DASTURIY TA’MINOTNI TESTLASHDA SUN’IY INTELLEKTLI YONDASHUVLARNING TAHLILI

Matkarimov Bobirjon Erkinboyevich

Urganch Ranch texnologiya universiteti, “Kompyuter injineringi” mutaxassisligi 2-bosqich magistranti
E-mail: matkarimovboburbek@gmail.com

K E Y W O R D S

dasturiy ta’minotni testlash, sun’iy intellekt, mashinaviy o‘rganish, mustahkamlangan o‘rganish, test holatlarini generatsiya qilish, nuqsonlarni bashorat qilish, avtomatlashtirilgan testlash, katta til modellari, dasturiy ta’minot sifati

A B S T R A C T

Maqolada dasturiy ta’minotni testlashda sun’iy intellekt (SI) yondashuvlarining nazariy va amaliy jihatlari tahlil qilinadi. Tadqiqot doirasida mashinaviy o‘rganish (ML), mustahkamlangan o‘rganish (RL), generativ modellar va katta til modellari (LLM) asosidagi metodlar qiyosiy tarzda o‘rganilgan. Tadqiqot natijalari sun’iy intellekt asosidagi yondashuvlar test holatlarini avtomatik generatsiya qilish, nuqsonlarni bashorat qilish va testlash strategiyalarini optimallashtirish samaradorligini an’anaviy usullarga nisbatan o‘rtacha 25–40% ga oshirishi mumkinligini ko‘rsatadi. Shu bilan birga, ma’lumotlarning sifati va hajmiga bog‘liqlik, hisoblash resurslariga yuqori talab hamda natijalarni interpretatsiya qilishdagi murakkabliklar kabi cheklovlar aniqlangan. Maqolada tahlil natijalari asosida turli SI yondashuvlarini integratsiyalash zarurati asoslab berilgan va kelajakdagi tadqiqot yo‘nalishlari taklif etilgan.

KIRISH

Zamonaviy bosqichda dasturiy ta’minot tizimlarining murakkabligi va miqyosi tobora ortib bormoqda. Buning natijasida ushbu tizimlarning sifatini ta’minlash, ishonchliligini kafolatlash va xavfsizligini saqlash bo‘yicha vazifalar dasturiy injeneriyaning eng dolzarb muammolaridan biriga aylanmoqda. Statistikaga ko‘ra, dasturiy ta’minotni ishlab chiqish loyihalarining umumiy byudjetining 30–50 foizigacha bo‘lgan qismi testlash jarayoniga sarflanadi, biroq an’anaviy yondashuvlar har doim ham yetarli darajada samaradorlikni ta’minlamaydi.

An’anaviy testlash usullari, asosan, qo‘lda ishlab chiqilgan test ssenariylariga asoslanadi. Bunday yondashuvlar dasturiy mahsulotning barcha funksional holatlarini va o‘zaro bog‘liqliklarini to‘liq qamrab olish imkonini bermaydi. Bundan tashqari, zamonaviy chevra (Agile) va doimiy integratsiya (CI/CD) metodologiyalari sharoitida dasturiy mahsulotlarning tez-tez yangilanib turishi testlash jarayonining samaradorligini sezilarli darajada pasaytiradi hamda qo‘shimcha vaqt va inson resurslarini talab qiladi.

So‘nggi yillarda testlash jarayonini avtomatlashtirish va optimallashtirish maqsadida sun’iy intellekt (SI) texnologiyalaridan keng foydalanilmoqda. Xususan, mashinaviy o‘rganish (Machine Learning, ML) usullari dasturiy komponentlardagi xatoliklarni bashorat qilish va testlash resurslarini samarali taqsimlashga yordam beradi. Mustahkamlangan o‘rganish (Reinforcement Learning, RL) yondashuvlari testlash jarayonini dinamik boshqarish va optimal strategiyalarni shakllantirishda qo‘llanilmoqda. Generativ sun’iy intellekt va katta til modellari (Large Language Models, LLM) esa test holatlarini avtomatik generatsiya qilishda yangi imkoniyatlarni ochib bermoqda.

