MĀKSLĪGAIS INTELEKTS UN TĀ IESPĒJAS PROGRAMMĒŠANĀ
ARTIFICIAL INTELLIGENCE AND ITS POSSIBILITIES IN PROGRAMMING

Autors: **Alvis Pastars**, e-mail: ap20147@edu.rta.lv
Zinātniskā darba vadītājs: **Dr.sc.ing., prof. Pēteris Grabusts**
Rēzeknes Tehnoloģiju Akadēmija, Rēzekne,
Atbrīvošanas aleja 115

Abstract. *The authors in this work provides insight into the significance of Artificial Intelligence (AI) and provide a general overview of its strengths, weaknesses, and potential applications. Drawing upon statistical data, the authors present a comprehensive description of AI systems and their impact on various industries.*

Keywords: *Artificial Intelligence, AI applications, Machine learning, Deep learning, Data analysis.*

Ievads

Mākslīgais intelekts (MI) ir izcēlies kā transformējošs spēks dažādās jomās, paātrinot nozares un paplašinot cilvēka spējas. Viena no interesantākajām MI lietojumprogrammām ir programmu koda ģenerēšana, kur MI algoritmi izmanto lielu datu kopu un sarežģītos algoritmos, lai automatizētu programmatūras koda veidošanu. Šajā darbā tiks aplūkota MI būtība un daudzsološās iespējas programmu koda ģenerēšanas jomā.

1. Mākslīgā intelekta būtības izpratne

Mākslīgais intelekts ir plašs un sarežģīts jēdziens, kas ietver cilvēka intelektuālo spēju un procesu emulēšanu ar mašīnām [4]. Šajā kontekstā "intelekts" attiecas uz cilvēka spēju domāt, saprast, mācīties un pieņemt lēmumus. MI galvenokārt tiek realizēts caur datoru sistēmām un programmētām algoritmam, kas ļauj mašīnām veikt sarežģītas kognitīvās funkcijas, kas tradicionāli bija raksturīgas tikai cilvēkiem.

Šīs datoru sistēmas ir nodrošinātas ar algoritmiskām struktūrām, kas ļauj tām mācīties no datiem, saprast un interpretēt cilvēka valodu, attēlus un citus datus, kā arī pieņemt lēmumus, pamatojoties uz analīzi un secinājumiem no šiem datiem. MI ietver vairākas tehnoloģijas un pieejas, tai skaitā [5]:

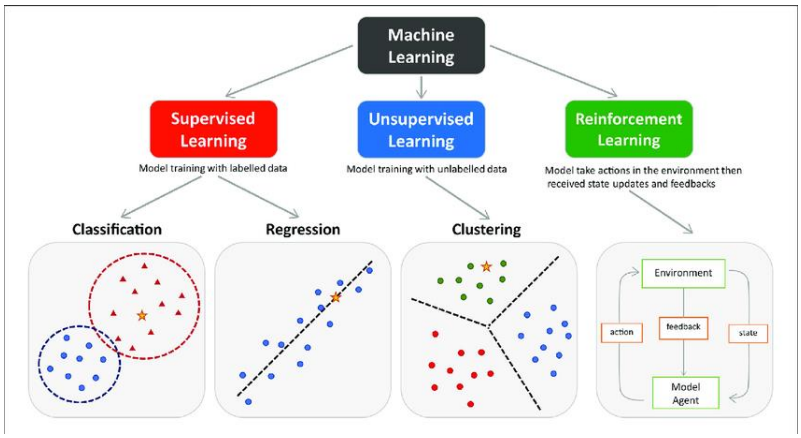
Mašīnmācība: Šī ir tehnoloģija, kas ļauj datoriem mācīties no datiem un uzlabot savas veiktspējas, veicot specifiskus uzdevumus bez specifiskas programmēšanas (skat.1.attēlu).

Dabisko valodu apstrāde: Šī tehnoloģija ļauj datoriem saprast cilvēka rakstīto un runāto valodu, kā arī komunicēt ar cilvēkiem šajā valodā.

Datorredze: Tas ir pielietojumu lauks, kas saistīts ar datoru spēju interpretēt attēlus un video, kā arī saprast un analizēt to saturu.

