

SUN'IY INTELLEKT TEXNOLOGIYALARINING DASTURLASH JARAYONIGA INTEGRATSIYALASHUVI VA UNING SAMARADORLIKKA TA'SIRI**Asatullayeva Hosila Sobir qizi**

Termiz davlat universiteti

60610300-Kompyuter injiniringi yo'nalishi talabasi.

<https://doi.org/10.5281/zenodo.18675246>

Annotatsiya. Ushbu maqolada sun'iy intellekt (SI) texnologiyalarining dasturlash jarayoniga integratsiyalashuvi va uning samaradorlikka ta'sirini tahlil qilinadi. Tadqiqot natijalari kod yozish tezligi va xatoliklarni kamaytirish bo'yicha o'tkazilgan tajribalarni o'z ichiga oladi. Shuningdek, maqolada global kompaniyalar — Google, Microsoft, Amazon va Meta — tajribalariga asoslanib, SI'ni dasturlash jarayoniga tatbiq etish amaliyoti va uning samaradorlikka ta'siri ko'rib chiqiladi.

Kalit so'zlar: Sun'iy intellekt, dasturlash, kod generatsiyasi, kod tahlili, xatoliklarni aniqlash, samaradorlik, AIOps, AI-assisted development, global kompaniyalar, dasturiy ta'minot.

Аннотация. В данной статье анализируется интеграция технологий искусственного интеллекта (ИИ) в процесс программирования и ее влияние на эффективность. В исследовании представлены результаты экспериментов, направленных на оценку скорости написания кода и сокращения числа ошибок. Кроме того, в работе рассматривается практика внедрения ИИ в процессы разработки программного обеспечения на основе опыта ведущих глобальных компаний — Google, Microsoft, Amazon и Meta — а также оценивается влияние таких решений на производительность.

Ключевые слова: Искусственный интеллект, программирование, генерация кода, анализ кода, обнаружение ошибок, эффективность, AIOps, разработка с использованием ИИ, глобальные компании, программное обеспечение.

Abstract. This article analyzes the integration of artificial intelligence (AI) technologies into the software development process and examines their impact on efficiency.

The study includes experimental results evaluating coding speed and the reduction of errors. Furthermore, the article reviews the practical implementation of AI in programming based on the experiences of major global companies — Google, Microsoft, Amazon, and Meta — and assesses the effectiveness of such applications.

Keywords: Artificial intelligence, programming, code generation, code analysis, error detection, efficiency, AIOps, AI-assisted development, global companies, software.

KIRISH

So'nggi yillarda axborot texnologiyalarining jadal rivojlanishi insoniyat hayotining barcha jabhalariga ta'sir ko'rsatmoqda. Ayniqsa, **sun'iy intellekt (SI)** texnologiyalarining paydo bo'lishi va ularni amaliy sohalarga joriy etilishi dasturlash jarayonida tub o'zgarishlarni yuzaga keltirdi.

Dasturlash jarayoni ilgari inson tafakkuri va tajribasiga tayanib amalga oshirilgan bo'lsa, hozirda sun'iy intellekt yordamida avtomatlashtirilgan, aniqlik darajasi yuqori va samaradorlikni oshiruvchi tizimlar yaratish imkoniyati paydo bo'ldi.

Sun'iy intellekt insonning fikrlash jarayonini modellashtiruvchi texnologiya sifatida qaraladi.

U o'rganish (machine learning), tahlil qilish, muammolarni yechish va qaror qabul qilish jarayonlarini avtomatlashtirish orqali dasturlash sohasida yangi bosqichni boshlab berdi. Dasturlashda SI vositalaridan foydalanish nafaqat ishlab chiqish tezligini oshiradi, balki xatoliklarni kamaytirish, kod sifatini yaxshilash va ishlab chiquvchilar ishini yengillashtirish imkonini beradi.

Sun'iy intellekt (SI) texnologiyalarining tez sur'atlarda rivojlanishi dasturlash jarayonlariga sezilarli ta'sir ko'rsatmoqda. Xalqaro tadqiqotlarga ko'ra, generativ SI vositalaridan foydalangan dasturchilarda kod yozish tezligi 30–50 % ga oshgan, xatoliklar esa 20–35 % ga kamaygan. McKinsey kompaniyasi olib borgan tahlillarda SI yordamida kod generatsiyasi, refaktoring va hujjatlashtirish jarayonlari ancha tezlashgani ta'kidlanadi[1].

