

**VILNIAUS UNIVERSITETO
KAUNO FAKULTETAS**

**SOCIALINIŲ MOKSLŲ IR TAIKOMOSIOS INFORMATIKOS
INSTITUTAS**

Finansų technologijų studijų programa

Kodas 6211BX022

JOKŪBAS KETURAKIS

MAGISTRO BAIGIAMASIS DARBAS

**GAUTINŲ SUMŲ VIDINIO AUDITO ROBOTINIS
AUTOMATIZAVIMO METODAS**

Kaunas 2025

**VILNIAUS UNIVERSITETO
KAUNO FAKULTETAS**

**SOCIALINIŲ MOKSLŲ IR TAIKOMOSIOS INFORMATIKOS
INSTITUTAS**

JOKŪBAS KETURAKIS

MAGISTRO BAIGIAMASIS DARBAS

**GAUTINŲ SUMŲ VIDINIO AUDITO ROBOTINIS
AUTOMATIZAVIMO METODAS**

Magistrantas _____

(parašas)

Darbo vadovas _____

(parašas)

_____ part. prof. dr. Darius Dilijonas _____

(darbo vadovo mokslo laipsnis, mokslo
pedagoginis vardas, vardas ir pavardė)

Darbo įteikimo data

Registracijos Nr.

Kaunas 2025

TURINYS

SANTRUMPŲ SĄRAŠAS	5
LENTELIŲ SĄRAŠAS.....	6
PAVEIKSLŲ SĄRAŠAS.....	6
SANTRAUKA (užsienio kalba).....	7
IVADAS.....	8
1. RPA TECHNOLOGIJOS TAIKYMO AUDITO PROCESUOSE TEORINIAI ASPEKTAI	14
1.1. RPA technologijos apžvalga	14
1.2. RPA integravimas į audito procesus	16
1.3. RPA privalumai audito srityje.....	17
1.4. RPA pritaikymo iššūkiai ir apribojimai audito srityje	18
1.5. RPA taikymo sistema audito procesuose	19
1.6. RPA taikymo sistema audito procesuose	20
1.7. Duomenų analitika ir elgsenos modeliai vidiniame audite	22
1.8. RPA taikymas gautinų sumų ir pajamų cikle	24
1.9. RPA ir vidinio audito vaidmenys, kompetencijos ir valdymas	25
2. TYRIMO METODIKA IR EPDA METODO MODELIS	28
2.1. Tyrimo objektas, tikslas, uždaviniai ir ribotumai.....	28
2.2. Naudojamas duomenų rinkinys („Payment Date Prediction“ duomenys)	29
2.3. Numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodo skaičiavimo algoritmas.....	32
2.4. Tradicinis sąskaitų senaties metodas ir lyginamosios metrikos	35
2.5. Tyrimo eiga ir eksperimento schema (.NET ir Power Automate Desktop prototipų palyginimas)	36
2.6. Tyrimo metodai ir vertinimo kriterijai	39
3. EPDA METODO REALIZACIJA .NET IR RPA PROTOTIPUOSE	41
3.1. .NET prototipo architektūra ir funkcionalumas	41
3.2. „Power Automate Desktop“ (RPA) prototipo architektūra ir funkcionalumas.....	46
3.3. Tradicinės sąskaitų senaties ir EPDA metodo palyginimai	51
3.3.1. Praktinis palyginimo pavyzdys	51
3.3.2. Tradicinės sąskaitų senaties ir EPDA rezultatų palyginimas visai imčiai	57
3.4. .NET ir RPA prototipų palyginimas.....	58
3.4.1. Matavimų metodika.....	58
3.4.2. „PerfMon“ rezultatų suvestinė	59

3.4.3. Normalizuotas vykdymo laikas ir interpretacija	60
3.5. Skyriaus apibendrinimas	62
IŠVADOS.....	65
LITERATŪROS SĄRAŠAS.....	67

SANTRUMPŲ SĄRAŠAS

RPA - robotinis procesų automatizavimas.

EPDA – numatomo ir faktinio mokėjimo vėlavimo metodas (angl. *Expected Payment Delay from Actual*), naudojamas kliento mokėjimo elgsenai įvertinti.

DSO – pardavimo dienos gautinose sumose (angl. *Days Sales Outstanding*), rodiklis, parodantis, per kiek vidutiniškai dienų įmonė surenka pajamas iš pardavimų kreditan.

.NET – „.NET“ programavimo platforma, Microsoft kuriama programinės įrangos kūrimo aplinka, skirta taikomosioms programoms kurti.

PAD – “Power Automate Desktop”, robotinio procesų automatizavimo (RPA) įrankis darbalaukio procesams automatizuoti.

FinTech – finansinės technologijos (angl. *Financial Technology*), apimančios skaitmeninius sprendimus finansų paslaugoms teikti ir valdyti.

IT – informacinės technologijos (angl. *Information Technology*), t.y. kompiuterinės ir programinės įrangos bei tinklų visuma, naudojama duomenims apdoroti ir saugoti.

CSV – kableliais atskirtų reikšmių (angl. *Comma – Separated Values*) tekstinis duomenų failo formatas, naudojamas lentelinėms struktūroms saugoti.

CPU – centrinis procesorius (angl. *Central Processing Unit*), pagrindinis kompiuterio skaičiavimų vykdymo įrenginys.

IO – įvesties – išvesties (angl. *Input/Output*) operacijos, apimančios duomenų skaitymą ir rašymą į diską ar kitus įrenginius.

MB – megabaitas, duomenų apimtys matavimo vienetas, naudojamas atminčiai ir failų dydžiui nusakyti.

ID – identifikatorius (pvz., kliento ar sąskaitos ID), unikalus objekto žymuo duomenų rinkinyje.

JAV – Jungtinės Amerikos Valstijos.

LINQ – integruotų užklausų kalba (angl. *Language Integrated Query*) .NET platformoje, leidžianti patogiai vykdyti užklausas į duomenų rinkinius programos kode.

PerfMon – „Windows Performance Monitor“ (PerfMon), sistemos našumo stebėsenos įrankis, naudojamas procesoriaus, atminties ir disko apkrovai matuoti.

n8n – atvirojo kodo automatizavimo ir darbo eigos (workflow) orkestravimo įrankis, leidžiantis kurti automatizuotus procesus be tradicinio programavimo.

LENTELIŲ SĄRAŠAS

1 lentelė Uždarytų sąskaitų paskirstymas senaties krepšelių metodu ir EPDA metodu.....	57
2 lentelė .NET prototipo našumo rodikliai.....	59
3 lentelė RPA („Power Automate Desktop“) prototipo našumo rodikliai	60
4 lentelė Normalizuotas vykdymo laikas vienam EPDA logikos paskaičiavimui 1 000 sąskaitų	61

PAVEIKSLŲ SĄRAŠAS

1 pav. Procentinė RPA rinkos dalis pagal regionus 2023 metais.....	15
2 pav. Kaggle duomenų rinkinio fragmentas	29
3 pav. Siūlomo EPDA metodo skaičiavimo veiklos diagrama.....	34
4 pav. Tyrimo eiga ir EPDA eksperimento schema (.NET ir „Power Automate Desktop“ prototipų palyginimas).....	38
5 pav. .NET prototipo EPDA skaičiavimo kodo ištrauka.....	43
6 pav. „Power Automate Desktop“ EPDA skaičiavimo logika (1 dalis).....	48
7 pav. „Power Automate Desktop“ EPDA skaičiavimo logika (2 dalis).....	50

SANTRAUKA (užsienio kalba)

Keturakis, Jokūbas (2025). *Robotic automation method for internal audit of receivables*. MBA Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty, Institute of Social Sciences and Applied Informatics. 73 p.

SUMMARY

The aim of this thesis is to develop and empirically evaluate an internal audit method for trade receivables based on the Expected Payment Delay from Actual (EPDA) indicator and to compare two technological implementations of this method – a custom .NET application and a Power Automate Desktop (RPA) prototype. To achieve this aim, the thesis analyses literature on receivables auditing, traditional aging and payment behaviour models, and the use of RPA in audit; defines the research problem, object, aim and tasks; and designs a conceptual EPDA model using a Kaggle invoices dataset. The empirical research involves cleaning and preparing the dataset (48,580 invoices), calculating EPDA and traditional aging buckets, implementing two functionally equivalent EPDA prototypes in .NET and Power Automate Desktop, and experimentally comparing them on several dataset sizes (1k, 10k, 25k, 48k records). Performance metrics (execution time, CPU, memory and I/O usage) are measured with Windows Performance Monitor and normalised per one EPDA run per 1,000 invoices.

The results show that EPDA provides additional behavioural insight compared to traditional aging, as it measures each invoice's deviation from the customer's typical payment delay and thus reveals behavioural changes and “hidden” risk in invoices that appear low-risk in standard aging buckets. Heavily overdue invoices remain high-risk under both methods, but EPDA additionally highlights a share of invoices in the 0–30 days past due bucket as behavioural outliers, which is important for audit risk assessment and sampling. The technological comparison confirms that both prototypes generate consistent EPDA results, while the .NET solution is several orders of magnitude faster and more resource-efficient, making it suitable as the core computation engine. Power Automate Desktop is more appropriate as a low-code orchestration and integration layer for automating data extraction and reporting. The thesis concludes that EPDA should be used as a complementary behavioural metric alongside traditional aging and DSO in receivables audit and that combining high-performance calculation platforms with RPA-based workflows is a promising direction for FinTech-oriented internal audit automation.

The work is comprised of 69 pages, contains 4 tables and 7 figures.

IVADAS

Temos aktualumas

Šiuolaikinėje verslo aplinkoje finansiniai srautai tampa vis sudėtingesni, o jų valdymas - vis labiau priklausomas nuo didelių duomenų apimčių ir skaitmeninių sistemų integracijos. Gautinos sumos yra viena iš pagrindinių įmonių turto sudedamųjų dalių, turinčių tiesioginę įtaką likvidumui, pinigų srautų planavimui ir kredito rizikai. Todėl vidinis auditas, vertindamas gautinų sumų kokybę ir valdymo procesus, tampa svarbia priemone organizacijų finansiniam stabilumui užtikrinti.

Tradicinėje praktikoje gautinų sumų analizė dažniausiai grindžiama sąskaitų senaties (angl. *accounts receivable aging*) ataskaitomis, kuriose neapmokėtos sąskaitos suskirstomos pagal tai, kiek dienų jos yra pradelstos. Tokiose ataskaitose sumos paprastai grupuojamos į 0–30, 31–60, 61–90 ir ilgesnio nei 90 dienų vėlavimo intervalus, siekiant įvertinti, „kiek laiko sąskaita yra pradelsta“ ir identifikuoti rizikingesnes gautinas sumas Upflow blog (2024). Tokie senaties raportai plačiai naudojami kaip pagrindinis įrankis pinigų srautų valdymui, kredito rizikos vertinimui ir probleminių klientų identifikavimui Zone and Co (2025).

Nors sąskaitų senaties metodas yra paprastas ir gerai suprantamas, jis remiasi tik „dienų pradelstumo“ matu ir neatsižvelgia į konkretaus kliento elgsenos istoriją. Tai reiškia, kad dvi sąskaitos, pradelstos, pavyzdžiui, 45 dienas, tradiciniame senaties raporte bus vertinamos panašiai, nepaisant to, ar klientas istoriškai visuomet vėluodavo keliomis savaitėmis, ar paprastai apmokėdavo sąskaitas laiku. Tokia agreguota perspektyva riboja galimybes anksti identifikuoti elgsenos pokyčius ir potencialius nuostolius, o vidiniam auditui apsunkina tikslinį rizikos vertinimą atskirų klientų ar sąskaitų lygmeniu Tabs platform (2025).

Tuo pačiu metu finansų technologijų ir duomenų analitikos pažanga sudaro prielaidas gautinų sumų valdymą ir susijusias analitines procedūras grįsti kliento elgsenos modeliais. Naujesni tyrimai rodo, kad, remiantis istorine sąskaitų ir mokėjimų informacija, galima prognozuoti, kada tikėtina, jog konkreti sąskaita bus apmokėta, ir šias prognozes panaudoti operatyvių sprendimų priėmimui kredito ir išieškojimo procesuose. Pavyzdžiui, Appel, Oliveira, Malfatti, De Santana, ir De Paula (2019), taikydami mašininio mokymosi metodus ir bendradarbiaudami su tarptautiniu banku, sukūrė prototipą, padedantį prognozuoti sąskaitų apmokėjimo laiką; tyrimo rezultatai parodė, kad toks modelis gali pasiekti iki 77–81 % prognozių tikslumą ir reikšmingai pagerinti klientų prioritetizavimą bei inkasavimo veiklos

efektyvumą. Vis dėlto šie sprendimai yra orientuoti į gautinų sumų administravimą ir pinigų srautų optimizavimą, o ne į vidinio audito analitinių procedūrų kūrimą ar elgsenos pagrindu grindžiamų rodiklių (tokių kaip numatomo ir faktinio mokėjimo vėlavimo palyginimas) realizavimą skirtingose technologinėse platformose.

Šie darbai rodo, kad elgsenos pagrindu grindžiami mokėjimų prognozavimo metodai gali suteikti gerokai daugiau informacijos nei vien tik sąskaitų senaties ataskaitos. Tačiau dauguma egzistuojančių sprendimų orientuoti į kredito valdymą ir kasdienės finansų funkcijas, o ne į vidinio audito analitines procedūras. Tuo tarpu vidinis auditas, ypač taikant duomenų analitiką ir automatizavimą, vis aktyviau ieško būdų naudoti naujus rodiklius ir metodus, kurie leistų objektyviau vertinti finansinių procesų kokybę ir riziką. AuditBoard (2022) hyperautomatizacijos pavyzdžiu grindžiami tyrimai parodo, kad audito komandos gali išnaudoti skaitmeninių technologijų derinį (duomenų analitiką, RPA, dirbtinį intelektą) pajamų ir gautinų sumų ciklo rizikoms identifikuoti ir sukčiavimo schemoms aptikti.

Robotinio procesų automatizavimo (RPA) technologijos šioje transformacijoje užima ypatingą vietą. Kaip pažymi FloQast (2021), RPA robotai gali automatizuoti didelių duomenų imčių analizes – sistemingai lyginti duomenų rinkinius, skaičiuoti variacijas ir rodiklius, generuoti grafikus ir tendencijų linijas, taip palengvindami audito analitinių procedūrų vykdymą. Moffitt, Rozario, ir Vasarhelyi (2018) literatūroje RPA įvardijama kaip vienas iš svarbių vidinio audito skaitmenizavimo įrankių, galinčių sumažinti rankinį darbą, padidinti audito aprėptį ir leisti auditoriams daugiau dėmesio skirti aukštesnio lygio analizei.

Atsižvelgiant į šias tendencijas, aktualu tirti, kaip vidiniame audite būtų galima taikyti numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodą, pagrįstą kliento istorine elgsena, ir kaip tokį metodą efektyviausiai realizuoti skirtingomis technologijomis – tiek tradicinės .NET aplikacijos, tiek RPA („Power Automate Desktop“) prototipo pavidalu. Toks tyrimas yra aktualus tiek moksliniu požiūriu (plečiant elgsenos analitika grįstų rodiklių taikymą audite), tiek praktiniu požiūriu – ieškant naujų būdų, kaip automatizuoti gautinų sumų audito procedūras ir pagerinti rizikos vertinimo kokybę, palyginti su tradiciniu sąskaitų senaties skirstymu į „aging buckets“.

Problemos ištyrimo lygis

Gautinų sumų kokybės vertinimas ir susijusi kredito rizika ilgą laiką buvo grindžiami tradiciniais apskaitos metodais. Upflow blog (2024) šaltinyje, sąskaitų senaties ataskaitos aprašomos kaip periodinės ataskaitos, kuriose pirkėjų skolos suskirstomos pagal laiką, kiek sąskaita yra išlaikyta neapmokėta; jos tapo standartiniu įrankiu, padedančiu stebėti pradelstas sumas, identifikuoti pavėluotus mokėjimus ir taikyti skirtingas išieškojimo strategijas priklausomai nuo pradelstumo trukmės. Tokios ataskaitos plačiai taikomos ir vidiniame audite kaip bazinė priemonė vertinant gautinų sumų portfelio struktūrą, tačiau jos nesuteikia informacijos apie tai, ar konkretaus kliento elgsena pasikeitė, lyginant su ankstesniais laikotarpiais, ir nepadaeda prognozuoti, kada tikėtina, kad sąskaita bus apmokėta.

Pastaraisiais metais mokslinėse studijose vis daugiau dėmesio skiriama sąskaitų apmokėjimo prognozavimui, taikant mašininio mokymosi ir statistinius metodus. Appel et al. (2019) pasiūlė mokymosi modelį, leidžiantį numatyti, kada sąskaitos bus apmokėtos, remiantis istorine gautinų sumų ir mokėjimų informacija; jų sukurtas prototipas, testuotas kartu su tarptautiniu banku, pasiekė iki 81 % prognozių tikslumą ir, simuliacijų duomenimis, galėtų sutaupyti reikšmingą dalį sąnaudų, optimizuojant išieškojimo veiklą. Schoonbee, Moore ir Van Vuuren (2022) atliko tyrimą, kuriame mašininio mokymosi modeliu siekė įvertinti, per kurį vėlavimo intervalą (pvz., 1 - 30, 31 - 60 ar daugiau nei 60 dienų) pateks sąskaita, ir pademonstravo, kad į modelį įtraukus iš istorinių duomenų gautus kliento mokėjimo elgsenos požymius, prognozavimo tikslumas reikšmingai pagerėjo, lyginant su paprastesniais metodais.

Be akademinų tyrimų, elgsenos pagrindu grįstas mokėjimų prognozavimas imamas plačiai taikyti ir komercinėse gautinų sumų valdymo sistemose. Įvairūs tiekėjai (pvz., „Invoiced“, „Billtrust“, „HighRadius“ ir kt.) savo produktuose siūlo funkcionalumą, kuris, naudodamas istorinius klientų mokėjimo duomenis, prognozuoja tikėtiną sąskaitų apmokėjimo laiką ir padeda finansų komandoms planuoti pinigų srautus bei prioritetizuoti išieškojimą kaip apžvelgia savo darbe Peiris, ir Premaratne (2025). Šios sistemos daugiausia orientuotos į operatyvų kredito valdymą ir kasdienį gautinų sumų administravimą, o ne į vidinio audito poreikius, tačiau rodo aiškią tendenciją – nuo statinių senaties ataskaitų pereinama prie elgsenos duomenimis grįstų prognozavimo modelių.

Kita tyrimų kryptis - robotinio procesų automatizavimo taikymas apskaitoje ir audite. AIMultiple (2025) RPA apžvalgos parodo, kad automatizuoti robotai gali perimti pasikartojančias užduotis, tokias kaip sąskaitų ir ataskaitų atsisiuntimas, duomenų sutikrinimas, išimčių paieška ir ataskaitų generavimas, taip sumažindami rankinį darbą ir klaidų tikimybę. FloQast (2021) pabrėžia, kad RPA gali būti taikomas ir analitinėms procedūroms – sistemingai lyginti duomenų rinkinius, ieškoti išskirčių ir skaičiuoti rodiklius, kas ypač aktualu vidiniam auditui, kai siekiama išplėsti tikrinamų operacijų apimtį nepadidinant auditoriaus darbo sąnaudų. Tyrimai, atlikti Kassar, ir Jizi (2025), analizuojantys dirbtinio

intelekto ir RPA sąveiką audito veikloje, akcentuoja, kad vidinio audito funkcija turi gebėti ne tik audituoti automatizuotus procesus, bet ir pati išnaudoti tokias technologijas efektyvesnėms analitinėms procedūroms.

Tačiau nepaisant to, kad:

- sąskaitų senaties metodas yra plačiai taikomas gautinų sumų kokybei ir rizikai vertinti,
- mokėjimų prognozavimo modeliai parodė didelį potencialą tiksliau numatyti apmokėjimo laiką,
- RPA sprendimai sėkmingai naudojami finansinių procesų automatizavimui,

mokslo literatūroje trūksta darbų, kurie integruotų šias kryptis vidinio audito kontekste. Ypač stokojama tyrimų, kurie:

- (1) formalizuotų elgsenos pagrindu grįstą numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodą kaip alternatyvą tradiciniam sąskaitų senaties skirstymui audito analitinėse procedūrose;
- (2) eksperimentiškai palygintų skirtingų technologinių platformų - tradicinės .NET aplikacijos ir RPA (pvz., „Power Automate Desktop“) - tinkamumą tokio metodo realizavimui, vertinant ne tik skaičiavimo rezultatus, bet ir našumą, lankstumą bei pritaikomumą audito praktikoje.

Šis tyrimų trūkumas sudaro pagrindą darbo problemai, kuri šiame magistro darbe formuluojama kaip klausimas, kaip vidiniame audite taikyti EPDA metodą, pasitelkiant skirtingus technologinius sprendimus, kad gautinų sumų rizika būtų vertinama patikimiau nei naudojant vien tradicinę sąskaitų senaties analizę.

Darbo problema - kaip vidiniame audite taikyti numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodą, pasitelkiant skirtingus technologinius sprendimus (.NET programėlę ir RPA „Power Automate Desktop“ prototipą), kad gautinų sumų kokybė būtų vertinama patikimiau nei naudojant tradicinį sąskaitų senaties modelį?

Darbo objektas – gautinų sumų audito metu taikomi numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodo technologiniai sprendimai.

Darbo tikslas - ištirti numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodo taikymo galimybes gautinų sumų vidiniame audite ir sukurti bei eksperimentiškai palyginti du šio metodo realizacijos prototipus - .NET programėlę ir RPA „Power Automate Desktop“ sprendimą.

