Įmonių bankroto prognozavimas naudojant gilųjį mokymą

Ieva Rizgelienė

Kauno technologijos universitetas,
K. Donelaičio g. 73, Kaunas
paulaviciute.ieva@gmail.com

Santrauka. Šiame straipsnyje pristatomi sukurti giliojo mokymosi modeliai skirti įmonių bankroto prognozavimui. Tyrimo metu, naudojant Lietuvos įmonių duomenis, sukurti du modeliai: daugiasluoksnis perceptronas ir konvoliucinis neuroninis tinklas. Modelių apmokymui ir hiperparametrų validavimui, buvo naudojami subalansuoti duomenų poaibiai. Siekiant įvertinti modelių gebėjimą atskirti bankroto atvejus išbalansuotoje duomenų aibėje, modelių testavimui buvo naudojamas poaibis, kuriame buvo išlaikytas pradinio duomenų rinkinio klasių disbalansas. Gauti rezultatai parodė, kad sukurti giliojo mokymosi modeliai atpažįsta bankroto atvejus duomenų aibėje su dideliu klasių disbalansu.

Raktiniai žodžiai: gilusis mokymasis, dirbtiniai neuroniniai tinklai, konvoliucinis neuroninis tinklas, bankroto prognozavimas, daugiasluoksnis perceptronas, klasifikavimo modelis.

1 Įvadas

Įmonių bankrotas – tai neišvengiamas ekonomikos reiškinys. Įmonės bankroto prognozė gali padėti identifikuoti įmones, susiduriančias su sunkumais, o tai gali užkirsti kelią įmonės bankrotui arba leisti tinkamai tam pasiruošti, taip sumažinant galimus nuostolius.

Ekonomikos inžinieriai bankrotą pradėjo nagrinėti 1930 m. Pirmiausiai buvo susikoncentruota į bankrutavusių ir nebankrutavusių įmonių finansinių rodiklių palyginimą. 1968 m. Edward'as I. Altman'as pirmą kartą bankrotui prognozuoti pritaikė daugiamatės diskriminantinės analizės modelį [1]. Logistinė regresija pirmą kartą bankrotui prognozuoti buvo pritaikyta 1980 m. Ohlson'o [2]. Gerokai vėliau (21a.) buvo pritaikyti tokie modeliai kaip pagrindinių komponenčių analizė, sprendimų medžiai ir atraminių vektorių klasifikatorius. Giliojo mokymosi algoritmai vis dar yra retesnis pasirinkimas bankroto prognozavimo uždaviniui spręsti, lyginant su kitais klasifikavimo metodais.

DOI: https://doi.org/10.15388/LMITT.2022.9

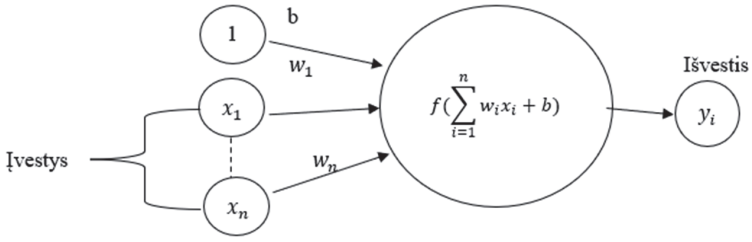
Gilusis mokymasis kiekviename sluoksnyje gali išmokti vis abstraktesnius duomenų bruožus, todėl parinkus tinkamą neuroninio tinklo architektūrą, galima gauti pakankamai tikslius prognozės rezultatus ir pranokti klasikinius bankroto prognozei taikomus algoritmus. Tyrėjai, bankroto prognozavimui pritaikę tiesioginio sklidimo neuroninius tinklus, pastebėjo, kad šio metodo klasifikavimo rezultatai rodo gerus rezultatus ir prilygsta klasikiniams klasifikavimo algoritmams [3]. Konvoliuciniai neuroniniai tinklai yra plačiai taikomi vaizdų atpažinimo uždaviniams spręsti, tačiau jau yra atlikta tyrimų ir įmonių bankroto prognozavimo tematika, kuriuose siūloma įmonių finansinius rodiklius išreikšti nespalvotų vaizdų pavidalu, pritaikius minėtą duomenų transformaciją, buvo pastebėta, kad toks metodas gali pranokti daugelį klasifikavimo algoritmų [4].

Šio tyrimo tikslas, naudojant giliuosius dirbtinius neuroninius tinklus, sukurti įmonių bankroto prognozės modelius, kurie prognozuotų įmonių bankrotą likus vienam mėnesiui iki šio įvykio. Tyrimo metu buvo išbandomi dviejų rūšių neuroniniai tinklai: daugiasluoksnis perceptronas ir konvoliucinis neuroninis tinklas. Kuriant neuroninių tinklų modelius, geriausios tinklo architektūros ir geriausių hiperparametrų paieškai buvo taikomas Bajeso optimizavimas, išbandomos 8 skirtingos aktyvacijos funkcijos ir 7 skirtingi optimizavimo algoritmai. Tyrimo metu naudojama atvirojo kodo programinės įrangos biblioteka TensorFlow ir Python programavimo kalba.

2 Metodai

2.1 Dirbtiniai neuroniniai tinklai

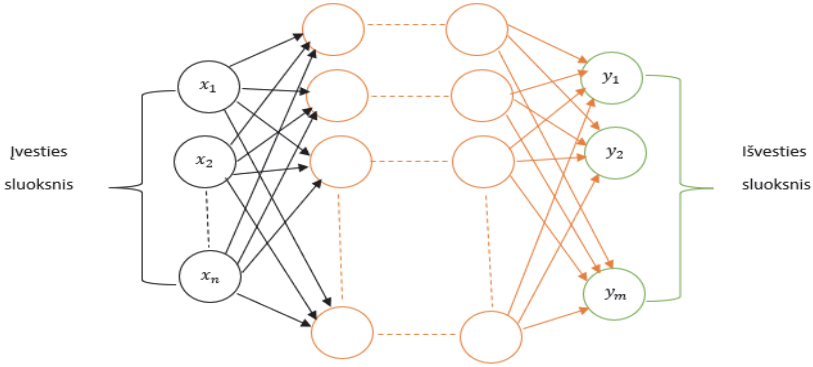
Dirbtiniai neuroniniai tinklai – tai informacijos apdorojimo sistemos, paremtos biologinio neurono veikimo principu. Kaip ir žmogaus smegenų atveju, dirbtiniai neuroniniai tinklai yra pagrįsti tarpusavyje sujungtais neuronais. Kiekvienai jungčiai priskirti koeficientai, vadinami neuronų svoriais. Į neuroninį tinklą įeinančios jungtys vadinamos įvestimis, išeinančios – išvestimis. Toliau pateikiama dirbtinio neurono schema.



1 pav. Dirbtinio neurono schema [5]

2.1.1 Tiesioginio sklidimo neuroniniai tinklai

Tiesioginio sklidimo neuroninio tinklo atveju, neuronai yra grupuojami į sluoksnius, o sluoksniai sujungiami taip, kad tinkle nesusidarytų ciklai. Šių neuroninių tinklų atveju, kiekvienais dirbtinis neuronas iš esamo sluoksnio yra sujungtas su kiekvienu dirbtiniu neuronu iš praeities sluoksnio. Šiame tinkle duomenys negali būti siunčiami į praeitus sluoksnius. Tiesioginio sklidimo neuroniniai tinklai, vadinami giliaisiais, tuomet, kai turi daugiau nei vieną paslėptąjį sluoksnį. Šiame tyrime yra išbandomi dviejų rūšių gilieji tiesioginio sklidimo neuroninio tinklai: daugiasluoksnis perceptronas ir konvoliucinis neuroninis tinklas, 2 pav. vaizduojamas daugiasluoksnio perceptrono modelis.



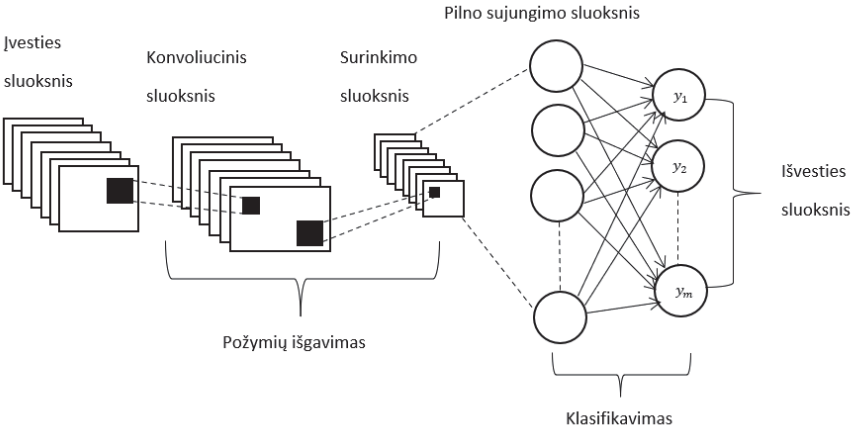
2 pav. Daugiasluoksnio perceptrono modelis [6]

2.1.2 Konvoliuciniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai, tai tiesioginio sklidimo neuroniniai tinklai, turintys konvoliucinius sluoksnius. Konvoliuciniai neuroniniai tinklai yra plačiai taikomi vaizdams analizuoti. Trys pagrindiniai konvoliucinių neuroninių tinklų sluoksniai:

- konvoliucinis sluoksnis,
- surinkimo sluoksnis (angl. *pooling*),
- pilnai sujungtas sluoksnis.

Konvoliuciniuose neuroniniuose tinkluose, neuronai išsidėstę trijose dimensijose: plotis, aukštis, gylis. Toliau pateikiama konvoliucinio neuroninio tinklo schema.



3 pav. Konvoliucinio neuroninio tinklo schema [6]

Trumpai aptarsime 3 pav. schemoje vaizduojamus neuroninio tinklo sluoksnius. Į neuroninį tinklą perduodamos įvestys, turi trimatį pavidalą. Pagrindinis konvoliucinio sluoksnio tikslas, naudojant apmokomus filtrus (angl. *kernels*), išgauti duomenų požymius. Surinkimo sluoksnyje yra sumažinamas įvesčių dydis. Pilno sujungimo sluoksnyje, gauti duomenys, konvertuojami į vienmatę erdvę, kitaip tariant šiam sluoksniui perduodamos įvestys yra paverčiamos vektoriumi. Tuomet gautoms reikšmėms pritaikoma aktyvacijos funkcija. Išvesties sluoksnyje, pritaikius pasirinktą aktyvacijos funkciją, gaunamos išvestys. Gauti rezultatai yra suklasifikuojami.

2.1.3 Aktyvacijos funkcijos

Daugiasluoksnių perceptronų modeliuose, aktyvacijos funkcija, naudojama kiekviename sluoksnyje, tuo tarpu konvoliucinių neuroninių tinklų atveju, aktyvacijos funkcija naudojama konvoliuciniuose ir pilno sujungimo sluoksniuose. Šiame tyrime yra išbandomos šios aktyvacijos funkcijos:

1. ReLU (angl. *Rectified linear unit*) funkcija, apibrėžiama: $f(x) = \max(0, x)$.
2. Softsign funkcija, apibrėžiama: $f(x) = \frac{x}{1+|x|}$.
3. Softplus funkcija, apibrėžiama: $f(x) = \ln(1 + e^x)$.
4. Leaky ReLU (angl. *Leaky Rectified Linear unit*) funkcija, apibrėžiama: $f(x) = \max(0.1 \cdot \mathrm{x},\ \mathrm{x})$.
5. Sigmoidinė funkcija, apibrėžiama: $f(x) = \frac{1}{1+e^{-x}}$.

6. Hiperbolinio tangento funkcija, apibrėžiama: $\tan(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.
7. Elu (angl. Exponential Linear Unit) funkcija, apibrėžiama:

   $$f(x) = \begin{cases} x & x > 0 \\ \alpha(e^x - 1) & x \leq 0 \end{cases}.$$

   Čia $\alpha < 0$. α – funkcijos hiperparametras, kuris kontroliuoja neigiamas įvestis. Ši funkcija mažina nykstančio gradiento efektą, taip pagreitindama giliųjų neuroninių tinklų mokymąsi ir gali užtikrinti didesnį klasifikavimo tikslumą [7].
8. Selu (angl. *Scaled Exponential Linear Unit*) funkcija, apibrėžiama:

   $$f(x) = \lambda \begin{cases} x & x > 0 \\ \alpha e^x - \alpha & x \leq 0 \end{cases}.$$

   Ši funkcija, gali padėti gauti išvestis, pasiskirsčiusias pagal normalųjį skirstinį [8]. Čia λ – standartizavimo konstanta, ši konstanta yra lygi 1.0507, α taip yra apibrėžta ir yra lygi 1.6732.

2.1.4 Dirbtinių neuroninių tinklų apmokymas

Daugiasluoksniui perceptronui, apmokymo metu, keičiami jungčių svoriai, konvoliuciniam neuroniniam tinklui, keičiami jungčių ir filtrų svoriai. Keičiant svorius, naudojamas atgalinės sklaidos metodas [9]. Šio metodu metu, pirmiausiai atsitiktinai sugeneruojami viso tinklo svoriai, tuomet iš apmokymo duomenų imties, parenkamas vienas elementas, kuris pateikiamas neuroniniam tinklui kaip įvesties sluoksnis. Atlikus neuroninio tinklo skaičiavimus, gaunamas išvesties sluoksnis, su pasirinkta klaidos funkcija apskaičiuojama klaidos reikšmė tarp gauto ir norimo rezultato. Tuomet, pasluoksniui, pradedant nuo išvesties sluoksnio, grįžtama atgal ir naudojant atgalinės sklaidos metodą, apskaičiuojami lokalūs gradientai. Pasirinktu optimizavimo metodu, ieškoma tikslo funkcijos optimumo – šiuo atveju, klaidos funkcijos minimumo.

2.1.4.1 Optimizavimo algoritmai

Tyrimo metu, ieškant labiausiai tinkančio optimizavimo algoritmo, išbandomi 7 skirtingi algoritmai:

1. Adam (angl. *Adaptive Moment Estimation*) algoritmas. Stochastinis optimizavimo metodas, kuris naudoja tik pirmos eilės gradientų reikšmes, taip sunaudodamas mažiau atminties resursų. Šis algoritmas, pirmojo ir antrojo gradientų įverčių skirtingiems parametrams, taiko

skirtingus mokymosi greičius. Adam algoritmas, sujungia dviejų kitų optimizavimo algoritmų privalumus: AdaGrad algoritmo, kuris rodo gerus rezultatus retiems gradientams ir RMSProp, kuri rodo gerus rezultatus nestacionariems gradientams [10], [11]. Šis algoritmas, mokymosi greitį, kiekvienam parametrui pritaiko atskirai, todėl ir yra priskiriamas prie adaptyvių metodų. Adam išsaugo eksponentiškai mažėjančius praeities kvadratinius gradientų vidurkius, tačiau skirtingai nei Adadelta. Ir RMSprop, šis algoritmas, taip pat išsaugo ir eksponentiškai mažėjantį praeities gradientų vidurkį [12].

2. Stochastinis gradientinis nusileidimas. Iteracinis optimizavimo metodas, kuris atlieka parametrų atnaujinimą kiekvienam mokymo poaibiui.
3. RMSProp (angl. *Root mean squared propagation*) algoritmas. Gradientinio nusileidimo algoritmas su impulsu. Šis algoritmas padalija mokymosi greitį iš eksponentiškai mažėjančio gradiento kvadratų vidurkio [12].
4. Adadelta (angl. *An Adaptive Learning Rate*) algoritmas. Stochastinis optimizavimo algoritmas, mažinantis monotoniškai mažėjantį mokymosi greitį. Adadelta algoritmas, užuot neefektyviai kaupęs ankstesnius kvadratinius gradientus, rekursiškai susumuoja ankstesnius mažėjančius kvadratinius gradientų vidurkius [12].
5. Adagrad (angl. *Adaptive Gradient Algorithm*) algoritmas. Gradientinio nusileidimo algoritmas, kuris taiko didesnius atnaujinimus retiems parametrams ir mažesnius dažniems parametrams, tokiu būdu, šis metodas skirtingiems parametrams pritaiko skirtingą mokymosi greitį. Šio metodo privalumas yra tai, kad jis rodo gerus rezultatus, retai išsidėsčiusiems duomenims [12].
6. AdaMax (angl. *Adaptive Moment Maximum*) algoritmas. Adam algoritmo plėtinys, kurio metu, metodas apibendrinamas iki begalinės normos (maksimumo) ir gali padėti efektyviau optimizuoti kai kurias problemas [12].
7. Nadam (angl. *The Nesterov-accelerated Adaptive Moment).* Adam algoritmo plėtinys, kurio metu yra pridedamas Nestrovo momentas. Pridėtas Nestrovo momentas, leidžia atlikti tikslesnį žingsnį, pasirenkant gradiento kryptį [12].

2.1.4.2 Bajeso optimizavimas

Tinkamas hiperparametrų įvedimas yra itin svarbus modelio tikslumui. Yra daugybė optimizavimo metodų hiperparametrų nustatymui, pradedant nuo atsitikinės paieškos iki sudėtingesnių metodų. Bajeso optimizavimas, tai globalaus optimizavimo metodas galintis optimizuoti sudėtingas, triukšmingas juodųjų dėžių funkcijas. Literatūroje, šis metodas yra išskiriamas iš kitų paieškos metodų, dėl savo lankstumo ir gebėjimo tirti neapibrėžtas funkcijas [13]. Dažniausiai naudojamas Bajeso optimizavimo metodas paremtas Gauso procesu. Tarkime turime parametrų aibe x ir tikslo funkciją f kurios pavidalo nežinome. Akivaizdu, kad f yra sudėtinga įvertinti, nes nežinome šios funkcijos formos, struktūros ir ypatybių. Taigi galime teigti, kad f yra juodoji dėžė. Tuomet kaip pakaitinis modelis yra naudojamas Gauso procesas. Pakaitinis modelis optimizuojamas tol kol yra patenkinama sustojimo sąlyga arba pasiekiamas konvergavimas. Bajeso optimizavimas kaip ir kiti hiperparametrų paieškos metodai, daug kartų apskaičiuoja modelį su skirtingomis hiperparametrų reikšmėmis, tačiau šio metodo pranašumas yra tai, kad vykdoma paieška nėra atsitiktinė, o paremta ankstesnės informacijos vertinimu, o tai leidžia atrasti modelį su didžiausiu tikslumu, sunaudojant mažiau resursų.

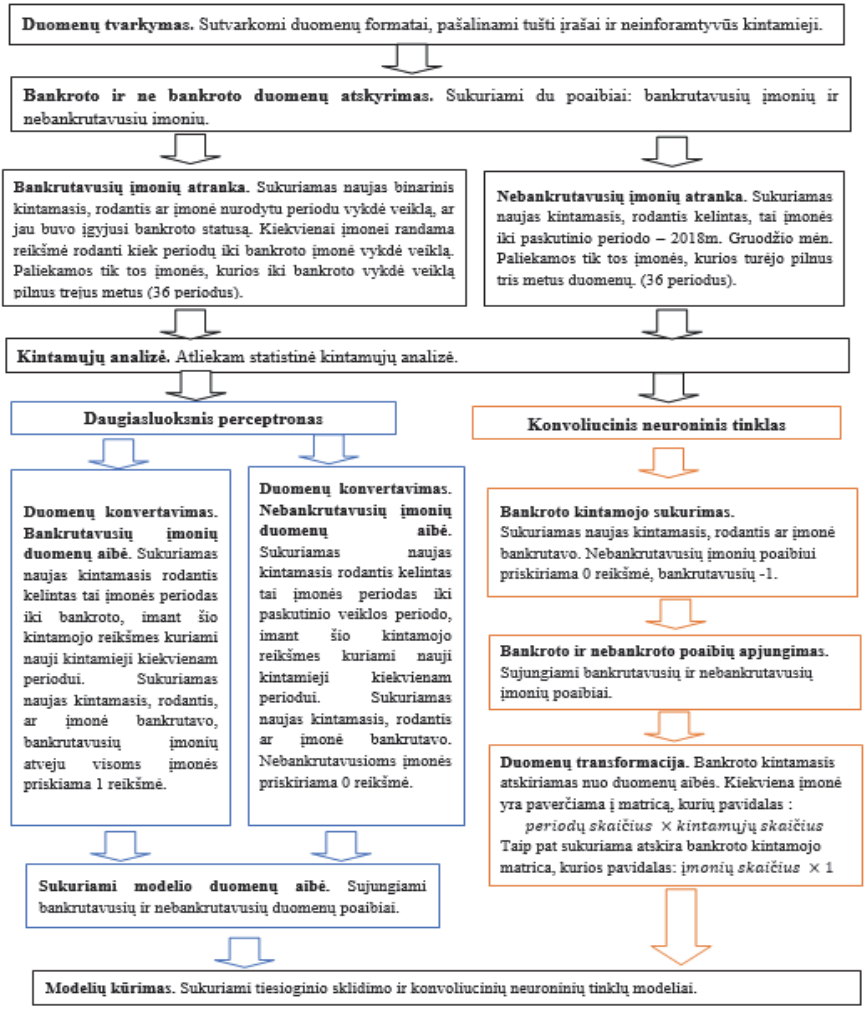
Tyrimas

2.2 Tyrimo planas

Tyrimą sudarė keli etapai, 4 pav. vaizduojamas tyrimo planas.

2.3 Duomenys

Tyrime naudojami duomenys yra gauti iš valstybinės mokesčių inspekcijos ir Sodros. Duomenų rinkinį sudarė visų Lietuvos įmonių duomenys nuo 2013-01-01 iki 2018-12-01 (imtinai), surinkti iš PVM deklaracijų, pelno – nuostolių ataskaitų. Įmonių informacija nebūtinai buvo pateikiama visam nurodytam laikotarpiui, į duomenų rinkinį buvo įtrauktos įmonės ir turėjusios vos keletą mėnesių duomenų. Bendras visų duomenų dydis 8.17 GB, visą duomenų rinkinį sudarė 65 kintamieji. Pirmiausiai buvo atlikta duomenų kokybės analizė ir buvo atsisakyta tokių kintamųjų, kurie nesuteikia reikšmingos informacijos (tušti įrašai, kintamieji įgiję vieną reikšmę visu laikotarpiu). Taip pat buvo pašalintos įmonės, kuriose nagrinėjamu laikotarpiu buvo vykdoma reorganizacija. Atlikus literatūros analizę, buvo pastebėta, kad įmonės bankroto ženklus pradeda rodyti jau dveji metai iki bankroto [15].



4 pav. Tyrimo planas

Tam, kad modeliai galėtų tinkamai atskirti įmones, modelių sudarymui, buvo reikalingos tokios įmonės, kurios vykdė veiklą iki bankroto tokį pat laikotarpį kaip ir nebankrutavusios įmonės. Kadangi pateikti duomenys buvo penkerių metų laikotarpiui, buvo nuspręsta nagrinėti tokias įmones, kurios iki bankroto vykdė veiklą mažiausiai trejus metus. Tam, kad įsitikinti, kad trejų metų pakanka modelio sudarymui, buvo atlikta statistinė kintamųjų analizė, atskyrus bankrutavusias ir nebankrutavusias įmones. Statistinė analizė, buvo atliekama trejiems, dvejiems ir vieneriems metams iki bankroto. Buvo analizuojamos kintamųjų minimumų, maksimumų, standartinių nuokrypių, bei vidurkių reikšmės. Atlikus statistinę analizę, buvo pastebėta, kad bankrutavusių įmonių rodikliai nebuvo stabilūs jau dveji metai iki bankroto, o likus vieniems metams iki bankroto, kai kurie rodikliai rodė ir didelius skirtumus, lyginant su nebankrutavusių įmonių rodikliais.

Atlikus duomenų tvarkymo žingsnius, modelių sudarymui buvo palikti 45 kintamieji. Toliau yra pateikiami metiniai ir mėnesiniai kintamieji ir pagrindinės kintamųjų grupės.

- Bendra įmonės informacija:
  - įmonės amžius – 1 kintamasis;
  - ekonominė veiklos rūšis – 3 kintamieji;
  - geografinė įmonės informacija – 1 kintamasis;
  - teisinis statusas ir kodas – 1 kintamasis;
  - mokesčių mokėtojo informacija.
- Kintamieji iš metinių deklaracijų:
  - įmonių veiklos rodikliai iš pvm deklaracijų – 6 kintamieji;
  - informacija apie darbuotojus – 8 kintamieji;
  - darbo užmokesčio rodikliai – 2 kintamieji;
  - informacija apie įmonių nepriemokas – 2 kintamieji.

2.3.1 Klasių disbalansas.

Pastebėsime, kad nebankrutavusių ir bankrutavusių įmonių klasės buvo išbalansuotos. Bankrutavusių įmonių nagrinėjamoje imtyje buvo 1%, nebankrutavusių 99%.

Kuriant modelį, kuris turės veikti išbalansuotoje duomenų aibėje, svarbu duomenis subalansuoti taip, kad modelis pakankamai apsimokytų atpažinti mažumos klasę. Norint ištestuoti modelio gebėjimą klasifikuoti nesubalan-

suotoje duomenų aibėje, testavimo poaibyje yra tikslinga išlaikyti tokį klasių disbalansą, koks yra pradinėje duomenų imtyje. Todėl prieš duomenų subalansavimą, iš įmonių duomenų rinkinio, atsitiktinai atrenkama 15% įmonių ir sudaromas testavimo poaibis, kurį sudaro 12009 įmonių. Testavimo poaibyje išlaikomas klasių disbalansas – tik 1,3% įmonių turi bankroto statusą. Likę duomenys, naudojant dominuojančios klasės išmetimo metodą (angl. *Undersampling*), subalansuojami, taip, kad duomenų rinkinį sudarytų 15% bankrutavusių ir 85% nebankrutavusių įmonių. Subalansuotas duomenų rinkinys padalijamas į apmokymo ir validavimo poaibius, apmokymo poaibiui priskiriant 85% duomenų, validavimo 15% duomenų. Apmokymo poaibį sudarė 4799 įmonės, validavimo 854.

2.3.2 Duomenų konvertavimas

Tam, kad galėtume duomenis naudoti neuroninių tinklų modeliams, duomenis reikėjo konvertuoti. Daugiasluoksnio perceptrono atveju, buvo kuriami nauji kintamieji, periodams atgal. Atlikus duomenų konvertavimą, buvo gauti apmokymo, validavimo ir testavimo duomenų poaibiai, kuriuos sudarė 960 kintamųjų, šiuo atveju, vienas duomenų rinkinio įrašas atitiko vienos įmonės duomenis. Konvoliucinio neuroninio tinklo atveju, apmokymo, validavimo ir testavimo poaibiai buvo transformuojami į trimatę erdvę, kiekvieną įmonę paverčiant į matricą, kurios pavidalas – 36 × 45 ×1. Po duomenų transformacijos, apmokymo, validavimo ir testavimo duomenų poaibiai turėjo tokį pavidalą: *įmonių skaičius* × 36 × 45 × 1.

2.3.3 Modelių sudarymas

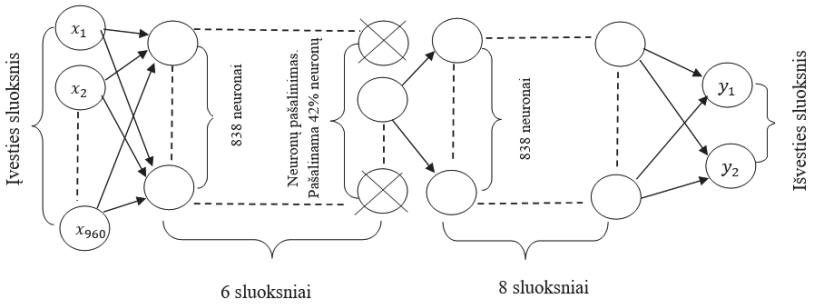
Konkrečiam uždaviniui spręsti tam, kad rasti geriausiai tinkantį neuroninio tinklo modelį, reikia rasti labiausiai tinkančią tinklo architektūrą ir tinkamiausius modelio hiperparametrus. Sudarius apmokymo, testavimo ir validavimo duomenų poaibius, kuriami neuroninių tinklų modeliai. Modelių apmokymui naudojami apmokymo, hiperparametrų validavimui – validavimo, modelių testavimui – testavimo duomenys. Geriausios neuroninio tinklo architektūros paieškai yra sukuriama funkcija, kuri sudaro neuroninio tinklo modelį. Ieškant geriausios tinklo architektūros, sukurta funkcija yra perduodama Bajeso optimizavimo metodui. Bajeso optimizavimo metu yra ieškoma maksimalaus modelio tikslumo – validavimo metu. Daugiasluoksniui perceptronui tiriami šie hiperparametrai: sluoksnių skaičius, neuronų

skaičius, aktyvacijos funkcija, optimizavimo algoritmas, mokymosi greitis, neuronų išmetimo sluoksnis ir išmetamų neuronų dydis, epochų skaičius, sluoksnių normalizavimas. Konvoliuciniam neuroniniam tinklui tiriami šie hiperparametrai: konvoliucinių sluoksnių skaičius, filtrų dydis, filtrų skaičius, aktyvacijos funkcija, neuronų išmetimo sluoksnis ir išmetimo dydis, neuronų skaičius, optimizavimo algoritmas. Daugiasluoksniui perceptronui išvesties sluoksnyje naudojama sigmoidinė aktyvacijos funkcija, konvoliucinio neuroninio tinklo modeliui – softmax funkcija.

3 Rezultatai

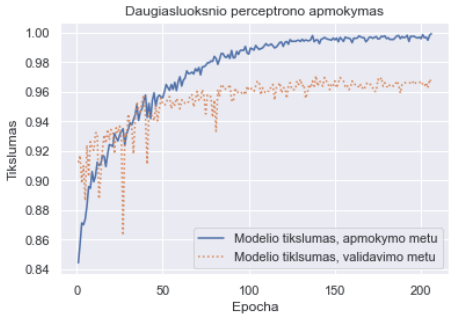
3.1 Daugiasluoksnis perceptronas

Atlikus hiperparametrų paiešką daugiasluoksniui perceptronui, buvo gautas neuroninis tinklas sudarytas iš 14 paslėptųjų sluoksnių. Paslėptuosius sluoksnius sudarė 838 neuronai, šeštame sluoksnyje pašalinama 42% neuronų. 5 pav. pateikiama daugiasluoksnio perceptrono modelio schema.

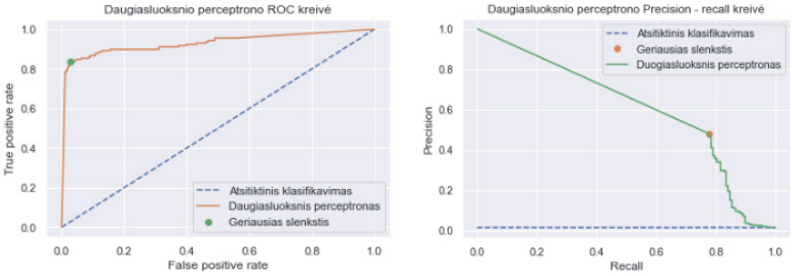


5 pav. Daugiasluoksnio perceptrono modelis

Gauto daugiasluoksnio perceptrono modelio 5 pav., paslėptuosiuose sluoksniuose buvo taikoma selu aktyvacijos funkcija. Modelio apmokymo metu taikomas Adadelta optimizavimo algoritmas. Apmokymo metu duomenys į priekį ir atgal buvo perduodami 206 kartus (epochų skaičius). Apmokymo metu gautas 99.9% modelio tikslumas, validavimo metu – 96.6%, testavimo metu – 97%. Toliau yra pateikiamas modelio apmokymo grafikas ir ROC, PRC kreivės.



6 pav. Daugiasluoksnio perceptrono apmokymas.

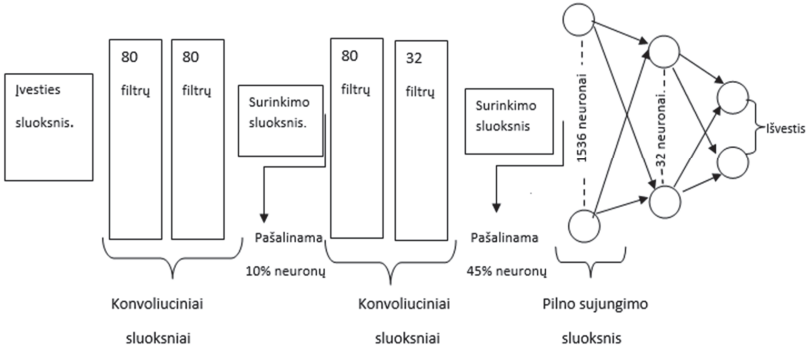


7 pav. Daugiasluoksnio perceptrono ROC ir PRC kreivės.

Iš gautų ROC ir PRC kreivių 7pav., galime teigti, kad modelis geba atskirti klases. Pastebėsime, kad gautas pakankamai geras AUC įvertis, tačiau, kadangi modelis testuojamas nesubalansuotoje duomenų aibėje, aukštas AUC įvertis gali būti gaunamas, dėl gero daugumos klasės klasifikavimo, neatsižvelgiant į prastą mažumos klasės klasifikavimą, tačiau iš gauto F1 įvertčio, kuris nėra labai prastas ir PRC kreivės, galime teigti, kad modelis geba prognozuoti bankroto atvejus.

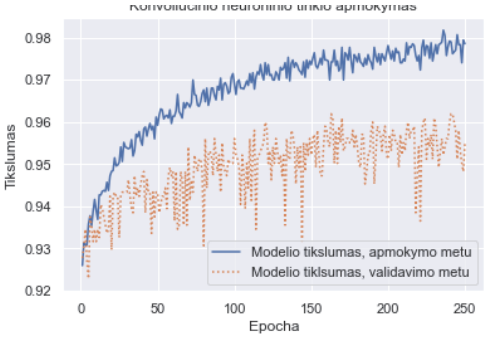
3.2 Konvoliucinis neuroninis tinklas

Atlikus hiperparametrų paiešką konvoliucinio neuroninio tinklo atveju, buvo gautas neuroninis tinklas sudarytas iš 8 paslėptųjų sluoksnių. Toliau pateiksime gautą konvoliucinio neuroninio tinklo schemą.

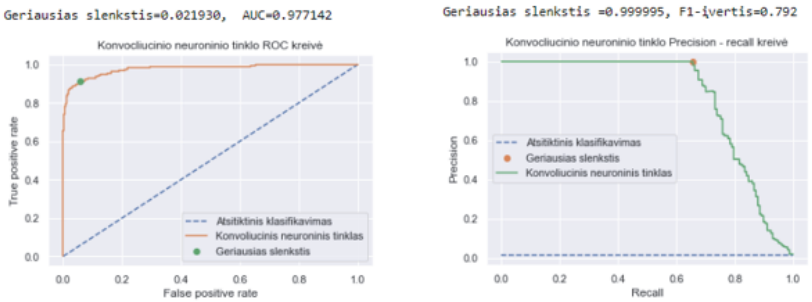


8 pav. Konvoliucinio neuroninio tinklo schema.

8 pav. pateikto modelio konvoliuciniuose sluoksniuose, buvo naudojama softsign aktyvacijos funkcija ir 3 × 3 filtrai. Surinkimo sluoksniuose, naudojama maksimumo surinkimo funkcija, o surinkimo žingsnis lygus 2. Pilno sujungimo sluoksnyje, naudojama elu aktyvacijos funkcija, o išvesties sluoksnyje softmax funkcija. Konvoliucinio tinklo apmokymui, naudojamas RMSprop optimizavimo algoritmas. Apmokymo metu duomenys buvo perduodi į priekį ir atgal 250 kartų. Apmokymo metu gautas 98.2% modelio tikslumas, validavimo metu – 96.3%, testavimo metu – 99%. Toliau pateikiama modelio apmokymo grafikas ir modelio ROC, PRC kreivės.



9 pav. Konvoliucinio neuroninio tinklo apmokymas.



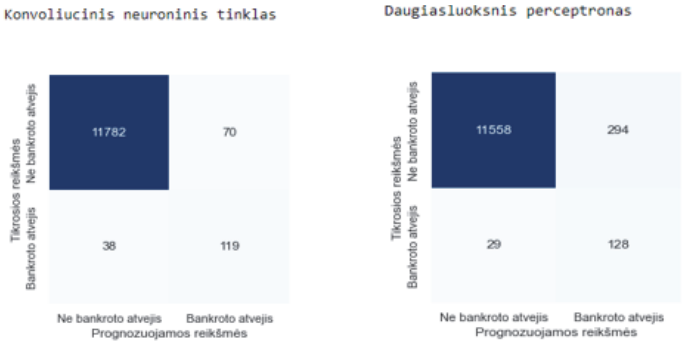
10 pav. Konvoliucinio neuroninio tinklo ROC ir PRC kreivės.

Iš gautų ROC ir PRC kreivių 10 pav., galime teigti, kad modelis gerai atskiria klases. Pastebėsime, kad gautas AUC įvertis yra labai arti idealaus klasifikavimo, taip pat pakankamai geras gaunamas ir F1 įvertis, vertinantis modelio gebėjimą klasifikuoti mažumos klasę.

3.3 Modelių palyginimas.

Norėdami palyginti modelių sumaišymo matricas ir klasifikavimo įverčius, klasifikavimui turime naudoti tą patį slenkstį. Iš gautų PRC ir ROC kreivių, pastebėjome, kad vertinant siūlomus slenksčius giliojo mokymosi modeliams, ROC kreivėse, gaunami slenksčiai yra pakankamai maži, tuo tarpu PRC kreivėse pakankamai dideli ir yra artimi maksimaliam galimam slenksčiui (vienetui). Didesnis slenkstis, mažintų prognozuojamų bankroto atvejų skaičių, o mūsų atveju, mums yra svarbiau identifikuoti bankroto atvejus, todėl yra geriau daugiau įmonių, kurioms negresia bankrotas, priskirti bankroto klasei, nei nepriskirti įmonių kurioms gresia bankrotas - bankroto klasei. Tačiau taip pat yra svarbu ir gerai atskirti nebankrutavusias įmones. Atsižvelgiant į tai, nusprendėme modelių klasifikavimo įverčių palyginimui, naudoti klasifikavimo slenkstį – 0.5. Vadinasi, įmonės, kurioms buvo gauta bankroto tikimybė didesnė nei 0.5, buvo priskiriamos bankroto klasei.

Iš 11 pav. pateiktų sumaišymo matricų, pastebėsime, kad daugiasluoksnis perceptronas teisingai bankroto klasei priskyrė 9 įmonės daugiau, lyginant su konvoliuciniu neuroniniu tinklu. Tačiau, konvoliucinis neuroninis tinklas, klaidingai bankroto klasei priskyrė 4 kartus mažiau įmonių, lyginant su daugiasluoksniu perceptronu.



11 pav. Modelių sumaišymo matricos

1 lentelė. Modelių klasifikavimo įverčiai.

Modelis	Klasė	Precision	Recall	f1-score	Accuracy
Daugiasluoksnis perceptronas	Ne Bankroto	1	0.98	0.99	0.97
	Bankroto	0.3	0.82	0.44	
Konvoliucinis neuroninis tinklas	Ne Bankroto	1	0.99	1	0.99
	Bankroto	0.63	0.76	0.69	

Iš 1 lentelėje pateiktų precision įverčių, matome, kad modeliai puikiai prognozavo ne bankroto klasę. Vertinant, giliojo mokymosi metodais, prognozuojamus bankroto atvejus, galime teigti, kad 30% daugiasluoksnio perceptrono bankroto prognozių buvo teisingos, tuo tarpu konvoliucinio neuroninio tinklo atveju 63% prognozių buvo teisingos. Vertinant Recall įvertį, galime pastebėti, kad daugiasluoksnis perceptronas, ne bankroto klasei priskyrė 98% visų nebankroto atvejų, tuo tarpu konvoliucinis neurononinis tinklas – 99%. Vertinant kiek visų bankroto atvejų buvo priskirtų bankroto klasei, matome, kad daugiasluoksnis perceptrontas teisingai priskyrė 82% įmonių, konvoliucinis neuroninis tinklas 76%. Vertinant f1 įvertį, kuris yra precision ir recall harmoninis vidurkis, galime teigti, kad konvoliucinis neuroninis tinklas klases atskyrė geriau. Vertinant bendrus modelių tikslumus, matome, kad konvoliucinio neuroninio tinklo bendras tikslumas buvo šiek tiek geresnis.

4 Išvados

Tyrimo metu sukurti giliųjų neuroninių tinklų modeliai, parodė gerus prognozavimo rezultatus. Gautų modelių bendri klasifikavimo tikslumai, tiek apmokymo, tiek testavimo metu buvo didesni nei 96%. Vertinant ROC ir PRC kreives, bei gautus klasifikavimo įverčius, galime teigti, kad bendrai įmonių bankrotą geriau prognozavo konvoliucinis neuroninis tinklas. Gautos sumaišymo matricos ir Recall įverčiai, rodo, tai, kad daugiasluoksnis perceptronas geriau atskyrė bankroto atvejus, bankroto klasei, tačiau daug daugiau klydo, atskirdamas bankroto atvejus, ne bankroto klasėje, lyginant su konvoliuciniu neuroniniu tinklu.

Bendrai paėmus, galime daryti išvadą, kad išbandyti giliojo mokymosi modeliai yra tinkami įmonių bankrotui prognozuoti ir geba atpažinti bankroto atvejus - duomenų aibėje su dideliu klasių disbalansu.

Literatūra

[1] Altman E. (1968). Financial ratios discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance 23(4) 589–609."

[2] Ohlson J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. Journal of Accounting Research. (internetinė prieiga : Financial Ratios and the Probabilistic Prediction of Bankruptcy on JSTOR).

[3] Shekar Shetty Mohamed Musa Xavier Brédart "Bankruptcy Prediction Using Machine Learning Techniques".

[4] Tadaaki Hosaka, Bankruptcy prediction using imaged financial ratios and convolutional neural networks, Expert Systems with Applications, Volume 117, 2019,Pages 287-299,ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2018.09.039

[5] Keiron O'Shea, Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. https://doi.org/10.48550/arXiv.1511.08458

[6] Azzouni, Abdelhadi & Boutaba, R. & Pujolle, Guy. (2017). NeuRoute: Predictive Dynamic Routing for Software-Defined Networks.

[7] Clevert, Djork-Arné & Unterthiner, Thomas & Hochreiter, Sepp. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). Under Review of ICLR2016 (1997).

[8] Nguyen, A., Khoa Pham, Dat Thanh Ngo, Thanh Ngo and Lam Dang Pham. "An Analysis of State-of-the-art Activation Functions For Supervised Deep Neural Network." 2021 International Conference on System Science and Engineering (ICSSE) (2021): 215-220.

[9] Cilimkovic, Mirza. "Neural Networks And Back Propagation Algorithm." (2010).

[10] Kingma, Diederik P. and Ba, Jimmy. „Adam: A Method for Stochastic Optimization.." CoRR abs/1412.6980 (2014).

[11] Diederik P. Kingma, Jimmy Ba. 2017. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. https://doi.org/10.48550/arXiv.1412.6980

[12] Ruder, Sebastian. (2016). An overview of gradient descent optimization algorithms.

[13] Jasper Snoek Oren Rippel Kevin Swersky Ryan Kiros Nadathur Satish Narayanan Sundaram Md. Mostofa Ali Patwary Prabhat Ryan P. Adams (2015). Scalable Bayesian Optimization Using Deep Neural Networks. the Natural Sciences and Engineering Research Council of Canada. NSF IIS-1421780.
[14] Flavio Barboza, Herbert Kimura, Edward Altman, Machine learning models and bankruptcy prediction, Expert Systems with Applications, Volume 83, 2017, Pages 405-417, ISSN 0957-4174.
[15] Becerra-Vicario, Rafael & Alaminos, David & Llamas, Eva & Fernández-Gámez, Manuel. (2020). Deep Recurrent Convolutional Neural Network for Bankruptcy Prediction: A Case of the Restaurant Industry. Sustainability. 12. 5180. 10.3390/su12125180.