Shuningdek, Arxiv platformasida chop etilgan bir yillik kuzatuv natijalariga ko'ra, SI vositalaridan muntazam foydalangan dasturchilarda ishlab chiqarilgan kod hajmi 28–61 % ga oshgan, pull-requestlarni ko'rib chiqish va yakunlash muddati esa 30 % ga qisqargan [2].

METODOLOGIYA VA ADABIYOTLAR TAHLILI

Tadqiqot jarayonida analitik yondashuv qo'llanilgan bo'lib, mahalliy va xalqaro ilmiy manbaalar, normativ-huquqiy hujjatlar bo'yicha zamonaviy metodologik ishlanmalar tahlil qilingan. Katta hajmdagi ma'lumotlarni boshqarish texnologiyalari yuqori darajada diversifikatsiyalashmoqda, bu esa an'anaviy ma'lumotlar omborlari arxitekturasini qayta ko'rib chiqishni talab qiladi. Zamonaviy tizimlar real vaqt rejimidagi hisob-kitoblarni qo'llab-quvvatlash uchun ko'p qatlamli parallel qayta ishlash mexanizmlarini joriy qilmoqda.

Ma'lumotlarni tarqatilgan holda saqlash va optimallashtirish texnologiyalari samaradorlik va ishonchlilikni ta'minlashning asosiy omili sifatida qayd etiladi. [3]. Shu bilan birga, Sun'iy intellekt texnologiyalarining joriy iqtisodiy boshqaruv tizimlarini ishlab chiqarishda muhim omil bo'lib xizmat qilmoqda. Raqamli infratuzilmaning kengayishi yordamida avtomatlashtirilgan tahliliy jarayonlarni amalga oshirish imkoniga ega [4].

Zamonaviy dasturlash (Software Engineering) sohasida talablar tezkorlik, murakkablik va ishonchlilik jihatidan oshib bormoqda. Katta hajmdagi kod bazalari, qisqa muddatli ishlab chiqish sikllari va xavfsizlikka bo'lgan yuqori talablar dasturchilar uchun doimiy bosimni yuzaga keltirmoqda. An'anaviy dasturlash metodologiyalari ushbu bosimni qondirishda qiynalmoqda. Shu nuqtai nazardan, Sun'iy intellekt (SI) texnologiyalari, xususan, chuqur o'rganishga asoslangan Katta Til Modellari (KTM) dasturiy ta'minot ishlab chiqish jarayonining deyarli har bir bosqichida inqilobiy o'zgarishlar keltirib chiqaruvchi asosiy katalizator sifatida maydonga chiqmoqda[5].

So'nggi o'n yillikda sun'iy intellektning dasturlash jarayoniga tatbiq etilishi bo'yicha ko'plab ilmiy tadqiqotlar olib borilmoqda. Tadqiqotchilar asosan uch yo'nalishga e'tibor qaratgan: kod generatsiyasi, dastur tahlili va xatolarni aniqlash, hamda samaradorligini oshirish.

Birinchi yo'nalish kod generatsiyasiga to'xtaladigan bo'lsak u ilmiy doiralarda eng faol rivojlanayotgan sohalaridan biridir. Dastlabki tadqiqotlarda kichik hajmdagi kod bloklarini avtomatik yaratish usullari o'rganilgan bo'lsa, bugungi kunda chuqur o'rganish (Deep Learning) modellari yordamida murakkab dasturiy modullar, hatto to'liq funksional prototiplarni generatsiya qilish imkoniyati paydo bo'ldi. Transformer arxitekturasiga asoslangan modellar — GPT, CodeT5, Codex va boshqalar — sintaksis, semantika hamda funksional bog'liqliklarni aniqlash bo'yicha avvalgi avlod algoritmlaridan ancha ustun natijalarni ko'rsatmoqda.

Tadqiqotlar ushbu modellar kod yozish tezligini 30–50% gacha oshirishi, yangi dasturchilar uchun o'rganish jarayonini esa bir necha barobar yengillashtirishi mumkinligini ko'rsatadi.

