ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)

«ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҰЛТТЫҚ ҒЫЛЫМ
АКАДЕМИЯСЫ» РҚБ

«ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҰЛТТЫҚ ҒЫЛЫМ АКАДЕМИЯСЫ» РҚБ

ХАБАРЛАРЫ

ИЗВЕСТИЯ

РОО «НАЦИОНАЛЬНОЙ
АКАДЕМИИ НАУК РЕСПУБЛИКИ
КАЗАХСТАН»

NEWS

OF THE NATIONAL ACADEMY
OF SCIENCES OF THE REPUBLIC
OF KAZAKHSTAN

SERIES OF PHYSICS AND MATHEMATICS

1 (353)

JANUARY – MARCH 2025

PUBLISHED SINCE JANUARY 1963
PUBLISHED 4 TIMES A YEAR

ALMATY, NAS RK

«ҚР ҰҒА Хабарлары. Физика-математика сериясы».
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)
Меншіктеуші: «Қазақстан Республикасының Ұлттық ғылым академиясы» РҚБ (Алматы).
Ақпарат агенттігінің мерзімді баспасөз басылымын, ақпарат агенттігін және желілік басылымды қайта есепке қою туралы ҚР Мәдениет және Ақпарат министрлігі «Ақпарат комитеті» Республикалық мемлекеттік мекемесі **28.02.2025** ж. берген **№KZ20VPY00113741** Куәлік.
Тақырыптық бағыты: *ақпараттық-коммуникациялық технологиялар*
Қазіргі уақытта: *«ақпараттық-коммуникациялық технологиялар» бағыты бойынша ҚР БҒМ БҒСБК ұсынған журналдар тізіміне енді.*
Мерзімділігі: *жылына 4 рет.*
Редакцияның мекен-жайы: *050010, Алматы қ., Шевченко көш., 28, 219 бөл., тел.: 272-13-19*
http://www.physico-mathematical.kz/index.php/en/

«Известия НАН РК. Серия физико-математическая».
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)

Собственник: *Республиканское общественное объединение «Национальная академия наук Республики Казахстан» (г. Алматы).*

Свидетельство о постановке на переучет периодического печатного издания, информационного агентства и сетевого издания **№ KZ20VPY00113741.** Дата выдачи **28.02.2025**

Тематическая направленность: *информационно-коммуникационные технологии.*

В настоящее время: *вошел в список журналов, рекомендованных КОКСНВО МНВО РК по направлению «информационно-коммуникационные технологии».*

Периодичность: *4 раза в год.*

Адрес редакции: *050010, г. Алматы, ул. Шевченко, 28, оф. 219, тел.: 272-13-19*
http://www.physico-mathematical.kz/index.php/en/

News of the National Academy of Sciences of the Republic of Kazakhstan.
Series of Physics and Mathematics
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)
Owner: RPA «National Academy of Sciences of the Republic of Kazakhstan» (Almaty).
Certificate **No. KZ20VPY00113741** on the re-registration of the periodical printed and online publication of the information agency, issued on **28.02.2025** by the Republican State Institution «Information Committee» of the Ministry of Culture and Information of the Republic of Kazakhstan
Subject area*: information and communication technologies.*
Currently: *included in the list of journals recommended by the CCSES MSHE RK in the direction of «Information and communication technologies».*
Periodicity: *4 times a year.*
Editorial address: *28, Shevchenko str., of. 219, Almaty, 050010, tel. 272-13-19*
http://www.physico-mathematical.kz/index.php/en/

NEWS OF THE NATIONAL ACADEMY OF SCIENCESOF THE REPUBLIC OF KAZAKHSTAN
PHYSICO-MATHEMATICAL SERIES
ISSN 1991-346X
Volume 1. Namber 353 (2025). 17–31

https://doi.org/10.32014/2025.2518-1726.322

IRSTI 50.05.09
UDC 004.378

R. Abdualiyeva, L. Smagulova*, A. Yelepbergenova, 2025.
Zhetysu university named after I. Zhansugurov, Taldykorgan, Kazakhstan.
E-mail: jgu_laura@mail.ru

THE EFFECTIVENESS OF USING CHATGPT IN PROGRAMMING

Abdualiyeva Rima – Ph.D, Zhetysu University named after I. Zhansugurov, Taldykorgan, Kazakhstan, e-mail: rimaergaliyevna09@gmail.com, ORCID ID: https://orcid.org/0000-0002-4161-9825;
Smagulova Laura – Candidate of Pedagogical Sciences, Zhetysu University named after I. Zhansugurov, Taldykorgan, Kazakhstan, e-mail: jgu_laura@mail.ru, ORCID ID: https://orcid.org/0000-0002-1359-2119;
Yelepbergenova Aigul – master's degree, Zhetysu University named after I. Zhansugurov, Taldykorgan, Kazakhstan, e-mail: aigul_eu@mail.ru, ORCID ID: https://orcid.org/0000-0002-3525-1825.

Abstract. This study evaluates the efficacy of OpenAI's ChatGPT models, GPT-3.5 and GPT-4, in programming and software development. A quantitative approach was employed, using the Mostly Basic Python Problems (MBPP) dataset to assess their ability to handle coding tasks.

Beyond direct evaluation, a comparative analysis was conducted to benchmark ChatGPT against other large language models, including Google's Bard and Anthropic's Claude. This analysis provided a broader perspective on ChatGPT's programming capabilities within the AI landscape.

The findings highlight ChatGPT's advantages in coding tasks, demonstrating its ability to enhance efficiency and productivity. Developers and researchers can leverage these insights to integrate AI-driven solutions into software development. Additionally, the study sheds light on key strengths and limitations of ChatGPT compared to competing models, offering a foundation for further refinement of AI-based coding assistants.

As AI continues to transform various industries, understanding its role in programming becomes increasingly relevant. This research underscores ChatGPT's potential in optimizing workflows, fostering innovation, and streamlining software development. By identifying areas for further improvement, it also contributes to ongoing advancements in AI-driven programming tools.

Ultimately, this study serves as a valuable resource for AI researchers and developers, guiding future improvements in language models and their applications

in coding, automation, and broader technological advancements. The continuous evolution of AI-powered programming tools promises to reshape software development, making it more efficient and accessible.

Keywords: ChatGPT, GPT-3.5, GPT-4, Python programming, OpenAI, Google's Bard,Claude from Anthropic, Software Development, MBPP Dataset, Performance, Comparative Analysis, Efficiency.

Р.Е. Абдуалиева, Л.А. Смагулова*, А.У. Елепбергенова, 2025.
І. Жансүгіров атындағы Жетісу Университеті, Талдықорған, Қазақстан.
E-mail: jgu_laura@mail.ru

БАҒДАРЛАМАЛАУДА CHATGPT ҚОЛДАНУ ТИІМДІЛІГІ

Абдуалиева Рима Ергалиевна – Ph.D, І. Жансүгіров атындағы Жетісу университеті, Талдықорған, Қазақстан, e-mail: rimaergaliyevna09@gmail.com, ORCID ID: https://orcid.org/0000-0002-4161-9825;
Смагулова Лаура Амангельдиевна – п.ғ.к., І. Жансүгіров атындағы Жетісу университеті, Талдықорған, Қазақстан, e-mail: jgu_laura@mail.ru, ORCID ID: https://orcid.org/0000-0002-1359-2119;
Елепбергенова Айгүл Узбековна – магистр, І. Жансүгіров атындағы Жетісу университеті, Талдықорған, Қазақстан, e-mail: aigul_eu@mail.ru, ORCID ID: https://orcid.org/0000-0002-3525-1825.