Darbo uždaviniai:

1. Išnagrinėti mokslinę literatūrą ir praktines rekomendacijas apie gautinų sumų auditą, tradicinį sąskaitų senaties (aging buckets) metodą ir mokėjimų vėlavimo modeliavimą bei rodiklius.
2. Apibendrinti .NET ir RPA („Power Automate Desktop“) technologijų taikymo galimybes finansinių procesų ir vidinio audito automatizavimui, identifikuojant jų privalumus ir apribojimus kuriant audito prototipus.
3. Sukurti ir pagrįsti EPDA metodo taikymo gautinų sumų audite konceptualų modelį ir skaičiavimo algoritmą, remiantis viešai prieinamu sąskaitų ir mokėjimų duomenų rinkiniu.
4. Sukurti du EPDA metodo realizacijos prototipus - .NET programėlę ir RPA „Power Automate Desktop“ srautą - ir atlikti eksperimentinį tyrimą, kuriame, naudojant pasirinktą duomenų rinkinį, būtų palygintas EPDA ir tradicinio sąskaitų senaties metodo informatyvumas bei tikslumas ir įvertintas .NET ir RPA prototipų našumas, lankstumas ir taikymo ribos.
5. Įvertinus gautus rezultatus, suformuluoti rekomendacijas, kaip EPDA metodą ir skirtingus technologinius sprendimus (.NET ir RPA) taikyti vidiniame audite, siekiant didesnio efektyvumo ir audito įrodymų kokybės.

Darbo tyrimo metodai

Tyrime taikyti šie pagrindiniai mokslinio tyrimo metodai: sisteminė mokslinė literatūros ir norminių šaltinių analizė, leidusi atskleisti gautinų sumų valdymo, sąskaitų senaties ir mokėjimų elgsenos modeliavimo teorinius pagrindus (1 skyrius); lyginamoji analizė, skirta tradicinio sąskaitų senaties metodo ir EPDA rodiklio palyginimui konceptualių ir praktinių aspektu (2–3 skyriai); eksperimentinis skaitmeninis tyrimas, kuriuo Kaggle „Payment Date Prediction“ duomenų rinkinyje įgyvendintas EPDA algoritmas ir įvertintas jo pritaikomumas gautinų sumų auditui (3 skyrius); statistinė duomenų analizė ir našumo matavimai, taikant „PerfMon“ priemonę .NET ir „Power Automate Desktop“ prototipams bei interpretuojant gautas metrikas vidinio audito kontekste.

Eksperimentinių tyrimų apimtis

Eksperimentinė dalis apima EPDA metodo realizavimą dviem technologinėmis priemonėmis – .NET aplikacijoje ir RPA „Power Automate Desktop“ sraute – naudojant Kaggle duomenų rinkinį, iš kurio po duomenų valymo liko 48 580 sąskaitų įrašų. EPDA skaičiavimas ir tradicinės sąskaitų senaties analizė buvo atlikti kelių skirtingų duomenų rinkinio dydžių imtims (1 tūkst., 10 tūkst., 25 tūkst. ir 48

tūkst. eilučių), siekiant įvertinti metodo stabilumą bei rezultatų jautrumą duomenų apimčiai. Papildomai buvo atliktas .NET ir RPA prototipų našumo palyginimas, matuojant vykdymo trukmę, CPU, atminties ir IO sąnaudas ir normalizuojant rezultatus vienam EPDA logikos paskaičiavimui 1 000 sąskaitų.

Svarbiausi darbo rezultatai ir jų pritaikomumas FinTech srityje

Darbe sukurta ir praktiškai išbandyta EPDA skaičiavimo schema parodė, kad elgsenos pagrindu grindžiamas numatomo ir faktinio mokėjimo vėlavimo rodiklis reikšmingai papildo tradicinę sąskaitų senaties analizę: leidžia identifikuoti sąskaitas, kurios kalendorine prasme nėra pačios seniausios, tačiau ryškiai nukrypsta nuo įprasto kliento atsiskaitymo modelio ir todėl yra svarbios audito bei kredito rizikos vertinimui. Sukurti .NET ir RPA prototipai patvirtino, kad EPDA rodiklis gali būti integruotas į gautinų sumų analizės ir skoringo sprendimus, naudojamus finansinių technologijų (FinTech) srityje – pavyzdžiui, automatizuotam klientų prioritizavimui išieškojimo procesuose, vėlavimo rizikos nustatymui, pinigų srautų prognozavimo modelių papildymui ar išmaniųjų įspėjimų generavimui kredito valdymo sistemose.

Darbo apimtis - darbą sudaro 72 puslapiai, 7 numeruoti paveikslai ir 4 numeruotos lentelės. Darbas sudarytas iš 3 dalių.

1. RPA TECHNOLOGIJOS TAIKYMO AUDITO PROCESUOSE TEORINIAI ASPEKTAI

RPA technologijos pastaraisiais metais tapo vienu iš svarbiausių inovacijų verslo procesų automatizavime, įskaitant audito sritį. RPA technologijos naudojimas vidiniame audite suteikia galimybę automatizuoti pasikartojančias ir taisyklėmis pagrįstas užduotis, leidžiant auditoriams sutelkti dėmesį į sudėtingesnes ir didesnę vertę kuriančias veiklas. Šis skyrius nagrinėja RPA technologijų teorinius aspektus, įskaitant jų apžvalgą, integravimo į audito procesus metodus, privalumus, iššūkius ir apribojimus, bei RPA taikymo audito procesuose sistemą. Analizė remiasi pagrindiniais moksliniais straipsniais ir tyrimais, siekiant pateikti išsamią RPA technologijų taikymo audito procesuose analizę ir įvertinti jų poveikį.

1.1. RPA technologijos apžvalga

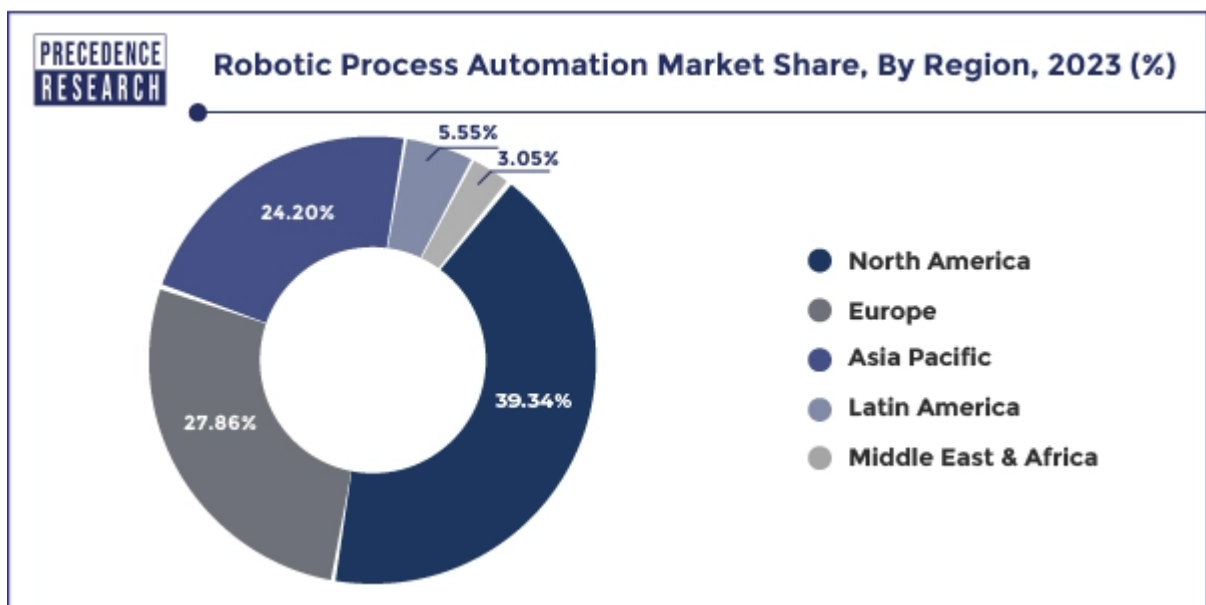
RPA yra technologija, skirta automatizuoti verslo procesus, kurie paprastai reikalauja žmogaus įsikišimo. RPA apima programinės įrangos robotų naudojimą, kurie gali atlikti taisyklėmis pagrįstas, pasikartojančias užduotis greičiau ir tiksliau nei žmonės. Kapur ir Singh (2022) teigė, kad RPA gali apdoroti dideles duomenų apimtis, atlikti skaičiavimus ir atlikti kitas audito užduotis daug greičiau ir tiksliau nei tradiciniai metodai. Autoriai pabrėžė, kad RPA ne tik padidina duomenų apdorojimo greitį, bet ir užtikrina didesnę tikslumą bei sumažina klaidų tikimybę, kas ypač svarbu audito srityje.

Tuo tarpu Kogan, Kokina, Stampone ir Boyle (2024) savo tyrimuose pritarė šiam požiūriui, pabrėždami, kad RPA technologija gali automatizuoti sudėtingus, pasikartojančius procesus, kurie anksčiau buvo atliekami rankiniu būdu. Jie teigė, kad RPA gali pakeisti rankiniu būdu atliekamus procesus, pavyzdžiui, duomenų rinkimą, analizę ir ataskaitų rengimą, tokiu būdu leidžiant auditoriams sutelkti dėmesį į aukštesnės pridėtinės vertės užduotis. Tai ne tik padidina darbo našumą, bet ir užtikrina nuolatinį ir nenutrūkstamą audito procesų vykdymą.

Xin, Asokan ir Balasingam (2022) savo darbe pridūrė, kad RPA yra ypač naudinga efektyvumui didinti ir klaidų mažinimui audito procesuose. Jie pažymėjo, kad RPA technologija gali atlikti užduotis, kurios anksčiau užtrukdavo daug laiko ir buvo imlios darbo jėgai, tokiu būdu leidžiant organizacijoms sumažinti operacines išlaidas ir padidinti bendrą produktyvumą. Autoriai taip pat pabrėžė, kad RPA

naudojimas gali pagerinti duomenų apdorojimo kokybę ir tikslumą, nes programinės įrangos robotai gali atlikti užduotis pagal nustatytas taisykles ir procedūras be jokių nukrypimų.

Lyginant šių autorių požiūrius, galima pastebėti, kad visi jie pripažįsta RPA technologijos privalumus, tokius kaip efektyvumas, tikslumas ir greitis. Tačiau Kapur ir Singh (2022) labiau akcentavo technologijos gebėjimą atlikti didelės apimties duomenų apdorojimą ir skaičiavimus, tuo tarpu Kogan, Kokina, Stampone ir Boyle (2024) daugiau dėmesio skyrė procesų automatizavimui ir rankinio darbo pakeitimui. Xin, Asokan ir Balasingam (2022) papildomai pabrėžė RPA naudą efektyvumo didinimui ir operacinių išlaidų mažinimui.



Šaltinis: Precedence Research (2024)

1 pav. Procentinė RPA rinkos dalis pagal regionus 2023 metais

Papildomai, pagal naujausius duomenis, RPA rinka sparčiai auga, ypač Šiaurės Amerikoje ir Azijos Ramiojo vandenyno regione. 2023 metais Šiaurės Amerika dominavo rinką, užimdama 39.34% visos rinkos dalies. Tai daugiausia dėl to, kad JAV yra viena iš pagrindinių robotikos inovacijų ir diegimo lyderių. Azijos Ramiojo vandenyno regionas tikimasi bus sparčiausiai augantis, nes vis daugiau įmonių šioje srityje diegia RPA technologijas farmacijos, sveikatos apsaugos, IT, telekomunikacijų, mažmeninės prekybos ir gamybos sektoriuose. Šie regioniniai skirtumai pabrėžia, kad RPA technologijos pritaikymas ir nauda gali skirtis priklausomai nuo geografinės vietovės ir pramonės sektoriaus.

Apibendrinant, RPA technologija yra galingas įrankis, kuris gali žymiai pagerinti audito procesų efektyvumą ir tikslumą. Autorių darbai rodo, kad RPA technologija turi platų pritaikymą ir gali būti naudinga įvairiose audito srityse, nuo duomenų rinkimo ir analizės iki ataskaitų rengimo ir rizikos valdymo. Nors požiūriai šiek tiek skiriasi, visi autoriai sutaria, kad RPA technologija yra esminė šiuolaikinio audito proceso dalis, suteikianti reikšmingų privalumų ir galimybių.

1.2. RPA integravimas į audito procesus

RPA technologijos integravimas į audito procesus reikalauja kruopštaus planavimo, organizacinės struktūros pritaikymo ir atitinkamos infrastruktūros sukūrimo. Kapur ir Singh (2022) pabrėžė aiškaus plano ir vaidmenų nustatymo svarbą, siekiant užtikrinti sėkmingą RPA diegimą. Autoriai teigė, kad organizacijos, kurios iš anksto planuoja ir nustato aiškias atsakomybes, dažniau pasiekia geresnių rezultatų diegiant RPA technologijas. Kogan, Kokina, Stampone ir Boyle (2024) savo tyrimuose taip pat pabrėžė valdymo modelių svarbą, pažymėdami, kad centralizuota priežiūra gali užtikrinti didesnę procesų valdymo efektyvumą ir sumažinti rizikas, susijusias su RPA diegimu. Jie akcentavo, kad tinkamos valdymo struktūros sukūrimas yra būtinas norint išvengti dažnų RPA projektų nesėkmių, kurios dažnai atsiranda dėl prasto valdymo ir rizikos kontrolės trūkumo.

Xin, Asokan ir Balasingam (2022) savo darbe nagrinėjo technologinius ir žmogiškuosius iššūkius, su kuriais susiduria organizacijos, integruodamos RPA į vidinius audito procesus. Jie pažymėjo, kad viena iš pagrindinių kliūčių yra darbuotojų pasipriešinimas technologijų diegimui ir organizacijos kultūros pokyčiams. Tai dažnai lemia lėtesnį diegimo procesą ir sumažina technologijos pritaikymo efektyvumą. Autoriai pabrėžė, kad organizacijos turi nuolat mokytis savo darbuotojus ir skatinti juos adaptuotis prie naujų technologijų, siekiant užtikrinti sėkmingą integraciją.

Kokina ir Blanchette (2019) tyrime nurodė, kad sėkmingam RPA diegimui būtina standartizuoti ir optimizuoti procesus, sukuriant įrankius, panašius į balų korteles, kurie padeda įvertinti užduočių tinkamumą automatizavimui. Jie taip pat pabrėžė, kad reikia pritaikyti valdymo struktūras, kad būtų įtraukti skaitmeniniai darbuotojai, ir peržiūrėti vidines kontrolės priemones. Pasak jų, organizacijos gauna naudą automatizuodamos tik tuos procesus, kurie yra struktūruoti, pasikartojantys, taisyklėmis pagrįsti ir turi skaitmeninius įvestis. Tai leidžia ne tik sumažinti išlaidas, bet ir pagerinti procesų dokumentaciją, sumažinti klaidų skaičių, tiksliau įvertinti procesų našumą ir gerinti ataskaitų kokybę.

Gotthardt, Koivulaakso, Paksoy, Saramo, Martikainen ir Lehner (2020) savo darbe išskyrė pagrindinius RPA diegimo iššūkius, tokius kaip duomenų kokybės problemos, pokyčių valdymas ir nuolatinės stebėsenos bei atnaujinimų būtinybė. Jie teigė, kad sėkmingai integracijai reikia ne tik technologinės pasirengimo, bet ir organizacinės paramos bei nuolatinio darbuotojų mokymo. Autoriai taip pat pabrėžė, kad RPA integracija turi būti paremta aiškiomis strategijomis ir gerai apgalvotu planavimu, kad būtų pasiektas maksimalus efektyvumas.

Apibendrinant, RPA technologijos integravimas į audito procesus yra sudėtingas ir daugialypis procesas, reikalaujantis ne tik technologinės infrastruktūros, bet ir organizacinės struktūros pritaikymo, aiškių valdymo modelių sukūrimo bei nuolatinės darbuotojų adaptacijos ir mokymų. Sėkmingai integracijai būtina užtikrinti procesų standartizavimą, tinkamą rizikos valdymą ir nuolatinę stebėseną, siekiant maksimaliai išnaudoti RPA technologijos teikiamus privalumus ir išvengti galimų iššūkių.

1.3. RPA privalumai audito srityje

Robotinio procesų automatizavimo (RPA) integravimas į vidaus audito procesus gerokai padidina audito komandų galimybes ir efektyvumą. Rawashdeh ir kt. (2022) teigia, kad pagrindinis RPA indėlis yra pasikartojančių ir daug laiko reikalaujančių užduočių, kurios tradiciškai atimdavo daug auditorių laiko, automatizavimas. Šis automatizavimas leidžia audito specialistams nukreipti dėmesį į sudėtingesnes ir reikšmingesnes sritis, pavyzdžiui, strateginę rizikos vertinimą ir paramos vadovybės sprendimams teikimą. RPA transformacinį potencialą dar labiau išryškina žmogiškųjų klaidų, susijusių su duomenų tvarkymu, mažinimas ir galimybė patikimiau ir greičiau apdoroti didelius sandorių kiekius.

Be to, RPA palengvina perėjimą nuo tradicinių periodinių audito peržiūrų prie dinamiškesnio ir nuolatinio audito proceso. Šis perėjimas nėra tik procedūrinis pokytis, bet reiškia esminę transformaciją, kaip audito funkcijos gali teikti vertę. Jis leidžia realiuoju laiku stebėti ir peržiūrėti sandorius ir verslo procesus, taip padidinant gebėjimą aktyviai valdyti riziką ir veiksmingiau užtikrinti atitiktį teisės aktų reikalavimams. Kapur ir Singh (2022) rašo apie efektyvumo padidėjimą, ypač laiko taupymą ir tikslumą, sustiprina RPA naudą, kad audito procesai taptų efektyvesni ir lanksčiau reaguotų į sparčius verslo aplinkos pokyčius.

Robotizuotų procesų automatizavimo (RPA) diegimas vidaus audito srityje ne tik didina veiklos efektyvumą, bet ir keičia auditorių įgūdžius. Kaip pabrėžia Xin, Asokan ir Suresh (2022), nauja rizika,

susijusi su pasiklojimu RPA, apima būtinybę griežtai kontroliuoti automatizuotus procesus ir galimą priklausomybės nuo šios technologijos riziką. Ši besikeičianti aplinka reikalauja, kad auditoriai ne tik suprastų techninius RPA aspektus, bet ir gerai išmanytų pagrindinius verslo procesus, kuriuos jie automatizuoja.

1.4. RPA pritaikymo iššūkiai ir apribojimai audito srityje

Pereinant prie automatizavimo, ypač svarbūs įgūdžiai, susiję su duomenų analize, procesų projektavimu ir RPA sistemų priežiūra. Iš auditorių vis dažniau tikimasi, kad jie prisidės prie automatizuotų procesų projektavimo ir valdymo, užtikrindami, kad jie atitiktų verslo tikslus ir reguliavimo reikalavimus. Šis pokytis rodo, kad auditorių vaidmuo plečiasi nuo tradicinių vertinimo užduočių iki labiau technologiškai integruotų pareigų, kurios apima nuolatinį mokymąsi ir IT bei konkrečių RPA priemonių išmanymą. Ši raida rodo ateitį, kai nuolatinis mokymasis ir technologijų pritaikymas yra neatsiejama audito profesijos dalis, ir pabrėžia naujos auditorių mokymo ir profesinio tobulėjimo paradigmos poreikį.

Robotizuotų procesų automatizavimo (RPA) diegimas vidaus audito srityje kelia įvairių iššūkių ir pabrėžia veiksmingo rizikos valdymo svarbą. Vis labiau plintant RPA, vidaus audito skyriai turi susidurti su naujais sudėtingais uždaviniais, susijusiais su automatizuotų sistemų integravimu į esamus procesus. Tiek Xin ir kt. (2022), tiek Kapur ir Singh (2022) teigia, jog sėkmingam RPA įgyvendinimui reikalingi aukštos kokybės duomenys ir patikima procesų dokumentacija. Be jų automatizavimo procese gali atsirasti klaidų arba neveiksmingumo, kurie gali pakenkti RPA siekiamai naudai.

Kitas svarbus iššūkis yra automatizuotų procesų kontrolės išlaikymas. Įdiegus RPA, kyla rizika, kad šios automatizuotos sistemos gali veikti ne taip, kaip tikėtasi, galimai nukrypti nuo numatytų procesų arba sukurti naujų pažeidžiamumų. Auditoriai turi užtikrinti, kad šios sistemos veiktų pagal nustatytus parametrus, o tam reikia gerai išmanyti ir technologiją, ir su ja susijusius verslo procesus. Šis budrumas yra labai svarbus, nes automatizuotuose procesuose daromos klaidos gali išlikti ar net padidėti, jei nebus greitai nustatytos ir ištaisytos.

Be to, didėjant automatizavimui, dėl priklausomybės nuo technologijų gali kilti rizika, susijusi su duomenų saugumu ir vientisumu. Dėl šio rūpesčio vidaus auditoriai turi priimti platesnę perspektyvą, sutelkti dėmesį į valdymą ir užtikrinti, kad RPA sistemos būtų ne tik veiksmingos, bet ir saugios bei

atitiktų teisės aktų reikalavimus. Iššūkis - subalansuoti RPA efektyvumo padidėjimą ir poreikį išlaikyti aukštą priežiūros ir rizikos valdymo lygį, užtikrinant, kad perėjimas prie automatizavimo nepakenktų audito kokybei ar verslo vientisumui.

1.5. RPA taikymo sistema audito procesuose

Robotic Process Automation (RPA) technologijos diegimas ir integravimas į audito procesus yra kompleksinis procesas, reikalaujantis išsamaus planavimo ir strateginio požiūrio. Siekiant sėkmingai pritaikyti RPA technologijas, būtina sukurti veiksmingą sistemą, kuri apimtų visus esminius diegimo ir valdymo aspektus. Šiame skyriuje aptariama, kaip RPA technologija gali būti integruota į audito procesus, kokie iššūkiai gali kilti šio proceso metu ir kokie yra pagrindiniai sėkmės veiksniai.

RPA taikymo audito procesuose sistema apima kelis esminius etapus, nuo planavimo iki įgyvendinimo ir nuolatinės stebėsenos. Kapur ir Singh (2022) pabrėžė aiškaus plano ir strategijos svarbą pradiniam etape, siekiant užtikrinti, kad RPA diegimas būtų efektyvus ir naudingas. Jie teigė, kad organizacijos, kurios iš anksto planuoja ir nustato aiškias atsakomybes, dažniau pasiekia geresnių rezultatų diegiant RPA technologijas. Kapur ir Singh taip pat akcentavo būtinybę turėti aiškią valdymo struktūrą ir griežtas kontrolės priemones, kurios padėtų sumažinti rizikas ir užtikrintų skaidrumą diegimo procese.