Dziļā mašīnmācība: Šī ir īpaša mašīnmācības apakšnozare, kas izmanto dziļo neironu tīklu arhitektūru, lai veiktu sarežģītus un augsta līmeņa datu analīzes uzdevumus.

https://doi.org/10.17770/het2024.28.8263



1.attēls. **Mašīnmācības shēma**

Kopumā MI ir atvēris iespējas jaunām iespējām cilvēces attīstībā, piedāvājot līdz šim nebijušas iespējas, lai uzlabotu produktivitāti, inovācijas un cilvēku dzīves kvalitāti dažādās jomās, sākot no medicīnas līdz finansēm un no izglītības līdz izklaidei.

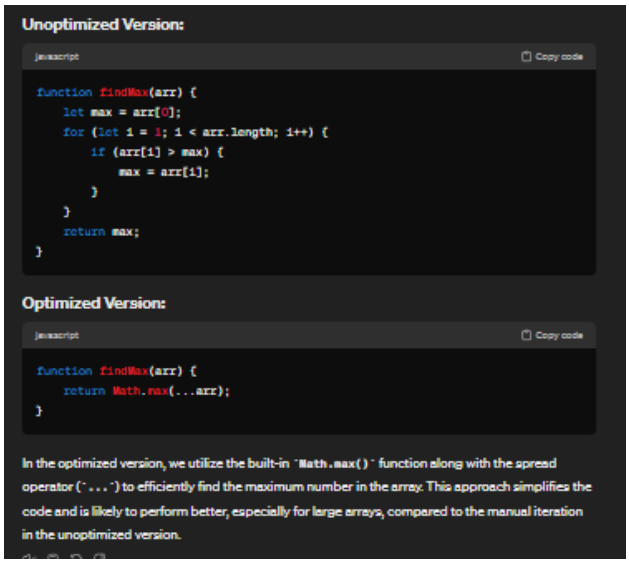
2. MI potenciāls programmu koda ģenerēšanā

Programmu koda ģenerēšana ietver programmatūras koda automatizētu veidošanu, uzdevumu, ko tradicionāli veic cilvēki programmētāji. Tomēr ar progresējošajiem MI sasniegumiem, īpaši mašīnmācībā un dabisko valodu apstrādē, situācija strauji mainās. MI vadītā koda ģenerēšana piedāvā vairākas pievilcīgas iespējas:[3.1][3.2]

Ātrā prototipēšana un iterācija: MI vadītā koda ģenerēšana paātrina programmatūras izstrādes ciklu, automatizējot atkārtotus uzdevumus. Tas ļauj ātri prototipēt (skat.2.attēlu) un veikt daudzas iterācijas, ļaujot izstrādātājiem bez pūlēm eksperimentēt ar dažādām koda struktūrām un funkcionalitātēm.

2.attēls. **ChatGPT koda prototips**

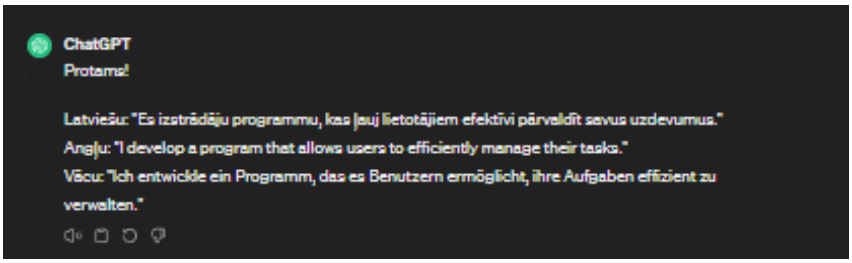
Koda Optimizācija: MI algoritmi var analizēt esošās koda bāzes, lai identificētu ne-efektīvus koda fragmentus un liekos elementus (skat.3.attēlu). Pielietojot optimizācijas tehnikas, MI vadītie koda ģeneratori var radīt efektīvāku kodu, uzlabojot veiktspēju un samazinot resursu patēriņu.