Аннотация. Бұл зерттеуде OpenAI ChatGPT модельдерінің, атап айтқанда GPT-3.5 және GPT-4-тің, бағдарламалау мен бағдарламалық жасақтама әзірлеудегі тиімділігі бағаланады. Сандық әдіс қолданылып, негізінен Basic Python Problems (MBPP) деректер жинағы арқылы модельдердің кодтау тапсырмаларын орындау қабілеті зерттелді. Тікелей бағалаудан бөлек, ChatGPT-ті Google Bard және Anthropic Claude сияқты басқа да ірі тілдік модельдермен салыстырмалы талдау жүргізілді. Бұл талдау ChatGPT-тің бағдарламалау мүмкіндіктерін жасанды интеллект ортасындағы басқа модельдермен салыстыра отырып, оның орнын анықтауға мүмкіндік берді. Алынған нәтижелер ChatGPT-тің кодтау тапсырмаларын орындаудағы артықшылықтарын көрсетіп, оның тиімділік пен өнімділікті арттыру қабілетін дәлелдейді. Әзірлеушілер мен зерттеушілер бұл мәліметтерді жасанды интеллект негізіндегі шешімдерді бағдарламалық жасақтама әзірлеу үдерісіне енгізу үшін пайдалана алады. Сонымен қатар, зерттеу ChatGPT-тің бәсекелес модельдермен салыстырғанда негізгі артықшылықтары мен шектеулерін анықтап, жасанды интеллект негізіндегі кодтау көмекшілерін одан әрі жетілдіруге негіз қалайды.

Жасанды интеллект әртүрлі салаларды трансформациялауды жалғастырып жатқандықтан, оның бағдарламалаудағы рөлін түсіну барған сайын маңызды болуда. Бұл зерттеу ChatGPT-тің жұмыс процестерін оңтайландырудағы, инновацияларды ынталандырудағы және бағдарламалық жасақтаманы әзірлеуді жетілдірудегі әлеуетін көрсетеді. Сондай-ақ, ол болашақта

жетілдіруді қажет ететін негізгі бағыттарды анықтауға көмектесіп, жасанды интеллект негізіндегі бағдарламалау құралдарының үздіксіз дамуына үлес қосады. Қорытындылай келе, бұл зерттеу жасанды интеллектті зерттеушілер мен әзірлеушілер үшін құнды ресурс болып табылады. Ол тілдік модельдерді және олардың бағдарламалау, автоматтандыру және технологиялық даму салаларындағы қолданылуын жетілдіруге ықпал етеді. Жасанды интеллект негізіндегі бағдарламалау құралдарының эволюциясы бағдарламалық жасақтама әзірлеу үдерісін неғұрлым тиімді әрі қолжетімді ете отырып, оны жаңа деңгейге көтеруге ықпал етеді.

Түйін сөздер: ChatGPT, GPT-3.5, GPT-4, Python бағдарламалау, OpenAI, Google Bard, Anthropic's Claude, бағдарламалық жасақтама жасау, MBPP деректер жиынтығы, өнімділік, салыстырмалы талдау, тиімділік.

Р.Е. Абдуалиева, Л.А. Смагулова*, А.У. Елепбергенова, 2025.
Жетысуский университет им. И. Жансугурова, Талдыкорган, Казахстан.
E-mail: jgu_laura@mail.ru

ЭФФЕКТИВНОСТЬ ИСПОЛЬЗОВАНИЯ CHATGPT В ПРОГРАММИРОВАНИИ

Абдуалиева Рима Ергалиевна – PhD, Жетысуский университет им. И. Жансугурова, Талдыкорган, Казахстан, E-mail: rimaergaliyevna09@gmail.com, ORCID ID: https://orcid.org/0000-0002-4161-9825;
Смагулова Лаура Амангельдиевна – к.п.н., Жетысуский университет им. И. Жансугурова, Талдыкорган, Казахстан, E-mail: jgu_laura@mail.ru, ORCID ID: https://orcid.org/0000-0002-1359-2119;
Елепбергенова Айгуль Узбековна – магистр, Жетысуский университет им. И. Жансугурова, Талдыкорган, Казахстан, E-mail: aigul_eu@mail.ru, ORCID ID: https://orcid.org/0000-0002-3525-1825.

Аннотация. В этом исследовании оценивается эффективность моделей OpenAI ChatGPT, GPT-3.5 и GPT-4, в программировании и разработке программного обеспечения. Был применен количественный подход, в основном с использованием набора данных Basic Python Problems (MBPP) для оценки их способности справляться с задачами кодирования.

Помимо прямой оценки, был проведен сравнительный анализ, чтобы сравнить ChatGPT с другими крупными языковыми моделями, включая Google Bard и Anthropic Claude. Этот анализ позволил получить более широкое представление о возможностях программирования ChatGPT в среде искусственного интеллекта. Полученные результаты подчеркивают преимущества ChatGPT в задачах программирования, демонстрируя его способность повышать эффективность и продуктивность. Разработчики и исследователи могут использовать эти знания для интеграции решений, основанных на искусственном интеллекте, в разработку программного обеспечения.

Кроме того, исследование проливает свет на ключевые преимущества и ограничения ChatGPT по сравнению с конкурирующими моделями, предлагая основу для дальнейшего совершенствования помощников по кодированию на основе искусственного интеллекта. Поскольку искусственный интеллект продолжает трансформировать различные отрасли, понимание его роли в программировании становится все более актуальным.

Это исследование подчеркивает потенциал ChatGPT в оптимизации рабочих процессов, стимулировании инноваций и усовершенствовании разработки программного обеспечения. Определяя области для дальнейшего совершенствования, он также способствует постоянному совершенствованию инструментов программирования, управляемых искусственным интеллектом. В конечном счете, это исследование является ценным ресурсом для исследователей и разработчиков ИИ, который поможет в дальнейшем совершенствовать языковые модели и их применение в программировании, автоматизации и более широком технологическом прогрессе. Непрерывная эволюция инструментов программирования на базе ИИ обещает изменить процесс разработки программного обеспечения, сделав его более эффективным и доступным.

Ключевые слова: ChatGPT, GPT-3.5, GPT-4, программирование на Python, OpenAI, Google Bard, Claude от Anthropic, разработка программного обеспечения, набор данных MBPP, производительность, сравнительный анализ, эффективность.

Introduction. In recent years, artificial intelligence (AI) has experienced remarkable growth, driven by advancements in machine learning and natural language processing. This progress has significantly impacted numerous industries and applications, reshaping traditional approaches. One of the areas garnering significant attention is AI-assisted programming. Advanced language models hold the potential to revolutionize how developers create, manage, optimize, and test code. The primary objective of this study is to evaluate ChatGPT, a popular large language model that encompasses GPT-4 and GPT-3.5 versions. By testing these models on various coding tasks, the research aims to analyze their capabilities and potential applications.

OpenAI's release of GPT models, particularly ChatGPT, has marked a significant step forward in AI's development and accessibility (An, et al., 2023). Each iteration of these models has brought increased adaptability and performance, sparking interest in their application across diverse fields. Specifically, in programming, these models exhibit immense potential for automating tasks, enhancing code quality, and providing valuable insights to developers. AI's influence on programming is both profound and transformative. By streamlining workflows, reducing human error, and automating repetitive processes, AI has demonstrated its ability to significantly enhance productivity. Tasks such as code generation, documentation creation, and bug detection are prime examples of how AI optimizes the development process, enabling programmers to focus on more complex and creative aspects of their work (Adamson, et al., 2023).

This study employs quantitative testing methods to explore the capabilities of ChatGPT, with three key objectives:

• Evaluating ChatGPT's efficiency in executing diverse coding tasks.

• Comparing GPT-4 and GPT-3.5 performance against other prominent large language models.

• Identifying challenges and limitations in the application of these models to programming tasks.

Recent advancements in automated code generation have been largely fueled by the development of large language models like GPT-3. These innovations have surpassed previous deep learning approaches, significantly enhancing AI's role in programming.

For instance, OpenAI Codex (Biswas, 2023), a specialized variant of GPT-3, demonstrated the ability to produce fully accurate solutions for 29% of unseen programming challenges in a single sample, with accuracy rising to 72% when extended to 100 samples. Additionally, a study investigating the Python coding capabilities of GPT models revealed a 28% success rate on problem-solving within limited sample sets.

Research by Hammond et al. (Kalla, et al., 2023) examined the application of OpenAI Codex and other large language models in addressing software vulnerabilities. Their findings highlighted that these models resolved 67% of security flaws within a dataset of historical bugs from open-source projects. On the other hand, Refs prioritized evaluating the practical usability of LLM-generated code rather than its correctness. (Firat, 2023) assessed GPT-Neo, GPT-J, and GPT-NeoX, large language models trained on diverse datasets containing code samples across 12 programming languages. (Rabbi, et al., 2024) provided a comprehensive review of existing LLMs designed for NL2Code (natural language to code), discussing their features from various perspectives. However, neither study focused on evaluating the quality or accuracy of the generated code.