Kogan, Kokina, Stampone ir Boyle (2024) taip pat aptarė, kaip svarbu yra tinkamai nustatyti valdymo modelius ir kontrolės priemones, siekiant užtikrinti procesų skaidrumą ir saugumą. Jie pabrėžė, kad centralizuota priežiūra gali užtikrinti didesnę procesų valdymo efektyvumą ir sumažinti rizikas, susijusias su RPA diegimu. Autoriai pažymėjo, kad tinkamos valdymo struktūros sukūrimas yra būtinas norint išvengti dažnų RPA projektų nesėkmių, kurios dažnai atsiranda dėl prasto valdymo ir rizikos kontrolės trūkumo.

Xin, Asokan ir Balasingam (2022) akcentavo nuolatinės stebėsenos ir atnaujinimų būtinybę, siekiant užtikrinti, kad RPA sistemos veiktų efektyviai ir atitiktų organizacijos poreikius. Jie taip pat teigė, kad viena iš pagrindinių kliūčių yra darbuotojų pasipriešinimas technologijų diegimui ir organizacijos kultūros pokyčiams. Tai dažnai lemia lėtesnį diegimo procesą ir sumažina technologijos pritaikymo efektyvumą. Autoriai pabrėžė, kad organizacijos turi nuolat mokytis savo darbuotojus ir skatinti juos adaptuotis prie naujų technologijų, siekiant užtikrinti sėkmingą integraciją.

Fernandez ir Aman (2021) išskyrė pagrindinius RPA diegimo iššūkius, tokius kaip duomenų kokybės problemos, pokyčių valdymas ir nuolatinės stebėsenos bei atnaujinimų būtinybę. Jie teigė, kad sėkmingai integracijai reikia ne tik technologinės pasirengimo, bet ir organizacinės paramos bei nuolatinio darbuotojų mokymo. Autoriai taip pat pabrėžė, kad RPA integracija turi būti paremta aiškiais strategijomis ir gerai apgalvotu planavimu, kad būtų pasiektas maksimalus efektyvumas. Savo darbe jie taip pat pažymėjo, kad RPA diegimas reikalauja detalaus procesų vertinimo ir optimizavimo prieš pradedant automatizaciją, siekiant išvengti esamų problemų automatizavimo.

Kokina ir Blanchette (2019) tyrime nurodė, kad sėkmingam RPA diegimui būtina standartizuoti ir optimizuoti procesus, sukuriant įrankius, panašius į balų korteles, kurie padeda įvertinti užduočių tinkamumą automatizavimui. Jie pabrėžė, kad organizacijos gauna naudą automatizuodamos tik tuos procesus, kurie yra struktūruoti, pasikartojantys, taisyklėmis pagrįsti ir turi skaitmeninius įvestis. Tai leidžia ne tik sumažinti išlaidas, bet ir pagerinti procesų dokumentaciją, sumažinti klaidų skaičių, tiksliau įvertinti procesų našumą ir gerinti ataskaitų kokybę.

Apibendrinant, RPA technologijos integravimas į audito procesus yra sudėtingas ir daugialypis procesas, reikalaujantis ne tik technologinės infrastruktūros, bet ir organizacinės struktūros pritaikymo, aiškių valdymo modelių sukūrimo bei nuolatinės darbuotojų adaptacijos ir mokymų. Sėkmingai integracijai būtina užtikrinti procesų standartizavimą, tinkamą rizikos valdymą ir nuolatinę stebėseną, siekiant maksimaliai išnaudoti RPA technologijos teikiamus privalumus ir išvengti galimų iššūkių.

1.6. RPA taikymo sistema audito procesuose

Gautinų sumų ciklas prasideda pardavimo kreditan momentu ir baigiasi tada, kai klientas visiškai atsiskaito arba skola nurašoma kaip nebeatgaunama. Šis ciklas yra viena pagrindinių įmonės apyvartinio kapitalo dalių: kuo ilgiau lėšos „užstringa“ gautinose sumose, tuo labiau ribojamas įmonės likvidumas ir investicinės galimybės. Empiriniai tyrimai, atlikti Asman, Fernandez, Jamaludin, Zaki, ir Ghazali (2022), analizuojantys piniginių srautų konversijos ciklą, pabrėžia, kad gautinų sumų poskyris – dažniausiai matuojamas „days sales outstanding“ (DSO) rodikliu – yra kritiškas įmonės veiklos rezultatams ir finansiniam stabilumui. Trumpesnis DSO ir efektyvesnis gautinų sumų valdymas siejami su geresne įmonių pelningumo dinamika bei mažesne likvidumo rizika.

Klasikinė priemonė šiam ciklui valdyti – gautinų sumų senaties ataskaita (accounts receivable aging report). Ji grupuoja neapmokėtas sąskaitas pagal tai, kiek dienų praėjo nuo jų apmokėjimo termino arba išrašymo datos, dažniausiai naudojant intervalus „dabartinės“, 1–30, 31–60, 61–90 ir daugiau kaip 90 dienų. Tokia ataskaita leidžia greitai identifikuoti labiausiai pavėluotas skolas, įvertinti blogų skolų riziką ir spręsti, kuriuos klientus reikia prioritetiškai informuoti ar perduoti išieškojimui Schmidt, J. (2024). Senaties analizė taip pat padeda formuoti kredito politiką – pavyzdžiui, griežtinti sąlygas klientams, kurių skolos nuolat patenka į seniausius krepšelius, bei apskaičiuoti tikėtinus nuvertėjimo ir atidėjinių dėl abejotinų skolų dydžius.

Kartu su senaties ataskaita plačiai taikomi keli kiekybiniai rodikliai. Vienas svarbiausių – dienų pardavimų gautinose sumose rodiklis (DSO), kuris parodo, kiek vidutiniškai dienų reikia, kol pardavimai kreditan paverčiami pinigais. Jis dažniausiai skaičiuojamas taip:

$$DSO = \frac{\text{Vidutinės gautinos sumos}}{\text{Grynieji pardavimai kreditan}} \times \text{dienų skaičius laikotarpiu.}$$

Šis rodiklis yra pinigų konversijos ciklo dalis ir plačiai naudojamas likvidumo bei kreditų valdymo efektyvumui vertinti, aiškina Kenton, W. (2025). Paul, Devi, ir Teh(2012) mokslininkų empiriniai tyrimai rodo, kad įmonės, kurios palaiko trumpesnę DSO, dažniau pasižymi didesniu pelningumu ir mažesne blogų skolų dalimi, o nuosekliai didėjantis DSO interpretuojamas kaip signalas apie prastėjančią klientų mokumo kokybę ar neefektyvią išieškojimo praktiką.

Gautinų sumų ciklo rodikliai yra esminė vidinės kontrolės ir audito procedūrų dalis. Tyrimai, atlikti Chen, Wang, Yang, ir Zhou (2025) mokslininkų, taikant COSO vidaus kontrolės modelį, rodo, kad aukštesnė vidaus kontrolės kokybė tiesiogiai siejasi su didesniu gautinų sumų apyvartumu, mažesniu senstančių sąskaitų kiekiu ir mažesniais vertės sumažėjimo nuostoliais. Praktiniuose audito vadovuose pabrėžiama EFRP Group (2024), kad gautinų sumų auditas paprastai prasideda nuo senaties ataskaitų peržiūros, DSO ir kitų rodiklių analizės – šie duomenys naudojami reikšmingoms rizikos sritims identifikuoti, atrankai į detalesnę testavimą, įvertinti atidėjinių dėl abejotinų skolų pakankumą ir nustatyti galimos sukčiavimo ar pajamų dirbtinio „išpūtimo“ požymius.

Tačiau tiek senaties ataskaitos, tiek DSO ir panašūs rodikliai turi ribotumą: jie daugiausia vertina absoliutų pradelstumo dydį ir bendrą portfelio efektyvumą, bet menkai atskleidžia individualių klientų

elgsenos modelius. Vienodai 60 dienų pradelstą sąskaitą jie traktuos vienodai tiek klientui, kuris istoriškai visada atsiskaito su dideliu vėlavimu, tiek klientui, kuris įprastai moka laiku, bet šįkart gerokai vėluoja. Pastaraisiais metais mokslinėje literatūroje, pavyzdžiui mokslininkų Bahrami, Bozkaya, ir Balcisoy (2020) darbe, pradėti siūlyti pažangesni mokėjimo elgsenos modeliai, kurie remiasi istorinių sąskaitų ir mokėjimų duomenimis ir prognozuoja tikėtiną apmokėjimo momentą konkrečiam klientui. Tradiciniai senaties rodikliai vis dar yra pagrindinė vidinio audito atskaitos priemonė, tačiau jie sudaro tik pradinį gautinų sumų rizikos įvertinimo sluoksnį ir sudaro prielaidas tokio tipo elgsenos rodikliams, kaip šiame darbe nagrinėjamas numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodas, papildomai praturtinti audito analizę.

1.7. Duomenų analitika ir elgsenos modeliai vidiniame audite

Pastaraisiais metais duomenų analitika tapo viena svarbiausių vidinio audito transformacijos krypčių. Tradiciškai auditas rėmėsi istorinių duomenų analize ir ribota atranka – tikrinamos pavienės operacijų imtys, o išvados daromos remiantis statistine ekstrapoliacija į visą populiaciją. KPGM (2012) tyrimai ir profesinių organizacijų rekomendacijos rodo, kad tokio požiūrio nepakanka sudėtingoje, didelių duomenų apimčių ir realaus laiko sandorių aplinkoje. Duomenų analitika ir nuolatinis auditas (angl. *continuous auditing*) leidžia vidiniam auditui pereiti nuo retrospektyvaus patikrinimo prie nuolatinio rizikų stebėjimo, automatizuoto anomalijų aptikimo ir operatyvesnio kontrolės trūkumų identifikavimo.

Profesiniai šaltiniai pabrėžia, kad duomenų analitika auditui suteikia galimybę analizuoti ne tik nedidelę imtį, bet ir visą operacijų populiaciją, taip didinant audito kokybę ir patikimumą. Brown (2024) bei Joshi, ir Marthandan (2020) pažymi, jog vidinio audito padaliniai vis plačiau taiko analitiką išlaidų, darbo užmokesčio, tiekėjų atsiskaitymų ir pajamų srityse, o pajamų pusėje ypač svarbi tampa sąskaitų faktūrų ir atsiskaitymų duomenų analizė – nuo bilingo tikslumo iki neįprastų tendencijų ir galimų sukčiavimo atvejų identifikavimo. Nuolatinio audito tyrimai rodo, kad taikant didžiųjų duomenų analitiką galima automatizuotai testuoti kontrolės veiksmingumą realiu laiku ar artimu realiam laikui režimu, o tai leidžia anksčiau pastebėti klaidas ir sukčiavimą bei sumažinti nuostolių tikimybę.

Kartu su bendrąja duomenų analitika plėtojami ir elgsenos modeliai, paremti klientų mokėjimo istorija ir kitais rizikos veiksniais. Kredito rizikos literatūroje išskiriami taikomieji (angl. *application*)

modeliai, vertinantys naujų klientų riziką pagal statinius požymius, ir elgsenos (angl. *behavioral*) modeliai, prognozuojantys jau esamų klientų vėlesnį mokėjimo elgesį remiantis jų istorija, kaip aprašo savo darbe Didkovskiy, Jean, Pera, ir Nordio (2024). Pastarųjų tikslas – numatyti tikėtiną vėlavimą, nemokumo tikimybę ar sąskaitos „sveikatos“ indeksą, panaudojant tokias charakteristikas kaip ankstesni vėlavimai, ginčų istorija, skolos lygis, demografiniai ir elgsenos požymiai. Praktiniuose gautinų sumų valdymo sprendimuose ir FinTech produktų aprašymuose dažnai minimos mašininio mokymosi platformos, kurios, apdorodamos istorinius duomenis, prognozuoja būsimą mokėjimo datą, priskiria klientams elgsenos reitingus ar „account health“ balus ir taip padeda optimizuoti inkasavimo prioritetus bei pinigų srautų prognozes, aptaria S. Nanda (2018).

Moksliniuose darbuose vis dažniau nagrinėjamas konkrečiai sąskaitų apmokėjimo vėlavimo prognozavimas. Pavyzdžiui, E. C. Malesiou (2021) tyrime mašininio mokymosi modeliai naudojami mėnesinių mokėjimų vėlavimo statusui ir dienų skaičiui iki apmokėjimo prognozuoti, remiantis istorinių sąskaitų ir mokėjimų duomenimis. M. Martikainen (2023) darbe sistemingai apžvelgiami statistiniai klasifikavimo metodai, skirti pardavimo sąskaitų vėlavimo rizikai prognozuoti, ir parodoma, kad tokie modeliai gali patikimai identifikuoti vėluojančius klientus ir padėti efektyviau paskirstyti išieškojimo resursus. Pramoniniuose šaltiniuose, pavyzdžiui, HighRadius ir Nanda (2018), pabrėžiama, kad regresijos ir klasifikavimo modeliai gali prognozuoti tiek konkrečią mokėjimo datą, tiek vėlavimo krepšelius, ir taip suteikti pagrindą aktyviam, o ne reakciniam skolų valdymui. Kong, ir Brintrup (2025) tyrimai taip pat žengia link priežastinių (causal) dirbtinio intelekto modelių, kurie siekia ne tik prognozuoti, bet ir suprasti, kokie veiksniai lemia mokėjimų vėlavimą ir kaip juos būtų galima valdyti.

Vidinio audito kontekste tokie elgsenos modeliai atveria naujas galimybes rizikos vertinimui ir procedūrų planavimui, tačiau iki šiol dauguma jų sukurti pirmiausia operaciniams tikslams – kredito valdymui, inkasavimui ar pinigų srautų planavimui, o ne kaip integruota audito analitinė priemonė. Profesiniai šaltiniai, Caseware (2023), apie duomenų analitiką vidiniame audite akcentuoja, kad auditoriai vis dažniau naudoja analitines priemones rizikos pagrindu atrinkdami testuotinas operacijas ir taikydami nuolatinį stebėjimą, tačiau konkrečių, elgsenos pagrindu sukurtų rodiklių gautinų sumų audite aprašoma palyginti nedaug. Dėl to atsiranda tyrimų niša, kurią siekia užpildyti šis darbas: įvertinti, kaip elgsenos modelio logika – numatomo ir faktinio mokėjimo vėlavimo (EPDA) principu – gali būti integruojama į vidinio audito analitiką ir naudojama gautinų sumų rizikos vertinimui kartu su tradiciniais senaties ir DSO rodikliais.

1.8. RPA taikymas gautinų sumų ir pajamų cikle

Gautinų sumų ir pajamų (order-to-cash) ciklas yra viena dažniausių sričių, kur taikoma robotinė procesų automatizacija (RPA) finansuose ir apskaitoje. Wewerka, ir Reichert (2020) tyrimų rezultatai rodo, kad šios technologijos ypač tinka struktūruotoms, taisyklėmis apibrėžtomis užduotims, būdingoms finansų funkcijai – duomenų nuskaitymui, tikrinimui, registravimui ir sutikrinimui. RPA naudojimo atvejų analizė finansų skyriuose išskiria sąskaitų išrašymo, debitorių ir kreditorių valdymo, pajamų ir sąnaudų registravimo bei ataskaitų rengimo procesus kaip vienus svarbiausių kandidatų automatizacijai, pažymi savo darbe C. Dilmegani (2025). Kokina ir Blanchette (2019) teigia, kad RPA diegimas leidžia didelę dalį pasikartojančių užduočių perkelti „skaitmeniniams darbuotojams“, o finansų specialistams susitelkti į analitinę ir konsultacinę veiklą.

Specifiškai gautinų sumų ir pajamų cikle RPA dažniausiai taikomas visoje order-to-cash grandinėje: nuo užsakymo priėmimo ir kredito limito patikrinimo iki sąskaitos išrašymo, priminimų siuntimo ir gautų mokėjimų sudengimo. Ramsey, R. (2024) pabrėžia kelias tipines taikymo sritis: automatinį sąskaitų generavimą ir išsiuntimą, mokėjimų priskyrimą sąskaitoms (cash application), išimčių ir neatitikimų identifikavimą, kreditų ir nuolaidų taikymo tikrinimą bei standartizuotų priminimų klientams siuntimą. RPA platformos, integruotos su dokumentų atpažinimo (OCR) ir dirbtinio intelekto moduliais, leidžia automatizuoti net ir dalį nestruktūrotų duomenų apdorojimo (pavyzdžiui, skaityti mokėjimo nurodymus ar el. pašto atsakymus ir pagal juos atnaujinti skolos statusą), tačiau pagrindinis dėmesys dažniausiai išlieka ties procesine, o ne elgsenos analize, kaip nagrinėja Fomin, S. (2020).

Gautinų sumų automatizavimo tyrimai ir praktiniai atvejai rodo aiškią RPA naudą pinigų srautams ir operaciniam efektyvumui. Corcentric (2023) aprašyme pabrėžiama, kad automatizavus sąskaitų ir inkasavimo procesus galima reikšmingai sutrumpinti order-to-cash ciklą, sumažinti klaidų ir rankinio taisymo apimtį, pagerinti duomenų kokybę ir laikytis kredito politikos taisyklių nuosekliau nei tai darant vien rankiniu būdu. Pavyzdžiui, BlackLine (2022) AR automatizavimo sprendimų aprašymuose teigiama, kad el. sąskaitų siuntimo, mokėjimų sekimo ir priminimų procesų robotizavimas tiesiogiai prisideda prie greitesnio pinigų įplaukimo, mažesnio pradelstų sąskaitų skaičiaus ir geresnės klientų aptarnavimo patirties.

Vidinio audito ir rizikos kontrolės literatūra RPA gautinų sumų ir pajamų cikle vertina dviem aspektais: kaip audituotiną objektą ir kaip pačiam auditui pasitelkiamą priemonę. Kokina ir Blanchette (2019), nagrinėdamos „skaitmeninį darbą“ apskaitoje, parodė, kad RPA diegimas keičia apskaitos

specialistų funkcijas, o dalį kontrolės veiksmų perima programinės „botų“ procedūros. Kogan ir bendraautoriai (2022), apklausę RPA programų vadovus, atskleidė, kad diegiant robotus finansų ir apskaitos procesuose būtina sistemingai valdyti su RPA susijusią kontrolės aplinką – nustatyti prieigos teises, pakeitimų valdymo taisykles, logų fiksavimą ir stebėseną, nes klaidingai sukonfigūruoti robotai gali sistemingai kartoti tą pačią klaidą dideliu mastu. Naujesni tyrimai ir profesinės publikacijos, atlikti Almayyahi, Fotouh, ir Metawee (2025), apie RPA audite pažymi, kad robotai gali automatizuoti duomenų surinkimą ir testų vykdymą, tačiau auditoriams tenka užduotis įvertinti, ar pačių RPA darbo eigos logika, įskaitant gautinų sumų ir pajamų ciklo automatizavimą, neiškreipia finansinės informacijos ir nekuria naujų rizikos šaltinių.

Vis dėlto tiek mokslinėje literatūroje, tiek praktiniuose RPA gautinų sumų sprendimų aprašymuose daugiausia dėmesio skiriama procesų efektyvumo ir įprastų kontrolės žingsnių automatizavimui, o ne naujų, elgsenos pagrindu grindžiamų rizikos rodiklių kūrimui. Daugelis rinkoje siūlomų AR automatizavimo ir kredito rizikos valdymo sprendimų jau taiko mokėjimo istorijos duomenis ir elgsenos kredito scoringą, siekdami geriau įvertinti klientų riziką ir parinkti tinkamas inkasavimo strategijas, kaip rašo Verma D. (2025) tinklaraštyje. Tačiau šie modeliai dažniau integruojami į kredito valdymo ir išieškojimo procesus, o ne į vidinio audito analitines procedūras ir paprastai nėra suprantami kaip audito rodikliai. Todėl šiame darbe nagrinėjamas numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodas užpildo konkrečią spragą: siūloma, kad RPA ir kitos automatizavimo priemonės gautinų sumų ir pajamų cikle naudotų ne tik tradicinius senaties krepšelius ir DSO, bet ir elgsenos rodiklius, kurie tiesiogiai pritaikomi vidinio audito rizikos vertinimui.

1.9. RPA ir vidinio audito vaidmenys, kompetencijos ir valdymas

Robotinė procesų automatizacija (RPA) literatūroje vis dažniau apibūdinama kaip „skaitmeninis darbo jėgos“ tipas, kai programiniai robotai perima pasikartojančias, taisyklėmis grindžiamas užduotis, imituodami darbuotojų veiksmus informacinėse sistemose, teigia savo darbe PwC (2017). Vidinis auditas šiame kontekste atsiduria dvejopoje pozicijoje: viena vertus, RPA tampa nauju audituotinu objektu, kuris keičia organizacijos rizikos profilį ir kontrolės aplinką; kita vertus, pats vidinis auditas vis dažniau naudoja RPA kaip priemonę savo procedūroms automatizuoti, didinant testavimo apimtį ir mažinant rankinio darbo kiekį. Internal Audit Foundation (2019) pabrėžia, kad, norėdamas išlikti aktualus, vidinis

auditas turi ne tik vertinti RPA diegimo rizikas, bet ir pats aktyviai įsitraukti į šių technologijų panaudojimą audito veikloje.

Kuriant ir taikant RPA sprendimus vidiniame audite, išskiriami keli pagrindiniai auditoriaus vaidmenys. Pirma, vidinis auditas padeda identifikuoti procesus, tinkamus automatizavimui, ir įvertinti, kaip RPA pakeis esamą kontrolės struktūrą bei rizikų pasiskirstymą. PwC (2017) ir Deloitte (2025) publikacijose pažymima, kad automatizavimas gali būti taikomas visame audito gyvavimo cikle – nuo rizikos vertinimo ir plano sudarymo iki duomenų rinkimo, testų vykdymo ir ataskaitų rengimo – taip didinant audito aprėptį ir mažinant testavimo laiką. Antra, kaip teigia Toor H. (2020), brandesnėse organizacijose vidinis auditas pats kuria arba bendrai su IT ir verslo padaliniais prižiūri robotus, kurie reguliariai vykdo kontrolės testus (pavyzdžiui, ieško išskirčių sandorių, tikrina prieigos teises ar lygina sandorių duomenis su išoriniais šaltiniais). Tokiu atveju auditorius tampa ne vien „roboto naudotoju“, bet ir jo logikos bei rezultatų kokybės prižiūrėtoju.