3.attēls. **ChatGPT optimizē kodu**

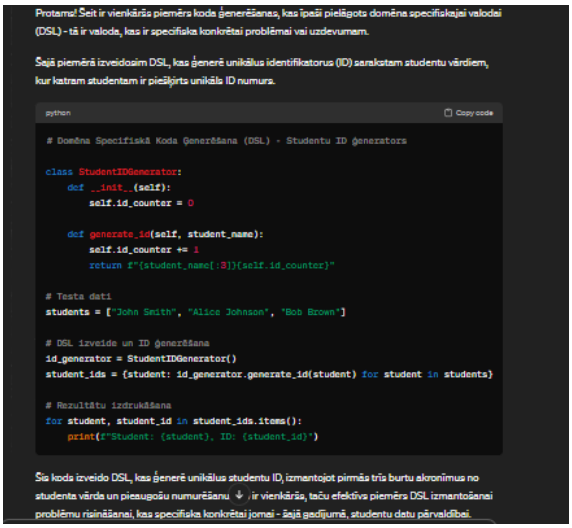
Prasmju nepietiekamības pārvarēšana: Globālā pieprasījums pēc prasmīgiem programmatūras izstrādātājiem pārsniedz pieejamo talantu bāzi. MI vadītā koda ģenerēšana piedāvā risinājumu, automatizējot programmatūras izstrādi. Lietotāji, kas nav eksperti, var izmantot MI rīkus, lai radītu funkcionālu kodu, samazinot atkarību no ļoti specializētām programmēšanas prasmēm.

Dabiskās valodas programmēšana: Progress dabisko valodu apstrādē ir ļāvis MI modeļiem interpretēt un ģenerēt kodu no cilvēkiem saprotamiem aprakstiem (skat. 4.attēlu). Šāda paradigma, pazīstama kā dabiskās valodas programmēšana, atvieglo komunikāciju starp izstrādātājiem un mašīnām, optimizējot kodēšanas procesu.



4.attēls. **ChatGPT valodas apstrāde**

Domēna specifiskā koda ģenerēšana: MI modeļi, kuri ir apmācīti ar domēnam specifiskiem datu kopumiem, var radīt kodu, kas pielāgots konkrētiem pielietojumiem vai nozarēm (skat. 5.attēlu). Vai nu tās ir tiešsaistes tirdzniecības platformas, veselības aprūpes sistēmas vai finanšu programmatūra, MI vadītā koda ģenerēšana var radīt pielāgotus risinājumus, kas optimizēti konkrētai nozarei.



5.attēls. **ChatGPT domēna specifiska koda ģenerēšana**

3. Iespējamās grūtības un aspekti

Neskatoties uz MI iespējām, tā vadītā koda ģenerēšana rada vairākas grūtības un izaicinājumus, kuriem ir jāpievērš īpaša uzmanība. Šie izaicinājumi ietver jautājumus par kodu kvalitāti, drošības ievainojamībām, ētiskām sekām un nepieciešamību pēc cilvēka uzraudzības. Koda kvalitāte: Lai gan MI var automātiski ģenerēt kodu, tas ne vienmēr nodrošina optimālu kvalitāti. Ģenerētais kods var būt sarežģīts, neefektīvs vai nepietiekami pielāgots konkrētajam uzdevumam. Tas var radīt sarežģītības nākamajās izstrādes stadijās un palielināt turpmākos riskus.

Drošības ievainojamības: MI vadītā koda ģenerēšana var radīt drošības riskus, ja algoritmi neņem vērā potenciālās drošības ievainojamības vai neļauj identificēt drošības trūkumus koda izstrādes laikā. Ģenerētais kods var saturēt neaizsargātas koda fragmentus, kas pakļauti ļaunprātīgai izmantošanai vai uzbrukumiem [1.3].

Ētiskās sekas: MI vadītā koda ģenerēšana var radīt ētiskus jautājumus saistībā ar atbildīgu izmantošanu un iespējamām sekām sabiedrībai. Piemēram, ja algoritmi tiek apmācīti ar datiem, kuriem ir iebildumi saistībā ar privātumu vai diskrimināciju, ģenerētais kods var atspoguļot šos ierobežojumus vai nevienlīdzību [1.1].

Nepieciešamība pēc cilvēka uzraudzības: Lai novērstu problēmas un novērstu neparedzētus rezultātus, MI vadītajam koda ģenerēšanai nepieciešama regulāra cilvēka uzraudzība un pārbaude. Cilvēka pārraudzība ir nepieciešama, lai novērtētu ģenerētā koda kvalitāti, identificētu drošības riskus un novērtētu ētiskos aspektus [2.1].

