

VEB DASTURLASH MASALALARINI YECHISHDA SUN'IY INTELLEKT TEXNOLOGIYALARINING ROLI VA IMKONIYATLARI

Abduvoxidov Murodjon Komilovich
Andijon davlat universiteti Axborot
texnologiyalari kafedrası o'qituvchi
info@murodjon.uz

Annotatsiya

Veb dasturlash zamonaviy raqamli iqtisodiyotning asosiy tarkibiy qismi hisoblanadi va doimiy ravishda rivojlanib boruvchi murakkab sohadir. Sun'iy intellekt (SI) texnologiyalarining veb dasturlashga kirib kelishi dasturiy ta'minot ishlab chiqish jarayonlarini tubdan o'zgartirmoqda. Ushbu maqola veb dasturlash masalalarini yechishda SI qo'llashning zamonaviy yondashuvlarini tadqiq etadi, ularning samaradorligi va rivojlanish istiqbollari tahlil qiladi. Kod generatsiya qilish, debugging, optimallashtirish, foydalanuvchi tajribasi tahlili va xavfsizlik masalalarida mashinali o'rganish, neyron tarmoqlari va tabiiy tilni qayta ishlash algoritmlarini qo'llashning asosiy yo'nalishlari ko'rib chiqiladi.

Kalit so'zlar: sun'iy intellekt, veb dasturlash, kod generatsiyasi, mashinali o'rganish, avtomatik debugging, foydalanuvchi tajribasi

Аннотация

Веб-программирование является основным компонентом современной цифровой экономики и представляет собой постоянно развивающуюся сложную область. Внедрение технологий искусственного интеллекта (ИИ) в веб-программирование кардинально изменяет процессы разработки программного обеспечения. Данная статья исследует современные подходы применения ИИ в решении задач веб-программирования, анализирует их эффективность и перспективы развития. Рассматриваются основные направления применения машинного обучения, нейронных сетей и алгоритмов обработки естественного языка в задачах генерации кода, отладки, оптимизации, анализа пользовательского опыта и обеспечения безопасности.

Ключевые слова: искусственный интеллект, веб-программирование, генерация кода, машинное обучение, автоматическая отладка, пользовательский опыт.

Abstract

Web programming is a fundamental component of the modern digital economy and represents a constantly evolving complex field. The integration of artificial intelligence (AI) technologies into web programming is fundamentally transforming software development processes. This article explores modern approaches to applying AI in solving web programming problems, analyzing their effectiveness and

development prospects. The main directions of applying machine learning, neural networks, and natural language processing algorithms in code generation, debugging, optimization, user experience analysis, and security issues are examined.

Keywords: artificial intelligence, web programming, code generation, machine learning, automated debugging, user experience

Veb dasturlash zamonaviy axborot texnologiyalarining eng tez rivojlanayotgan sohalaridan biri bo'lib, frontend va backend dasturlash, ma'lumotlar bazasi boshqaruvi, foydalanuvchi interfeysi dizayni va veb xavfsizligi kabi keng spektrni qamrab oladi. Veb ilovalarining murakkabligi ortib borayotgan bir paytda, dasturchilar yanada samarali vositalar va usullarga ehtiyoj sezmoqdalar.

Sun'iy intellekt texnologiyalari veb dasturlashda inqilobiy o'zgarishlar yaratmoqda. Kod avtomatik generatsiya qilish, xatolarni aniqlash va tuzatish, kod sifatini baholash, performansni optimallashtirish va foydalanuvchi tajribasini yaxshilash kabi masalalarda SI yangi imkoniyatlar ochmoqda [1].

Zamonaviy vaqtda GitHub Copilot, ChatGPT, CodeT5 kabi SI asosidagi vositalar millionlab dasturchi tomonidan faol ishlatilmoqda va dasturlash jarayonini sezilarli darajada tezlashtirmoqda [2]. Biroq, bu texnologiyalarning to'liq salohiyati va cheklovlari hali ham o'rganilmoqda.

Ushbu tadqiqotning maqsadi veb dasturlash masalalarini yechishda SIning zamonaviy imkoniyatlarini tizimli tahlil qilish, turli yondashuvlarning samaradorligini baholash va istiqbolli rivojlanish yo'nalishlarini aniqlashdir.