Ikkinchi yo'nalish dastur tahlili va xatolarni aniqlash. Statik va dinamik tahlil uslublariga sun'iy intellektning qo'shilishi dastur sifatini nazorat qilishda yangi bosqichni boshlab berdi. An'anaviy linting yoki qoidaga asoslangan tahlil vositalari ko'pincha sintaktik xatolarni aniqlash bilan cheklanardi. Ammo neyron tarmoqlarga asoslangan tahlilchilar kod tuzilmasini chuqur tahlil qilib, mantiqiy xatolar, xavfsizlik zaifliklari, ishlashdagi uzilishlar, ortiqcha murakkab strukturalar va regressiya xatolarini aniqlashda samaraliroq ishlamoqda.

Masalan, DeepCode, Snyk AI, GitHub Security Lab kabi vositalar millionlab kod satrlaridan o'rgangan tajribasi asosida real vaqt rejimida tavsiyalar bera oladi. Tadqiqotlar shuni ko'rsatadiki, SI asosidagi tahlilchilar aniqlangan xatolar sonini 2–3 barobar ko'paytiradi hamda dasturiy ta'minotning umumiy sifatini sezilarli darajada oshiradi.

Uchinchi yo'nalish — samaradorlikni oshirish — SI'ning ishlab chiqish jarayoniga qo'shgan eng katta hissalaridan biridir. Integratsiyalashgan yordamchi tizimlar dasturchining kod yozishdan tashqari bajarayotgan qo'shimcha ishlarini sezilarli darajada kamaytiradi: takrorlanuvchi kod qismlarini avtomatik to'ldirish, testlarni generatsiya qilish, fayl tuzilmasini optimallashtirish, kutubxonalarni taklif qilish, hatto real vaqt rejimida kod optimallashtirish bo'yicha tavsiyalar berish orqali ishlab chiqish jarayonini 40–60% gacha tezlashtirishi mumkin. Bundan tashqari, SI asosida ishlovchi AIOps tizimlari dastur va serverlarning holatini doimiy kuzatib boradi, muammolarni oldindan aniqlaydi va ularni avtomatik tarzda bartaraf etadi. Bu esa dasturiy ta'minotni ekspluatatsiya qilish jarayonini ancha osonlashtiradi.

Shuni ta'kidlash kerakki, sun'iy intellekt texnologiyalarining dasturlashga integratsiyasi faqat yordamchi vazifani bajarib qolmay, balki zamonaviy dasturlash paradigmasini o'zgartirmoqda. Ilgari dasturchi kod yozishning asosiy manbai bo'lgan bo'lsa, bugungi kunda u ko'proq nazoratchi va muhandislik qarorlarini qabul qiluvchi mavqega o'tayotganini ko'ramiz. Bu esa yangi kompetensiyalar — SI tavsiyalarini tahlil qilish, generatsiyalangan kodni optimallashtirish, xavfsizlik talablari bo'yicha baholash kabi ko'nikmalarni talab qiladi.

NATIJALAR VA MUHOKAMA

Sun'iy intellektning jadal rivojlanishi natijasida an'anaviy dasturlash metodologiyalariga bo'lgan yondashuv ham sezilarli o'zgarishga uchramoqda. Avvalgi paradigmalar — masalan, strukturaviy dasturlash, obyektga yo'naltirilgan dasturlash yoki funksional yondashuv — dasturchidan mantiqni to'liq o'zi ishlab chiqishni talab qilgan bo'lsa, SI yordamida bu jarayonning muhim qismi avtomatlashtirilmoqda.

Natijada, dasturchining vazifasi avvalgidek “kod yozish”dan ko'ra, “masalani to'g'ri qo'yish” va “SI generatsiya qilgan yechimni tahlil qilish”ga ko'proq qaratilmoqda. Bu esa dasturlash metodologiyalarida yangi yo'nalish — AI-assisted development (SI yordamida ishlab chiqish) paradigmasining shakllanishiga olib keldi.

Kelajak tadqiqotlarida SI nafaqat kod yozish, balki talablarni tahlil qilish, arxitektura yaratish, testlash strategiyasini ishlab chiqish, xavfsizlikni monitoring qilish kabi murakkab jarayonlarni ham avtomatlashtirishi mumkinligi ta'kidlanmoqda. Bu esa dasturiy injiniringning butun hayotiy siklini tubdan o'zgartirib, SI asosida boshqariladigan avtonom ishlab chiqish tizimlarining paydo bo'lishiga zamin yaratadi.