Dinamiskās programmatūras izstrādes dabas dēļ ir nepieciešama pastāvīga MI modeļu pielāgošana un pilnveidošana, lai saglabātu saskaņotību ar mainīgajām prasībām un tehnoloģiju attīstības tempu. Tas prasa ievērojamus resursus un laiku, kā arī prasa nepārtrauktu uzmanību no attiecīgajiem izstrādes komandu locekļiem.

Kopumā, lai gan MI vadītā koda ģenerēšana piedāvā daudzas priekšrocības un inovācijas iespējas programmatūras izstrādē, ir jābūt uzmanīgiem pret šiem izaicinājumiem un nodrošināt, ka tiek veikti atbilstoši pasākumi, lai mazinātu negatīvās sekas un uzlabotu šīs tehnoloģijas izmantošanas drošību un efektivitāti.

Ir svarīgs aspekts, kas jāņem vērā, runājot par MI un tā darbību. Jāsaprot, ka MI, tādā veidā kā tas eksistē šodien, nav pilnībā autonoms tādā nozīmē kā cilvēka intelekts. MI sistēmas ir izstrādātas un uzprogrammētas ar cilvēka palīdzību, lai veiktu konkrētus uzdevumus un pieņemtu lēmumus, pamatojoties uz iepriekš definētām vadlīnijām, algoritmiem un datu kopām, ko definējis izstrādātājs.

Tādēļ, kad MI rīkojas kādā konkrētā veidā, tas nedarbojas autonomi vai pieņem lēmumus tāpat kā cilvēki. Tā vietā tas seko instrukcijām un modeļiem, kas definēti tā programmēšanas laikā. Šis atšķirīgais aspekts ir būtisks, it īpaši ņemot vērā ētiskās sekas un atbildību par MI rīcību. Ja MI algoritms pieņem tendenciozu lēmumu vai rada negaidītas sekas, atbildība galu galā gulstas uz dizaineriem, izstrādātājiem un tiem, kuri apmācīja MI modeli, nevis pašu MI sistēmu. Šī atšķirība palīdz nodrošināt to, ka, izmantojot MI tehnoloģijas dažādās jomās, ir izveidoti atbilstoši drošības pasākumi, sistēmas pārredzamība un atbildība.

Secinājumi

MI piedāvā nepieredzētas iespējas paātrināt programmatūras izstrādes procesus, uzlabot koda kvalitāti un pārvarēt prasmju trūkumu programmatūras inženierijā. Kamēr MI tehnoloģijas turpina attīstīties, tās integrēšana programmatūras izstrādes darbplūsmās sniedz lielas inovāciju un efektivitātes priekšrocības. Tomēr ir svarīgi MI vadītai koda ģenerēšanai pieiet pārdomāti, risinot jautājumus saistībā ar kvalitāti, drošību un ētiskajām sekām, lai izmantotu visas MI priekšrocības.

Summary

Artificial Intelligence has emerged as a transformative force in program code generation, offering unprecedented opportunities to expedite development processes, enhance code quality, and overcome skill gaps in software engineering. As AI technologies continue to evolve, their integration into software development workflows provides significant advantages in innovation and efficiency. However, it is crucial to approach AI-driven code generation thoughtfully, addressing concerns related to quality, security, and ethical implications, to leverage all its benefits.

Literatūras avoti

1. https://www.coe.int/en/web/bioethics/common-ethical-challenges-in-ai
2. https://www.captechu.edu/blog/ethical-considerations-of-artificial-intelligence
3. https://www.unesco.org/en/artificial-intelligence/recommendation-ethics/cases
4. https://www.walkme.com/blog/ai-risks/
5. https://www.techtarget.com/searchitoperations/feature/The-promises-and-risks-of-AI-in-software-development
6. https://www.revelo.com/blog/ai-generated-code
7. https://circleci.com/blog/risks-rewards-generative-ai/
1. AI Prompt Engineering - ELEVEL Webinar
2. https://permutable.ai/navigating-the-challenges-and-opportunities-of-artificial-intelligence/
3. https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence