[Įvadas]

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

[Pagrindinė problema]

Darbe pagrindinė tiriama problema yra duomenų išsaugojimas mašininiame mokyme. Turint sukurtą modelį, neturi būti galima atgaminti duomenų, pagal kuriuos jis buvo mokomas, bei negali būti identifikuoti asmenys, kurių duomenys yra naudojami.

Vienas iš pavyzdžių gali būti teksto atpažinimo modelis. Tokie modeliai yra naudojami analizuojant parašytą tekstą ir pasiūlant sekantį žodį ar pataisymą.

Tarkime, kad kalbos modelis buvo mokomas su programuotojo pateiktu tekstu, kuriame yra privačios informacijos. Jeigu modelis yra nesaugus, jis gali kitiems vartotojams atskleisti privačias detales. Duomenų atskleidimo pavyzdžiu gali būti tokia situacija, jeigu vartotojas pradės rašyti kokį nors adresą, o modelis užbaigs ir parašys programuotojo adresą.

[Modelių duomenų lyginimas]

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

Pirmas metodas po kaire, naudoja rangus, skaičiuojant metriką. Šiame kontekste, elemento rangas yra jo vieta tam tikroje distribucijoje. 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.

Kiek praktiškesnis variantas yra pateiktas dešinėje. Reikia aproksimuoti duomenų pasiskirstymą, pagal didėjančią duomenų entropiją. Turint aproksimuotą distribuciją, reikia apskaičiuoti integralą tarp 0 ir ieškomo elemento, tai yra rasti distribucijos plotą iki ieškomo elemento. Šis metodas yra paprastesnis, nes galima aproksimuoti atsakymą.

[Pasiūlyta tyrimo metodika]

Darbe pateiktas pasiūlymas naudoti šią DMDK metriką, kuri leidžia lyginti skirtingus mašininio mokymosi modelius tarpusavyje.

Tokias metrikos pavadinimas yra pasirinktas dėl pačios skaičiavimo logikos. 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.

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

Tiksliau apie metodą, 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. Turint tik tas eilutes, su kuriomis modelis išvedė teisingą atsakymą, galima į nelygybę įstatyti kintamuosius.

[Metrikos validacija (1)]

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

Pirmas iliustracinis eksperimentas – turime du modelio parametrus KMI ir gimdymų skaičių ir išvestį – ar moteris serga cukriniu diabetu.

Buvo sukurti keli modeliai – su dideliu DMDK rodikliu ir mažu.

Modelis su maža DMDK reikšme, kuris yra mažiau saugus, atgamino pradinius duomenis pateiktus antroje lentelėje. Buvo pasirinkti duomenys artimi tikriems duomenims, jie yra pateikti pirmoje lentelėje ir buvo analizuojama modelio išvestis, kol bus atgaminti tikri duomenys.

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. Tai yra todėl, kad modelis nieko neprisimena ir neįmano atskleisti kokių duomenų.

[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.

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.

[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.

[PyTorch neuroninio tinklo tyrimas]

Trečias atliktas tyrimas yra su PyTorch karkasu pilnai sujungtų trijų sluoksnių neuroniniu tinklu. PyTorch modeliui esant 50-85% tikslumui, jis yra saugesnis už homomorfinį šifravimą.

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

[Išvados]

Esant aukštam modelio tikslumui, rekomenduojama naudoti homomorfinį šifravimą. Esant mažesniam, nei 70% tikslumui, rekomenduojama naudoti PyTorch neuroninius tinklus.

Esant didesniam modelio parametrų skaičiui, PyTorch neuroniniai tinklai labiau prisimena pradinius mokymosi duomenis ir juos galima lengviau atskleisti.

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.