Mākslīgā intelekta ietekme uz mācību priekšmetu “Datorika” un “Programmēšana” apguvi

Saturs

MI priekšrocības

MI trūkums

Pašvadīta mācīšanās “Datorika”

Pašvadīta mācīšanās “Programmēšana I” un “Programmēšana II”

Izaicinājumi

Diskusija

Priekšrocības izglītībā

Palīdz pašvadītai mācīšanai

Ātrs un ērts informācijas apkopojums

Dažādas platformas var pielietot dažādiem uzdevumiem:

- Darbs ar MS Office datnēm
- Foto un Video rediģēšana
- Programmēšana

Trūkumi

Zūd prasme informācijas meklēšanā un analīzē

Var mazināt zināšanu un prasmju kvalitāti

Datorikā:

- Zūd prasmes pielietot MS Office lietotņu standarta funkcijas. Jo MI platfortmas spēj sagatavot gatavas datnes ar noformējumu, formulām un t.t.
- Zūd izpratne par datora komponenšu funkcijām

Programmēšana:

- Zūd prasme veikt atkļūdošanu
- Zūd nepieciešamība izmantot standarto un nestandarto domāšanu
- Zūd prasmes meklēt nepieciešamās tehnoloģijas

Pašvadīta mācīšanās “Datorika” (7.–9. klase)

Izmantojot MI var apgūt sekojošās galvenās tēmas

- MS Office un Google lietotnēs
- HTML, CSS, JavaScript pamati
- Datortīkli

MS Office un Google lietotnēs

Skaidro funkciju darbību soli pa solim

Spēj izskadrot konkrētas darbības veikšanu ar tekstu un ilustrācijām

Skolotājiem:

- MI palīdz veidot prezentācijas, tabulas, dokumentus.
- Var ģenerēt tekstus, diagrammas, pārbaudes jautājumus.
- Google Docs/Slides piedāvā MI funkcijas, piemēram, “Help me write”.

HTML, CSS, JavaScript pamatos

Var izveidot “road map” pēc kuras ir saprotams kādā secībā vajag apgūt tēmas.

Var veidot teorētiskus piemērus ar plašu skaidrojumu.Piemēru sarežģītību var pašam norādīt.

Var izveidot uzdevumus ar kuriem praktizēt iegūtās prasmes.

Darbu automatizācijas ,kuras skolotājs var izmantot mācību stundās:

- ChatGPT vai Copilot palīdz rakstīt un saprast kodu.
- Var izskaidrot kļūdas un piedāvāt piemērus.

Datortīkli , Foto un video apstrādē

MI var izskaidrot tīkla darbības principus, terminus vai zīmēt tīkla topoloģijas shēmas.

MI rīki palīdz uzlabot attēlus (izšķirtspēju), izgriezt fonu, veidot video klipus.

Piemērs

Kā pievienot atsauci MS Word dokumentā

Lieliski jautāts 👍 — **atsauces pievienošana MS Word dokumentā** ir diezgan vienkārša, un to var izdarīt vairākos veidos atkarībā no tā, vai tu gribi:

1. **Automātisku atsauci (citātu ar avotu datu bāzi)** — izmantojot Word iebūvēto "References" jeb "Atsauces" rīku,
   vai
2. **Manuālu atsauci** — ievadot to pašam tekstā vai kā zemsvītras piezīmi.

Zemāk — **soli pa solim abiem variantiem**.

🧩 1. Automātiska atsauce, izmantojot "References" cilni

📝 Soļi:

1. Atver **cilni "References" (Atsauces)** augšējā rīkjoslā.
2. Noklikšķini uz **"Insert Citation" (Ievietot citātu)**.
3. Izvēlies **"Add New Source..." (Pievienot jaunu avotu)**.
4. Atveras logs "Create Source" (Izveidot avotu):
   - Izvēlies **avota tipu** (piem., Book, Journal Article, Website utt.).
   - Aizpildi laukus: autora vārds, nosaukums, gads utt.
5. Spied **OK**, un atsauce parādīsies tekstā, piemēram: *(Smith, 2022)*.
6. Lai beigās izveidotu **literatūras sarakstu**, spied:
   - "Bibliography" → izvēlies "Bibliography" vai "Works Cited" stilu.
   - Word automātiski ģenerēs sarakstu no visiem ievadītajiem avotiem.

Pašvadīta mācīšanās “Programmēšana I” un “Programmēšana II”

Izmantojot MI var apgūt sekojošās tēmas

Programmēšana valodas bāze (mainīgie,nosacījumi,cikli,datu tipi)

Tīmekļa lietotņu izstrāde (padziļināti)

Padziļinātā kursa tēmas (OOP,darbs ar datubāzēm, API,grafiskā lietotāja saskarsme,servera izveide)

Bāzes apgūšana

MI var izveidot “road map” jebkurā izvēlētajā programmēšanas valodā.

Var saprotamā valodā paskaidrot teoriju. Sniedz koda piemērus.

Var noģenerēt dažādas sarežģītības pakāpes uzdevumus prasmju pilnveidei.

Tīmekļa lietotņu izstrāde

Var paskaidrot un parādīt kāda ir pareiza lietotnes arhitektūra

Var dot teorētisku aprakstu ar koda piemēriem efektīvai HTML,CSS,JS pielietošanai.

Var dot uzdevumus kuros jāpielieto praktiskās prasmes un radošā domāšana(veidot dizainu pēc apraksta)

Padiziļinātā kursa tēmu apgūšana

Var izskaidrot OOP koncepciju

Var paskaidrot un parādīt kā strādā API un kā izveidot savu API.

Paskaidro par datubāzēm,datu tipiem, relācijām.

Parāda un apraksta kā veidot GLS.

Paskaidro kā izveidot lokālo serveri. Salīdzina dažādas bibliotēkas un to projektu arhitektūru.

Parāda un paskaidro kā veikt vienību testēšanu.

Veido sarežģītus uzdevumus (tēmu apvienojumu) prasmju pārbaudei un pilnveidei.

Piemērs

Kā izveidot lokālo serveri izmantojot Flask?

Labi! Es tev parādīšu, kā izveidot vienkāršu lokālo serveri, izmantojot **Flask** Pythonā. Es sniegšu **koda piemēru** un detalizētu paskaidrojumu par katru soli.

1️⃣ Instalācija

Pirmkārt, pārliecinies, ka Flask ir instalēts. Atver terminālī un izpildi:

bash — Kopēt kodu

```
pip install Flask
```

2 Koda piemērs

```python
# faila nosaukums: app.py

from flask import Flask

# 1. Izveidojam Flask aplikācijas instanci
app = Flask(__name__)

# 2. Definējam maršrutu (route) mājas lapai
@app.route("/")
def home():
    return "Sveiks! Tas ir mans lokālais Flask serveris."

# 3. Palaid serveri
if __name__ == "__main__":
    app.run(debug=True)
```

3 Ko katrs solis dara

1. `from flask import Flask`

   Importējam Flask klasi, kas ļauj mums izveidot web serveri.

2. `app = Flask(__name__)`

   Izveidojam Flask aplikāciju. `__name__` palīdz Flask noteikt, kur meklēt failus, piemēram, statiskos resursus vai šablonus.

3. `@app.route("/")`

   Tas ir maršruts (URL ceļš). Šajā gadījumā tas ir mājas lapas ceļš `/`. Kad lietotājs apmeklē `http://127.0.0.1:5000/`, tiks izsaukta funkcija `home()`.

4. `def home(): return "..."`

   Funkcija, kas atgriež tekstu lietotājam pārlūkā. Šeit vienkārši atgriežam tekstu "Sveiks! Tas ir mans lokālais Flask serveris."

5. `if __name__ == "__main__": app.run(debug=True)`

   Pārbauda, vai skripts tiek izpildīts tieši (nevis importēts citur).

   - `app.run()` palaid serveri.
   - `debug=True` ieslēdz "debug mode", kas automātiski pārlādē serveri, ja maini kodu, un rāda detalizētus kļūdu ziņojumus.

Noderīgi MI rīki

Joma	Rīks	Ko palīdz darīt
Teksts / prezentācijas	ChatGPT, Microsoft Copilot, Google Gemini	Veidot idejas, slaidus, labot tekstus
Tabulas	Excel Copilot, Google Sheets SmartFill	Analizēt datus, veidot formulas
Kods	ChatGPT, Replit Ghostwriter	Izskaidrot HTML/CSS/JS, labot kļūd
Foto / video	Canva AI, Adobe Express, Pika Labs	Rediģēt attēlus un video
Tīkli	ChatGPT, Google Bard	Izskaidrot tīklu darbību, terminus

Ko skolēns iegūst

- Mācās patstāvīgi plānot darbu.
- Uzlabo digitālās prasmes.
- Attīsta radošumu un kritisko domāšanu.
- Iemācās atbildīgi lietot MI tehnoloģijas.

Kaitnieciski MI rīki WormGPT un FraudGPT

WormGPT un FraudGPT ir bīstami un nelikumīgi mākslīgā intelekta rīki, kas tiek izmantoti kibernoziegumos.

