integration"", 2023, Journal of Education and Educational Research, 3(2), 62–65. doi: 10.54097/jeer.v3i2.9019.

- [5] A.N. Subbotin, N.A. Zhukova and P.T. Anh, "Building a Model of a Patient State based on Machine Learning Methods in a Fog Computing Environment," 2023 XXVI International Conference on Soft Computing and Measurements (SCM), Saint Petersburg, Russian Federation, 2023, pp. 322-325, doi: 10.1109/SCM58628.2023.10159074.
- [6] Rusdiman AB, Nathanael Sitanggang, and Muhammad Joharis Lubis, "Problems Of Higher Education, Curriculum and Education In Indonesia", 2023, IJHES, vol. 2, no. 4, Feb. 2023, doi: <https://doi.org/10.55227/ijhess.v2i4.344>.
- [7] M. Chervontsev, A. Subbotin, A. Vodyaho and N. Zhukova, "Use of Dynamic Models in Cognitive Cyber-Physical Systems", 2023 Presented at the 15th International Conference "Intelligent Systems" (INTELS'22), Moscow, Russia, Eng. Proc. 2023, 33(1), 14; <https://doi.org/10.3390/engproc2023033014>.
- [8] Amelia F., & Darmadja S. (2019). Konfirmasi Faktor yang Berpengaruh terhadap Pengambilan Keputusan dalam Pemenuhan Nutrisi Ibu Hamil. Citra Delima, 2(2): 101-110. doi:10.33862/citradelima.v2i2.37.
- [9] A. Subbotin, N. Zhukova, M. Tianxing, "Video Processing Algorithm in Foggy Environment for Intelligent Video Surveillance," Intelligent Systems and Applications, Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 2, August 2021, doi: 10.1007/978-3-030-82196-8_52.
- [10] L. Kornilav, "Kant's pedagogical thought and current problems of education", 2023 SHS Web of Conferences, Volume 161, 2023, doi: 10.1051/shsconf/202316106008.
- [11] V. Osipov, N. Zhukova, A. Subbotin, "Intelligent escalator passenger safety management," Scientific Reports, April 2022.
- [12] X. Lu, "Problems And Countermeasures in The Development of Preschool Integrated Education in China", EHSS, vol. 10, pp. 117–125, Apr. 2023, doi: 10.54097/ehss.v10i.6901.
- [13] A. Vodyaho, E. Stankova, N. Zhukova, A. Subbotin, "Use of Digital Twins and Digital Threads for Subway Infrastructure Monitoring," Computational Science and Its Applications – ICCSA 2022 Workshops, January 2022.
- [14] A.N. Subbotin, "Determining the Type of Interference in a Receiving Antenna Using Machine Learning in Fog Computing Environments," 2022 International Conference on Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS), Saint Petersburg, Russian Federation, 2022, pp. 311-314, doi: 10.1109/ITQMIS56172.2022.9976645.
- [15] 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).