RPA taikymas iš esmės keičia ir vidinių auditorių kompetencijų profilį. Stoica, ir Ionescu-Feleagă (2021) tyrimai apie skaitmenizaciją apskaitoje ir audite parodo, kad tradicinės finansinės atskaitomybės ir kontrolės žinios turi būti papildytos gilesniu procesų supratimu, duomenų analitikos įgūdžiais ir bent bazinėmis technologinėmis kompetencijomis – gebėjimu suprasti automatizavimo darbo eigas, parametrus ir duomenų srautus. Naujose studijose apie dirbtinio intelekto ir RPA kompetencijas viešajame sektoriuje, autoriai Aldemir, ir Uysal (2024), išskiria, kad auditoriai turi mokėti ne tik interpretuoti technologijų rezultatus, bet ir suprasti jų ribotumus, šališkumo rizikas, modelių mokymo logiką ir sprendimų paaiškinamumą. Tai nereiškia, kad kiekvienas auditorius turi tapti programuotoju, tačiau iš jo tikimasi gebėjimo diskutuoti su IT specialistais „bendru technologiniu kalbos lygiu“, suprasti robotų logiką ir įvertinti, ar automatizuotos procedūros iš tikrųjų atitinka audito tikslus ir profesinius standartus.

Augant RPA svarbai, vis daugiau dėmesio skiriama ir specifiniam RPA valdymui (governance) bei jo sąsajai su vidaus kontrolės modeliais. Euleric M., Gruene J., ir Wood D. (2024) autoriai pabrėžia, kad robotų diegimas be struktūruoto valdymo gali sukurti naujas rizikas – nuo neteisingai suprogramuotų ir plačiu mastu klaidas kartojančių robotų iki nepakankamos prieigos kontrolės ir pakeitimų valdymo. Eulerich, Waddoups, Wagener, ir Wood (2024) autorių publikacijoje siūloma RPA valdymo struktūrą susieti su COSO vidaus kontrolės sistema, aiškiai apibrėžiant „botų“ gyvenimo ciklą (sukūrimą, testavimą, diegimą, pakeitimų valdymą ir nutraukimą), atsakomybių paskirstymą, veiklos monitoringą ir žurnalinimą. Vidinis auditas šioje sistemoje atlieka dvejopą funkciją: konsultacinę – padeda kurti ir

tobulinti RPA valdymo politiką – ir užtikrinimo – vertina, ar ši politika realiai veikia ir ar RPA nepažeidžia vidaus kontrolės tikslų.

Empiriniai tyrimai, analizuojantys RPA įtaką vidiniam auditui, rodo, kad automatizacija gali tiek sustiprinti audito funkciją, tiek sukurti naujų iššūkių. Pavyzdžiui, Armijo-Perea, Sepúlveda, Velásquez, ir Villanueva (2025) tyrimas apie RPA diegimą didelėse įmonėse parodė, kad robotų įdiegimas vidiniame audite leidžia padidinti audito apimtį ir sumažinti laiko sąnaudas, tačiau tuo pačiu išryškina kompetencijų trūkumą ir valdymo spragas ten, kur RPA projektai vykdomi be aiškos strategijos ir kontrolės struktūros. Uklańska (2023) pastebi, kad skaitmeninis darbas apskaitoje ir audite keičia atsakomybių ribas tarp žmonių ir technologijų: dalis tradiciškai auditoriaus atliktų tikrinimo žingsnių persikelia į robotus, o auditorius vis labiau tampa „skaitmeninės kontrolės architektu“, vertinančiu ne tik dokumentus ir įrašus, bet ir algoritmus bei darbo eigas.

Šiame darbe nagrinėjamas EPDA metodas ir jo realizacija .NET bei RPA aplinkose natūraliai įsilieja į minėtą rolę ir kompetencijų kaitą. EPDA, kaip elgsenos rodiklis, sukuriamas automatizuotai – tačiau vidinis auditas turi suprasti, kaip jis skaičiuojamas, kokiais duomenimis remiasi, kokios daromos prielaidos ir kokią riziką gali kelti neteisingas modelio ar duomenų pritaikymas. Tokiu būdu RPA ir kiti automatizavimo įrankiai tampa ne tik technine priemone gautinų sumų ir pajamų ciklą vykdyti, bet ir nauju audito objektu, kuriam reikalingas aiškus valdymas, atskaitomybė ir atitinkamos auditoriaus kompetencijos.

2. TYRIMO METODIKA IR EPDA METODO MODELIS

Antrajame skyriuje nuosekliai aprašoma tyrimo metodika ir EPDA metodo modelis, kuriuo remiantis siekiama įgyvendinti darbo tikslą – ištirti numatomo ir faktinio mokėjimo vėlavimo (EPDA) taikymo galimybes gautinų sumų vidiniame audite ir palyginti skirtingus jo technologinius realizavimo būdus. Čia apibrėžiami tyrimo objektas, tikslas, uždaviniai ir ribotumai, pristatomas naudojamas Kaggle „Payment Date Prediction“ duomenų rinkinys ir atliktos duomenų paruošimo procedūros, detaliai išdėstomas EPDA skaičiavimo algoritmas ir tradicinis sąskaitų senaties metodas, o taip pat suformuojama tyrimo eiga bei eksperimento schema, numatant .NET ir „Power Automate Desktop“ prototipų palyginimą pagal pasirinktus vertinimo kriterijus.

2.1. Tyrimo objektas, tikslas, uždaviniai ir ribotumai

Šiame skyriuje detalizuojama tyrimo metodika, kuria remiantis siekiama įgyvendinti įvade suformuluotą darbo tikslą ir uždavinius. Darbo objektu pasirenkami gautinų sumų vidinio audito metu taikomi skaitiniai rodikliai ir technologiniai sprendimai, skirti mokėjimų vėlavimo rizikai įvertinti, ypatingą dėmesį skiriant numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodui ir jo realizacijai skirtingose technologinėse platformose - .NET programėlėje ir robotinio procesų automatizavimo „Power Automate Desktop“ (RPA) sprendime.

Darbo tikslas - ištirti EPDA metodo taikymo galimybes gautinų sumų vidiniame audite ir sukurti bei eksperimentiškai palyginti du šio metodo realizacijos prototipus - .NET programėlę ir RPA „Power Automate Desktop“ sprendimą. Šiam tikslui pasiekti įvade suformuluoti penki uždaviniai: ištirti literatūrą ir praktiką apie gautinų sumų auditą ir mokėjimų vėlavimo modeliavimą; apibendrinti .NET ir RPA taikymo galimybes audite; sukurti EPDA konceptualų modelį ir skaičiavimo algoritmą; sukurti .NET ir RPA prototipus ir atlikti eksperimentinį tyrimą; suformuluoti rekomendacijas vidiniam auditui.

Tyrimas turi kelis aiškius ribotumus. Pirma, naudojamas vienas viešai prieinamas, sintetinis, tačiau realistiškais duomenimis pagrįstas sąskaitų ir mokėjimų duomenų rinkinys, todėl rezultatai nėra tiesiogiai generalizuojami visoms šakoms ar konkrečioms įmonėms. Antra, EPDA metodas šiame darbe realizuojamas palyginti paprastu statistiniu principu (remiantis istoriniu vidutiniu vėlavimu), nenaudojant sudėtingesnių mašininio mokymosi algoritmų, nors literatūroje jie plačiai tiriami. Trečia,

lyginamos tik dvi technologinės platformos - .NET ir „Power Automate Desktop“, todėl gautos išvados apie technologijų privalumus ir ribotumus galioja šioms pasirinktoms priemonėms ir nagrinėjamam scenarijui.

2.2. Naudojamas duomenų rinkinys („Payment Date Prediction“ duomenys)

Empirinei tyrimo daliai naudojamas viešai prieinamas Kaggle platformoje paskelbtas „Payment Date Prediction for Invoices“ („Customer Invoices Dataset“) duomenų rinkinys. Kaggle aprašyme nurodoma, kad šiame rinkinyje pateikiami tokie laukai kaip kliento identifikaciniai duomenys, sąskaitos ir mokėjimo terminų datos bei mokėtina suma – tai yra, „kliento duomenys, sąskaitų datos, mokėjimo terminai ir mokėtinos sumos“ Kaggle (2023). Mokslinėje literatūroje šis rinkinys apibūdinamas kaip „realistic but synthetic structured invoices“, t. y. realistišką sąskaitų struktūrą ir elgseną imituojantis, tačiau dėl konfidencialumo ne iš konkrečios įmonės apskaitos paimtas duomenų rinkinys Charabuddi (2025).

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
business_code	cust_number	name_customer	clear_date	business_year	doc_id	posting_date	document_create_date	document_create_date.1	due_in_date	invoice_currency	document_type	posting_id	total_open_amount	baseline_create_date	cust_payment_terms	invoice_id	isOpen
U001	200769623	WAL-MAR corp	2/11/2020 0:00	2020	1930438491	1/26/2020	20200125	20200126	20200210	USD	RV	1	54273.28	20200126	NAH4	1930438491	0
U001	200980828	BEN E	8/8/2019 0:00	2019	1929646410	7/22/2019	20190722	20190722	20190811	USD	RV	1	79656.6	20190722	NAD1	1929646410	0
U001	200792734	MDW/ trust	12/30/2019 0:00	2019	1929873765	9/14/2019	20190914	20190914	20190929	USD	RV	1	2253.86	20190914	NAA8	1929873765	0
CA02	140105686	SYSC lic		2020	2960623488	3/30/2020	20200330	20200330	20200410	CAD	RV	1	3299.7	20200331	CA10	2960623488	1
U001	200769623	WAL-MAR foundation	11/25/2019 0:00	2019	1930147974	11/13/2019	20191113	20191113	20191128	USD	RV	1	33133.29	20191113	NAH4	1930147974	0
CA02	140106181	THE corporation	12/4/2019 0:00	2019	2960581231	9/20/2019	20190920	20190920	20190904	CAD	RV	1	22225.84	20190924	CA10	2960581231	0
U001	200769623	WAL-MAR in	11/12/2019 0:00	2019	1930083373	11/1/2019	20191031	20191101	20191116	USD	RV	1	7358.49	20191101	NAH4	1930083373	0
U001	200744019	TARIG us		2020	1930659387	3/19/2020	20200318	20200319	20200403	USD	RV	1	11173.02	20200319	NAH4	1930659387	1
U001	200769623	WAL-MAR corp	6/18/2019 0:00	2019	1929439637	6/7/2019	20190605	20190607	20190622	USD	RV	1	15995.04	20190607	NAH4	1929439637	0
U001	200762301	C&S WH systems	3/6/2019 0:00	2019	1928819386	2/20/2019	20190219	20190220	20190307	USD	RV	1	28.63	20190220	NAC6	1928819386	0
U001	200418007	AM		2020	1930610806	3/11/2020	20200306	20200311	20200326	USD	RV	1	3525.59	20200311	NAH4	1930610806	1
U001	200743129	BROOKS co	1/22/2019 0:00	2019	1928550622	1/2/2019	20190102	20190102	20190117	USD	RV	1	103147.37	20190102	NAH4	1928550622	0
U001	200188937	SYSC corporation	5/6/2019 0:00	2019	1929151655	4/15/2019	20190415	20190415	20190430	USD	RV	1	16381.45	20190415	NAH4	1929151655	0
U001	200721222	GO corporation	11/1/2019 0:00	2019	1930022117	10/17/2019	20191017	20191017	20191031	USD	RV	1	19581.57	20191017	NAH4	1930022117	0
U001	200739534	OK systems		2020	1930788296	4/15/2020	20200415	20200415	20200430	USD	RV	1	121105.65	20200415	NAH4	1930788296	1
U001	200533024	DECA corporation		2020	1930817482	4/23/2020	20200423	20200423	20200426	USD	RV	1	3726.06	20200416	NAM2	1930817482	1
U001	200794332	COST in	11/12/2019 0:00	2019	1930052739	10/25/2019	20191025	20191025	20191109	USD	RV	1	5181.51	20191025	NAH4	1930052739	0
U001	200881076	ALBERT corp	12/17/2019 0:00	2019	1930209407	12/2/2019	20191202	20191202	20191217	USD	RV	1	19082.14	20191202	NAH4	1930209407	0
U001	200769623	WAL-MAR trust	11/26/2019 0:00	2019	1930153511	11/15/2019	20191114	20191115	20191130	USD	RV	1	12819.5	20191115	NAH4	1930153511	0
U001	200769623	WAL-MAR systems	2/5/2020 0:00	2020	1930438462	1/24/2020	20200124	20200124	20200210	USD	RV	1	53349.54	20200124	NAH4	1930438462	0
U013	190053554	SYSTEMS systems	2/11/2020 0:00	2020	193037617	1/11/2020	20200107	20200111	20200210	USD	RV	1	5683.5	20200111	NAH4	193037617	0
U001	200783734	FAREW us	9/4/2019 0:00	2019	1929773400	8/21/2019	20190822	20190821	20190905	USD	RV	1	11845.19	20190821	NAH4	1929773400	0

2 pav. Kaggle duomenų rinkinio fragmentas

2 paveiksle pateikiamas naudojamo Kaggle duomenų rinkinio fragmentas, iliustruojantis pagrindinių laukų struktūrą. Šiame darbe toliau analizei ir EPDA metodo skaičiavimui reikšmingiausi yra šie stulpeliai:

business_code - verslo vieneto ar segmento kodas, leidžiantis identifikuoti, kuriam įmonės padaliniui priklauso sąskaita; šiame tyrime naudojamas tik pagalbinei duomenų segmentavimo analizei.

cust_number - kliento identifikatorius, naudojamas grupuoti sąskaitas pagal klientą ir apskaičiuoti istorinį tipinį mokėjimo vėlavimą, kuris yra EPDA metodo pagrindas.

name_customer - kliento pavadinimas; padeda identifikuoti klientą ataskaitose, tačiau tiesiogiai nenaudojamas skaičiavimuose.

doc_id - sąskaitos faktūros identifikatoriai, naudojami atskirų įrašų atsekamumui ir rezultatų siejimui su pradiniu duomenų rinkiniu.

business_year - sąskaitos finansiniai metai, leidžiantys atlikti analizę pagal metus ar laikotarpius, jei reikia.

posting_date - sąskaitos užregistravimo apskaitoje data; pagal ją galima vertinti laiką nuo užregistravimo iki apmokėjimo.

document_create_date / document_create_date.1 - dokumento sukūrimo datos, naudojamos duomenų kokybės patikrai ir, jei reikia, termino logikos analizei.

due_in_date - sąskaitos mokėjimo terminas; nuo jo skaičiuojamas tradicinis pradelstumo rodiklis sąskaitų senaties analizėje ir faktiškas vėlavimas, reikalingas EPDA skaičiavimui.

invoice_currency - sąskaitos valiuta, reikalinga užtikrinti, kad analizuojamos sumos būtų palyginamos (šiuo tyrimu naudojami tik įrašai viena valiuta).

total_open_amount - mokėtina sąskaitos suma; tai pagrindinis finansinis rodiklis, naudojamas vertinant, kokios sumos tenka skirtingiems senaties ir EPDA segmentams, taip pat, jei reikia, skaičiuojant sumomis svertinį tipinį vėlavimą.

baseline_create_date - bazinė data, nuo kurios skaičiuojamas mokėjimo terminas; naudojama tikrinti žingsnių logiką ir duomenų nuoseklumą.

cust_payment_terms - kliento atsiskaitymo sąlygų kodas, leidžiantis papildomai interpretuoti termino ir faktinio apmokėjimo santykį (pvz., ar klientui suteiktas ilgesnis ar trumpesnis terminas).

clear_date - faktinė sąskaitos apmokėjimo data, būtina apskaičiuoti faktišką mokėjimo vėlavimą ir EPDA nuokrypį.

isOpen - indikatorius, rodantis, ar sąskaita dar atidaryta (neapmokėta); šis laukas naudojamas atskirti apmokėtas ir neapmokėtas sąskaitas bei skirtingai skaičiuoti EPDA nuokrypį uždarytoms ir atviroms pozicijoms.

Likę stulpeliai šiame tyrime naudojami tik kontekstinei informacijai arba duomenų kokybės patikrai, todėl tiesiogiai neįtraukiami į EPDA ir tradicinio sąskaitų senaties metodo skaičiavimo algoritmus.

Šio duomenų rinkinio struktūra iš esmės atitinka realiose apskaitos sistemose ir gautinų sumų senaties ataskaitose naudojamą informaciją. Praktiniuose gautinų sumų senaties metodikos aprašymuose nurodoma, kad sudarant senaties ataskaitą pirmiausia surenkami pagrindiniai sąskaitų duomenys - „invoice date, customer name, amount due“ (sąskaitos data, kliento pavadinimas ir mokėtina suma) Tabs platform (2025). Apskaitos programų dokumentacijoje pažymima, kad standartinėse AR senaties (aging) ataskaitose kiekvienai sąskaitai pateikiamas „invoice number, invoice date, amount, due date“ (sąskaitos numeris, data, suma ir mokėjimo terminas), o toliau sumos grupuojamos į senaties intervalus Net Suite (2022). Taigi Kaggle rinkinyje esantys laukai - kliento identifikatorius, sąskaitos data, mokėjimo terminas, mokėtina suma ir faktinė mokėjimo data - visiškai sutampa su tais, kurie realiai naudojami gautinų sumų apskaitoje ir senaties ataskaitose.

Dėl šių priežasčių šiame darbe naudojamą Kaggle rinkinį laikome realistišką gautinų sumų situaciją imituojančiais sintetiniais duomenimis: jie neatspindi konkrečios įmonės, tačiau jų struktūra ir duomenų tipai atitinka tai, kas rekomenduojama sudarant senaties ataskaitas ir kas iš tiesų pateikiama komercinėse apskaitos sistemose Tabs platform (2025). Tokia prielaida leidžia pagrįstai naudoti šį rinkinį EPDA metodo ir technologinių prototipų (.NET ir „Power Automate Desktop“) testavimui, kartu aiškiai suvokiant ribotumą – gautos išvados iliustruoja metodo ir prototipų veikimą tipiškoje, bet ne konkrečiu ūkio subjektą atspindinčioje gautinų sumų situacijoje.

Papildomai šiame darbe buvo atliktos kelios tikslingos duomenų valymo ir struktūrizavimo procedūros, susijusios su Kaggle rinkinio ypatumais ir tyrimo tikslu. Pirmiausia buvo pašalintas stulpelis „area_business“, kadangi šiame tyrime EPDA metodo skaičiavimas ir gautinų sumų analizė remiasi kliento identifikatoriumi, sąskaitų datomis ir sumomis, o „area_business“ reikšmės nėra naudojamos nei EPDA algoritme, nei tradicinės senaties rodiklių skaičiavime. Siekiant išvengti nereikalingų, su nagrinėjamu reiškiniu tiesiogiai nesusijusių kintamųjų, šis stulpelis buvo eliminuotas iš tolesnės analizės.

Antra, buvo pasirinkta klientų identifikatorių vienodumu. Originaliame duomenų rinkinyje stulpelyje „cust_number“ dalis klientų buvo žymimi vien tik skaitiniais identifikatoriais, o dalis – skaičių ir simbolių deriniais (pvz., „CCCA02“, „CCU001“). Kadangi tyrime EPDA metodas taikomas kliento lygmeniu, grupuojant istorines sąskaitas pagal tą patį identifikatorių, tokie mišraus formato kodai galėjo reikšti kitokio tipo įrašus ar testinius duomenis ir apsunkinti rezultatų interpretaciją. Dėl šios priežasties

eilutės, kuriose „cust_number“ reikšmė nebuvo grynai skaitinė, buvo pašalintos, paliekant tik tuos įrašus, kuriuose klientų identifikatorius atitinka nuoseklų, vienodo tipo numeravimą. Tai leido užtikrinti, kad EPDA skaičiavimas būtų atliekamas homogeniškam klientų rinkiniui ir kad išvados apie mokėjimo elgseną nebūtų iškraipomos dėl mišraus kodavimo ar galimų dirbtinių įrašų. Po šių duomenų valymo veiksmų nagrinėjamo rinkinio apimtis sumažėjo nuo maždaug 50 000 įrašų iki 48 580 sąskaitų, kurios laikomos tinkamomis tolesniam EPDA ir tradicinės senaties metodų taikymui.

2.3. Numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodo skaičiavimo algoritmas

Numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodas šiame darbe suprantamas kaip elgsenos analitika grįstas skaičiavimo algoritmas, leidžiantis kiekvienam klientui nustatyti tipinį mokėjimo vėlavimą ir, remiantis juo, apskaičiuoti numatomą sąskaitos apmokėjimo datą. Tokiu būdu sukuriamas rodiklis, parodantis, kiek konkrečios sąskaitos elgsena nukrypsta nuo istorinio kliento mokėjimo modelio. Panašus principas – numatyti, kada tikėtina, jog konkreti sąskaita bus apmokėta, remiantis istorine atsiskaitymų informacija – taikomas ir ankstesniuose mokėjimų prognozavimo tyrimuose, kuriuose mašininio mokymosi modeliai naudojami prognozuoti sąskaitų apmokėjimo laiką bei paremti išieškojimo sprendimus Appel et al. (2019).

EPDA algoritmas sudaromas keliais nuosekliais žingsniais. Pirmiausia analizuojamame duomenų rinkinyje apskaičiuojamas faktiškas vėlavimas nuo termino datos kiekvienai apmokėtai sąskaitai. Tam naudojama termino ir faktinio apmokėjimo datų skirtumo metrika:

$$\text{DelayFromDue}_i = \text{PaymentDate}_i - \text{DueDate}_i,$$

kur DelayFromDue_i išreiškiamas dienomis ir gali būti tiek teigiamas (mokėjimas po termino), tiek neigiamas (ankstyvas apmokėjimas). Toliau visi įrašai grupuojami pagal klientą, ir kiekvienam klientui apskaičiuojama jo istorinė „tipinė“ vėlavimo reikšmė $E[\text{DelayFromDue}]_c$. Šiame darbe tipinis vėlavimas apibrėžiamas kaip vidurkis arba mediana, priklausomai nuo pasirinktų parametrų, o dalyje eksperimentų taip pat vertinamas mokėtina suma svertinis vidurkis, siekiant labiau pabrėžti didesnes sąskaitas.