Sun’iy intellekt asosidagi yondashuvlar testlash jarayonini sezilarli darajada takomillashtirayotgan bo‘lsa-da, ular bir qator muammolarga ham ega. Jumladan, SI modellarining ma’lumotlarga yuqori darajada bog‘liqligi, hisoblash resurslariga bo‘lgan yuqori talab va modellarning ichki ishlash jarayonini interpretatsiya qilishdagi murakkabliklar (“qora quti” muammosi) bu yondashuvlarning amaliy qo‘llanilishini cheklaydi.

© Matkarimov Bobirjon Erkinboyevich

MAVZUGA DOIR ADABIYOTLAR TAHLILI

Dasturiy ta’minotni testlash sohasida olib borilgan ilmiy tadqiqotlar an’anaviy yondashuvlardan tortib zamonaviy sun’iy intellekt texnologiyalarigacha bo‘lgan keng doiradagi usullarni qamrab oladi. Klassik testlash metodlari (qora quti, oq quti, kulrang quti testlashi) dasturiy mahsulotning funksional to‘g‘riligini tekshirishga qaratilgan bo‘lib, ular dastlabki bosqichlarda samarali hisoblanadi. Biroq, dasturiy tizimlarning murakkablashuvi ushbu yondashuvlarning samaradorligini cheklab qo‘ydi.

Modelga asoslangan testlash (Model-Based Testing) va qidiruvga asoslangan testlash (Search-Based Software Testing, SBST) yondashuvlari klassik usullarning kamchiliklarini bartaraf etishga qaratilgan. T.Y. Chen va hammualliflari tomonidan test holatlarini generatsiya qilish texnikalari tizimli ravishda o‘rganilgan [1]. M. Harman va uning hammualliflari qidiruvga asoslangan testlash yondashuvlari dasturiy testlash jarayonini optimallashtirishda muhim rol o‘ynashini ko‘rsatib bergan [2]. A. Panichella tomonidan ko‘p maqsadli evolyutsion algoritmlar asosida avtomatik test generatsiyasi usullari taklif etilgan bo‘lib, ular test qamrovini sezilarli darajada oshirishga xizmat qiladi [3].

Sun’iy intellekt texnologiyalarining rivojlanishi bilan dasturiy ta’minotni testlash jarayoniga yangi yondashuvlar kirib keldi. Mashinaviy o‘rganish asosidagi metodlar dasturiy komponentlarda nuqsonlarni bashorat qilish va testlash resurslarini samarali taqsimlash imkonini beradi. N.S.A. Abu Bakar tomonidan mashinaviy o‘rganishning avtomatlashtirilgan testlashdagi qo‘llanilishi tizimli ravishda o‘rganilgan [4]. M.J. Kadhim tadqiqotlarida ushbu yondashuvlar nafaqat testlash samaradorligini oshirishi, balki testlash xarajatlarini sezilarli darajada kamaytirishi mumkinligi ko‘rsatib berilgan [5].

So‘nggi yillarda generativ sun’iy intellekt va katta til modellari yordamida test holatlarini avtomatik generatsiya qilish bo‘yicha tadqiqotlar jadal rivojlanmoqda. M. Andrzejewski tomonidan katta til modellari (GPT, BERT va shu kabi) asosida test generatsiyasi usullari taklif etilgan bo‘lib, ular murakkab tizimlar uchun samarali yechim sifatida qaralmoqda [6]. W. Li va G. Kathiresan tadqiqotlarida sun’iy intellekt yordamida test holatlarini generatsiya qilishning turli usullari (transformer arxitekturasi, kontekstual o‘rganish va boshqalar) tahlil qilingan [7].

Mustahkamlangan o‘rganish asosidagi yondashuvlar testlash jarayonida agentning tizim bilan o‘zaro ta’sirini optimallashtirish imkonini beradi. R.S. Sutton va A.G. Barto tomonidan ushbu yondashuvning nazariy asoslari ishlab chiqilgan [8]. A. Abo-Eleneen va hammualliflari tadqiqotlarida mustahkamlangan o‘rganishning dasturiy testlashdagi turli ssenariylari ko‘rib chiqilgan, jumladan, GUI testlashi, fuzzing va regression testlash [9].