Tie nepalīdz mācībās — tie ir veidoti, lai:

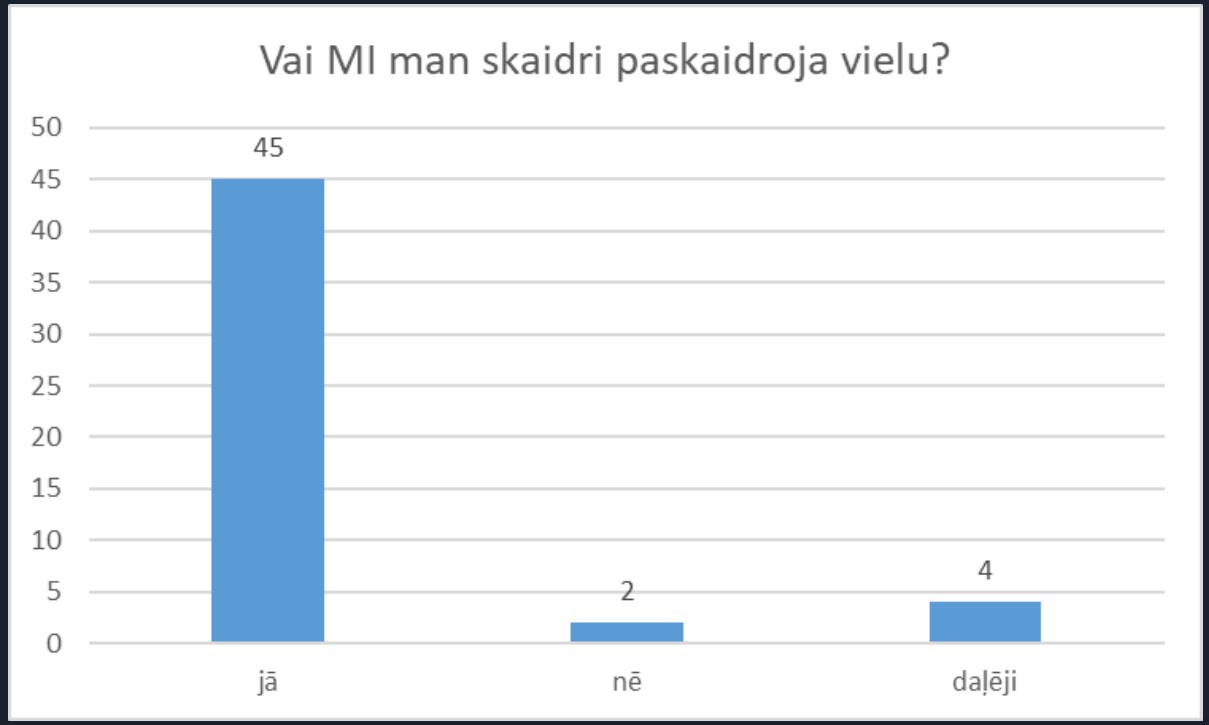
- rakstītu krāpnieciskas e-pasta vēstules (phishing),
- lauztu paroles un izmantotu drošības ievainojamības,
- maldinātu cilvēkus un izmantotu datus ļaunprātīgi.