Antrame etape ši istorinė informacija paverčiama numatoma apmokėjimo data. Kiekvienai analizuojamai sąskaitai i , kuriai žinoma termino data ir klientas c , apskaičiuojamas numatomas apmokėjimo momentas:

$$\text{ExpectedPaymentDate}_i = \text{DueDate}_i + E[\text{DelayFromDue}]_c.$$

Trečiame etape lyginama numatoma ir faktinė apmokėjimo data bei skaičiuojamas EPDA nuokrypis. Jei sąskaita jau apmokėta, naudojamas faktinis mokėjimo momentas:

$$\text{EPDADeviation}_i = \text{PaymentDate}_i - \text{ExpectedPaymentDate}_i.$$

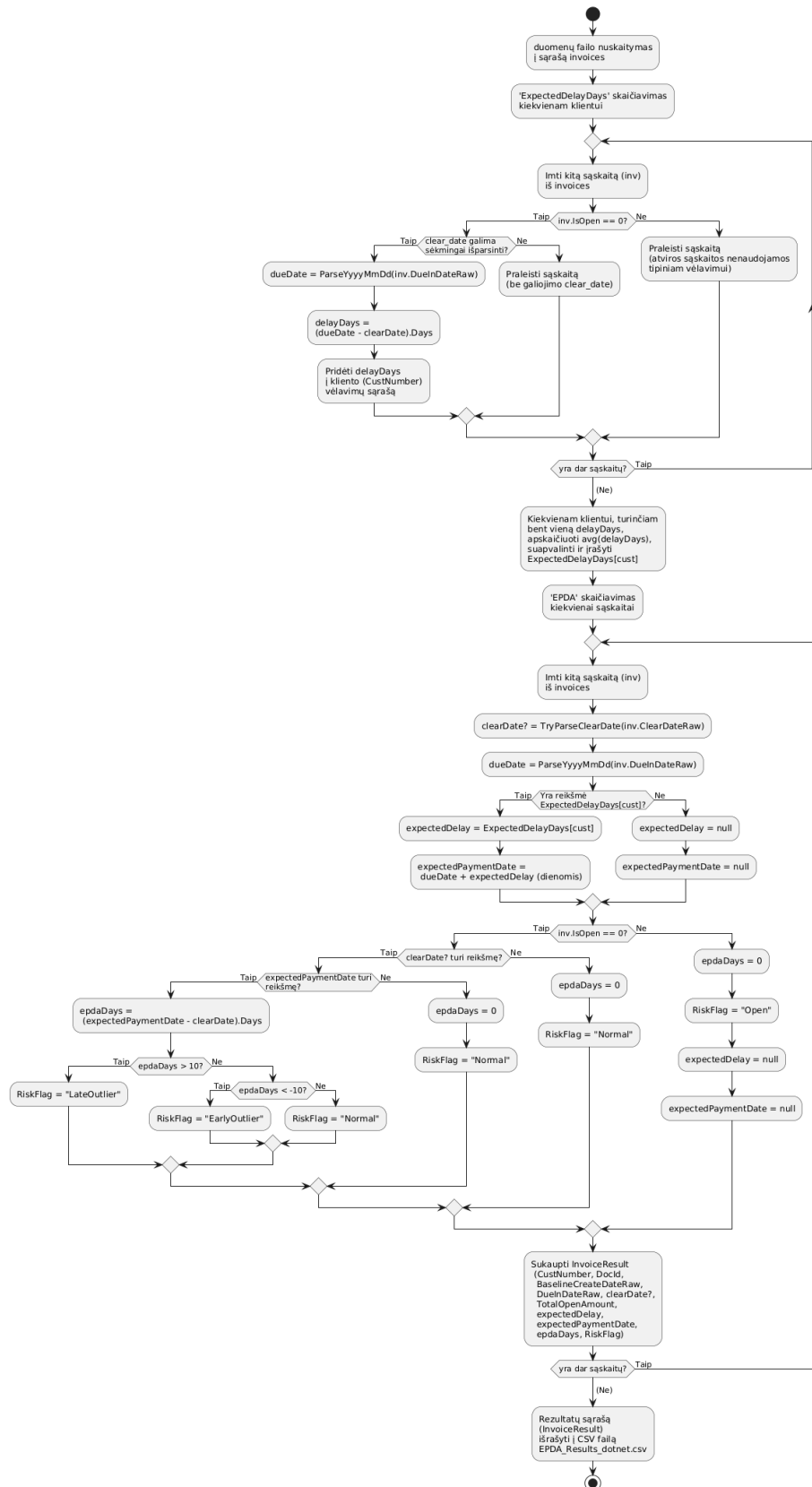
Jeigu sąskaita dar neapmokėta, EPDA nuokrypis apskaičiuojamas naudodant ataskaitinę datą (pavyzdžiui, pasirinktą analizės momentą):

$$\text{EPDADeviation}_i = \text{ReferenceDate} - \text{ExpectedPaymentDate}_i.$$

Teigiama EPDA nuokrypio reikšmė reiškia, kad sąskaita mokama vėliau, nei būtų tikėtina pagal istorinius kliento duomenis; neigiama reikšmė rodo, kad klientas atsiskaito greičiau nei įprastai. Absoliutus nuokrypio dydis parodo elgsenos anomalijos stiprumą – kuo didesnė reikšmė, tuo labiau konkreti situacija skiriasi nuo įprasto kliento mokėjimo modelio.

Praktikoje ne visi klientai turi pakankamą istorinių duomenų kiekį. Tyrime numatyta atsarginė taisyklė: jei klientui nustatyti patikimą $E[\text{DelayFromDue}]_c$ neįmanoma (pavyzdžiui, yra tik viena sąskaita), tokiam klientui pritaikomas bendras visų klientų tipinis vėlavimas arba, jei reikia, atitinkamos segmentuotos grupės (pvz., pagal kliento tipą) vidurkis. Taip užtikrinama, kad EPDA rodiklis būtų apskaičiuotas visoms sąskaitoms, o metodika išliktų vientisa. Be to, atsižvelgiant į audito praktikoje taikomą detalumo lygį, EPDA nuokrypio reikšmės galutiniuose rezultatuose dažniausiai pateikiamos dienomis, suapvalintomis iki sveikų skaičių, kad auditorius galėtų jas lengviau interpretuoti ir naudoti analitinėse procedūrose.

EPDA metodo skaičiavimo veiklos diagrama



3 pav. Siūlomo EPDA metodo skaičiavimo veiklos diagrama

3 paveiksle pateikta siūlomo EPDA metodo skaičiavimo veiklos diagrama. Pirmiausia iš duomenų failo nuskaitomas sąskaitų sąrašas, o tuomet apskaičiuojamas tipinis kiekvieno kliento mokėjimo vėlavimas: naudojamos tik uždarytos ($IsOpen = 0$) sąskaitos su teisingai užpildyta „clear_date“ reikšme, iš jų gaunamas vėlavimo dienų skirtumas ($due_date - clear_date$) ir apskaičiuojamas suapvalintas vidurkis kiekvienam klientui. Klientams, kurie neturi nė vienos apmokėtos sąskaitos, tipinis vėlavimas neskaičiuojamas – jų „ExpectedDelayDays“ ir „ExpectedPaymentDate“ vėliau paliekami tušti.

Antroje diagramos dalyje kiekvienai sąskaitai apskaičiuojamos EPDA reikšmės: pagal kliento tipinį vėlavimą (jei toks yra) nustatoma numatoma apmokėjimo data, ji lyginama su faktine apmokėjimo data, o skirtumas dienomis verčiamas EPDA rodikliu. Uždarytoms sąskaitoms, turinčioms ir tipinį vėlavimą, ir faktinę mokėjimo datą, pagal EPDA reikšmę priskiriama rizikos kategorija („Normal“, „EarlyOutlier“ arba „LateOutlier“). Sąskaitoms, kurios yra atviros ($IsOpen \neq 0$), EPDA dienų skaičius nenustatomas (laikomas 0), o rizikos požymis nustatomas kaip „Open“; tokiu atveju expected delay ir numatoma mokėjimo data paliekami tušti ir į rezultatų CSV failą išrašomi kaip tušti laukai.

2.4. Tradicinis sąskaitų senaties metodas ir lyginamosios metrikos

Siekiant objektyviai įvertinti EPDA metodo naudą, šiame tyrime jis lyginamas su tradiciniu sąskaitų senaties (angl. *accounts receivable aging*) metodu. Sąskaitų senaties ataskaita apibrėžiama kaip apskaitos dokumentas, kuris „klasifikuoja neapmokėtas klientų sąskaitas pagal laikotarpį, kiek ilgai jos yra neapmokėtos“, ir padeda įmonėms stebėti pradelstus mokėjimus, vertinti kredito riziką ir valdyti pinigų srautus HighRadius (2025). Tokiose ataskaitose gautinos sumos paprastai skirstomos į standartinius laikotarpius – 0–30, 31–60, 61–90 ir daugiau nei 90 dienų pradelstumo krepšelius Stripe (2024).

Remiantis Kaggle duomenų rinkiniu ir pasirinktąja ataskaitine data, šiame darbe kiekvienai sąskaitai apskaičiuojamas dienų pradelstumo rodiklis:

$$DaysPastDue_i = ReferenceDate - DueDate_i,$$

kur ReferenceDate - analizei pasirinktas momentas. Pagal šią reikšmę sąskaitos priskiriamos senaties intervalams (0–30, 31–60, 61–90, >90 dienų). Gauta senaties struktūra atspindi bendrą portfelio „senėjimo“ vaizdą ir leidžia nustatyti, kuri dalis gautinų sumų yra labiausiai pradelsta ir tikėtina, kad rizikingiausia. Tokia senaties analizė yra plačiai taikoma praktikoje ir rekomenduojama kaip pagrindinė priemonė organizuojant išieškojimo procesą bei vertinant gautinų sumų kokybę Zone and Co (2025).

EPDA ir senaties metodai šiame tyrime traktuojami kaip du tarpusavyje lyginami požiūriai į tą patį duomenų rinkinį. Pagrindinės lyginamosios metrikos yra:

- tradicinis pradelstumo rodiklis (DaysPastDue) ir jo pasiskirstymas tarp senaties krepšelių;
- EPDA nuokrypis (EPDADeviation) ir jo pasiskirstymas klientų ir sąskaitų grupėse.

Analizė orientuota į situacijas, kuriose senaties ir EPDA signalai skiriasi – pavyzdžiui, sąskaitas, kurios nėra seniausios pagal dienų pradelstumą, bet turi didelį teigiamą EPDA nuokrypį (rodo reikšmingą elgsenos pablogėjimą), ir atvirkščiai, senas sąskaitas, kurios EPDA požiūriu yra artimos kliento istoriniam vidurkiui. Tokie atvejai leidžia įvertinti, ar EPDA suteikia papildomą informaciją, palyginti su vien tradicine senaties ataskaita.

2.5. Tyrimo eiga ir eksperimento schema (.NET ir Power Automate Desktop prototipų palyginimas)

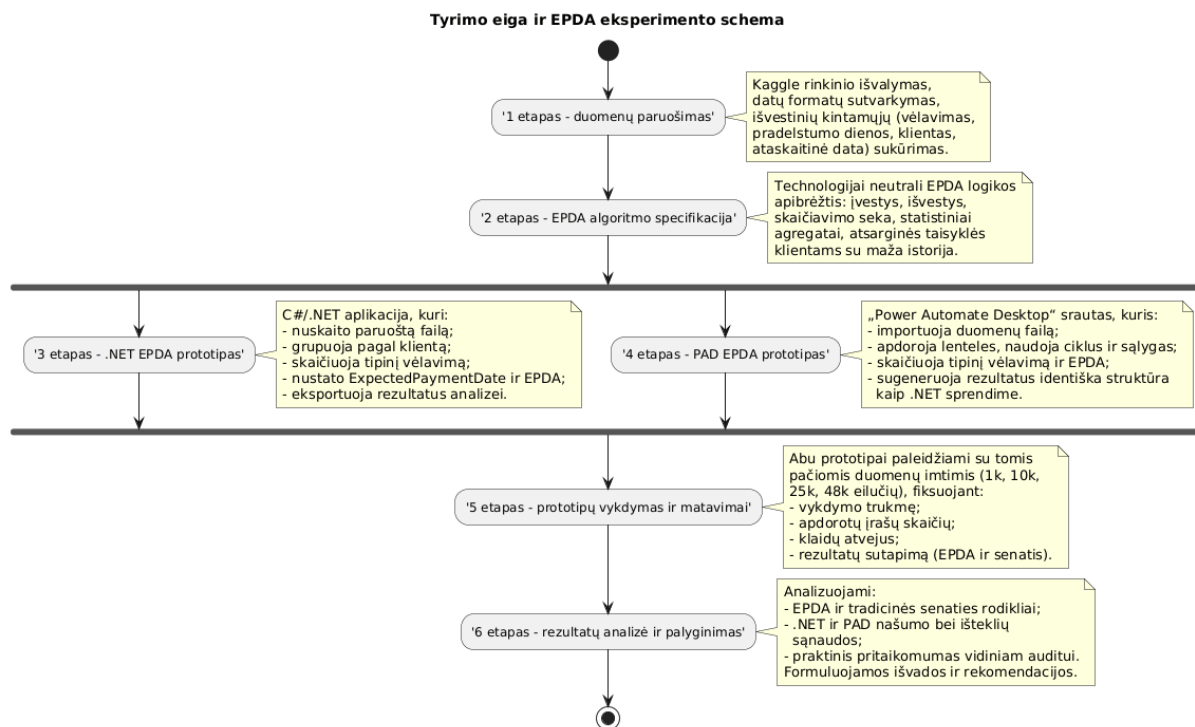
Tyrimo eiga suplanuota taip, kad leistų sistemiškai įgyvendinti EPDA metodą ir palyginti skirtingas jo technologines realizacijas vidinio audito kontekste. Pirmame etape atliekamas duomenų apdorojimas: Kaggle rinkinys išvalomas nuo akivaizdžių anomalijų, sutvarkomi datų formatai, sukuriami išvestiniai kintamieji (faktiškas vėlavimas, pradelstumo dienų skaičius, klientų identifikatoriai, ataskaitinė data). Šis paruoštas duomenų failas tampa bendru pagrindu abiem prototipams.

Antrame etape suformuluojamas EPDA skaičiavimo algoritmas technologijų atžvilgiu neutralia forma. Apibrėžiami įvesties ir išvesties laukai, skaičiavimo seka ir naudotini statistiniai agregatai (vidurkis, mediana, svertinis vidurkis), taip pat atsarginės taisyklės klientams su mažu istorinių duomenų kiekiu. Ši specifikacija vėliau ta pati taikoma tiek .NET programėlėje, tiek „Power Automate Desktop“ sraute, kad būtų užtikrintas rezultatų palyginamumas.

Trečiame etape sukuriamas .NET prototipas, realizuojantis EPDA algoritmą naudojant tipizuotas klases ir kolekcijų apdorojimą. Šiame prototipe įgyvendinamas duomenų nuskaitymas, grupavimas pagal klientą, tipinio vėlavimo skaičiavimas, numatomos apmokėjimo datos ir EPDA nuokrypio apskaičiavimas bei rezultatų eksportas į analizei tinkamą formatą. Ketvirtame etape kuriamas analogiškas „Power Automate Desktop“ prototipas: EPDA logika įgyvendinama naudojant šio įrankio veiksmus (lentelių apdorojimas, ciklai, sąlyginių išraiškų panaudojimas), taip, kad galutinis rezultatas (EPDA reikšmės ir senaties rodikliai) būtų identiškas .NET sprendimo rezultatams.

Penkto etapo metu abu prototipai paleidžiami su tomis pačiomis duomenų imtimis, fiksuojant kelis rodiklius: vykdymo trukmę, sėkmingai apdorotų įrašų skaičių, klaidų atsiradimo atvejus ir galutinio rezultato sutapimą (ar EPDA ir senaties reikšmės abiejuose prototipuose sutampa iki nustatyto tikslumo). Šie duomenys naudojami lyginamajai analizei. Galiausiai, šeštame etape EPDA ir tradicinės senaties išvadai lyginami analitiniu požiūriu, vertinant, ar EPDA rodiklis leidžia vidiniam auditui geriau identifikuoti rizikingas sąskaitas ir klientus, bei aptariama, kuris prototipas (.NET ar RPA) yra labiau tinkamas praktiniam taikymui, atsižvelgiant į audito aplinkos ypatumus.

Technologinio įgyvendinimo analizė remiasi bendromis RPA taikymo audite apžvalgomis, kuriose nurodoma, kad robotai gali automatizuoti duomenų surinkimą, sutikrinimą ir ataskaitų parengimą, o tai leidžia auditoriams daugiau dėmesio skirti analitiniam darbui PwC (2017). .NET platforma vertinama kaip tradicinis programinės įrangos kūrimo pasirinkimas, leidžiantis pasiekti didesnę našumą ir išplečiamumą, tačiau reikalaujantis daugiau IT kompetencijų.



4 pav. Tyrimo eiga ir EPDA eksperimento schema (.NET ir „Power Automate Desktop“ prototipų palyginimas)

Siekiant aiškiau pavaizduoti šiame poskyryje aprašytą tyrimo eigą ir eksperimento schemą, 4 pav. pateikta scheminė tyrimo žingsnių diagrama. Joje nuosekliai atvaizduojami pagrindiniai etapai: nuo Kaggle duomenų rinkinio paruošimo iki galutinės EPDA ir tradicinės senaties rezultatų analizės bei technologinių sprendimų palyginimo.

Pirmasis etapas apima duomenų paruošimą – Kaggle duomenų rinkinio išvalymą, datų formatų sutvarkymą ir išvestinių kintamųjų (faktiško vėlavimo, pradelstumo dienų, kliento identifikatoriaus, ataskaitinės datos) sukūrimą, taip suformuojant bendrą įvesties failą abiem prototipams. Antrajame etape suformuluojama technologijai neutrali EPDA algoritmo specifikacija: apibrėžiami įvesties ir išvesties laukai, skaičiavimo seka, naudojami statistiniai agregatai ir atsarginės taisyklės klientams su ribota istorija.

Trečiajame ir ketvirtajame etapuose ši specifikacija realizuojama dviejose technologinėse aplinkose – .NET aplikacijoje ir „Power Automate Desktop“ sraute. .NET prototipas EPDA logiką įgyvendina per tipizuotas klases ir kolekcijų apdorojimą, o RPA prototipas – per vizualius veiksmų blokus (lentelių apdorojimą, ciklus, sąlygines išraiškas), siekiant, kad abiejų sprendimų skaičiavimų

rezultatai sutaptų. Penktajame etape abu prototipai paleidžiami su tomis pačiomis duomenų imtimis, fiksuojant vykdymo trukmę, sėkmingai apdorotų įrašų skaičių, klaidų atvejus ir galutinio rezultato sutapimą; šie duomenys naudojami lyginamajai analizei. Galiausiai, šeštajame etape atliekama EPDA ir tradicinės senaties rezultatų analizė ir technologinių sprendimų palyginimas, vertinant, ar EPDA metodas tikrai suteikia papildomą naudą vidiniam auditui ir kuris iš prototipų (.NET ar RPA) geriau atitinka praktinius audito aplinkos reikalavimus.

2.6. Tyrimo metodai ir vertinimo kriterijai

Šiame darbe taikomas kelių tyrimo metodų derinys, apimantis tiek teorinę, tiek empirinę analizę. Pirmiausia atliekama mokslinės literatūros ir praktinių šaltinių analizė, skirta gautinų sumų audito, sąskaitų senaties metodikos, mokėjimų prognozavimo ir procesų automatizavimo temoms. Literatūra apie sąskaitų senaties ataskaitas padeda identifikuoti tradicinius rodiklius ir jų ribotumus – daugelyje šaltinių pabrėžiama, kad senaties ataskaita klasifikuoja neapmokėtas sąskaitas pagal pradelstumo trukmę ir padeda „įvertinti kredito riziką bei valdyti pinigų srautus“, bet neapima elgsenos analizės HighRadius (2025). Tuo tarpu mokėjimų prognozavimo tyrimai rodo, kad remiantis istorine sąskaitų ir mokėjimų informacija galima sukurti modelius, prognozuojančius, kada tikėtina, jog sąskaita bus apmokėta, ir taip geriau prioritetizuoti išieškojimą Appel et al. (2020).

Antra, taikomas konstruktyvusis (artefaktų kūrimo) metodas: sukuriami du EPDA realizacijos prototipai - .NET ir „Power Automate Desktop“ sprendimai, kuriuose įgyvendinama ta pati skaičiavimo logika. Tai atitinka dizaino mokslo principus, kai naujas arba adaptuotas metodas tikrinamas kuriant veikiančią artefaktą ir vertinant jo tinkamumą praktiniame kontekste.

Trečia, vykdomas eksperimentinis tyrimas ir lyginamoji analizė. Abiem prototipams pateikiamas tas pats Kaggle duomenų rinkinys; palyginama, ar skiriasi EPDA skaičiavimo rezultatai, taip pat matuojamas vykdymo laikas ir resursų sąnaudos. Papildomai atliekama rezultatų analizė, lyginant EPDA rodiklį su tradiciniu senaties skirstymu ir vertinant, ar EPDA padeda išryškinti tokias elgsenos anomalijas, kurias senaties ataskaita „užmaskuoja“. RPA taikymo audite apžvalgose nurodoma, kad automatizacija gali didinti audito efektyvumą ir aprėptį, ypač kai naudojami dideli duomenų rinkiniai ir pasikartojančios procedūros PwC (2017). Šio tyrimo kontekste tai leidžia įvertinti, ar pasirinktos technologinės priemonės iš tiesų suteikia vidiniam auditui laiko ir kokybės pranašumų.

Vertinimo kriterijai apima:

- funkcinį teisingumą - ar .NET ir RPA prototipai generuoja vienodas EPDA ir senaties reikšmes pagal bendrą algoritmą;
- našumą - laikas ir resursai, reikalingi apdoroti tas pačias duomenų imtis;
- lankstumą ir techninį išplečiamumą - prototipų pritaikymo kitiems duomenų rinkiniams ar papildomiems rodikliams galimybės;
- panaudojamumą audito praktikoje - kiek lengvai auditorius gali suprasti ir paaiškinti skaičiavimų logiką, integruoti sprendimą į savo darbo eigą, bei kiek sprendimas priklauso nuo IT specialistų.

Remiantis šiais metodais ir kriterijais, tolimesniuose darbo skyriuose vertinama EPDA metodo nauda gautinų sumų vidiniam auditui ir pateikiami argumentai, kada labiau tikslinga rinktis .NET, o kada - RPA pagrįstą realizaciją.

3. EPDA METODO REALIZACIJA .NET IR RPA PROTOTIPUOSE

Antrame skyriuje buvo aprašytas numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodas, Kaggle duomenų rinkinys ir planuojama tyrimo eiga. Šiame skyriuje pereinama prie praktinio lygmens - detaliau pristatomi sukurti .NET ir „Power Automate Desktop“ (RPA) prototipai, įgyvendinantys EPDA logiką, aptariama jų architektūra, funkcinės savybės ir praktinis taikymas. Taip pat aprašomas atliktas eksperimentas, kuriame lyginami EPDA ir tradicinio sąskaitų senaties metodo rezultatai, bei analizuojami dviejų technologinių sprendimų - .NET ir RPA - privalumai ir ribotumai vidinio audito kontekste.

Skyriaus struktūra atitinka tyrimo logiką: pirmiausia pristatomas .NET prototipas, tuomet - „Power Automate Desktop“ sprendimas, po to nagrinėjamas EPDA ir senaties metodų rezultatų palyginimas ir galiausiai aptariama, kaip, remiantis eksperimentu, vertintini skirtingi technologiniai sprendimai audito praktikai.

3.1. .NET prototipo architektūra ir funkcionalumas

.NET prototipas sukurtas kaip atskira aplikacija, skirta automatizuotai apskaičiuoti EPDA rodiklius ir sugeneruoti analizei tinkamas ataskaitas. Sprendimas paremtas sluoksnine architektūra, išskiriant duomenų nuskaitymo, duomenų apdorojimo ir ataskaitų generavimo dalis. Toks funkcijų atskyrimas atitinka gerąsias programų inžinerijos praktikas ir sudaro sąlygas ateityje nesunkiai plėsti ar keisti atskiras logikos dalis, nepaveikiant visos sistemos.

Duomenų nuskaitymo sluoksnis. Prototipas dirba su Kaggle duomenų rinkinio failais (CSV ar Excel formatu), kuriuose pateikiami klientų identifikatoriai, sąskaitų datos, mokėjimo terminai, faktinės apmokėjimo datos ir sumos. Duomenys nuskaityti naudojant .NET bibliotekas, leidžiančias patikimai apdoroti struktūrizuotus failus, atlikti tipų konvertavimą (pavyzdžiui, datų laukų) ir suformuoti tipizuotas objektų kolekcijas. Kiekviena sąskaita atitinkama `Invoice` ar panašios klasės egzemplioriui, kuriame laikoma visa reikalinga informacija vėlesniems skaičiavimams.

Duomenų apdorojimo ir EPDA skaičiavimo sluoksnis. Šio sluoksnio paskirtis – įgyvendinti antrame skyriuje aprašytą EPDA algoritmą. Pirmiausia, naudojant LINQ ir kolekcijų operacijas,

sąskaitos susiejamos su atitinkamais klientais ir apskaičiuojamas faktinis vėlavimas nuo termino datos kiekvienai sąskaitai, kuri turi mokėjimo datą. Tuomet įrašai grupuojami pagal klientą ir, remiantis istorine informacija, kiekvienam klientui nustatomas tipinis vėlavimo dydis (vidurkis, mediana ar svertinis vidurkis, priklausomai nuo pasirinktos konfigūracijos). Šis dydis naudojamas numatomos apmokėjimo datos skaičiavimui ir EPDA nuokrypio nustatymui.

Atskira logikos dalis skirta išimtiniais atvejams – klientams, neturintiems pakankamai istorinių duomenų. Tokiems klientams taikomas bendras visų klientų tipinis vėlavimas arba, esant poreikiui, bendros klientų grupės (pavyzdžiui, pagal tam tikrą segmentavimo kriterijų) vidurkis. Taip užtikrinama, kad EPDA rodiklis būtų apskaičiuojamas visoms sąskaitoms, o algoritmas neišskaidytų duomenų į „apdorotus“ ir „neapdorotus“ segmentus.

Rezultatų ir ataskaitų sluoksnis. Baigus skaičiavimus, prototipas suformuoja rezultatų lentelę, kurioje kiekvienai sąskaitai pateikiami šie laukai: klientas, sąskaitos numeris, sąskaitos ir termino datos, faktinė apmokėjimo data (jei yra), tradicinis pradelstumo rodiklis (DaysPastDue), tipinis kliento vėlavimas ($E[\text{DelayFromDue}]$), numatoma apmokėjimo data (ExpectedPaymentDate) ir EPDA nuokrypis (EPDADeviation). Šie duomenys eksportuojami į failą, kurį auditorius gali toliau analizuoti, filtruoti, vizualizuoti ar naudoti kaip įrodymą audito dokumentuose.

Architektūriškai .NET prototipas suprojektuotas taip, kad būtų lengva keisti EPDA skaičiavimo parametrus (pavyzdžiui, vidurkį į medianą, apvalinimo taisykles, ataskaitinę datą) nekeičiant visos programos struktūros. Tai suteikia lankstumo adaptuoti metodiką skirtingiems audito objektams ir naudotojų poreikiams.

```

static List<InvoiceResult> ComputeEpda(
    List<Invoice> invoices,
    Dictionary<long, double> expectedDelayByCustomer)
{
    var results = new List<InvoiceResult>();

    foreach (var inv in invoices)
    {
        // Bandom apdoroti faktinę apmokėjimo datą (clear_date);
        // jei jos nėra arba ji neteisinga - laikome null
        DateTime? clearDate = TryParseClearDate(inv.ClearDateRaw, out var cdt)
            ? cdt
            : (DateTime?)null;

        // Sąskaitos mokėjimo terminas (due date)
        var dueDate = ParseYyyyMmDd(inv.DueInDateRaw);

        // Surandame šio kliento tipinį vėlavimą (ExpectedDelayDays)
        double? expectedDelay = null;
        if (expectedDelayByCustomer.TryGetValue(inv.CustNumber, out var ed))
        {
            expectedDelay = ed;
        }

        // Numatomos apmokėjimo datos skaičiavimas:
        // ExpectedPaymentDate = due_date + ExpectedDelayDays
        DateTime? expectedPaymentDate = null;
        if (expectedDelay.HasValue)
        {
            expectedPaymentDate = dueDate.AddDays(expectedDelay.Value);
        }

        int epdaDays = 0;
        string riskFlag;

        if (inv.IsOpen == 0 && clearDate.HasValue && expectedPaymentDate.HasValue)
        {
            // Uždaryta sąskaita ir turime tiek faktinę, tiek numatomą datą:
            // EPDA = ExpectedPaymentDate - ClearDate
            epdaDays = (int)(expectedPaymentDate.Value - clearDate.Value).TotalDays;

            // Rizikos kategorijos priskyrimas pagal EPDA nuokrypį
            if (epdaDays > 10)
            {
                riskFlag = "LateOutlier"; // žymiai vėliau nei įprasta
            }
            else if (epdaDays < -10)
            {
                riskFlag = "EarlyOutlier"; // žymiai anksčiau nei įprasta
            }
            else
            {
                riskFlag = "Normal"; // elgsena artima istoriniam vidurkiui
            }
        }
        else if (inv.IsOpen == 0 && clearDate.HasValue && !expectedPaymentDate.HasValue)
        {
            // Uždaryta sąskaita, bet šiam klientui nebuvo galima apskaičiuoti tipinio vėlavimo
            // (pvz., trūksta istorinių duomenų). EPDA čia nenustatomas - laikome 0 ir "Normal".
            epdaDays = 0;
            riskFlag = "Normal";
        }
        else
        {
            // IsOpen != 0 -> atvira sąskaita:
            // pagal pasirinktą taisyklę EPDA = 0, rizikos žyma - "Open",
            // o tipinis vėlavimas ir numatoma apmokėjimo data neapskaičiuojami.
            epdaDays = 0;
            riskFlag = "Open";
            expectedDelay = null;
            expectedPaymentDate = null;
        }

        // Sukuriame rezultatų įrašą, kuris vėliau išrašomas į EPDA_Results_dotnet.csv
        results.Add(new InvoiceResult(
            CustNumber: inv.CustNumber,
            DocId: inv.DocId,
            BaselineCreateDateRaw: inv.BaselineCreateDateRaw,
            DueInDateRaw: inv.DueInDateRaw,
            ClearDate: clearDate,
            TotalOpenAmount: inv.TotalOpenAmount,
            ExpectedDelayDays: expectedDelay,
            ExpectedPaymentDate: expectedPaymentDate,
            EPDA_Days: epdaDays,
            RiskFlag: riskFlag
        ));
    }

    return results;
}

```

5 pav. .NET prototipo EPDA skaičiavimo kodo ištrauka

5 paveiksle pateikta ComputeEpda metodo kodo ištrauka realizuoja pagrindinę .NET prototipo verslo logiką – EPDA rodiklio apskaičiavimą kiekvienai sąskaitai ir rizikos kategorijos priskyrimą. Metodas priima du pagrindinius parametrus: sąrašą sąskaitų invoices ir žemėlapi expectedDelayByCustomer, kuriame pagal kliento identifikatorių (CustNumber) saugomas tipinis to kliento mokėjimo vėlavimas dienomis (ExpectedDelayDays), anksčiau apskaičiuotas pagal istorinius duomenis. Metodo grąžinama reikšmė – List<InvoiceResult> – yra struktūruotas rezultatų sąrašas, kuriame kiekvienai sąskaitai pateikiami tiek EPDA skaičiavimo, tiek rizikos įvertinimo rezultatai.

Pirmoje metodo dalyje, cikle foreach, einama per kiekvieną į sąrašą invoices patenkančią sąskaitą. Kiekvienam įrašui iš pradžių bandoma išparsinti faktinę apmokėjimo datą (clear_date) funkcija TryParseClearDate. Jei data yra korektiška, ji priskiriama kintamajam clearDate, kitu atveju priskiriama null. Taip iš anksto pasirūpinama, kad tolesniuose skaičiavimuose būtų aiškiai atskiriamos sąskaitos, kurios turi galiojančią apmokėjimo datą, nuo tų, kuriose šios informacijos trūksta. Toliau iš tekstinio lauko DueInDateRaw su funkcija ParseYyyyMmDd gaunama sąskaitos mokėjimo termino data dueDate, kuri naudojama tiek tradiciniam pradelstumo skaičiavimui, tiek numatomai apmokėjimo datai nustatyti.

Tuomet metodu TryGetValue bandoma žemėlapyje expectedDelayByCustomer rasti tipinį šio kliento vėlavimą dienomis. Jei toks įrašas egzistuoja, jo reikšmė priskiriama expectedDelay kintamajam, kitu atveju expectedDelay paliekamas null. Ši logika atitinka teorinę prielaidą, kad EPDA metodas gali būti patikimai taikomas tik tiems klientams, kuriems turima pakankamai istorinių apmokėjimų duomenų; jei istorijos nėra, tipinis vėlavimas neišvedamas. Turint dueDate ir kliento expectedDelay, skaičiuojama numatoma apmokėjimo data expectedPaymentDate – tai daroma prie termino datos pridendant tipinį vėlavimą dienomis (dueDate.AddDays(expectedDelay.Value)), kaip ir buvo aprašyta 2.3 skyriuje.

Toliau metoduose naudojamas kintamasis epdaDays ir tekstinis laukas riskFlag, kurie bus naudojami EPDA rodikliui ir rizikos kategorijai saugoti. Pagrindinė sprendimų logika sutelkta trijuose if/else blokuose. Pirmuoju atveju tikrinama, ar sąskaita yra uždaryta (IsOpen == 0), ar turi faktinę apmokėjimo datą (clearDate.HasValue) ir ar yra apskaičiuota numatoma apmokėjimo data (expectedPaymentDate.HasValue). Jeigu visos šios sąlygos tenkinamos, laikoma, kad turime idealų scenarijų EPDA skaičiavimui: tiek faktinę, tiek numatomą datas. Tokiu atveju EPDA nuokrypis apskaičiuojamas kaip šių datų skirtumas dienomis:

$$\text{EPDA_Days} = (\text{ExpectedPaymentDate} - \text{ClearDate}).$$

Jei rezultatas teigiamas (daugiau kaip 10 dienų), laikoma, kad tai „LateOutlier“ – sąskaita apmokėta gerokai vėliau, nei būtų tikėtina pagal istoriją. Jei rezultatas neigiamas ir mažesnis nei –10, ji klasifikuojama kaip „EarlyOutlier“ – apmokėjimas įvyko žymiai anksčiau nei įprasta. Tarpiniame intervale (nuo –10 iki +10 dienų) sąskaita laikoma „Normal“, t. y. kliento elgsena artima istoriniam vidurkiui. Šis slenksčių pagrindu veikiantis klasifikavimas leidžia automatizuotai išskirti netipinius atvejus, su kuriais auditorius gali dirbti detaliau.

Antrajame loginio bloko variante (else if) nagrinėjamos tokios sąskaitos, kurios yra uždarytos ir turi faktinę apmokėjimo datą, tačiau neturi apskaičiuotos ExpectedPaymentDate reikšmės, nes klientui nebuvo įmanoma nustatyti tipinio vėlavimo (trūksta istorinių duomenų). Tokiu atveju EPDA metodas negali įvertinti nuokrypio nuo įprasto elgesio, todėl kodo realizacijoje nusprendžiama tokioms sąskaitoms priskirti epdaDays = 0 ir rizikos žymą „Normal“. Tai konservatyvi ir aiškiai dokumentuota taisyklė: jei nėra pakankamai istorijos, sąskaita elgsenos požiūriu nelaikoma nei gerokai blogesne, nei gerokai geresne už vidurkį. Tokia situacija taip pat išryškina vieną iš EPDA ribotumų – būtinybę turėti pakankamą istorinių duomenų kiekį.

Trečiasis else blokas apima visas kitas situacijas, visų pirma – atviras sąskaitas (IsOpen != 0). Pagal pasirinktą metodikos logiką, tokioms sąskaitoms EPDA dienų skaičius nenustatomas (paliekamas 0), rizikos žyma nustatoma „Open“, o tipinis vėlavimas ir numatoma apmokėjimo data nustatomi į null. Taip aiškiai atskiriamos sąskaitos, kurios dar nėra apmokėtos, nuo tų, dėl kurių jau galima daryti išvadas apie kliento elgseną. Vėliau, generuojant rezultatų failą ir analizuojant EPDA reikšmes, auditorius gali atskirai filtruoti atviras sąskaitas ir vertinti jų riziką taikant kitus kriterijus.

Kiekvieno ciklo pabaigoje sukuriamas InvoiceResult struktūros egzempliorius, kuriame sujungiama pradinė sąskaitos informacija (kliento numeris, dokumento ID, datos, suma) ir naujai apskaičiuoti laukai – ExpectedDelayDays, ExpectedPaymentDate, EPDA_Days ir RiskFlag. Šie rezultatai kaupiami sąrašė results, kuris pasibaigus ciklui grąžinamas ir vėliau išrašomas į CSV failą EPDA_Results_dotnet.csv. Taip sukuriamas vienos išvestinės lentelės, kuri audito požiūriu gali būti naudojama kaip analitinių procedūrų pagrindas: pagal RiskFlag galima greitai identifikuoti vėlyvus outlier'ius, ankstyvus netipinius apmokėjimus ar tiesiog normalų klientų elgesį, o EPDA_Days leidžia kiekybiškai įvertinti, kiek dienų konkreti sąskaita nukrypsta nuo istorinio kliento mokėjimo modelio.

Apibendrinant, ši kodo ištrauka aiškiai parodo, kaip teoriniame 2 skyriuje aprašytas EPDA algoritmas – tipinio vėlavimo apskaičiavimas, numatomos apmokėjimo datos nustatymas ir nuokrypio įvertinimas – yra realizuojamas praktiniame .NET prototipe. Ji taip pat parodo, kaip algoritme tvarkomasi su tipinėmis duomenų situacijomis (uždarytos sąskaitos su istorija, uždarytos be istorijos, atviros sąskaitos) ir kaip šios techninės taisyklės virsta aiškiais, auditoriams interpretuojamais rizikos žymėjimais rezultatuose.

3.2. „Power Automate Desktop“ (RPA) prototipo architektūra ir funkcionalumas

Antrasis sukurtas prototipas – RPA sprendimas, paremtas „Power Automate Desktop“ (PAD) įrankiu. Ši technologija leidžia automatizuoti pasikartojančias, taisyklėmis grįstas užduotis, imituojuant vartotojo veiksmus kompiuterio ekrane: atidaryti failus, kopijuoti ir įklijuoti duomenis, vykdyti skaičiavimus, generuoti ataskaitas. Kaip nurodoma RPA ir vidinio audito apžvalgose, tokie robotai padeda sumažinti rankinį darbą, išplėsti audito aprėptį ir leisti auditoriams koncentruotis į analitinį vertinimą, o ne technines užduotis Metricstream (2020).

EPDA logika „Power Automate Desktop“ aplinkoje įgyvendinama per kelias logines blokų grandines:

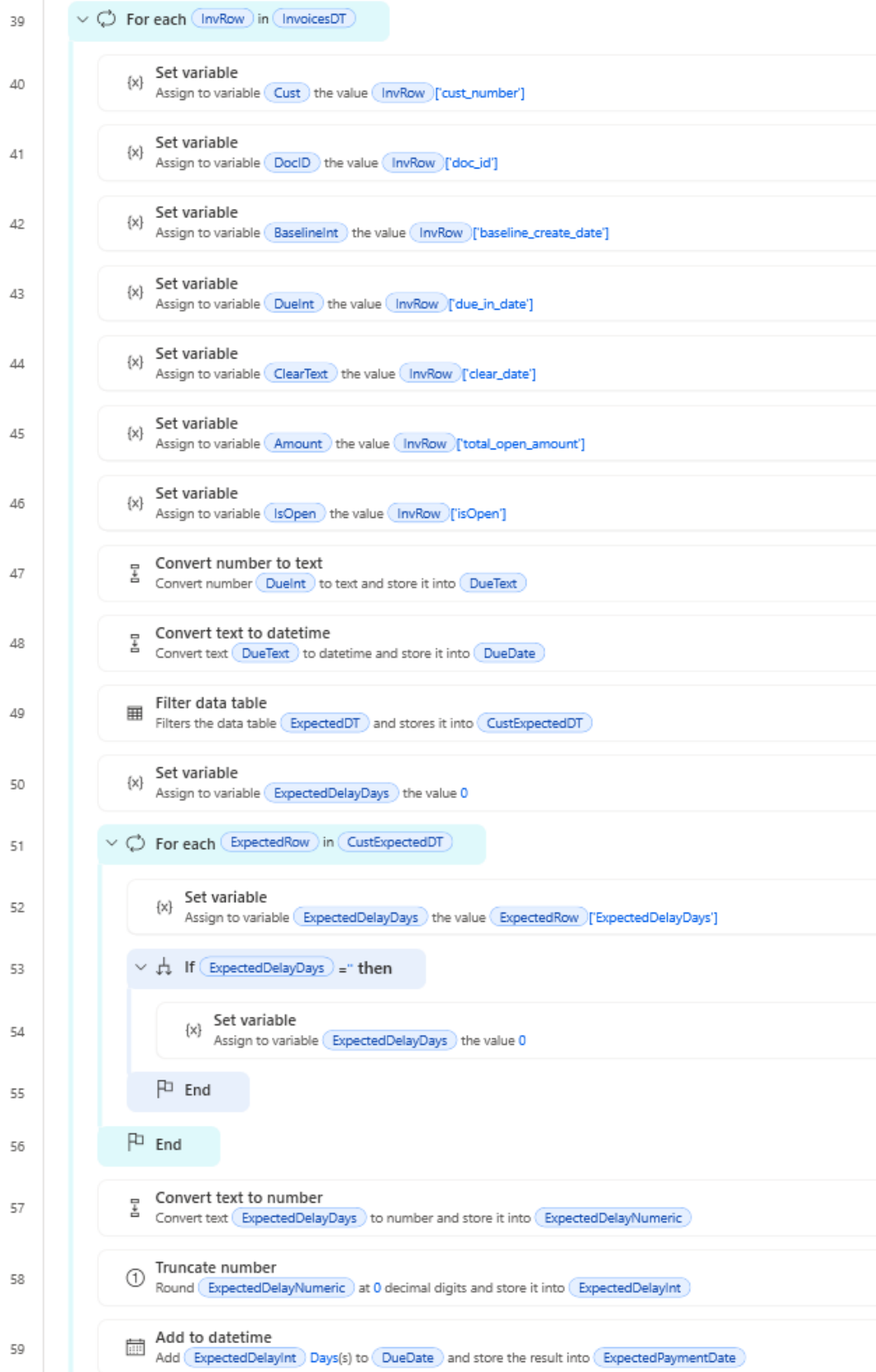
Duomenų nuskaitymas ir paruošimas. Pirmasis srauto blokas atsakingas už Kaggle duomenų failo atidarymą ir importavimą į VIDINES duomenų struktūras (lenteles). PAD veiksmas leidžia nuskaityti CSV ar Excel failą į kintamuosius, kuriuose toliau vykdomas filtravimas, datų konvertavimas ir pradinė validacija. Šiame etape sukuriami tarpiniai stulpeliai, tokie kaip faktiškas vėlavimas nuo termino datos, kurie vėliau naudojami tipinio vėlavimo skaičiavimui.

EPDA skaičiavimo logika. Kadangi „Power Automate Desktop“ nesiūlo tokio patogaus grupavimo kaip aukštesnio lygio programavimo kalbos, kliento istorijai apdoroti naudojami ciklai ir sąlyginės išraiškos. Vienas iš įgyvendinimo būdų – iš anksto susirūšiuoti lentelę pagal klientą ir datą, tuomet iteruojant per sąrašą agreguoti vėlavimus tam pačiam klientui (kaupiant sumas ir skaičių), o pasikeitus klientui – apskaičiuoti vidurkį ir priskirti jį atitinkamoms sąskaitoms. Tokios konstrukcijos gali būti techniškai sudėtingesnės nei LINQ užklausos .NET aplinkoje, tačiau jos leidžia EPDA logiką išreikšti vizualinių veiksmų sekomis, suprantamesnėmis auditoriams, neturintiems gilių programavimo žinių.

