[Tiriama sritis]

Trumpas įvadas apie tiriamą sritį. Darbe tiriama sritis yra mašininis mokymasis. Tai yra dirbtinio intelekto sritis, kuri pasitelkia statistinius algoritmus, kad apibrėžtų duomenų generavimo mechanizmą, ar egzistuojančius sąryšius

[Tikslas ir uždaviniai]

Darbo tikslas – palyginti privatumą saugančius dirbtinio intelekto algoritmus pagal jų saugumą, našumą ir panaudojamumą, bei pateikti rekomendacijas.

Darbe buvo iškelti keturi uždaviniai

- Pirmas – išanalizuoti esamus algoritmus pagal jų saugumą ir panaudojamumą.
- Antras – Identifikuoti kriterijus, pagal kuriuos būtų vertinami algoritmai.
- Trečias – Realizuoti dalį ištirtų algoritmų, kurie nėra realizuoti.
- Ketvirtas – Palyginti algoritmus pagal našumą ir pateikti rekomendacijas.

[Problematika]

Darbe yra nagrinėjama, kaip įvairūs algoritmai, metodai sprendžia duomenų privatumo problemas. Problemos yra keturios:

1. Turint sukurtą mašininio mokymosi modelį, negali būti įmanoma atgaminti pradinių mokymosi duomenų.
2. Trečios šalys negali matyti vartotojų įvedamų duomenų.
3. Modelio išvesties neturi matyti asmenys, kuriems šie duomenys nepriklauso.
4. Sukurtas modelis negali būti niekieno pasisavintas.

[Modelių duomenų lyginimas (1)]

Literatūroje aprašytos metrikos, skirtos modelių saugumui lyginti, atsižvelgia tik į duomenis, su kuriais modelis yra apmokomas. Čia yra pateikta formulė, skirta apskaičiuoti duomenų „atvirumo“ metriką. Kuo atvirumo reikšmė yra didesnė, tuo labiau yra nesaugūs duomenys.

Pirmas metodas naudoja rangus, skaičiuojant metriką. Šiame kontekste, elemento rangas yra jo vieta tam tikrame skirstinyje. Norint apskaičiuoti duomenų rangus, reikia sugeneruoti visus galimus duomenų variantus ir juos surūšiuoti didėjančia entropijos tvarka. Tai yra, sustatome visus galimus variantus didėjimo tvarka ir gauname ieškomo elemento vietą eilėje. Tokiu atveju, pati paprasčiausia galima duomenų eilutė turės rangą lygų 1.

Šis metodas nėra praktiškas, nes reikia sugeneruoti visus galimus duomenų variantus ir reikia daug resursų.

[Modelių duomenų lyginimas (2)]

Šis atvirumo metrikos skaičiavimo metodas yra praktiškesnis. Reikia aproksimuoti duomenų pasiskirstymą, pagal didėjančią duomenų entropiją. Turint aproksimuotą skirstinį, reikia apskaičiuoti integralą tarp 0 ir ieškomo elemento, tai yra rasti skirstinio plotą iki ieškomo elemento. Šis metodas yra paprastesnis, nes galima aproksimuoti atsakymą.

[Pasiūlyta tyrimo metodika]

Darbe pateiktas pasiūlymas naudoti šią DMDK, tai yra didžiausio modelio duomenų nuokrypio metriką, kuri leidžia lyginti skirtingus mašininio mokymosi modelius tarpusavyje. Metrika buvo mano paties pasiūlyta.

Taigi, ieškome maksimalaus epsilon reikšmės eilutėje, ją suvidurkiname, pridedame kitas eilutes, vėl suvidurkiname ir gauname didžiausią galimą epsilon reikšmę iš apskaičiuotų maksimalių eilučių reikšmių, kuri nurodo galima didžiausią duomenų nuokrypį, su kurio nepakis modelio išvestis. Prieš DMDK skaičiavimą reikia paimti visus modelio mokymui skirtus duomenis ir kiekvienai duomenų eilutei apskaičiuoti modelio išvestį. Skaičiavimus reikia atlikti tik su tomis eilutėmis, su kuriomis modelis išvedė teisingą atsakymą, tai yra, modelio išvestis sutampa su tikrai duomenimis. Turint tik tas eilutes, su kuriomis modelis išvedė teisingą atsakymą, galima į nelygybę įstatyti kintamuosius.

Kuo metrika yra mažesnė, tuo tiksliau galima nuspėti, kokie duomenys buvo naudojami modelio mokymui. Ši metrika leidžia lyginti skirtingus modelius su skirtingais duomenimis.