Sun’iy intellekt asosidagi testlash tizimlarini umumiy tahlil qilishga qaratilgan ishlar ham mavjud. A. Trudova tomonidan olib borilgan tizimli adabiyotlar sharhida sun’iy intellektning testlashdagi roli va rivojlanish tendensiyalari tahlil qilingan [10]. A. Faraji va N. Pombo tomonidan sun’iy intellekt asosidagi avtomatlashtirilgan testlash tizimlari samaradorligi sanoat sharoitida baholangan [11]. R. Wang va hammualliflari tadqiqotlarida chuqur o‘rganish (deep learning) asosidagi nuqsonlarni aniqlash modellari empirik tarzda baholangan [12].

Mahalliy tadqiqotlardan O‘zbekistonda dasturiy injeneriyaning rivojlanishi va sun’iy intellekt texnologiyalarini joriy etish bo‘yicha bir qator ilmiy ishlar olib borilmoqda. Biroq, dasturiy ta’minotni testlashda SI yondashuvlarini integratsiyalashgan holda qo‘llash bo‘yicha tadqiqotlar nisbatan kam, bu esa ushbu yo‘nalishda qo‘shimcha tadqiqotlar olib borish zaruratini tug‘diradi.

Yuqorida keltirilgan tadqiqotlar shuni ko‘rsatadiki, dasturiy ta’minotni testlashda sun’iy intellekt yondashuvlari muhim natijalar berayotgan bo‘lsa-da, mavjud ishlarda yondashuvlar ko‘pincha alohida yo‘nalishlar doirasida ko‘rib chiqilgan. Bu esa testlash jarayonini kompleks va integratsiyalashgan tarzda tashkil etish masalasi hali to‘liq hal etilmaganligini ko‘rsatadi.

TADQIQOT METODOLOGIYASI

(cc) BY © Matkarimov Bobirjon Erkinboyevich

Tadqiqot tizimli adabiyotlar sharhi (Systematic Literature Review, SLR) va qiyosiy tahlil metodlari asosida amalga oshirildi. Tahlil uchun 2018–2025 yillar oralig‘ida IEEE Xplore, ACM Digital Library, Springer, Elsevier (ScienceDirect) ma’lumotlar bazalarida nashr etilgan ilmiy maqolalar tanlab olindi. Saralash mezonlari sifatida ishlarning sun’iy intellekt va dasturiy ta’minotni testlash sohasiga to‘g‘ridan-to‘g‘ri aloqadorligi, iqtibos olish darajasi va metodologik aniqligi hisobga olindi.

Yondashuvlarni qiyosiy baholash uchun quyidagi mezonlar tanlangan: (a) testlash samaradorligini oshirishga ta’siri; (b) ma’lumotlarga bog‘liqligi; (c) hisoblash resurslariga bo‘lgan talab; (d) interpretatsiya qilish murakkabligi; (e) amaliy qo‘llash imkoniyati. Ushbu mezonlar adabiyotlar tahlili asosida shakllantirilgan va ekspert baholash usuli bilan tasdiqlangan. Ma’lumotlarni umumlashtirish jarayonida sifat ko‘rsatkichlari (1–5 balli shkala) bilan qiyosiy tahlil amalga oshirilgan, natijalar jadval va sxemalar orqali tasvirlangan.