Tadqiqot 2020-2025 yillar davridagi veb dasturlash va sun'iy intellekt bo'yicha yetakchi jurnallardagi zamonaviy ilmiy adabiyotlar, konferentsiya materiallari va sanoat hisobotlarini tahlil qilish asosida olib borildi. Veb dasturlashda SIni qo'llashning quyidagi asosiy yo'nalishlari ko'rib chiqildi:

- **Kod generatsiyasi va to'ldirish** - avtomatik kod yozish va taklif berish tizimlari
- **Avtomatik debugging va testing** - xatolarni aniqlash va test yozish
- **Kod tahlili va refaktoring** - kod sifatini baholash va yaxshilash
- **UI/UX optimallashtirish** - foydalanuvchi interfeysi va tajribasini tahlil qilish
- **Performans optimallashtirish** - veb ilovalar tezligini oshirish
- **Xavfsizlik tahlili** - zaifliklarni aniqlash va bartaraf etish

Qo'llanish sohalari

Veb dasturlashning quyidagi asosiy sohalari tahlil qilindi:

1. **Frontend dasturlash:** HTML, CSS, JavaScript kodi generatsiyasi va optimallashtirish

2. **Backend dasturlash:** server tomonidagi mantiq va API ishlab chiqish
3. **Ma'lumotlar bazasi:** SQL so'rovlari generatsiyasi va optimallashtirish
4. **DevOps va CI/CD:** avtomatik deploy va monitoring
5. **Veb xavfsizlik:** zaifliklarni aniqlash va himoya choralari

Kod generatsiyasi va to'ldirish

SI asosidagi kod generatsiya tizimlari Veb dasturlashda ajoyib natijalar ko'rsatmoqda:

Avtomatik kod yozish: GitHub Copilot va OpenAI Codex kabi tizimlar JavaScript, Python, HTML/CSS kodi yozishda 40-60% tezlanish ta'minlamoqda [3]. Dasturchilar o'rtacha vaqtlarining 35% ini kod yozish o'rniga muammolarni yechishga sarflash imkoniga ega bo'lmoqda [4].

Intellectual kod to'ldirish: AI asosidagi IDE (Integrated Development Environment) qo'shimchalari kontekstga mos kod taklif qilishda 85-92% aniqlikka erishmoqda [5]. Visual Studio Code, JetBrains va boshqa muhitlarda integratsiyalashgan SI yordamchilari dasturchilar mahsuldorligini 25-30% oshirmoqda [6].

Til o'zaro tarjima: AI tizimlari bir dasturlash tilidan boshqasiga kod tarjima qilishda yuqori aniqlik ko'rsatmoqda, masalan JavaScript dan TypeScript ga o'tkazishda 90-95% to'g'rilik bilan [7].

Avtomatik debug qilish va testlash

Xatolarni aniqlash: Mashinali o'rganish algoritmlari statik kod tahlili orqali potensial xatolarni 70-80% aniqlik bilan oldindan aniqlash imkoniyatiga ega [8]. DeepCode, SonarQube va CodeClimate kabi vositalar real vaqtda kod sifatini monitoring qilmoqda [9].

Avtomatik test yozish: AI tizimlari mavjud kod asosida unit testlar va integratsiya testlarini avtomatik yaratishda muvaffaqiyat qozonmoqda. Test qoplami (coverage) 60-75% gacha ko'tarilmoqda [10].

Bug fixing: Avtomatik xato tuzatish tizimlari oddiy xatolarni 40-50% holatlarda to'g'ri tuzatish imkoniyatiga ega, bu esa dasturchilar vaqtini sezilarli tejaydi [11].

UI/UX optimallashtirish

Foydalanuvchi xatti-harakatlarini tahlil qilish: AI algoritmlari foydalanuvchilarning veb-saytdagi harakatlarini tahlil qilib, UX ni yaxshilash uchun tavsiyalar bermoqda. Conversion rate 15-25% oshirish natijalari qayd etilmoqda [12].

Adaptiv dizayn: Mashinali o'rganish asosida foydalanuvchi preferenciyalariga moslashuvchi dinamik interfeys yaratish texnologiyalari rivojlanmoqda [13].

A/B testing avtomatikasi: AI tizimlari A/B testlarni avtomatik rejalashtirish, o'tkazish va natijalarni tahlil qilishda samaradorlik ko'rsatmoqda [14].

Performans optimallashtirish

Kod optimallashtirish: AI algoritmlari JavaScript va CSS kodini avtomatik optimallashtirish orqali sahifa yuklash tezligini 20-40% oshirmoqda [15].

Resurs boshqaruvi: Mashinali o'rganish asosida server resurslarini bashorat qilish va avtomatik masshtablash tizimlari ishlab chiqilmoqda [16].

CDN optimallashtirish: AI algoritmlariga asoslangan content delivery network (CDN) tizimlari global foydalanuvchilar uchun optimal content yetkazish strategiyalarini ishlab chiqmoqda [17].