This research provides a foundational exploration of programming with AI assistance, conducted during a period of rapid advancements in artificial intelligence. Its findings aim to inform developers, researchers, and the broader tech community, shaping the future trajectory of AI-driven programming.

The structure of this paper includes an overview of ChatGPT's development and the mechanisms underlying generative AI in Section 2. Experimental results are detailed in Section 4, while Section 5 concludes with a summary of the key findings.

Materials and methods. The evolution of modern generative AI has roots in the 1940s with the early development of artificial neural networks (ANNs). However, limited computational resources and insufficient understanding of the brain's biological mechanisms led to minimal interest in ANNs until the 1980s. This period marked significant progress due to advancements in hardware, neuroscience, and the introduction of the backpropagation algorithm, which made training ANNs much more practical. Prior to backpropagation, neural network training was a labor-intensive process, as no efficient method existed to compute error gradients relative to each neuron's weights. The adoption of backpropagation transformed this, unlocking the potential of ANNs (Ellis, et al., 2024).

In 2013, Kingma and Welling introduced variational autoencoders (VAEs) through their influential paper *Auto-Encoding Variational Bayes*. VAEs, a type of generative model based on variational inference, established a method for learning compressed data representations. This process involves encoding data into a lower-dimensional latent space and then reconstructing it into its original form via a decoding mechanism.

A pivotal advancement came in 2017 when Google researchers unveiled the Transformer architecture in their publication *Attention Is All You Need*. This innovative approach fundamentally altered language generation paradigms (Arias, et al., 2024). Unlike earlier models, such as long short-term memory (LSTM) networks (Hassani, et al., 2023) or recurrent neural networks (RNNs), Transformers utilized parallel processing to achieve superior contextual understanding and significantly improved performance (Campesato, 2024).

In 2021, OpenAI launched Codex, a specialized iteration of GPT trained on publicly available GitHub code. Early evaluations revealed that Codex could successfully solve about 30% of Python programming problems, a stark contrast to GPT-3, which achieved a 0% success rate on similar tasks. These results underscored the ability of large language models (LLMs) (Tomcsányiová, 2023) to effectively learn and generate code. Codex subsequently became the foundation for GitHub Copilot.

GitHub Copilot, powered by GPT-4, is an AI-driven programming assistant that integrates seamlessly with widely used code editors. It provides real-time code suggestions and can even generate entire segments of code. Controlled experiments revealed that developers using Copilot completed tasks approximately 55.8% faster than those who worked without it, illustrating the tool's potential to enhance programming efficiency (Ma. 2024).

Further studies have highlighted GPT's capabilities in Python code generation, particularly in aiding novice programmers with challenging problems using minimal input. However, these investigations also emphasize the necessity of human intervention to effectively guide ChatGPT's outputs (Siddiq, et al., 2024). Generative AI models utilize neural networks to analyze patterns and structures in existing datasets, enabling them to create original content . Inspired by human neuronal processes, these models learn from input data and generate outputs that align with the patterns they have internalized. Techniques such as generative adversarial networks (GANs) (Kaleemunnisa, et al., 2024), large language models (LLMs), variational autoencoders (VAEs), and Transformers are employed to produce content across diverse domains.

Methods like unsupervised and semi-supervised learning empower organizations to leverage vast amounts of unlabeled data for training, forming the basis for more advanced AI systems. These so-called foundation models, such as GPT-3 and Stable Diffusion, are capable of performing multiple tasks. For instance, they enable applications like ChatGPT to generate essays from brief prompts or Stable Diffusion to create lifelike images from textual descriptions (Porter, et al., 2024). Generative AI continuously improves its outputs through iterative training processes, analyzing data relationships to adjust parameters and reduce discrepancies between desired and generated outputs. This iterative refinement enhances the model's ability to produce high-quality, contextually relevant content. Typically, the content generation process begins with a user prompt, followed by repeated adjustments and explorations to fine-tune the results.

This section outlines the methodological framework employed to assess ChatGPT's performance and its comparison with other prominent large language models (LLMs). It begins with a description of the dataset utilized in the study, followed by an explanation of the evaluation strategy. The evaluation details include the testing procedures and the method for calculating the performance scores of the language models.

Selection of LLMs

With the rise in popularity of ChatGPT, competing companies have introduced their own large language models (LLMs) to stake their claim in the market. To provide a comprehensive evaluation, three additional widely recognized and accessible LLMs were selected for comparison based on their market presence and potential impact.

- Google Bard: Launched in February 2023, Bard represents Google's significant foray into the AI landscape. It is powered by the PaLM2 model and is known for its advanced reasoning capabilities, particularly in tasks such as mathematics and programming. Bard is freely available on its dedicated website.

- Microsoft Bing Chat: Released in February 2023, Bing Chat (referred to as Bing in this paper) is marketed as a "copilot for the web" and is also powered by GPT-4. As one of OpenAI's largest investors, Microsoft leverages this AI technology for its chatbot, which is publicly available via the Bing platform.

- Anthropic's Claude: Developed by a research company founded in 2021 by former OpenAI employees, Claude entered the market as an AI assistant in March 2023, with its upgraded version, Claude v2, released in July 2023. Similar to ChatGPT and Bard, Claude is designed for tasks including coding and other creative applications.

Dataset

This study utilized a dataset called Basic Python Programming (MPPP). which is widely recognized and tested for evaluating code generation capabilities in AI models. Developed by Google researchers, the dataset comprises approximately 1,000 crowd-sourced Python programming problems. These problems encompass a broad range of fundamental programming concepts and functionalities, making them suitable for assessment and designed to be solvable by entry-level programmers.

A subset of problems from this dataset was selected for analysis, ensuring representation across different programming topics and difficulty levels to effectively evaluate the performance of the tested LLMs.

Each problem in the Basic Python Programming (MPPP) dataset consists of the following components:

- Task_id: A unique identifier for each problem, ranging from 1 to 1000.
- Prompt: Instructions provided to the LLM, specifying the task the code should accomplish.
- Code: A proposed solution to the problem.
- Test_imports: Libraries that need to be imported for the code to function correctly.
- Test_list: Typically, three test cases used to validate whether the code produces the expected output.

An example of coding problem number 162 from the dataset, along with the corresponding test cases, is illustrated in Figure 1.This structure ensures that the AI models are tested for both their ability to generate functional code and their compatibility with various programming requirements (Tomcsányiová, 2023).

```
Task_id = "162"
prompt = """Write a function to calculate the sum (n - 2*i) from i=0 to n // 2,
           for instance n + (n-2) + (n-4)... (until n-x =< 0)."""
test_list = [
    "assert sum_series(6) == 12",
    "assert sum_series(10) == 30",
    "assert sum_series(9) == 25",
]
def sum_series(n):
  """Calculates the sum of the series (n - 2*i) from i=0 to n // 2.
  Args:
    n: The upper limit of the series.
  Returns:
    The sum of the series.
  """
  sum = 0
  for i in range(n // 2):
    sum += n - 2 * i
  return sum
```

Figure 1. Program code.

```
score = 0
for test in test_list:
    try:
        exec(test)
        score += 1
        print("Test passed\n")
    except AssertionError:
        # Extracting the code from the assert statement
        test_code = test.split("assert ")[1].split("==")[0]
        # Evaluate the extracted code and capture the computed result
        computed_result = eval(test_code)

        # Extracting the expected result from the assert statement
        expected_result = (test.split("==")[1])

        print("Test failed:")
        print("Test:", test)
        print("Expected:", expected_result)
        print("Computed:", computed_result)
        print()

print("Total score:", score, "/", len(test_list))
```

Figure 2. Program code.

Figure 1. Example of the used Python problems.

(a) The code problem number 162.

(b) The testing code.

This figure illustrates an example of a Python programming problem from the dataset, showing both the problem description and the testing code used to validate the solution generated by the AI models. The testing code includes specific test cases designed to ensure that the generated solution works correctly for various scenarios.