NATIJALAR VA ULARNING MUHOKAMASI

1. Mashinaviy o‘rganish asosidagi yondashuvlar. Mashinaviy o‘rganish dasturiy komponentlarda nuqsonlarni aniqlash va bashorat qilishda keng qo‘llanilmoqda. Ushbu yondashuvlarda dasturiy koddan olingan xususiyatlar (kod metrikalari, tarixiy nuqsonlar, o‘zgartirishlar tarixi va boshqalar) asosida model o‘qitiladi va xatolik ehtimoli yuqori bo‘lgan komponentlar aniqlanadi. Asosiy algoritmlar sifatida tasodifiy o‘rmon (Random Forest), gradient boosting (XGBoost, LightGBM), neyron tarmoqlar va konvolyutsion neyron tarmoqlar (CNN) qo‘llaniladi. Tadqiqotlarning ko‘rsatishicha, ML modellarining nuqson bashorat aniqligi (F1-score) o‘rtacha 0,75–0,90 oralig‘ida bo‘lib, bu klassik statistik modellardan sezilarli darajada yuqori. Biroq, ushbu yondashuvlarning asosiy cheklovi — ularning ma’lumotlar sifati va hajmiga yuqori darajada bog‘liqligidir [4, 5, 12].

2. Test holatlarini generatsiya qilish. Sun’iy intellekt yordamida test holatlarini generatsiya qilish testlash jarayonida muhim o‘rin tutadi. Generativ modellar (GAN, VAE) va katta til modellari (GPT, Codex, BERT) yordamida avtomatik tarzda test ssenariylari, kirish ma’lumotlari va kutilgan natijalar shakllantiriladi. LLM asosidagi yondashuvlar dasturiy talablar matnini va kod kontekstini tahlil qilib, semantik jihatdan to‘g‘ri test holatlarini generatsiya qilishga qodir. Empirik tadqiqotlar shuni ko‘rsatadiki, ushbu yondashuvlar test qamrovini (line coverage, branch coverage) o‘rtacha 25–40% ga oshiradi. Shu bilan birga, generatsiya qilingan test holatlarining sifatini nazorat qilish va ularning real tizimga mosligini ta’minlash muhim muammolardan biri bo‘lib qolmoqda [6, 7].

3. Mustahkamlangan o‘rganish asosidagi yondashuvlar. Mustahkamlangan o‘rganish testlash jarayonini dinamik boshqarish imkonini beradi. Bu yondashuvda agent sinov muhiti bilan o‘zaro ta’sir qilib, eng samarali test strategiyalarini bosqichma-bosqich o‘rganadi. Q-learning, Deep Q-Network (DQN) va Proximal Policy Optimization (PPO) algoritmlari bunday vazifalarda keng qo‘llaniladi. RL ayniqsa GUI testlashi, fuzzing (xaotik kirishlar yaratish) va regression testlashda foydali hisoblanadi. Tadqiqotlar shuni ko‘rsatadiki, RL asosidagi tizimlar an’anaviy testlash usullariga nisbatan xatoliklarni aniqlash tezligini o‘rtacha 30% ga oshiradi. Shunga qaramay, mukofot funksiyasini to‘g‘ri shakllantirish, o‘rganish jarayonining uzoq davom etishi va hisoblash resurslariga bo‘lgan talab ushbu yondashuvning asosiy cheklovlari hisoblanadi [8, 9].

4. Yondashuvlarni qiyosiy tahlili. Yuqoridagi tahlillarni umumlashtirish va yondashuvlarni qiyosiy baholash maqsadida 1-jadvalda asosiy mezonlar bo‘yicha SI yondashuvlarining qiyosiy ko‘rsatkichlari keltirilgan.

1-jadval.

Sun’iy intellekt yondashuvlarining qiyosiy ko‘rsatkichlari

© Matkarimov Bobirjon Erkinboyevich

Mezon	ML asosidagi	RL asosidagi	Generativ / LLM	An’anaviy usullar
Testlash samaradorligi	Yuqori	Yuqori	Juda yuqori	O‘rtacha
Ma’lumotlarga bog‘liqlik	Juda yuqori	O‘rtacha	Yuqori	Past
Hisoblash resurslariga talab	O‘rtacha	Yuqori	Juda yuqori	Past
Interpretatsiya qulayligi	O‘rtacha	Past	Past	Yuqori
Amaliy joriy etish darajasi	Yuqori	O‘rtacha	O‘rtacha	Yuqori
Test qamrovini oshirish (%)	≈20–30%	≈25–35%	≈30–40%	Bazaviy