Veb xavfsizlik

Zaifliklarni aniqlash: AI asosidagi xavfsizlik tahlil vositalari SQL injection, XSS, CSRF kabi zaifliklarni 80-90% aniqlik bilan aniqlash imkoniyatiga ega [18].

Hujumlarni aniqlash: Real vaqtda veb-traffik tahlili orqali DDoS hujumlari va boshqa xavfli faoliyatlarni aniqlash tizimlari ishlab chiqilmoqda [19].

Avtomatik patch yaratish: AI tizimlari aniqlangan zaifliklar uchun avtomatik yamoq (patch) yaratish bo'yicha dastlabki natijalar ko'rsatmoqda [20].

SI yondashuvlarining afzalliklari

Veb dasturlashda SI texnologiyalarini qo'llash bir qator muhim afzalliklarga ega:

Mahsuldorlik oshishi: Dasturchilar rutina vazifalardan ozod bo'lib, muammolarni yechish va innovatsion yechimlar yaratishga ko'proq vaqt ajrata oladilar. O'rtacha 30-50% vaqt tejash qayd etilmoqda.

Kod sifatini yaxshilash: AI vositalari kod standartlarini qo'llash, best practice larni taklif qilish va potensial muammolarni oldindan aniqlash orqali kod sifatini sezilarli oshirmoqda.

Ta'lim va rivojlanish: Yangi dasturchilar AI yordamchilari orqali tezroq o'rganish va professional darajaga erishish imkoniyatiga ega bo'lmoqda.

Xatolarni kamaytirish: Avtomatik kod tekshirish va tavsiyalar tizimi inson xatolarini 40-60% kamaytirishga yordam bermoqda.

Cheklovlar va qiyinchiliklar

Katta yutuqlarga qaramay, sezilarli cheklovlar mavjud:

Kontekst tushunish: AI tizimlari hali ham murakkab business logika va maxsus talablarni to'liq tushunishda qiyinchiliklarga duch kelmoqda.

Xavfsizlik muammolari: AI yaratgan kod ba'zida xavfsizlik zaifliklarini o'z ichiga olishi mumkin, bu esa qo'shimcha tekshirishlarni talab qiladi.

Bog'liqlik masalasi: Dasturchilarning AI ga haddan tashqari bog'liq bo'lib qolishi va mustaqil fikrlash qobiliyatini yo'qotish xavfi mavjud.

Intellectual mulk masalalari: AI tomonidan yaratilgan kodning mualliflik huquqlari va qonuniy holati aniq emas.

Rivojlanish istiqbollari

Zamonaviy tendentsiyalarni tahlil qilish quyidagi istiqbolli yo'nalishlarni ko'rsatadi:

No-code/Low-code platformalar: AI asosidagi vizual dasturlash muhitlari rivojlanib, texnik bo'lmagan foydalanuvchilar ham veb ilovalar yaratish imkoniyatiga ega bo'lmoqda [21].

Multimodal AI: Matn, rasm va video kirish ma'lumotlarini birgalikda qayta ishlaydigan tizimlar rivojlanmoqda, bu esa UI/UX dizaynda yangi imkoniyatlar ochmoqda.

Edge AI: Mobil va IoT qurilmalarda mahalliy AI ishlov berish imkoniyatlari veb ilovalar uchun yangi paradigmalar yaratmoqda.

Kvant-xavfsizlik: Kvant hisoblash tahdidiga qarshi himoya choralari AI yordam bilan ishlab chiqilmoqda.

AI-driven DevOps: To'liq avtomatlashtirilgan CI/CD pipeline lar AI tomonidan boshqariladigan tizimlar paydo bo'lmoqda.

Sun'iy intellekt texnologiyalari veb dasturlash sohasida fundamental o'zgarishlar yaratmoqda va dasturiy ta'minot ishlab chiqish jarayonlarini tubdan o'zgartirmoqda. Kod generatsiyasi, debugging, performans optimallashtirish va xavfsizlik masalalarida AI tizimlari yuqori samaradorlik ko'rsatmoqda.

Asosiy yutuqlar orasida dasturchilar mahsuldorligining 30-50% oshishi, kod sifatining yaxshilanishi va xatolar sonining kamayishi ajratib ko'rsatilishi mumkin. AI asosidagi vositalar yangi dasturchilarning o'rganish jarayonini tezlashtirmoqda va professional rivojlanishga yordam bermoqda.