[Metrikos validacija (1)]

Metrika yra naujai sugalvota, todėl ją reikia verifikuoti. Tai yra daroma keliai eksperimentais.

Pirmas iliustracinis eksperimentas – turime du modelio parametrus. Buvo sukurti keli modeliai – su dideliu DMDK rodikliu ir mažu.

Modelis su maža DMDK reikšme, atgamino pradinius duomenis. Su modeliais, kurie turėjo didelę DMDK reikšmę, nepavyko atgaminti pradinių duomenų.

[Metrikos validacija (2)]

Antras eksperimentas – eilučių duomenys buvo visi padauginti iš N. DMDK rodiklis išliko vienodas.

Paskutinis trečias eksperimentas, sukurtas modelis, kuris visą laiką grąžiną tapačią reikšmę. Jo apskaičiuota DMDK reikšmė yra arti begalybės.

[Pridėto triukšmo tyrimas]

Turint naują metriką, buvo atlikti keli tyrimai.

Pirmas – pridėto triukšmo tyrimas. Kairiame grafike yra pateikta tiesinė progresija tarp modelio tikslumo procentais ir pridedamo triukšmo. Rezultatas logiškas – triukšmas mažina modelio tikslumą. Rezultatai – vidutinis statistinis ryšys, statistiškai reikšminga.

Dešiniame grafike pateikta tiesinė priklausomybė tarp pridedamo triukšmo ir modelio DMDK reikšmės. Tai yra, kuo labiau yra didinamas triukšmas, tuo mažėja modelio tikslumas, tačiau didėja saugumas. Rezultatai – stiprus statistinis ryšys, statistiškai reikšminga.

[Pallier tyrimas]

Antras tyrimas – tiriama homomorfinio šifravimo pallier algoritmas.

Homomorfinis šifravimas, yra šifravimo algoritmų klasė, kuri yra grindžiama principu, leidžiančiu atlikti skaičiavimus su užšifruotais duomenimis, jų neatšifruojant.

Kituose darbuose yra patariama naudoti homomorfinį šifravimą, dėl duomenų saugumo.

Atliktas tyrimas parodo, kad modelio tikslumui artėjant 100%, DMDK reikšmė kyla, tačiau kai modelio tikslumas yra žemas, DMDK reikšmė yra maža ir modelis nėra saugus. Taigi, jeigu prognozuojame, kad modelis pasieks aukštą tikslumą procentais, derėtų apsvarstyti homomorfinį šifravimą.

Triukšmo ir DMDK rodiklio koreliacija turi labai stiprų statistinį ryšį, statistikai reikšminga.

[PyTorch neuroninio tinklo tyrimas]

Trečias atliktas tyrimas yra su PyTorch karkasu pilnai sujungtų trijų sluoksnių neuroniniu tinklu, kai parametrų skaičius yra mažiau nei 20. PyTorch modeliui esant 50-70% tikslumui, jis yra saugesnis už homomorfinį šifravimą apie 15 kartų.

Esant didesniam parametrų skaičiui, nei 20, homomorfinis šifravimas yra visą laiką saugesnis už neuroninius tinklus.

Modelio tikslumui artėjant 100%, modelis pradeda labiau atskleisti duomenis ir tampa nebesaugus.

Tikslumo ir DMDK reikšmė turi silpną statistinį ryšį ir statistiškai nereikšminga.

[Rezultatų aprobavimas]

Baigiamojo darbo rezultatai buvo pristatyti nacionalinėje konferencijoje, o pranešimo tezės publikuotos konferencijos leidinyje

[Išvados]

Esant aukštam modelio tikslumui, rekomenduojama naudoti homomorfinį šifravimą, nepriklausomai nuo kiek parametrų priima modelis. Esant mažesniam, nei 70% tikslumui, kai modelis priima < 20 parametrų, rekomenduojama naudoti PyTorch neuroninius tinklus.

Esant didesniam modelio parametrų skaičiui, PyTorch neuroniniai tinklai labiau prisimena pradinius mokymosi duomenis ir juos galima lengviau atskleisti. Todėl parametrų skaičiau esant daugiau nei 20, saugiau naudoti homomorfinį šifravimą.

Naudojant neuroninius tinklus be homomorfinio šifravimo ir modelio tikslumui esant daugiau nei 80%, rekomenduojama pridėti triukšmą prie pradinių modelio duomenų.

Eksperimentiniai tyrimai indikavo, jog pradinių duomenų kiekis neturi įtakos modelio duomenų saugumui.