1-jadvaldan ko‘rinib turibdiki, sun’iy intellekt asosidagi yondashuvlar testlash samaradorligi va test qamrovini oshirish bo‘yicha an’anaviy usullarga nisbatan sezilarli ustunliklarga ega. Shu bilan birga, ular ma’lumotlarga bog‘liqlik va hisoblash resurslariga talab kabi muhim cheklovlarga ham ega. Eng samaraliroq natija — generativ va LLM asosidagi yondashuvlarda kuzatilmoqda, biroq ularning interpretatsiya qilish murakkabligi va resurs talablari amaliy joriy etishni qiyinlashtiradi.

5. Integratsiyalashgan yondashuvning kontseptual modeli. Tahlil natijalari shuni ko‘rsatadiki, har bir SI yondashuvi muayyan vazifalarni samarali bajaradi, biroq ularning hech biri testlash jarayonining barcha bosqichlarini to‘liq qamrab olmaydi. Mashinaviy o‘rganish nuqson bashorat qilishda kuchli, mustahkamlangan o‘rganish strategik qaror qabul qilishda ustunlikka ega, generativ modellar va katta til modellari esa test qamrovini oshirishda samarali. Shu sababli zamonaviy testlash tizimlarini ishlab chiqishda ushbu yondashuvlarni integratsiyalash muhim ahamiyatga ega.

Taklif etilayotgan integratsiyalashgan intellektual testlash arxitekturasi quyidagi to‘rt qatlamdan iborat: (1) ma’lumotlar va xususiyatlar qatlami — kod metrikalari, tarixiy nuqsonlar, foydalanuvchi ssenariylarini yig‘ish; (2) bashorat qatlami — ML modellari yordamida xatolik ehtimoli yuqori komponentlarni aniqlash; (3) generatsiya qatlami — LLM va generativ modellar yordamida test holatlarini avtomatik shakllantirish; (4) optimallashtirish qatlami — RL agentlari yordamida testlash strategiyalarini dinamik boshqarish va yaxshilash. Bunday integratsiyalashgan yondashuv har bir metodning kuchli tomonlarini birlashtirib, testlash jarayonini yanada samarali, moslashuvchan va avtomatlashtirilgan tarzda tashkil etish imkonini beradi.

Shuni alohida ta’kidlash kerakki, ushbu integratsiyalashgan modelni amaliyotga joriy etish bir qator ishlab chiqarish va tashkiliy masalalarni hal qilishni talab qiladi. Birinchidan, sifatli va etarli hajmdagi ma’lumotlar bazasini shakllantirish zarur. Ikkinchidan, modellarni o‘qitish va ekspluatatsiya qilish uchun zamonaviy hisoblash infratuzilmasi (GPU/TPU klasterlari, bulutli xizmatlar) talab qilinadi. Uchinchidan, modellarning natijalarini izohlash va tasdiqlash uchun mutaxassis muhandislar ishtiroki zarur.

Xulosa

Ushbu maqolada dasturiy ta’minotni testlash jarayonida sun’iy intellekt yondashuvlari tizimli ravishda tahlil qilindi. Mashinaviy o‘rganish, mustahkamlangan o‘rganish, generativ modellar va katta til modellari asosidagi usullar testlash samaradorligini oshirishda muhim rol o‘ynashi aniqlandi.Tadqiqot natijalari shuni ko‘rsatdiki, sun’iy intellekt asosidagi yondashuvlar an’anaviy testlash usullariga nisbatan testlash samaradorligini o‘rtacha 25–40% ga, nuqson aniqlash tezligini esa 30% ga oshirish imkonini beradi. Shu bilan birga, har bir yondashuv

(cc) BY © Matkarimov Bobirjon Erkinboyevich

muayyan vazifalarni samarali bajaradi, biroq ular alohida qo‘llanilganda testlash jarayonining barcha talablarini to‘liq qondira olmaydi.