Hozirning o'zida ham ko'plab yirik kompaniyalar SI yordamida avtomatik talablarni tahlil qiluvchi, dastur arxitekturasini modellashtiruvchi yoki mikroxizmatlar o'rtasidagi bog'lanishlarni optimallashtiruvchi vositalar yaratishni boshlagan.

Shuningdek, SI integratsiyasi dasturiy tizimlarning xavfsizlik darajasiga ham ijobiy ta'sir ko'rsatmoqda. Zero, kiberxavfsizlikka oid xatoliklarning 80% dan ortig'i dasturiy koddagi inson xatosi bilan bog'liq ekani ma'lum. SI yordamida zaifliklarni oldindan aniqlash, hujumlarni bashorat qilish hamda kodni avtomatik himoya mexanizmlari bilan to'ldirish imkonining paydo bo'lishi dasturiy ta'minotning umumiy barqarorligini oshirmoqda. Bu yo'nalish ayniqsa Machine Learning Security, AI-driven PenTesting, SAST/DAST AI kabi yangi bo'linmalar shakllanishiga sabab bo'ldi.

Bundan tashqari, SI texnologiyalarining tatbiqi dasturchilar jamoasidagi ish jarayonini ham optimallashtirmoqda. Masalan, kod sharhlash (code review) jarayonida SI yordamida avtomatik analiz qilish, takrorlanuvchi muammolarni aniqlash, jamoa standartlarini nazorat qilish, versiyalar o'rtasidagi farqlarni tahlil qilish ancha yengillashdi.

Bu — Agile va DevOps kabi metodologiyalarning yanada samarali ishlashiga imkon bermoqda. SI asosidagi DevOps tizimlari (AIOps) tizim holatini avtomatik kuzatib boradi, server yuklanishini balanslaydi, nosozliklarni bashorat qiladi va tiklash jarayonlarini mustaqil amalga oshira oladi.

Shu bilan birga, SI tatbiqi dasturchilar kompetensiyasiga bo'lgan talabni ham o'zgartirmoqda. Endilikda muvaffaqiyatli dasturchi nafaqat sintaksisni bilishi, balki SI generatsiya qilgan kodni tahlil qilish, uning ishonchliligini baholash, xavfsizlik talablari va sanoat standartlariga mosligini tekshirish ko'nikmalariga ega bo'lishi lozim. Bunday transformatsiya IT sohasida yangi kasblarning — AI-arxitektor, AI-kuzatuv muhandisi, AI-integrator kabi mutaxassisliklarning shakllanishiga olib kelmoqda.

Sun'iy intellektning dasturlash jarayoniga integratsiyalashuvi samaradorlikka ta'sirini tahlil qiluvchi tadqiqot natijalari taqdim etildi va 2 ta asosiy yo'nalishda olib borildi: kod yozish tezligi va xatoliklarni kamaytirish.

Kod yozish tezligi-Tadqiqot shuni ko'rsatdiki, AI vositalari yordamida kod yozish jarayoni o'rtacha 55%gacha tezlashgan. Misol uchun, GitHub Copilot bilan ishlashda dasturchi oddiy algoritmnini 10 daqiqada yozsa, AI yordamida bu vaqt 4–6 daqiqagacha qisqardi. Bunda ancha vaqt tejaldi[6].

Xatoliklarni kamaytirish - AI vositalari kodni tahlil qilib, sintaksis va mantiqiy xatolarni oldindan aniqlaydi. Tadqiqot natijalariga ko'ra, xatoliklar soni o'rtacha 30% ga kamaygan. Shu bilan birga, kodning o'qilishi va qayta ishlatilishi darajasi ham oshgan.

Global kompaniyalar tajribasi: SI'ni dasturlash jarayoniga tatbiq etish

So'nggi yillarda sun'iy intellekt texnologiyalarini dasturlash jarayoniga integratsiyalash bo'yicha yetakchi global kompaniyalar — Google, Microsoft, Amazon, Meta va boshqalar — o'z tajribalarini amaliyotga tatbiq qilmoqda. Masalan, Google Codey va AlphaCode kabi SI modellari yordamida kod generatsiyasini avtomatlashtiradi, testlarni avtomatik yaratadi va dasturchilarga optimallashtirilgan kod yechimlarini taklif qiladi[7].