Numatomos datos ir EPDA nuokrypio skaičiavimas. Turint tipinį kliento vėlavimą, tolesni PAD veiksmai kiekvienai sąskaitai apskaičiuoja numatomą apmokėjimo datą, o tuomet – EPDA nuokrypį, naudojant pasirinktas datų operacijas ir aritmetines išraiškas. Kaip ir .NET prototipe, atliekamas atskiras apdorojimas klientams su mažu istorinių įrašų skaičiumi – jiems taikomas bendras vidutinis vėlavimas. Skaičiavimai gali būti atliekami tiek lentelių stulpelių lygmeniu, tiek per ciklus, užpildant naujus stulpelius su apskaičiuotomis reikšmėmis.

Rezultatų išvestis. Baigus skaičiavimus, RPA srautas sukuria naują rezultatų failą (pavyzdžiui, Excel ar CSV), kuriame analogiškai .NET prototipui pateikiami visi pagrindiniai laukai: klientas, sąskaitos numeris, datos, pradelstumo dienos, tipinis vėlavimas, numatoma apmokėjimo data ir EPDA nuokrypis. Šis failas gali būti automatiškai išsaugomas nurodytoje vietoje, išsiunčiamas el. paštu ar integruojamas į tolesnius audito darbo srautus.

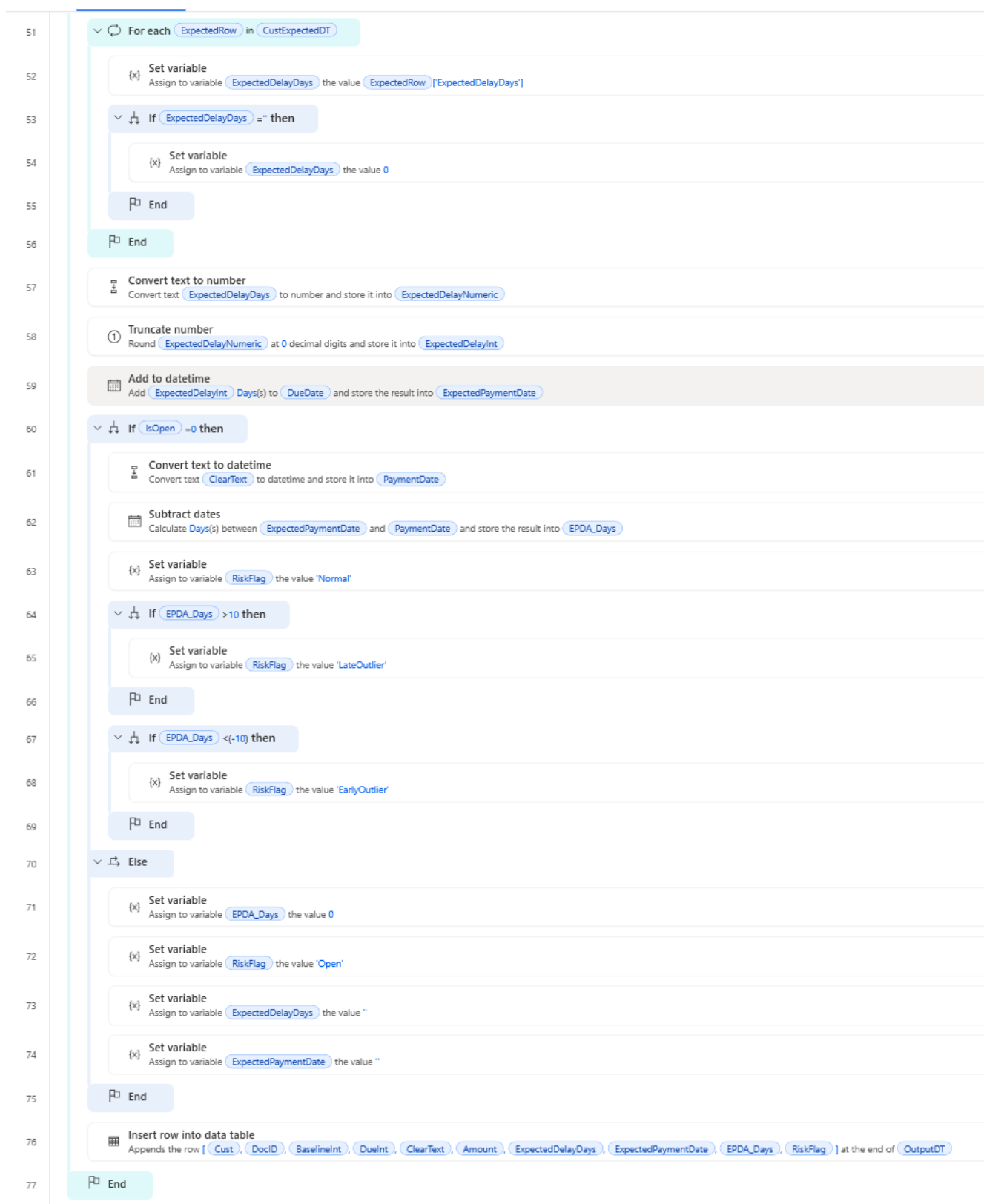
Svarbi „Power Automate Desktop“ pritaikymo ypatybė – tai, kad jis veikia kaip „pilietinės“ (angl. *citizen*) automatizacijos priemonė: auditoriai ar kiti verslo naudotojai, turintys bazinių IT žinių, gali patys kurti ir prižiūrėti srautus be pilno programavimo projekto. Tačiau literatūroje pažymima, kad tokie sprendimai ypač tinkami vidutinio sudėtingumo ir apimties procesams bei ten, kur daug sąveikos su grafinėmis vartotojo sąsajomis ar „Office“ aplinka, tuo tarpu didelio masto, našumo reikalaujantiems scenarijams dažnai pasirenkamos labiau programuotojams orientuotos platformos ar tradicinės aplikacijos.



6 pav. „Power Automate Desktop“ EPDA skaičiavimo logika (1 dalis)

6 paveiksle pavaizduota pirmoji EPDA skaičiavimo srauto dalis „Power Automate Desktop“ aplinkoje, kurioje apdorojamas kiekvienas duomenų rinkinio įrašas. Ciklas „For each (InvRow in InvoicesDT)“ paeiliui ima sąskaitas iš pradinės lentelės InvoicesDT ir jų reikšmes priskiria atskiriems kintamiesiems: Cust (kliento numeris), DocId (sąskaitos numeris), BaselineInt (bazinė data), DueInt (mokėjimo terminas), ClearText (faktinė apmokėjimo data tekstu), Amount (sąskaitos suma) ir IsOpen (indikatorius, ar sąskaita dar atvira). Tokiu būdu vienos lentelės eilutė išskaidoma į aiškiai įvardytus kintamuosius, su kuriais toliau galima atlikti EPDA skaičiavimus be tiesioginio darbo su lentele.

Toliau sraute atliekama termino datos paruošimo dalis. Veiksmai „Convert number to text“ ir „Convert text to datetime“ DueInt reikšmę paverčia pilna datos reikšme DueDate, kuri vėliau naudojama tiek numatomai apmokėjimo datai, tiek EPDA rodikliui skaičiuoti. Po to, naudojant komandą „Filter data table“, lentelė ExpectedDT, kurioje anksčiau sukaupiti klientų tipinių vėlavimų rezultatai, yra filtruojama pagal einamąjį klientą (Cust), o atrinkti įrašai išsaugomi CustExpectedDT lentelėje. Kintamasis ExpectedDelayDays iš pradžių nustatomas į 0, o vėliau cikle „For each (ExpectedRow in CustExpectedDT)“ iš atrinktų eilučių paimama šio kliento vidutinio vėlavimo reikšmė (ExpectedDelayDays1) ir priskiriama kintamajam ExpectedDelayDays. Jei reikšmė tuščia, ji pakeičiama nuliu, kad nekeltų klaidų kitame žingsnyje. Šitaip pirmoje srauto dalyje pasiruošiama EPDA skaičiavimui: iš pradinės sąskaitos eilutės išskiriami pagrindiniai laukai ir surandamas su tuo klientu susietas tipinis mokėjimo vėlavimas.



7 pav. „Power Automate Desktop“ EPDA skaičiavimo logika (2 dalis)

7 paveiksle vaizduojama antroji EPDA skaičiavimo logikos dalis, kurioje iš jau paruoštų reikšmių apskaičiuojama numatoma apmokėjimo data ir EPDA nuokrypis, o sąskaitai priskiriama rizikos kategorija. Pirmiausia tekstinė tikėtino vėlavimo reikšmė ExpectedDelayDays konvertuojama į skaičių (ExpectedDelayDaysNumeric), po to suapvalinama iki sveikų dienų (ExpectedDelayInt). Veiksmas „Add to datetime“ prie mokėjimo termino DueDate prideda šį dienų skaičių ir išveda ExpectedPaymentDate, kuri atitinka teorinėje dalyje apibrėžtą numatomą apmokėjimo momentą. Taip realizuojamas principas, kad kiekvienai sąskaitai apskaičiuojama, kada tikimasi, jog ji bus apmokėta, remiantis konkretaus kliento istoriniu mokėjimo vėlavimu.

Toliau srautas išsišakoja pagal sąskaitos būseną. Jei IsOpen = 0, laikoma, kad sąskaita yra apmokėta, todėl ClearText konvertuojama į datą PaymentDate, o veiksmas „Subtract dates“ apskaičiuoja dienų skirtumą tarp ExpectedPaymentDate ir PaymentDate, įrašydamas rezultatą į kintamąjį EPDA_Days. Iš pradžių rizikos žyma nustatoma kaip „Normal“, tačiau vėlesniuose patikrinimuose ji koreguojama: jei EPDA_Days > 10, nustatoma „LateOutlier“ (mokėjimas gerokai vėlesnis, negu būtų buvę tikėtina), o jei EPDA_Days < -10, – „EarlyOutlier“ (mokėjimas žymiai ankstesnis už įprastą kliento elgseną). Tuo atveju, kai sąskaita yra atvira (IsOpen ≠ 0), aktyvuojama „Else“ šaka: EPDA dienų skaičius priskiriamas 0, rizikos žyma – „Open“, o laukams ExpectedDelayDays ir ExpectedPaymentDate priskiriamos tuščios reikšmės, kad būtų aišku, jog šiai sąskaitai elgsenos nuokrypis dar neskaičiuojamas.

Skyriaus pabaigoje veiksmas „Insert row into data table“ suformuoja galutinę rezultatų eilutę lentelėje OutputDT. Į ją įrašomos visos pagrindinės analizės reikšmės: kliento numeris (Cust), dokumento numeris (DocId), bazinė data (BaselineInt), termino data (DueInt), faktinė apmokėjimo data (ClearText), suma (Amount), tikėtino vėlavimo dienos (ExpectedDelayDays), numatoma apmokėjimo data (ExpectedPaymentDate), EPDA nuokrypis dienomis (EPDA_Days) ir priskirta rizikos kategorija (RiskFlag). Ši lentelė vėliau eksportuojama į „Excel“ ir atitinka .NET prototipo generuojamą rezultatų rinkinį, todėl abu sprendimai gali būti tiesiogiai lyginami tiek pagal gautas EPDA reikšmes, tiek pagal patį duomenų pateikimo formatą.

3.3. Tradicinės sąskaitų senaties ir EPDA metodo palyginimai

3.3.1. Praktinis palyginimo pavyzdys

Svarbi šio darbo dalis – palyginti, kokią informaciją vidiniam auditui suteikia EPDA metodas ir tradicinė sąskaitų senaties (aging) analizė, paremta pradelstumo dienomis ir 30 dienų intervalų krepšeliais. Kaip nurodoma praktiniuose šaltiniuose, senaties ataskaita „kategorizuoja gautinas sumas pagal tai, kiek laiko sąskaitos yra neapmokėtos“, paprastai naudojant standartinius intervalus (pvz., 0–30, 31–60, 61–90, >90 dienų), ir yra „esminis įrankis valdant pinigų srautus ir vertinant kredito riziką“ Stripe (2024).

Norint aiškiai parodyti, kuo numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodas papildo tradicinę sąskaitų senaties (angl. *aging*) analizę, šiame poskyryje pateikiamas konkretus pavyzdys iš šiame darbe naudoto Kaggle duomenų rinkinio `dataset_less_data_3.csv`. Analizuojamos dviejų klientų – **THE corporation** (`cust_number = 140106181`) ir **WAL-MAR corp** (`cust_number = 200769623`) – sąskaitos.

1 pavyzdys: labai pavėluotas, bet klientui „tipinis“ mokėjimas

Klientui **THE corporation** išrašyta sąskaita (`doc_id = 2960581231`) turi šias reikšmes:

- sąskaitos mokėjimo terminas (`due_in_date`): **2019-10-04**;
- faktinė apmokėjimo data (`clear_date`): **2019-12-04**;
- mokėtina suma (`total_open_amount`): **22 225,84**;
- sąskaita uždaryta (`isOpen = 0`).

Tradicinis senaties skaičiavimas. Pradelstumo dienos skaičiuojamos kaip faktinės apmokėjimo ir termino datų skirtumas:

$$\text{DaysPastDue}_i = \text{PaymentDate}_i - \text{DueDate}_i = 2019-12-04 - 2019-10-04 = 61 \text{ diena.}$$

Naudojant įprastus 30 dienų senaties intervalus, ši sąskaita patenka į „61–90 dienų“ krepšelį. Auditoriui tradicinė senaties ataskaita duoda tokią informaciją:

- sąskaita yra labai pradelsta (61 diena po termino);
- suma yra santykinai didelė;

- tokios sąskaitos paprastai laikomos padidintos kredito rizikos.

Tačiau senaties metodas neatsako į klausimą, ar toks vėlavimas šiam klientui yra neįprastas, ar tai jo „įprastas“ mokėjimo elgesys.

EPDA skaičiavimas. Pirmiausia, remiantis šio kliento istoriniais duomenimis, apskaičiuojamas tipinis vėlavimas dienomis. Kadangi šiame susiaurintame rinkinyje klientas **THE corporation** turi vieną apmokėtą sąskaitą su 61 dienos vėlavimu, jo tipinis vėlavimas lygus:

$$E[\text{DelayFromDue}]_c = 61 \text{ diena.}$$

Tada šiai sąskaitai apskaičiuojama numatoma apmokėjimo data:

$$\text{ExpectedPaymentDate}_i = \text{DueDate}_i + E[\text{DelayFromDue}]_c = 2019-10-04 + 61 \text{ diena} = 2019-12-04.$$

EPDA nuokrypis:

$$\begin{aligned} \text{EPDADeviation}_i &= \text{PaymentDate}_i - \text{ExpectedPaymentDate}_i = 2019-12-04 - 2019-12-04 \\ &= 0 \text{ dienų.} \end{aligned}$$

Taigi, tradiciniu požiūriu sąskaita yra labai pradelsta (61 diena), o EPDA požiūriu tai yra visiškai tipinis šio kliento mokėjimas (nuokrypis = 0 dienų).

Šiuo atveju EPDA metodas parodo, kad, nors senaties ataskaitoje sąskaita atrodo rizikinga vien dėl didelės senaties, ji neatitinka elgsenos anomalijos – klientas taip elgiasi ir istorijoje.

2 pavyzdys: nežymiai pavėluotas, bet elgsenos požiūriu blogėjantis mokėjimas

Klientui **WAL-MAR corp** (cust_number = 200769623) nagrinėjama sąskaita (doc_id = 1930438491) turi šias reikšmes:

- sąskaitos mokėjimo terminas (due_in_date): **2020-02-10**;

- faktinė apmokėjimo data (clear_date): **2020-02-11**;
- mokėtina suma (total_open_amount): **54 273,28**;
- sąskaita uždaryta (isOpen = 0).

Tradicinis senaties skaičiavimas

Pradelstumo dienos:

$$\text{DaysPastDue}_i = 2020-02-11 - 2020-02-10 = 1 \text{ diena.}$$

Pagal tradicinius intervalus ši sąskaita patenka į „0–30 dienų“ senaties krepšelį. Senaties ataskaitos požiūriu:

- sąskaita yra tik 1 dieną pavėluota;
- ji paprastai vertinama kaip mažos rizikos;
- elgsenos pokyčių vien iš šio skaičiaus nematyti.

EPDA skaičiavimas

Šio klientų grupės tipinis vėlavimas skaičiuojamas iš kelių ankstesnių apmokėtų sąskaitų. To paties kliento 200769623 istorijoje šiame duomenų rinkinyje yra dar bent trys uždarytos sąskaitos su šiomis vėlavimo reikšmėmis (dienomis nuo termino):

- sąskaita doc_id = 1930147974: vėlavimas –3 dienos (apmokėta anksčiau už terminą);
- sąskaita doc_id = 1930083373: vėlavimas –4 dienos;
- sąskaita doc_id = 1930438491 (analizuojama) – čia jau 1 diena vėlavimo, bet tipiniam vėlavimui skaičiuoti naudojamos ankstesnės sąskaitos.

Apskaičiuotas tipinis šio kliento vėlavimas pagal ankstesnes sąskaitas:

$$E[\text{DelayFromDue}]_c = \frac{1 + (-3) + (-4)}{3} = \frac{-6}{3} = -2 \text{ dienos,}$$

t. y. tipiška šis klientas moka maždaug 2 dienomis anksčiau už terminą.

Tada nagrinėjamai sąskaitai numatoma apmokėjimo data:

$$\text{ExpectedPaymentDate}_i = \text{DueDate}_i + E[\text{DelayFromDue}]_c = 2020-02-10 - 2 \text{ dienos} = 2020-02-08.$$

EPDA nuokrypis:

$$\begin{aligned} \text{EPDADeviation}_i &= \text{PaymentDate}_i - \text{ExpectedPaymentDate}_i = 2020-02-11 - 2020-02-08 \\ &= 3 \text{ dienos.} \end{aligned}$$

Taigi, tradiciniu požiūriu sąskaita vėluoja tik 1 dieną, todėl ataskaitoje atrodo beveik nerizikinga, tačiau EPDA požiūriu klientas šią sąskaitą apmokėjo 3 dienomis vėliau, nei yra įprasta pagal jo istoriją (nuokrypis = +3 dienos), t. y. elgsena pablogėjo.

Remiantis pateiktais pavyzdžiais, galima aiškiai palyginti tradicinį sąskaitų senaties metodą ir EPDA metodą, atskleidžiant, kokią papildomą informaciją pastarasis suteikia vidiniam auditui.

Tradicinis sąskaitų senaties metodas matuoja absoliutų pradelstumo dydį dienomis. Kiekvienai sąskaitai apskaičiuojamas pradelstumas pagal formulę

$$\text{DaysPastDue}_i = \text{ReferenceDate} - \text{DueDate}_i,$$

gauta reikšmė priskiriama vienam iš senaties krepšelių (0–30, 31–60, 61–90, >90 dienų). Taip sudaroma gautinų sumų senaties ataskaita, kuri padeda greitai identifikuoti seniausias ir labiausiai pavėluotas skolas, įvertinti bendrą portfelio „senėjimo“ struktūrą ir nustatyti didžiausios kredito rizikos segmentus. Šis metodas yra paprastas, lengvai suprantamas ir lengvai sugeneruojamas praktiškai bet kurioje apskaitos sistemoje. Tačiau esminis jo trūkumas yra tas, kad senatis neatsižvelgia į konkretaus kliento istoriją: 61 dienos vėlavimas visais atvejais vertinamas vienodai, nepriklausomai nuo to, ar klientui toks vėlavimas būdingas ir pasikartojantis, ar tai naujas ir išskirtinis elgsenos pablogėjimas.

EPDA metodas į šią problemą žvelgia kitaip ir analizę praplečia elgsenos dimensija. Pirmiausia kiekvienam klientui apskaičiuojamas tipinis mokėjimo vėlavimas, remiantis istorinių sąskaitų duomenimis. Tam naudojama tokia bendra idėja: iš ankstesnių apmokėtų sąskaitų apskaičiuojamas vėlavimo nuo termino vidurkis ar mediana,

$$E[\text{DelayFromDue}]_c = \text{vidurkis/mediana iš}(\text{PaymentDate}_i - \text{DueDate}_i),$$

ir ši reikšmė interpretuojama kaip įprastas kliento elgesys. Remiantis šiuo tipiniu vėlavimu, kiekvienai naujai ar analizuojamai sąskaitai nustatoma numatoma apmokėjimo data:

$$\text{ExpectedPaymentDate}_i = \text{DueDate}_i + E[\text{DelayFromDue}]_c.$$

Tuomet EPDA metodas skaičiuoja nuokrypį nuo įprasto elgesio:

$$\text{EPDADeviation}_i = \text{PaymentDate}_i - \text{ExpectedPaymentDate}_i,$$

kuris parodo, kiek dienų konkreti sąskaita yra „geresnė“ (mokama anksčiau) arba „blogesnė“ (mokama vėliau) už tai, kas tipinė pagal kliento istoriją. Teigiamą EPDADeviation_i reikšmę rodo, kad klientas moka vėliau, nei įprasta, o neigiamą – kad jis atsiskaito greičiau nei įprastai.

Tokiu būdu EPDA metodas leidžia atskirti kelias svarbias situacijas. Pirmą, jis padeda atpažinti labai pradelstas, bet klientui tipiškas sąskaitas – pavyzdžiui, THE corporation atveju 61 dienos vėlavimas tradicinėje senaties ataskaitoje atrodo labai didelis, tačiau EPDA nuokrypis lygus nuliui, nes toks vėlavimas sutampa su istoriniu kliento vidurkiu. Antra, EPDA leidžia išryškinti nedaug pavėluotas, bet elgsenos požiūriu blogėjančias sąskaitas – kaip WAL-MAR corp pavyzdyje, kur tradicinė senatis rodo tik 1 dienos vėlavimą, tačiau EPDA nuokrypis +3 dienos rodo, kad klientas moka reikšmingai vėliau, nei buvo įprasta ankstesniuose laikotarpiuose. Dėl šių priežasčių EPDA suteikia vidiniam auditui gilesnį vaizdą apie kliento mokėjimo elgsenos pokyčius, neapsiribojant vien absoliučiu pradelstumo dienų skaičiumi ir leidžiant aiškiau identifikuoti tikrąsias rizikos pokyčių vietas.