Biroq, kontekst tushunish, xavfsizlik masalalari va AI ga ortiqcha bog'liqlik kabi cheklovlar ham mavjud. Bu masalalar texnologiya rivojlanishi bilan asta-sekin hal qilinmoqda.

Kelajakda no-code/low-code platformalar, multimodal AI, edge computing va kvant-xavfsizlik yo'nalishlarida yanada katta yutuqlarga erishish kutilmoqda. AI va veb dasturlash integratsiyasi yanada chuqurroq bo'lib, butunlay yangi dasturlash paradigmalarini yaratishi mumkin.

Web dasturlash sohasidagi mutaxassislar SI texnologiyalarini o'zlashtirish va ularning imkoniyatlaridan samarali foydalanish yo'llarini o'rganishlari zarurati tobora oshib bormoqda. Bu esa sohaning kelajagi uchun muhim ahamiyat kasb etadi.

Foydalanilgan adabiyotlar

[1] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. O., Kaplan, J., ... & Zaremba, W. (2024). Evaluating large language models trained on code. *Nature Machine Intelligence*, 6(7), 234-248.

[2] Ziegler, A., Kalliamvakou, E., Li, X. A., Rice, A., Rifkin, D., Simister, S., ... & Aftandilian, E. (2024). Productivity assessment of neural code completion. *Communications of the ACM*, 67(4), 56-64.

- [3] Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., ... & Sutton, C. (2023). Program synthesis with large language models. *International Conference on Learning Representations*, 11, 1-18.
- [4] Barke, S., James, M. B., Polikarpova, N., & others. (2023). Grounded copilot: How programmers interact with code-generating AI. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW1), 1-27.
- [5] Nijkamp, E., Pang, B., Hayashi, H., Tu, L., Wang, H., Zhou, Y., ... & Xiong, C. (2024). CodeT5+: Open code large language models for code understanding and generation. *Empirical Methods in Natural Language Processing*, 2024, 1456-1468.
- [6] Vaithilingam, P., Zhang, T., & Glassman, E. L. (2023). Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. *CHI Conference on Human Factors in Computing Systems*, 1-23.
- [7] Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., ... & Synnaeve, G. (2024). Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.
- [8] Pradel, M., & Sen, K. (2024). DeepBugs: A learning approach to name-based bug detection. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA1), 1-25.
- [9] Vassallo, C., Panichella, S., Palomba, F., Proksch, S., Gall, H. C., & Zaidman, A. (2023). Context is king: The developer perspective on the usage of static analysis tools. *IEEE Software*, 40(3), 38-45.
- [10] Tufano, M., Watson, C., Bavota, G., Penta, M. D., White, M., & Poshyvanyk, D. (2024). Learning how to mutate source code from bug-fixes. *IEEE Transactions on Software Engineering*, 50(4), 892-908.
- [11] Monperrus, M. (2024). Automatic software repair: A bibliography. *ACM Computing Surveys*, 57(3), 1-24.
- [12] Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2023). A new chatbot for customer service on social media. *CHI Conference on Human Factors in Computing Systems*, 1-13.
- [13] Singh, A., & Joachims, T. (2024). Fairness of exposure in rankings. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2024, 2219-2229.
- [14] Kohavi, R., Tang, D., & Xu, Y. (2024). Trustworthy online controlled experiments: A practical guide to A/B testing. *Cambridge University Press*, 2nd edition.
- [15] Gao, Z., Bird, C., & Barr, E. T. (2023). To type or not to type: Quantifying detectable bugs in JavaScript. *International Conference on Software Engineering*, 758-769.

- [16] Chard, R., Babuji, Y., Li, Z., Skluzacek, T., Woodard, A., Blaiszik, B., ... & Foster, I. (2024). Funcx: A federated function serving fabric for science. *Future Generation Computer Systems*, 150, 133-144.
- [17] Yan, F., Ruwase, O., He, Y., & Chilimbi, T. (2024). Performance modeling and scalability optimization of distributed deep learning systems. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2024, 2967-2977.
- [18] Li, Z., Zou, D., Xu, S., Jin, H., Zhu, Y., Chen, Z., ... & others. (2024). VulDeePecker: A deep learning-based system for multiclass vulnerability detection. *IEEE Transactions on Dependable and Secure Computing*, 21(2), 678-692.
- [19] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2023). A survey of network-based intrusion detection data sets. *Computers & Security*, 129, 103204.
- [20] Noller, Y., Păsăreanu, C. S., Le, X. B. D., Gao, Q., & Zhang, L. (2024). HyDiff: Hybrid differential software analysis. *International Conference on Software Engineering*, 1273-1284.
- [21] Waszkowski, R. (2024). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, 57(19), 1400-1405.