Maqolada taklif etilgan integratsiyalashgan intellektual testlash arxitekturasi mashinaviy o‘rganish (bashorat), generativ modellar (test holatlarini yaratish) va mustahkamlangan o‘rganish (strategik optimallashtirish) qatlamlarini birlashtirib, testlash jarayonini yanada samarali, moslashuvchan va avtomatlashtirilgan tarzda tashkil etish imkonini beradi.

Kelgusi tadqiqotlar quyidagi yo‘nalishlarga qaratilishi maqsadga muvofiq: (1) taklif etilgan integratsiyalashgan modelni real dasturiy loyihalarda sinab ko‘rish; (2) modellarning hisoblash samaradorligini oshirish; (3) sun’iy intellekt asosidagi testlash tizimlarining natijalarini interpretatsiya qilish (Explainable AI) bo‘yicha yangi yondashuvlarni ishlab chiqish; (4) mahalliy IT-kompaniyalar amaliyotida ushbu tizimlarning iqtisodiy samaradorligini baholash.

Foydalanilgan adabiyotlar ro‘yxati

1. Chen T.Y., Kuo F.-C., Liu H., Poon P.-L., Towey D., Tse T.H., Zhou Z.Q. Metamorphic testing: A review of challenges and opportunities // ACM Computing Surveys. — 2018. — Vol. 51, № 1. — P. 1–27.
2. Harman M., Mansouri S.A., Zhang Y. Search-based software engineering: Trends, techniques and applications // ACM Computing Surveys. — 2012. — Vol. 45, № 1. — P. 1–61.
3. Panichella A., Kifetew F.M., Tonella P. Automated test case generation as a many-objective optimisation problem with dynamic selection of the targets // IEEE Transactions on Software Engineering. — 2018. — Vol. 44, № 2. — P. 122–158.
4. Abu Bakar N.S.A., et al. Machine learning implementation in automated software testing: A systematic review // Journal of Software: Evolution and Process. — 2025. — Vol. 37, № 3. — P. 1–24.
5. Kadhim M.J. Machine learning-driven software testing: Approaches and challenges // International Journal of Software Engineering and Applications. — 2025. — Vol. 16, № 2. — P. 45–62.
6. Andrzejewski M., et al. Automated test generation using large language models: A comparative study // Empirical Software Engineering. — 2025. — Vol. 30, № 1. — P. 1–35.
7. Li W., Kathiresan G. AI in automated test case generation: A survey of techniques and tools // ACM Transactions on Software Engineering and Methodology. — 2025. — Vol. 34, № 2. — P. 1–42.
8. Sutton R.S., Barto A.G. Reinforcement learning: An introduction. — 2nd ed. — Cambridge, MA: MIT Press, 2018. — 552 p.
9. Abo-Eleneen A., et al. Reinforcement learning in software testing: A systematic mapping study // Information and Software Technology. — 2023. — Vol. 158. — P. 107–123.
10. Trudova A., Dolezel M., Buchalcevova A. Artificial intelligence in software testing: Systematic literature review // Information and Software Technology. — 2023. — Vol. 156. — P. 1–18.
11. Faraji A., Pombo N. AI-driven software test automation: An empirical evaluation in industrial settings // IEEE Transactions on Software Engineering. — 2025. — Vol. 51, № 4. — P. 1234–1252.
12. Wang R., Chen X., Liu Y. Deep learning-based defect prediction models: An empirical comparison // Journal of Systems and Software. — 2024. — Vol. 207. — P. 1–19.
13. Pacheco C., Ernst M.D. Randoop: Feedback-directed random testing for Java // Companion to the 22nd ACM SIGPLAN Conference on Object-Oriented Programming Systems and Applications Companion. — 2007. — P. 815–816.
14. Fraser G., Arcuri A. EvoSuite: Automatic test suite generation for object-oriented software // Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering (ESEC/FSE’11). — 2011. — P. 416–419.
15. Zhang J.M., Harman M., Ma L., Liu Y. Machine learning testing: Survey, landscapes and horizons // IEEE Transactions on Software Engineering. — 2022. — Vol. 48, № 1. — P. 1–36.

(cc) BY © *Matkarimov Bobirjon Erkinboyevich*