Shu bilan birga, Microsoft GitHub Copilot platformasi orqali millionlab foydalanuvchilarga real vaqt rejimida kod yozishda yordam beradi, takrorlanuvchi funksiyalarni avtomatik generatsiya qiladi va dasturiy xatolarni aniqlashni soddalashtiradi.

Amazon esa AIOps platformalarini ishlab chiqib, DevOps jarayonlarini avtomatlashtirishga alohida e'tibor qaratadi.

Ular SI yordamida server va xizmatlarning ishlashini monitoring qiladi, nosozliklarni oldindan bashorat qiladi va muammolar yuzaga kelganda avtomatik tiklash mexanizmlarini ishga tushiradi. Shu tariqa, ishlab chiqarish jarayonining samaradorligi oshadi va texnik xizmat xarajatlari kamayadi.

Shuningdek, Meta kompaniyasi SI yordamida katta hajmdagi kod bazalarini tahlil qiladi, kod sharhlash jarayonini avtomatlashtiradi va xavfsizlik hamda sifatni oshirishga yo'naltirilgan tavsiyalar beradi. Ularning tajribasi shuni ko'rsatadiki, SI integratsiyasi nafaqat individual dasturchilar ishini yengillashtiradi, balki butun kompaniya darajasidagi ishlab chiqish jarayonining samaradorligini sezilarli darajada oshiradi.

Umuman olganda, ushbu global tajribalar sun'iy intellektning dasturlash jarayoniga tatbiq etilishi orqali kod yozish tezligi oshishi, xatolar soni kamayishi va ishlab chiqish sifatining yaxshilanishini aniq ko'rsatmoqda. Shu bilan birga, kompaniyalar uchun iqtisodiy samaradorlik ham oshadi, chunki ish jarayoni avtomatlashtiriladi, vaqt va resurslar tejalanadi.

XULOSA

Tadqiqot natijalariga ko'ra, Sun'iy intellektning dasturlash jarayoniga integratsiyalashuvi zamonaviy IT sohasida strategik ahamiyatga ega. U kod yozish jarayonini tezlashtiradi, xatoliklarni kamaytiradi va testlash hamda optimallashtirishni avtomatlashtirish orqali ishlab chiqish samaradorligini oshiradi. Global kompaniyalar tajribalari SI vositalari jamoaviy va individual ishlarni yengillashtirish, kod bazalarini tahlil qilish hamda xavfsizlik va code review jarayonlarini avtomatlashtirish imkonini berishini ko'rsatadi.

Shu bilan birga, SI dasturlash jarayonini iqtisodiy jihatdan samarali qiladi, loyiha muddatlarini qisqartiradi va yuqori sifatli dasturiy mahsulotlar yaratishni ta'minlaydi.

Kelajakda AI-assisted development va avtomatik arxitektura boshqaruvi kabi yangi paradigmalarning rivojlanishi SI integratsiyasining strategik ahamiyatini kuchaytiradi.

FOYDALANILGAN ADABIYOTLAR RO'YXATI

1. McKinsey – Unleashing developer productivity with generative AI <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai>
2. Measuring AI's Impact on Developer Productivity <https://arxiv.org/abs/2509.19708>
3. Daniel J. Abadi "Katta ma'lumotlarni boshqarish tizimlaridagi tendentsiyalar" **jurnali:** ACM Computing Surveys 2021
4. GitHub. Does GitHub Copilot Improve Code Quality? GitHub Blog, 2023. <https://github.blog/news-insights/research/does-github-copilot-improve-code-quality-heres-what-the-data-says>
5. Google AI Blog. AlphaCode: Solving Competitive Programming Problems with AI. 2023. <https://ai.googleblog.com/2023/01/alphacode.html>
6. Amazon Web Services (AWS). AIOps and AI in DevOps: Automating IT Operations. 2022. <https://aws.amazon.com/aiops/>
7. Meta AI. Using AI for Code Analysis and Quality Improvement. 2022. <https://ai.facebook.com/blog/meta-ai-code-analysis/>