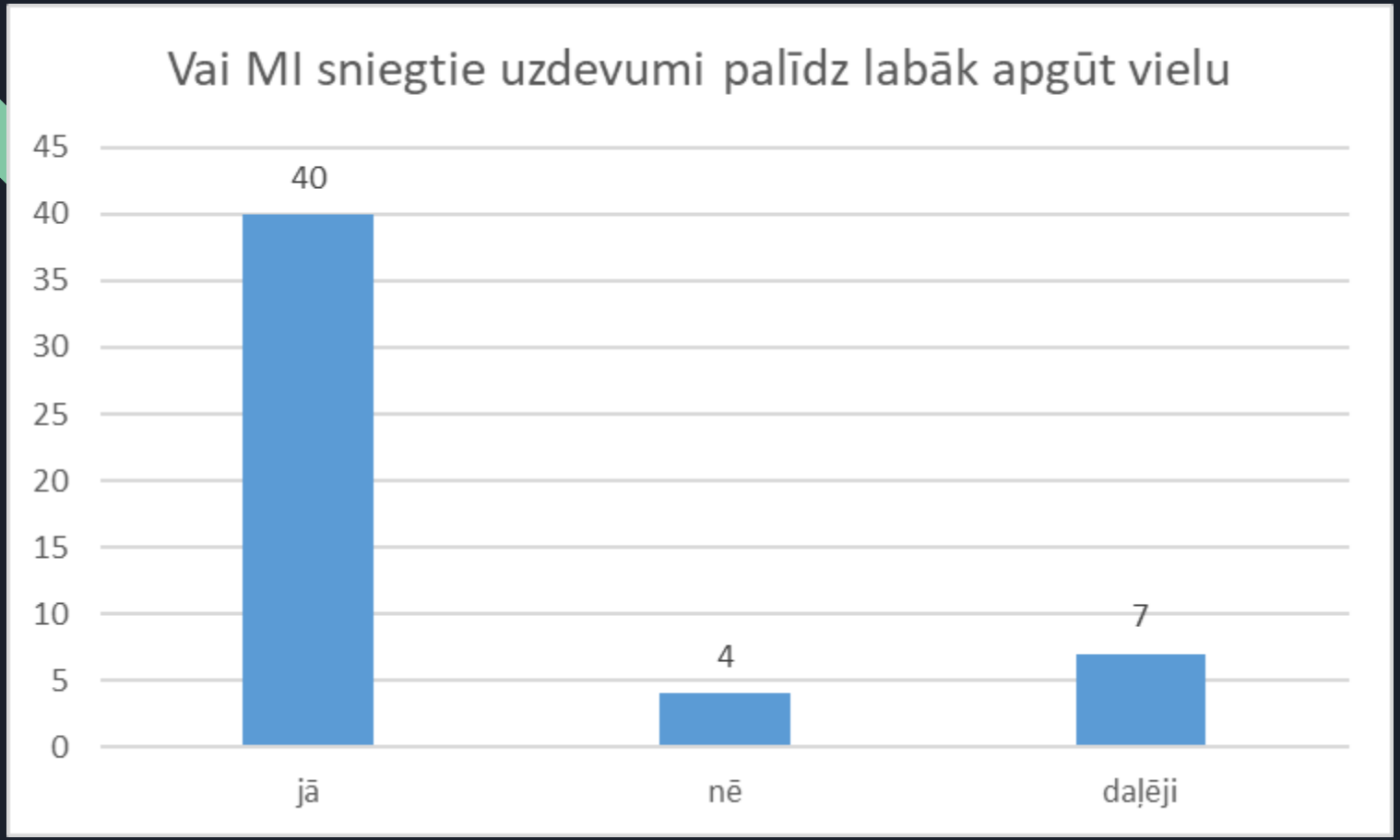
Kāpēc no tiem jāizvairās

Tie rada risku datiem, datoram un cilvēkiem.

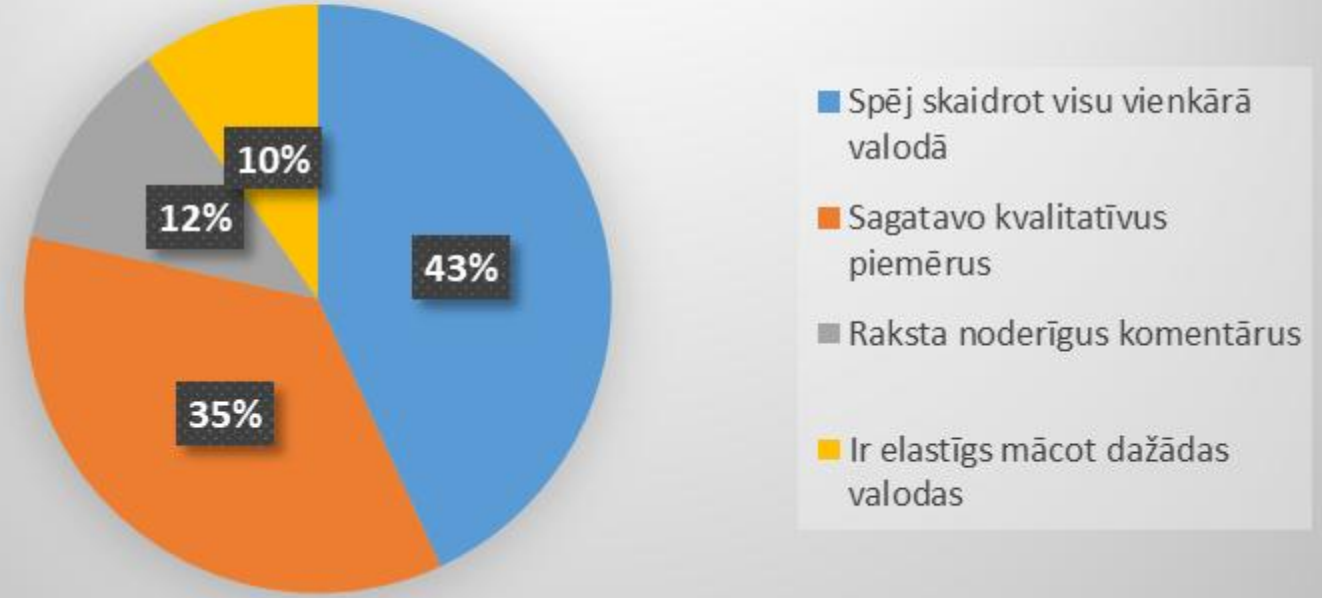
Izmantojot šādus rīkus, var nejauši iesaistīties kibernoziegumā.

Skolēnu rezultāti




Vai MI sniegtie uzdevumi palīdz labāk apgūt vielu
45
40
35
30
25
20
15
10
5
0
40
4
7
jā
nē
daļēji

Kas ir noderīgs MI pašvadītā mācīšanā



Diskusija