Evaluation Strategy

For this study, a purely quantitative approach was employed to assess the performance of the selected LLMs. Despite OpenAI offering paid API access to its GPT-3.5 and GPT-4 models, the decision was made to use the ChatGPT web interface for all tests. This choice was made because the web interface is the most accessible and user-friendly option, and is likely what most people will use. All models are accessible via their web interfaces, except for GPT-4, which is available only through a paid subscription to "ChatGPT Plus."

Bing and Bard do not offer API access, while Claude does; however, at the time of testing, Claude's API access was restricted by a waitlist. This led to the decision to conduct tests solely through the web interfaces to maintain consistency and ensure that all models were evaluated under similar conditions.

In the tests, each model was provided with programming prompts from the MBPP dataset. Only the "prompt" and the name of the function were given, to match the function name used in the test cases. Figure 2 illustrates an example of the prompt provided to the LLM, which served as input for the model's code generation. This evaluation strategy allowed for a clear, comparative analysis of the models' ability to generate functional Python code based on the given instructions.

```
I will give you a prompt, follow the prompt to write the python code that was
instructed. Do this in one code block, not in parts. If you need to import
something, do it in the code.

"prompt": "Write a function to find the shared elements from the given two
lists.",
name the function: similar_elements

Write ONLY the function
```

Figure -3. Example of the prompt used as an input to the LLM systems.

In the evaluation process, the generated code from the LLM systems was tested using assertion tests that verified whether the output passed or failed based on predefined test cases. The process involved multiple stages of testing and retesting to evaluate the models' capabilities in generating correct Python code.

1. Stage 1: Initial Evaluation

- The generated code for each task was tested using a list of test cases (assertion tests). A point was awarded for each successful test, with a score between 3 and 6 points per task.

- If the LLM produced code that passed all test cases, it received the maximum score of 3 points per task, with a possible total score of 305 points for 100 tasks.

2. Stage 2: Retesting of Low-Scoring Models

- The LLMs that scored the lowest in Stage 1 were retested with tasks they had failed or not completed successfully.

- This phase aimed to assess whether the models could generate correct code after receiving additional feedback or after multiple attempts.

3. Stage 3: New Conversation Evaluation

- To eliminate the influence of prior responses, the same prompt was provided in a new conversation. This stage tested whether the model could still generate correct code without any historical context.

- If a model struggled, feedback was provided after each response, and multiple retries (up to 10 attempts) were allowed to reach the correct code.

- The number of messages exchanged and attempts made before reaching a correct solution was recorded.

The goal of the evaluation was to determine whether the LLMs could generate correct and effective code across a variety of tasks, and to measure the efficiency of the models in completing these tasks when provided with adequate human feedback.

Results and discussion. The test was conducted using 460 certified Python programming tasks with a total possible score of 1,225 points, with each model being tested on all of these tasks. You can see a picture of the description in Figure 4. This allowed us to evaluate their performance on a representative sample of coding problems.- Claude: Scored the lowest among the five LLMs, achieving 873 points, which is equivalent to 70.43% of the total possible score. Although Claude showed competence in handling various tasks, it demonstrated the least overall success in generating correct Python code.

- Google Bard: Performed similarly to Claude, scoring 930 points, or 74.16%. While Bard showed a generally good understanding of the programming problems, there remains significant room for improvement, especially in more complex tasks.

- Comparative Performance: The results suggest that both Claude and Bard, while effective in some areas, still need enhancement to reliably solve a broader range of Python programming problems. Their performance indicates a need for further development in understanding specific coding tasks and generating error-free solutions.

These results provide valuable insights into the capabilities and limitations of current LLMs in solving coding problems. The findings highlight the importance of refining AI models to improve their problem-solving accuracy, particularly in more intricate programming scenarios.

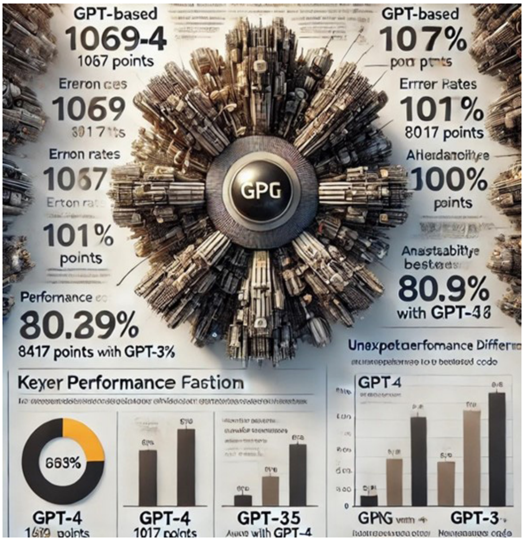

Figure- 4. Performance comparison between the different LLMs.

Code Quality

While the GPT-based models demonstrated strong performance, particularly GPT-4, the quality of the generated code remains a critical factor to evaluate. GPT-4 scored an impressive 1069 points (84.31%), outperforming the other models in terms of task completion success rate. Bing, powered by GPT-4, also performed remarkably well, scoring 1001 points (80.93%), which was slightly lower than GPT-3.5's score of 1017 points (81.18%).

Despite the high scores, the difference in performance between GPT-3.5 and Bing was surprising, as Bing is based on the newer GPT-4 model. This suggests that while GPT-4 demonstrates higher raw performance, the quality of generated code can still vary depending on specific implementation nuances or training methods between different platforms using the same underlying model.

The high performance of GPT-4 reflects its advanced capabilities in handling complex programming problems. However, code quality also depends on factors like error rates, readability, and adherence to best practices. While GPT-4 shows the highest success rate, it's important to consider not only whether the code passes tests but also how efficiently and clearly the code is written, which remains a challenge for all models. Further refinement may be needed to improve both correctness and the maintainability of the generated code across all LLMs.

Providing Feedback to the Model

In the second phase of the tests, the models were retested on tasks they had failed during the first phase, with feedback provided to help them understand and correct the errors in their code. The results of this phase revealed that GPT-4, the highest-scoring model, showed significant improvement. Out of the 19 tasks it had failed initially, GPT-4 was able to successfully complete 17 tasks after receiving feedback. This demonstrates its strong ability to learn from errors and adapt its responses.

In contrast, Bard, which scored lower in the initial tests, showed a much more limited ability to improve after feedback. Bard was only able to complete 6 of the previously failed tasks. This stark difference suggests that while both models are capable of learning and improving through feedback, GPT-4's performance is more robust and reliable when it comes to correcting mistakes.

Figure 5 and table 1 highlights the superiority of GPT-4 in adaptation and feedback-based learning. which compares the results of both models in this phase, highlights this disparity, emphasizing GPT-4's superior adaptability and problem-solving capabilities when provided with guidance. This phase of testing further supports GPT-4's position as the top performer in terms of both initial success rate and the ability to learn from feedback.

Table 1- highlights the superiority of GPT-4 in adaptation and feedback-based learning.

Model	Initially Failed Tasks	Successfully Corrected After Feedback	Improvement Rate (%)
GPT-4	19	17	89.47%
Bard	–	6	–

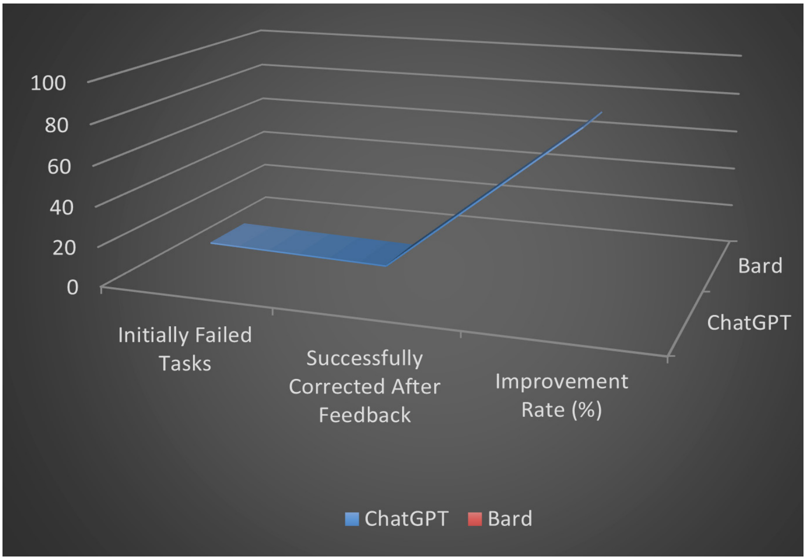

Figure -5. Comparison of GPT-4 and Bard's score when providing feedback.

These tests demonstrated a significant difference in how GPT-4 and Bard processed and applied feedback to the generated code. GPT-4 showed an impressive ability to comprehend the feedback and effectively incorporate fixes into the code. With each new output, GPT-4's code improved until it performed the desired task without errors, highlighting its strong adaptability and problem-solving skills.

On the other hand, Google's Bard, although occasionally able to generate code effectively, did not exhibit the same level of feedback comprehension. While Bard's responses typically acknowledged the errors, the "fixed" versions of the code it generated often consisted of just a repeat of the same code without meaningful adjustments. This suggests that Bard struggled to internalize the feedback and make the necessary corrections.

Overall, these results underscore the proficiency of AI language models, particularly GPT-4, in generating code and adapting to real-world coding scenarios, including bug fixing. When properly used, GPT-4 demonstrated the ability to solve nearly 100% of tasks, reinforcing its potential as an excellent coding assistant. However, the testing also revealed that LLMs, like GPT-4 and Bard, are most effective as coding assistants rather than replacements for human programmers. These models benefit significantly from human feedback, which helps them improve and refine their outputs.

The findings also highlight the transformative potential of LLMs in programming and software development. By providing valuable insights into how these models can assist with coding tasks, the study suggests that they could play a crucial role in enhancing productivity and solving complex programming challenges, as long as they are paired with effective human guidance.

Conclusion. This research evaluated various LLMs, including GPT-3.5, GPT-4, Bard, Bing, and Claude, using the Mostly Basic Python Problems (MBPP) dataset for code generation tasks. The study provided insightful results into the strengths and weaknesses of these models and highlighted areas where they could improve. Here are the key takeaways and suggestions for future research:

1. Proficiency of GPT Models: GPT-4 demonstrated the highest proficiency in code generation tasks, achieving a success rate of 80.23% on the tested subset of problems. GPT-based models consistently outperformed the other models (Bard and Claude) in terms of code generation accuracy and problem-solving abilities.

2. Performance of Competitor Models: Bard and Claude performed the worst among the models tested. This indicates a need for improvement in their coding capabilities, despite being the most recently released models. These models struggled with generating accurate code and often produced multiple solutions to a given problem, leading to inefficiencies.

3. Feedback and Monitoring Requirement: The study found that LLMs, including ChatGPT, require constant human feedback and monitoring to improve their code generation outputs. While they can assist programmers in writing code, they are not replacements for human developers, as they often produce bugs, errors, and multiple solutions when there are slight changes or incorrect queries.

4. Room for Improvement: The results indicated that while LLMs are good at generating code, there are still significant areas for improvement. Bugs and errors in the generated code, inconsistent solutions, and inefficiencies in resource usage are common issues that need addressing.

1. Evaluating Code Quality: Future studies should focus on evaluating generated

code itself using new metrics like code cleanliness, execution time, resource usage, and memory consumption. These metrics would provide a more comprehensive view of code performance beyond correctness.

2. Testing with More Complex Tasks: The MBPP dataset contains basic coding problems. Future research can explore more complex tasks, including algorithms, data structures, and real-world scenarios to test the capabilities of LLMs in handling more sophisticated coding challenges.

3. Real-World Applications: Future studies can investigate how LLMs like GPT-4can be applied to professional software development processes beyond code generation. For example, researchers can explore their use in code reviews, bug fixing, collaborative coding, and automating documentation. This would help determine their potential to enhance productivity in the industry and their practical application in real-world software development.

4. Investigating Novel Feedback Mechanisms: Given the feedback requirement highlighted by this research, exploring novel feedback mechanisms could improve the LLMs' code generation process. For example, integrating human-in-the-loop systems, real-time code reviews, and iterative feedback cycles could enhance the LLM's ability to learn from feedback and improve code output.

5. Comparative Studies with Human Programmers: Conducting comparative studies between human programmers and LLMs in terms of productivity, code quality, and problem-solving abilities would provide valuable insights into how these AI systems can complement human efforts in software development.

6. Ethical and Societal Implications: As LLMs become more embedded in software development, there are ethical and societal implications to consider, such as bias in generated code, security vulnerabilities, and privacy concerns. Future research should address these issues and propose guidelines for using LLMs responsibly.

7. Interdisciplinary Research: Combining expertise from fields like software engineering, human-computer interaction, cognitive science, and AI ethics could offer new insights into the use of LLMs in software development, potentially leading to more robust and practical applications.

All things considered, our study showed that LLMs like as GPT-4 have a lot of promise as programming assistants since they can provide code that successfully resolves a variety of programming issues. But for them to work, human supervision and ongoing feedback are necessary. Future studies should concentrate on improving these models by taking into account increasingly difficult tasks, novel metrics for evaluation, creative feedback systems, and real-world software development process applications. This will help academics and developers better understand the advantages and disadvantages of LLMs and how to use them to improve software development and code production.

References

An J., Ding W., Lin C. ChatGPT. tackle the growing carbon footprint of generative AI. – 2023. – T. 615. – C. 586. (in English)

Adamson V., Bägerfeldt J. Assessing the effectiveness of ChatGPT in generating Python code. – 2023. (in English)

Arias R. et al. Limitations and Benefits of the ChatGPT for Python Programmers and Its Tools for Evaluation. Computer Science On-line Conference. – Cham: Springer Nature Switzerland, 2024. – C. 171-194. (in English)

Biswas S.S. Potential use of chat gpt in global warming. Annals of biomedical engineering. – 2023. – T. 51. – №. 6. – C. 1126-1127. (in English)

Campesato O. Python 3 and Machine Learning Using ChatGPT/GPT-4. – Walter de Gruyter GmbH & Co KG, 2024. (in English)

Ellis M. E., Casey K. M., Hill G. ChatGPT and Python programming homework. Decision Sciences Journal of Innovative Education. – 2024. – T. 22. – №. 2. – C. 74-87. (in English)

Firat, Mehmet. "How Chat GPT Can Transform Autodidactic Experiences and Open Education?" OSF Preprints, 12 Jan. 2023. Web. (in English)

Hassani H., Silva E. S. The role of ChatGPT in data science: how ai-assisted conversational interfaces are revolutionizing the field. Big data and cognitive computing. – 2023. – T. 7. – №. 2. – C. 62. (in English)

Kalla D. et al. Study and analysis of chat GPT and its impact on different fields of study. International journal of innovative science and research technology. – 2023. – T. 8. – №. 3. (in English)

Kaleemunnisa F.N.U. et al. ChatGPT in the Classroom: Experimentation in a Python Class for Non-Computing Majors, 2024 IEEE Integrated STEM Education Conference (ISEC). – IEEE, 2024. – C. 1-7. (in English)

Ma B., Chen L., Konomi S. Enhancing programming education with ChatGPT: a case study on student perceptions and interactions in a Python course. International Conference on Artificial Intelligence in Education. – Cham: Springer Nature Switzerland, 2024. – C. 113-126. (in English)

Porter L., Zingaro D. Learn AI-assisted Python Programming. – Simon and Schuster, 2024. (in English)

Rabbi M. F. et al. AI writes, we analyze: The ChatGPT python code saga. Proceedings of the 21st International Conference on Mining Software Repositories. – 2024. – C. 177-18. (in English)

Siddiq M.L. et al. Quality assessment of chatgpt generated code and their use by developers. Proceedings of the 21st International Conference on Mining Software Repositories. – 2024. – C. 152-156. (in English)

Tomcsányiová M. ANALYSIS OF MISCONCEPTIONS IN PYTHON PROGRAMS USING CHATGPT. ICERI2023 Proceedings. – IATED, 2023. – C. 281-288. (in English)

CONTENTS

INFORMATION AND COMMUNICATION TECHNOLOGIES

A.Abdiraman, L.Aldasheva, A.Zakirova, B.Mukhametzhanova, I.Orman
GLOBAL ANALYSIS OF MOBILE BROADBAND NETWORK PERFORMANCE: INSIGHTS INTO 5G DEPLOYMENT AND FUTURE 6G CHALLENGES..5

R. Abdualiyeva, L. Smagulova, A. Yelepbergenova
THE EFFECTIVENESS OF USING CHATGPT IN PROGRAMMING.............17

A.B. Aben, N.M. Zhunissov, G.N. Kazbekova, A.N. Amanov, A.A. Abibullayeva
DEEPFAKE ARTIFICIAL VOICE DETECTION. COMPARISON OF THE EFFECTIVENESS OF THE LSTM AND CNN MODELS..................................32

A.A. Aitkazina, N.O. Zhumazhan
DEVELOPMENT OF A BIOTECHNICAL SYSTEM FOR LASER TREATMENT OF SUNFLOWER SEEDS..49

G. Aksholak, A. Bedelbayev, R. Magazov
SECURING KUBERNETES: AN ANALYSIS OF VULNERABILITIES, TOOLS, AND FUTURE DIRECTIONS...66

A.T. Akynbekova, A.A. Mukhanova, Salah Al-Majeed, A.G. Altayeva
PROBLEMS OF IMPLEMENTATION OF FUZZY MODELS OF DECISION MAKING IN SOCIAL PROCESSES..78

K.M. Aldabergenova, M.A. Kantureyeva, A.B. Kassekeyeva, A. Akhmetova, T.N. Esikova
FEATURES AND PROSPECTS FOR THE USE OF DIGITAL PLATFORMS AND INTERNET MARKETING IN THE DEVELOPMENT OF AGRICULTURAL PRODUCTION..93

A. Yerimbetova, M. Sambetbayeva, E. Daiyrbayeva, B. Sakenov, U. Berzhanova
CREATING A MODEL FOR RECOGNIZING THE KAZAKH SIGN LANGUAGE USING THE DEEP LEARNING METHOD...............................108

A.N. Zhidebayeva, S.T. Akhmetova, A.O. Aliyeva, B.O. Tastanbekova, G.S. Shaimerdenova
REVIEW OF DETECTION AND PREVENTION OF OFFENSIVE LANGUAGE VIA SOCIAL MEDIA DATA MINING..124

K.S. Ivanov, D.T. Tulekenova
ENSURING THE DETERMINABILITY OF MOTION OF AN ADAPTIVE SPACECRAFT DRIVE BY INTRODUCING AN ADDITIONAL VELOCITY CONSTRAINT FORCE.......136

M.N. Kalimoldayev, Z.D. Ormansha, K.B. Begalieva, A.S. Ainagulova, A.O. Aukenova
A BLOCKCHAIN MODEL FOR AGRICULTURAL PRODUCT TRACKING THAT SUPPORTS FEDERAL TRAINING.......151

I. Massyrova, O. Joldasbayev, S. Joldasbayev, A. Bolysbek, S. Mambetov
AUTOMATION OF THE SYSTEM FOR INDUSTRIAL PRACTICE AND INTERNSHIPS FOR STUDENTS IN ORGANIZATIONS OUTSIDE OF THE UNIVERSITY.......168

A.B. Mimenbayeva, G.O. Issakova, G.K. Bekmagambetova, A.B. Aruova, E.K. Darikulova
DEVELOPMENT OF DEEP LEARNING MODELS FOR FIRE SOURCES PREDICTION.......185

K. Momynzhanova, S.Pavlov, Sh. Zhumagulova
MATHEMATICAL MODELS AND PRACTICAL IMPLEMENTATION OF AN OPTICAL-ELECTRONIC EXPERT SYSTEM FOR GLAUCOMA DETECTION.......202

B.O. Mukhametzhanova, L.N. Kulbaeva, Z.B. Saimanova, E.K. Seipisheva, B.M. Sadanova
OPTIMIZATION AND INTEGRATION OF DOCKER TECHNOLOGY IN MODERN INFORMATION SYSTEMS.......218

A.R. Orazayeva, J.A. Tussupov, A.K. Shaikhanova, G.B. Bekeshova, A.D. Galymova
FUZZY EXPERT SYSTEM FOR ASSESSING DYNAMIC CHANGES IN BIOMEDICAL IMAGES OF BREAST CANCER TUMORS.......227

D. Oralbekova, O. Mamyrbayev, A. Akhmediyarova, D. Kassymova
USING KAZAKH NER DATASETS FOR MULTICLASS CLASSIFICATION IN THE LEGAL DOMAIN: A COMPARATIVE STUDY OF BERT, GPT, AND LSTM MODELS.......242

A. Ospanov, A.J. Pedro, T. Turymbetov, K. Dyussekeyev, A. Zhumadillayeva
ADVANCEMENTS IN ERP SYSTEMS THROUGH EMERGING

TECHNOLOGIES, MACHINE LEARNING AND HYBRID OPTIMIZATION TECHNIQUES..........259

K. Rabbany, A. Bekarystankyzy, A. Shoiynbek, D. Kuanyshbay, A. Mukhametzhanov
DETECTION OF SUICIDAL TENDENCIES IN REDDIT POSTS USING MACHINE LEARNING..........270

A. Taukenova
PERSONALIZED ARCHITECTURE: CREATING UNIQUE SPACES WITH DIGITAL TECHNOLOGIES..........283

МАЗМҰНЫ

АҚПАРАТТЫҚ-КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР

Ә. Әбдіраман, Л. Алдашева, А. Закирова, Б. Мухаметжанова, И. Орман
МОБИЛЬДІ КЕН ЖОЛАҚТЫ ЖЕЛІЛЕРДІҢ ТИІМДІЛІГІНІҢ ЖАҺАНДЫҚ ТАЛДАУ: 5G ЕНГІЗУ ЖӘНЕ 6G БОЛАШАҚ МӘСЕЛЕЛЕРІ..5

Р.Е. Абдуалиева, Л.А. Смагулова, А.У. Елепбергенова
БАҒДАРЛАМАЛАУДА CHATGPT ҚОЛДАНУ ТИІМДІЛІГІ........................17

А.Б. Абен, Н.М. Жунисов, Г.Н. Казбекова, А.Н. Аманов, А.А. Абибуллаева
DEEPFAKE ЖАСАНДЫ ДАУЫСТЫ АНЫҚТАУ. LSTM ЖӘНЕ CNN МОДЕЛЬДЕРІНІҢ ТИІМДІЛІГІ САЛЫСТЫРУ........................32

Ә.А. Айтқазина, Н.Ө. Жұмажан
КҮНБАҒЫС ТҰҚЫМДАРЫН ЛАЗЕРМЕН ӨҢДЕУГЕ АРНАЛҒАН БИОТЕХНИКАЛЫҚ ЖҮЙЕНІ ДАМЫТУ........................49

Г.И. Ақшолақ, А.А. Бедельбаев, Р.С. Мағазов
KUBERNETES-ТІ ҚОРҒАУ: ОСАЛДЫҚТАРДЫ, ҚҰРАЛДАРДЫ ЖӘНЕ БОЛАШАҚ БАҒЫТТАРДЫ ТАЛДАУ........................66

А.Т. Ақынбекова, А.А. Муханова, Salah Al-Majeed, Г.С. Алтаева
ӘЛЕУМЕТТІК ПРОЦЕСТЕРДЕ ШЕШІМДЕР ҚАБЫЛДАУДЫҢ БҰЛДЫР МОДЕЛЬДЕРІН ЕНГІЗУ МӘСЕЛЕЛЕРІ........................78

К.М. Алдабергенова, М.А. Кантуреева, А.Б. Касекеева, А.Ж. Ахметова, Т.Н. Есикова
АУЫЛ ШАРУАШЫЛЫҒЫ ӨНДІРІСІН ДАМЫТУДА ЦИФРЛЫҚ ПЛАТФОРМАЛАР МЕН ИНТЕРНЕТ-МАРКЕТИНГТІ ҚОЛДАНУДЫҢ ЕРЕКШЕЛІКТЕРІ МЕН ПЕРСПЕКТИВАЛАРЫ........................93

А.С. Еримбетова, М.А. Сэмбетбаева, Э.Н. Дайырбаева, Б.Е. Сэкенов, У.Г. Бержанова
ТЕРЕҢ ОҚЫТУ ӘДІСІН ҚОЛДАНУ АРҚЫЛЫ ҚАЗАҚ ЫМ ТІЛІН ТАНУҒА АРНАЛҒАН МОДЕЛЬ ҚҰРУ........................108

А.Н. Жидебаева, С.Т. Ахметова, А.О. Алиева, Б.О. Тастанбекова, Г.С. Шаймерденова
ӘЛЕУМЕТТІК ЖЕЛІЛЕРДЕН DATA MINING АРҚЫЛЫ БЕЙӘДЕП СӨЗДЕРДІ АНЫҚТАУ ЖӘНЕ АЛДЫН АЛУҒА ШОЛУ..............................124

К.С. Иванов, Д.Т. Тулекенова
ЖЫЛДАМДЫҚ БАЙЛАНЫСЫНЫҢ ҚОСЫМША КҮШІН ЕНГІЗУ АРҚЫЛЫ ҒАРЫШ АППАРАТЫНЫҢ БЕЙІМДЕЛГЕН ЖЕТЕК ҚОЗҒАЛЫСЫНЫҢ АЙҚЫНДЫЛЫҒЫН ҚАМТАМАСЫЗ ЕТУ................136

М.Н. Калимолдаев, З.Д. Орманша, К.Б. Бегалиева, А.С. Айнагулова, А.О. Аукенова
ФЕДЕРАТИВТІ ОҚЫТУДЫ ҚОЛДАЙТЫН АУЫЛШАРУАШЫЛЫҚ ӨНІМДЕРІН БАҚЫЛАУҒА АРНАЛҒАН БЛОКЧЕЙН МОДЕЛІ.................151

И. Масырова, О.К. Джолдасбаев, С.К. Джолдасбаев, А. Болысбек, С.Т. Мамбетов
УНИВЕРСИТЕТТЕН ТЫС ҰЙЫМДАРДА СТУДЕНТТЕРДІҢ ӨНДІРІСТІК ПРАКТИКАСЫ МЕН ТАҒЫЛЫМДАМАСЫН АВТОМАТТАНДЫРУ ЖҮЙЕСІ..168

А.Б. Мименбаева, Г.О. Исакова, Г.К. Бекмагамбетова, Ә.Б. Аруова, Е.Қ. Дәріқұлова
ӨРТ КӨЗДЕРІН БОЛЖАУ ҮШІН ТЕРЕҢ ОҚЫТУ МОДЕЛЬДЕРІН ӘЗІРЛЕУ...185

К.Р. Момынжанова, С.В. Павлов, Ш.П. Жұмағұлова, М.Т. Тұңғышбаев
ГЛАУКОМАНЫ АНЫҚТАУҒА АРНАЛҒАН ОПТИКАЛЫҚ-ЭЛЕКТРОНДЫҚ САРАПТАМАЛЫҚ ЖҮЙЕНІҢ МАТЕМАТИКАЛЫҚ МОДЕЛЬДЕРІ МЕН ПРАКТИКАЛЫҚ ІСКЕ АСЫРЫЛУЫ.........................202

Б.О. Мухаметжанова, Л.Н. Кулбаева, З.Б. Сайманова, Э.К. Сейпишева, Б.М. Саданова
ЗАМАНАУИ АҚПАРАТТЫҚ ЖҮЙЕЛЕРДЕГІ DOCKER ТЕХНОЛОГИЯСЫН ОҢТАЙЛАНДЫРУ ЖӘНЕ ИНТЕГРАЦИЯЛАУ.......218

А.Р. Оразаева, Д.А. Тусупов, А.К. Шайханова, Г.Б. Бекешова, Ә.Д. Ғалымова
СҮТ БЕЗІ ҚАТЕРЛІ ІСІГІ КЕЗІНДЕ БИОМЕДИЦИНАЛЫҚ КЕСКІНДЕРІНДЕГІ ДИНАМИКАЛЫҚ ӨЗГЕРІСТЕРДІ БАҒАЛАУҒА АРНАЛҒАН АНЫҚ ЕМЕС САРАПТАМА ЖҮЙЕСІ....................................227

Д. Оралбекова, О. Мамырбаев, А. Ахмедиярова, Д. Қасымова
ҚАЗАҚ ТІЛІНДЕГІ NER ДЕРЕКТЕР ЖИНАҒЫН ҚҰҚЫҚТЫҚ САЛАДА КӨПСАНАТТЫ ЖІКТЕУ ҮШІН ПАЙДАЛАНУ: BERT, GPT ЖӘНЕ LSTM МОДЕЛЬДЕРІНІҢ САЛЫСТЫРМАЛЫ ЗЕРТТЕУІ...........................242

А. Оспанов, П. Алонсо-Жорда, Т. Тұрымбетов, К. Дүйсекеев, А. Жұмаділлаева
ERP ЖҮЙЕЛЕРІНІҢ ЖЕТІЛДІРІЛУІ: ЗАМАНАУИ ТЕХНОЛОГИЯЛАР, МАШИНАЛЫҚ ОҚЫТУ ЖӘНЕ ГИБРИДТІ ОПТИМИЗАЦИЯ ӘДІСТЕРІ...259

К. Раббани, А. Бекарыстанқызы, Д. Қуанышбай, А. Шойынбек, А. Мұхаметжанов
МАШИНАЛЫҚ ОҚЫТУДЫ ПАЙДАЛАНУ АРҚЫЛЫ REDDIT ПОСТТАРЫНДАҒЫ СУИЦИДТІК ТЕНДЕНЦИЯЛАРЫН АНЫҚТАУ...270

Ә. Таукенова
ЖЕКЕЛЕНДІРІЛГЕН АРХИТЕКТУРА: ДИДЖИТАЛ ТЕХНОЛОГИЯЛАРМЕН ЕРЕКШЕ КЕҢІСТІКТЕР ЖАРАТУ...................283

СОДЕРЖАНИЕ

ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

А. Абдираман, Л. Алдашева, А. Закирова, Б. Мухаметжанова, И. Орман
ГЛОБАЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ МОБИЛЬНОЙ ШИРОКОПОЛОСНОЙ СЕТИ: ВНЕДРЕНИЕ 5G И БУДУЩИЕ ЗАДАЧИ 6G.........5

Р.Е. Абдуалиева, Л.А. Смагулова, А.У. Елепбергенова
ЭФФЕКТИВНОСТЬ ИСПОЛЬЗОВАНИЯ CHATGPT В ПРОГРАММИРОВАНИИ.........17

А.Б. Абен, Н.М. Жунисов, Г.Н. Казбекова, А.Н. Аманов, А.А. Абибуллаева
ОБНАРУЖЕНИЕ ИСКУССТВЕННОГО ГОЛОСА DEEPFAKE. СРАВНЕНИЕ ЭФФЕКТИВНОСТИ МОДЕЛЕЙ LSTM И CNN.........32

А.А. Айтказина, Н.О. Жумажан
РАЗРАБОТКА БИОТЕХНИЧЕСКОЙ СИСТЕМЫ ДЛЯ ЛАЗЕРНОЙ ОБРАБОТКИ СЕМЯН ПОДСОЛНЕЧНИКА.........49

Г.И. Акшолак, А.А. Бедельбаев, Р.С. Магазов
ЗАЩИТА KUBERNETES: АНАЛИЗ УЯЗВИМОСТЕЙ, ИНСТРУМЕНТОВ И НАПРАВЛЕНИЙ НА БУДУЩЕЕ.........66

А.Т. Акынбекова, А.А. Муханова, Salah Al-Majeed, Г.С. Алтаева
ПРОБЛЕМЫ РЕАЛИЗАЦИИ НЕЧЕТКИХ МОДЕЛЕЙ ПРИНЯТИЯ РЕШЕНИЙ В СОЦИАЛЬНЫХ ПРОЦЕССАХ.........78

К.М. Алдабергенова, М.А. Кантуреева, А.Б. Касекеева, А.Ж. Ахметова, Т.Н. Есикова
ОСОБЕННОСТИ И ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ЦИФРОВЫХ ПЛАТФОРМ И ИНТЕРНЕТ-МАРКЕТИНГА В РАЗВИТИИ СЕЛЬСКОХОЗЯЙСТВЕННОГО ПРОИЗВОДСТВА.........93

А.С. Еримбетова, М.А. Самбетбаева, Э.Н. Дайырбаева, Б.Е. Сакенов, У.Г. Бержанова
СОЗДАНИЕ МОДЕЛИ ДЛЯ РАСПОЗНАВАНИЯ КАЗАХСКОГО ЖЕСТОВОГО ЯЗЫКА С ИСПОЛЬЗОВАНИЕМ МЕТОДА ГЛУБОКОГО ОБУЧЕНИЯ.........108

А.Н. Жидебаева, С.Т. Ахметова, А.О. Алиева, Б.О. Тастанбекова, Г.С. Шаймерденова
ОБЗОР ОБНАРУЖЕНИЯ И ПРЕДОТВРАЩЕНИЯ ОСКОРБИТЕЛЬНОЙ ЛЕКСИКИ С ПОМОЩЬЮ DATA MINING В СОЦИАЛЬНЫХ СЕТЯХ…124

К.С. Иванов, Д.Т. Тулекенова
ОБЕСПЕЧЕНИЕ ОПРЕДЕЛИМОСТИ ДВИЖЕНИЯ АДАПТИВНОГО ПРИВОДА КОСМИЧЕСКОГО АППАРАТА С ПОМОЩЬЮ ВВЕДЕНИЯ ДОПОЛНИТЕЛЬНОЙ СИЛЫ СКОРОСТНОЙ СВЯЗИ……………......……..136

М.Н. Калимолдаев, З.Д. Орманша, К.Б. Бегалиева, А.С. Айнагулова, А.О. Аукенова
БЛОКЧЕЙН-МОДЕЛЬ ДЛЯ ОТСЛЕЖИВАНИЯ СЕЛЬСКОХОЗЯЙСТВЕННОЙ ПРОДУКЦИИ С ПОДДЕРЖКОЙ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ………...…..151

И. Масырова, О.К. Джолдасбаев, С.К. Джолдасбаев, А. Болысбек, С.Т. Мамбетов
АВТОМАТИЗАЦИЯ СИСТЕМЫ ДЛЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ И СТАЖИРОВКИ СТУДЕНТОВ В ОРГАНИЗАЦИЯХ ВНЕ ВУЗА……………..…168

А. Мименбаева, Г. Исакова, Г.К. Бекмагамбетова, А.Б. Аруова, Е.К. Дарикулова
РАЗРАБОТКА МОДЕЛЕЙ ГЛУБОКОГО ОБУЧЕНИЯ ПРОГНОЗИРОВАНИЯ ИСТОЧНИКОВ ПОЖАРОВ…………………….....185

К.Р. Момынжанова, С.В. Павлов, Ш.П. Жумагулова, М.Т. Тунгушбаев
МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ОПТИКО-ЭЛЕКТРОННОЙ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ВЫЯВЛЕНИЯ ГЛАУКОМЫ……..202

Б.О. Мухаметжанова, Л.Н. Кулбаева, З.Б. Сайманова, Э.К. Сейпишева, Б.М. Саданова
ОПТИМИЗАЦИЯ И ИНТЕГРАЦИЯ ТЕХНОЛОГИИ DOCKER В СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ……….……….218

А.Р. Оразаева, Д.А. Тусупов, А.К. Шайханова, Г.Б. Бекешова, А.Д. Галымова
НЕЧЕТКАЯ ЭКСПЕРТНАЯ СИСТЕМА ДЛЯ ОЦЕНКИ ДИНАМИЧЕСКИХ ИЗМЕНЕНИЙ В БИОМЕДИЦИНСКИХ ИЗОБРАЖЕНИЯХ ОПУХОЛЕЙ ПРИ РАКЕ МОЛОЧНОЙ ЖЕЛЕЗЫ………..………………………………..227

Д. Оралбекова, О. Мамырбаев, А. Ахмедиярова, Д. Касымова
ИСПОЛЬЗОВАНИЕ НАБОРОВ ДАННЫХ NER НА КАЗАХСКОМ ЯЗЫКЕ ДЛЯ МУЛЬТИКЛАССИФИКАЦИИ В ПРАВОВОЙ СФЕРЕ: СРАВНИТЕЛЬНОЕ ИССЛЕДОВАНИЕ МОДЕЛЕЙ BERT, GPT И LSTM……………….......................................…………………242

А. Оспанов, П. Алонсо-Жорда, Т. Турымбетов, К. Дюсекеев, А. Жумадиллаева
ПРОДВИЖЕНИЕ ERP СИСТЕМ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ, МАШИННОГО ОБУЧЕНИЯ И ГИБРИДНЫХ МЕТОДОВ ОПТИМИЗАЦИИ……………….............……………………………….259

К. Раббани, А. Бекарыстанкызы, Д. Куанышбай, А. Шойынбек, А. Мухаметжанов
ОБНАРУЖЕНИЕ СУИЦИДАЛЬНЫХ ТЕНДЕНЦИЙ В ПУБЛИКАЦИЯХ НА REDDIT С ИСПОЛЬЗОВАНИЕМ МАШИННОГО ОБУЧЕНИЯ…..…..270

А. Таукенова
ПЕРСОНАЛИЗИРОВАННАЯ АРХИТЕКТУРА: СОЗДАНИЕ УНИКАЛЬНЫХ ПРОСТРАНСТВ С ПОМОЩЬЮ ЦИФРОВЫХ ТЕХНОЛОГИЙ………..…….283

Publication Ethics and Publication Malpracticein
the journals of the National Academy of Sciences of the Republic of Kazakhstan

For information on Ethics in publishing and Ethicalguidelines for journal publication see http://www.elsevier.com/publishingethics and http://www.elsevier.com/journal-authors/ethics.

Submission of an article to the National Academy of Sciences of the Republic of Kazakhstan implies that the described work has not been published previously (except in the form of an abstract or as part of a published lecture or academic thesis or as an electronic preprint, see http://www.elsevier.com/postingpolicy), that it is not under consideration for publication elsewhere, that its publication is approved by all authors and tacitly or explicitly by the responsible authorities where the work was carried out, and that, if accepted, it will not be published elsewhere in the same form, in English or in any otherlanguage, including electronically without the written consent of the copyright-holder. In particular, translations into English of papers already published in another language are not accepted.

No other forms of scientific misconduct are allowed, such as plagiarism, falsification, fraudulent data, incorrect interpretation of other works, incorrect citations, etc. The National Academy of Sciences of the Republic of Kazakhstan follows the Code of Conduct of the Committee on Publication Ethics (COPE), and follows the COPE Flowcharts for Resolving Cases of Suspected Misconduct (http://publicationethics.org/files/u2/New_Code.pdf). To verify originality, your article may be checked by the Cross Check originality detection service http://www.elsevier.com/editors/plagdetect.

The authors are obliged to participate in peer review process and be ready to provide corrections, clarifications, retractions and apologies when needed. All authors of a paper should have significantly contributed to the research.

The reviewers should provide objective judgments and should point out relevant published works which are not yet cited. Reviewed articles should be treated confidentially. The reviewers will be chosen in such a way that there is no conflict of interests with respect to the research, the authors and/or the research funders.

The editors have complete responsibility and authority to reject or accept a paper, and they will only accept a paper when reasonably certain. They will preserve anonymity of reviewers and promote publication of corrections, clarifications, retractions and apologies when needed. The acceptance of a paper automatically implies the copyright transfer to the National Academy of Sciences of the Republic of Kazakhstan.

The Editorial Board of the National Academy of Sciences of the Republic of Kazakhstan will monitor and safeguard publishing ethics.

Правила оформления статьи для публикации в журнале смотреть на сайтах:
www:nauka-nanrk.kz
http://physics-mathematics.kz/index.php/en/archive
ISSN2518-1726 (Online),
ISSN 1991-346X (Print)

Директор отдела издания научных журналов НАН РК *А. Ботанқызы*
Редакторы: *Д.С. Аленов, Ж.Ш.Әден*
Верстка на компьютере *Г.Д.Жадыранова*

Подписано в печать 20.03.2025.
Формат 60x881/8. Бумага офсетная. Печать –ризограф.
20,0 п.л. Заказ 1.

Национальная академия наук РК
050010, Алматы, ул. Шевченко, 28, т. 272-13-19