3.3.2. Tradicinės sąskaitų senaties ir EPDA rezultatų palyginimas visai imčiai

Analizei naudojami tik uždarytas sąskaitas (isOpen = 0), kurioms galima apskaičiuoti tiek tradicinį pradelstumą (DaysPastDue), tiek EPDA. Tradicinis pradelstumas skaičiuotas taip:

$$\text{DaysPastDue}_i = \text{PaymentDate}_i - \text{DueDate}_i,$$

kur PaymentDate – faktinė apmokėjimo data (clear_date), o DueDate – mokėjimo terminas (due_in_date). Pagal šį rodiklį sąskaitos suskirstytos į senaties krepšelius: ≤ 0 , 1–30, 31–60, 61–90 ir >90 dienų. EPDA rizika vertinama pagal tavo .NET modelio RiskFlag reikšmę („Normal“, „EarlyOutlier“, „LateOutlier“).

Žemiau – visų 40 368 uždarytų sąskaitų pasiskirstymas:

1 lentelė Uždarytų sąskaitų paskirstymas senaties krepšelių metodu ir EPDA metodu

Senaties krepšelis (DaysPastDue)	Sąskaitų skaičius	EPDA – „Normal“	EPDA – „Early outlier“	EPDA – „LateOutlier“	EPDA rizikingų dalis, procentais *
1-30	15 954	13 914	2 032	8	12.8
31-60	187	0	187	0	100
61-90	155	0	155	0	100
≤ 0	24 027	21 757	29	2 241	9.4
> 90	45	0	45	0	100

* EPDA rizikingos sąskaitos čia laikomos tos, kurioms RiskFlag yra „EarlyOutlier“ arba „LateOutlier“.

Tradiciškai praktikoje rizikingomis dažniausiai laikomos sąskaitos, kurių pradelstumas viršija 30 dienų (krepšeliai 31–60, 61–90, >90). Pagal tavo duomenis tokių sąskaitų yra 387 (187 + 155 + 45). Visos jos EPDA požiūriu yra elgsenos outlier'iai – nė viena iš jų neturi RiskFlag = "Normal". Tai reiškia,

kad šioje imtyje nėra sąskaitų, kurios pagal tradicinę senaties ataskaitą būtų vertinamos kaip labai pavėluotos ir rizikingos, bet EPDA laikytų jas normaliu, tipiniu kliento elgesiu. Kitaip tariant, tradicinis senaties metodas čia nesukuria „klaidingų pavojaus signalų“ (>30 dienų senatis ir kartu EPDA = „Normal“ atvejų yra 0).

Kur EPDA parodo savo pridėtinę vertę, yra priešinga kryptis. Jei laikysime, kad tradicinis metodas „nerizikingomis“ laiko sąskaitas, kurių pradelstumas neviršija 30 dienų (t. y. krepšeliai ≤ 0 ir 1–30), tokių sąskaitų yra 39 981. Iš jų 4 310 sąskaitų (apie 10,7 % visų uždarytų sąskaitų) senaties požiūriu atrodo nereikšmingai vėluojančios arba net apmokėtos laiku/anksti, tačiau EPDA jas klasifikuoja kaip elgsenos outlier'ius („EarlyOutlier“ arba „LateOutlier“). Iš šių 4 310 atvejų maždaug 2 270 sąskaitų patenka į „ ≤ 0 dienų“ krepšelį – tradicinėje senaties ataskaitoje jos būtų laikomos visiškai saugiomis, nes neatsilieka nuo termino, tačiau EPDA rodo, kad mokėjimo momentas reikšmingai skiriasi nuo įprasto kliento istorinio modelio. Likusios maždaug 2 040 sąskaitų yra „1–30 dienų“ krepšelyje – pagal senatį jos dar būtų priskiriamos „normaliai“ vėluojančioms, tačiau EPDA identifikuoja čia elgsenos blogėjimą ar neįprastai ankstyvus mokėjimus.

Apibendrinant, pagal šiuos rezultatus tradicinis senaties metodas nepervertina rizikos (nenustatyta atvejų, kai >30 dienų pradelstumas būtų EPDA požiūriu „normalus“), tačiau jis nepakankamai ją atskleidžia: apie dešimtadalis uždarytų sąskaitų, tradiciškai laikomų nerizikingomis, EPDA metodu identifikuojamos kaip reikšmingai nukrypstančios nuo kliento įprasto atsiskaitymo elgesio. Tai argumentuoja EPDA metodo naudą vidiniam auditui – jis padeda pamatyti elgsenos pokyčius ir galimus rizikos signalus anksčiau ir tiksliau, nei tai leidžia vien tik kalendorinė senaties analizė.

3.4. .NET ir RPA prototipų palyginimas

3.4.1. Matavimų metodika

Abiejų prototipų – .NET aplikacijos ir „Power Automate Desktop“ (RPA) srauto – našumas buvo vertinamas naudojant „Windows Performance Monitor“ (PerfMon). Stebėti šie rodikliai:

- % Processor Time – procesoriaus užimtumas, priskiriamas konkrečiam procesui;
- IO Read Bytes/sec – disko skaitymo srautas baitais per sekundę;
- IO Write Bytes/sec – disko rašymo srautas baitais per sekundę;

- Working Set – Private – proceso naudojama fizinė atmintis (privatus working set).

Kadangi .NET aplikacija vieną kartą suveikusi visą duomenų rinkinį apdoroja labai greitai, siekiant gauti pakankamai „PerfMon“ matavimo taškų buvo pritaikyta tokia metodinė priemonė: kiekvienam duomenų rinkinio dydžiui logika buvo kartojama kelis šimtus ar tūkstančius kartų. Pakartotinai buvo vykdomi tik skaičiavimai (ExpectedDelay per klientą ir EPDA per sąskaitą), o rezultatų įrašymas į failą atliktas vieną kartą. RPA prototipui tokie kartojimai nebuvo reikalingi – vienas srauto vykdymas ir taip trunka pakankamai ilgai, kad PerfMon spėtų surinkti stabilias metrikas.

3.4.2. „PerfMon“ rezultatų suvestinė

2 lentelė .NET prototipo našumo rodikliai

Duomenų kiekis faile	Logikos kartojimai	Vykdymo trukmė, s	Vid. CPU užimtumas, procentais	Vid. IO read, KB/s	Vid. IO write, KB/s	Vid. „Working set - private“, MB
1 000	10 000	14	36,2	0	1 987,5	55,3
10 000	1 000	17	55,5	0	1 302,1	57,6
25 000	500	19	64,0	0	1 088,9	57,9
48 000	200	40	72,2	0	823,0	81,3

3 lentelė RPA („Power Automate Desktop“) prototipo našumo rodikliai

Duomenų kiekis faile	Logikos kartojimai	Vykdyto trukmė, s	Vid. CPU užimtumas, procentais	Vid. IO read, KB/s	Vid. IO write, KB/s	Vid. „Working set - private“, MB
1 000	1	14	122,3	288.0	4 252,7	43,0
10 000	1	78	124,0	44.2	3 587,2	98,2
25 000	1	228	119,2	23.4	2 743,0	162,1
48 000	1	706	113,7	12.6	1 690,4	267,6

Iš rezultatų matyti, kad didėjant duomenų rinkinio dydžiui, .NET prototipo vykdymo trukmė auga labai nuosaikiai, CPU užimtumas didėja nuo ~36 % iki ~72 %, o atminties poreikis – nuo ~55 MB iki ~81 MB. RPA prototipas, priešingai, rodo žymiai spartesnį išteklių augimą – vykdymo trukmė išauga nuo 14 s iki 706 s, o „Working set – private“ padidėja nuo ~43 MB iki ~268 MB. RPA procese CPU užimtumas nuolat svyruoja apie 110–125 % (t. y. procesas intensyviai naudoja kelis loginius branduolius), o disko rašymo srautai yra kelis kartus didesni nei .NET atveju, kas atspindi intensyvią darbą su tarpinėmis lentelėmis ir Excel aplinka.

Tiesioginis šių trukmių palyginimas (pvz., 40s .NET prieš 706s RPA) būtų metodologiškai netikslus, nes .NET atveju per tą patį laiką logika perskaičiuojama šimtus ar tūkstančius kartų. Todėl toliau pateikiamas normalizuotas palyginimas.

3.4.3. Normalizuotas vykdymo laikas ir interpretacija

Siekiant sąžiningai palyginti abiejų prototipų našumą, bendras matuotas laikas buvo normalizuotas pagal vieną pilną EPDA logikos prasukimą 1 000 sąskaitų. Tam naudojama tokia formulė:

$$t_{1000} = \frac{T_{\text{elapsed}}}{\frac{N_{\text{eilučių}}}{1000} \cdot N_{\text{kartojimų}}},$$

kur:

- t_{1000} – normalizuotas laikas (sekundėmis) vienam EPDA logikos prasukimui 1 000 sąskaitų;
- T_{elapsed} – bendras matuotas vykdymo laikas sekundėmis;
- $N_{\text{eilučių}}$ – eilučių (sąskaitų) skaičius duomenų rinkinyje;
- $N_{\text{kartojimų}}$ – kiek kartų ta pati logika buvo iškviesta (.NET atveju – 200–10 000, RPA atveju – 1).

Normalizuoti rezultatai apibendrinti 4 lentelėje.

4 lentelė Normalizuotas vykdymo laikas vienam EPDA logikos paskaičiavimui 1 000 sąskaitų

Duomenų kiekis faile	.NET t_{1000} , s	RPA t_{1000} , s
1 000	0,001	14 000
10 000	0,002	7 800
25 000	0,002	9 120
48 000	0,004	14 708

Matyti, kad .NET prototipo normalizuotas laikas svyruoja maždaug nuo 0,001 iki 0,004 sekundės vienam 1 000 sąskaitų, t. y. kiekvienam tūkstančiui įrašų sugaištama tik 1–4 milisekundės. Tuo tarpu RPA (Power Automate Desktop) prototipo laikas yra 8–15 sekundžių vienam 1 000 sąskaitų, priklausomai nuo rinkinio dydžio. Tai reiškia, kad vieną EPDA skaičiavimo iteraciją .NET sprendimas atlieka tūkstančius kartų greičiau nei analogiškas RPA srautas.

Šis normalizavimas yra svarbus metodologinis žingsnis: be jo gali susidaryti klaidingas įspūdis, kad, pavyzdžiui, .NET prototipas 48 tūkst. eilučių rinkiniui užtrunka 40 s, o RPA – 706 s, tarsi skirtumas būtų „tik“ apie 18 kartų. Tačiau atsižvelgus į tai, kad .NET per tas 40 s perskaičiuoja EPDA logiką 200 kartų, o RPA – tik kartą, realus algoritmų efektyvumo santykis yra gerokai didesnis. Normalizuotas t_{1000} rodiklis eliminuoja šį kartojimų efektą ir leidžia moksliškai pagrįstai lyginti vieną EPDA skaičiavimo „vienetą“.

Apibendrinant, atlikta analizė rodo, kad .NET prototipas yra iš esmės tinkamesnis kaip didelių duomenų kiekių skaičiavimų variklis: jis pasižymi labai trumpu vykdymo laiku vienam įrašų kiekiui, nuosaikiu atminties poreikiu ir minimaliais disko skaitymo srautais. Tuo tarpu RPA (Power Automate Desktop) prototipas yra žymiai lėtesnis, labiau apkrauna procesorių ir atmintį, bet suteikia kitą privalumą – vizualų, auditoriams lengviau suprantamą darbo srautą, kurį galima kurti ir keisti be tradicinio programavimo. Dėl to šio darbo kontekste .NET sprendimas gali būti rekomenduojamas kaip pagrindinis skaičiavimų komponentas, o RPA – kaip papildomas orkestravimo ir integracijos sluoksnis, kai reikia įtraukti EPDA logiką į platesnį vidinio audito procesų automatizavimą.

3.5. Skyriaus apibendrinimas

Trečiajame skyriuje atliktas tyrimas parodė, kad numatomo ir faktinio mokėjimo vėlavimo (EPDA) metodas gali būti sėkmingai pritaikytas gautinų sumų vidiniame audite ir realizuotas skirtingose technologinėse platformose – tiek tradicinėje .NET programėlėje, tiek RPA „Power Automate Desktop“ (PAD) sprendime. EPDA metodas, skirtingai nei tradicinis sąskaitų senaties (aging) skirstymas pagal pradelstumo dienas, leidžia analizuoti ne tik absoliutų vėlavimo dydį, bet ir kliento elgsenos nuokrypį nuo jo istorinio mokėjimo modelio. Atliekant palyginamąją analizę su Kaggle duomenų rinkiniu nustatyta, kad EPDA rodiklis leidžia identifikuoti reikšmingą dalį sąskaitų, kurios tradicinėje senaties ataskaitoje būtų laikomos nerizikingomis, tačiau elgsenos požiūriu rodo aiškius mokėjimo blogėjimo ar anomalijų požymius. Tai patvirtina, kad EPDA metodas papildo įprastą senaties analizę papildoma, vidiniam auditui svarbia elgsenos dimensija.

Tradicinės senaties ir EPDA rezultatų palyginimas visai imčiai atskleidė, jog sąskaitos, kurios pagal senaties ataskaitą laikomos labiausiai pavėluotomis (31–60, 61–90, >90 dienų krepšeliai), EPDA požiūriu taip pat identifikuojamos kaip elgsenos išskirtys – nenustatyta atvejų, kai didelė senatis būtų „normali“ pagal kliento istoriją. Kur kas svarbesnė išvalga yra tai, kad maždaug dešimtadalis uždarytų sąskaitų, patenkančių į „≤0“ ir „1–30 dienų“ krepšelius, EPDA metodu priskiriamos prie elgsenos outlier'ių. Kitaip tariant, EPDA leidžia išryškinti „paslėptą“ riziką – sąskaitas, kurios kalendorine prasme nėra pačios seniausios, tačiau rodo ryškų nukrypimą nuo kliento įprasto atsiskaitymo modelio. Vidinio audito požiūriu tai sudaro sąlygas tiksliau parinkti analitinių procedūrų imtį, fokusuotis į elgsenos pokyčius ir laiku identifikuoti galimus nuostolių ar mokumo problemų signalus, kurių tradicinė senaties ataskaita gali neparodyti.

Technologinių sprendimų palyginimas parodė, kad abu sukurti prototipai funkciškai teisingai įgyvendina EPDA algoritmą, tačiau esmingai skiriasi savo našumo ir pritaikomumo charakteristikomis. Normalizavus matavimus pagal vieną EPDA logikos prasukimą 1 000 sąskaitų, nustatyta, kad .NET prototipo vykdymo trukmė siekia vos 0,001–0,004 sekundės, o RPA (Power Automate Desktop) prototipo – apie 8–15 sekundžių. Tai reiškia, kad skaičiavimo požiūriu .NET sprendimas yra tūkstančius kartų efektyvesnis už analogišką RPA realizaciją. Be to, .NET aplikacija pasižymi nuosaikiu atminties poreikiu ir minimaliais disko skaitymo srautais, todėl yra tinkama naudoti didelėms duomenų imtims ir daugkartiniams perskaičiavimams. RPA prototipas, priešingai, nors ir gerokai lėtesnis bei ištekliams imlesnis, pasižymi dideliu prieinamumu ir skaidria, vizualia logika, kuri auditoriams yra lengviau suprantama ir prižiūrima be gilių programavimo žinių. Tai leidžia jį vertinti kaip patogų įrankį greitai kurti ir adaptuoti automatizuotus audito srautus, ypač ten, kur svarbi sąveika su „Office“ aplinka ar kitomis vartotojo sąsajomis.

Papildomai šiame darbe buvo mėginta EPDA metodo logiką realizuoti ir „n8n“ tipo AI/low-code orkestravimo įrankyje. Sukurtas bandomasis srautas turėjo atkartoti pagrindinius duomenų paruošimo ir EPDA skaičiavimo žingsnius, tačiau praktikoje susidurta su keliais reikšmingais apribojimais: sudėtinga efektyviai grupuoti ir agreguoti dideles sąskaitų imtis kliento lygiu, ribotos galimybės patogiai valdyti tarpinius duomenų rinkinius bei realizuoti sudėtingesnę ciklą ir išimčių logiką, o našumo ir resursų valdymo kontrolė yra gerokai menkesnė nei tradicinėje .NET aplinkoje. Dėl šių priežasčių „n8n“ pagrindu sukurtas prototipas liko eksperimentinis ir nebuvo įtrauktas į detalesnę našumo analizę kartu su .NET ir „Power Automate Desktop“ sprendimais. Vis dėlto pats bandymas parodė, kad platesnis EPDA metodo taikymo AI/low-code orkestratoriuose, integruojant jį su mašininio mokymosi modeliais ar kitomis FinTech sistemomis, galėtų būti savarankiškas ateities tyrimų objektas.

Tyrimo rezultatai taip pat atskleidė išorinius ir vidinius ribotumus, kuriuos svarbu įvertinti interpretuojant gautas išvadas. Pirma, taikytas vienas sintetinis, bet realistiškais duomenimis paremtas Kaggle rinkinys, todėl EPDA ir technologinių sprendimų vertinimas atspindi tipinę, tačiau ne konkrečią realios įmonės situaciją. Antra, EPDA metodas buvo realizuotas kaip santykinai paprastas statistinis modelis, paremtas tipinio vėlavimo (vidurkio ar medianos) skaičiavimu, nenaudojant sudėtingesnių mašininio mokymosi algoritmų, kurie galėtų dar tiksliau modeliuoti elgseną. Trečia, technologinių požiūriu lygintos tik dvi platformos – .NET ir „Power Automate Desktop“, todėl išvados apie našumą ir pritaikomumą galioja būtent šioms priemonėms ir nagrinėjamam scenarijui. Galiausiai, našumo matavimams naudotas PerfMon įrankis ir .NET logikos kartojimai, kurie reikalingi patikimam matavimui, tačiau reikalauja papildomos normalizavimo procedūros.

Nepaisant šių ribotumų, atliktas tyrimas leidžia daryti išvadą, kad EPDA metodas yra perspektyvus gautinų sumų vidinio audito rodiklis, papildantis tradicinę sąskaitų senaties analizę elgsenos dimensija, o skirtingi technologiniai sprendimai suteikia alternatyvius kelius šiam metodui įdiegti praktikoje. .NET sprendimas gali būti traktuojamas kaip didelio našumo „skaičiavimų variklis“, tinkamas centralizuotam EPDA rodiklių generavimui, tuo tarpu RPA prototipas – kaip lankstus orkestravimo ir integracijos sluoksnis, leidžiantis EPDA logiką įterpti į platesnį vidinio audito automatizuotų procedūrų rinkinį. Šios išvalgos sudaro prielaidą Išvadų skyriuje suformuluoti konkrečias rekomendacijas, kada ir kaip EPDA metodą bei atitinkamas technologijas tikslingiausia taikyti skirtingose organizacijų audito aplinkose.

IŠVADOS

Remiantis atliktu tyrimu, sukurtomis prototipinėmis sistemomis ir gautais eksperimentiniais rezultatais, galima suformuluoti šias pagrindines išvadas:

1. Išnagrinėjus mokslinę literatūrą apie gautinų sumų auditą, tradicinį sąskaitų senaties metodą ir mokėjimų vėlavimo modeliavimą, patvirtinta darbo problemos aktualumas. Literatūroje senaties (aging) ataskaitos aprašomos kaip standartinis įrankis, kuriame pirkėjų skolos grupuojamos į 30 dienų intervalų krepšelius pagal tai, kiek dienų sąskaita yra neapmokėta, ir taip vertinama kredito rizika bei planuojami pinigų srautai. Tuo pačiu naujesni tyrimai rodo, kad, remiantis istorine sąskaitų ir mokėjimų informacija, įmanoma prognozuoti faktinę apmokėjimo datą ir modeliuoti mokėjimo vėlavimą, pasitelkiant elgsenos pagrindu kuriamus rodiklius. Tačiau vidinio audito literatūroje stokojama darbų, kurie šias dvi kryptis – tradicinę senaties analizę ir elgsenos pagrindu grindžiamą mokėjimų modeliavimą – integruotų į vieną aiškų metodinį sprendimą, pritaikomą auditui.
2. Apibendrinus .NET ir RPA („Power Automate Desktop“) technologijų taikymo galimybes, nustatyta, kad jos tarpusavyje ne tiek konkuruoja, kiek papildo viena kitą audito automatizavimo kontekste. RPA literatūroje apibūdinama kaip technologija, leidžianti automatizuoti struktūruotas, taisyklėmis pagrįstas ir pasikartojančias užduotis – sąskaitų atsisiuntimą, duomenų perkėlimą tarp sistemų, paprastas analitines procedūras – taip mažinant rankinio darbo ir klaidų riziką bei plečiant tikrinamų operacijų apimtį. .NET aplinka, priešingai, pasižymi didesniu našumu, geresniu tipizuotų duomenų valdymu ir lankstumu kuriant sudėtingesnius skaičiavimo algoritmus ir modulinius sprendimus. Praktinė analizė parodė, kad RPA yra patogus, kai reikia greitai integruotis į esamas sistemas be papildomos IT infrastruktūros, o .NET labiau tinka, kai svarbūs dideli duomenų kiekiai, ilgaamžis palaikymas ir platesnės integracijos galimybės.
3. Sukurtas EPDA conceptualus modelis ir skaičiavimo algoritmas parodė, kad elgsenos pagrindu grindžiamas numatomo ir faktinio mokėjimo vėlavimo rodiklis gali būti praktiškai pritaikytas naudojant Kaggle „Payment Date Prediction“ duomenų rinkinį. Duomenys buvo nuosekliai parengti tyrimui: pašalintas nereikalingas stulpelis „area_business“, taip pat eilutės, kuriose klientų kodai buvo mišraus (raidžių ir skaičių) formato; po šių valymo žingsnių liko 48 580 sąskaitų, tinkamų EPDA skaičiavimui. EPDA algoritmas remiasi tuo, kad kiekvienam klientui iš ankstesnių apmokėtų sąskaitų apskaičiuojamas tipinis vėlavimas nuo termino iki faktinės apmokėjimo dienos, o tuomet kiekvienai naujai sąskaitai nustatoma numatoma apmokėjimo data ir EPDA nuokrypis – skirtumas

tarp faktinio ir tikėtino elgesio. Tai leidžia vertinti ne tik absoliutų pradelstumą, bet ir tai, ar klientas elgiasi geriau ar blogiau nei savo istorinis „standartas“.

4. Eksperimentinis tyrimas parodė, kad EPDA metodas suteikia papildomą rizikos vertinimo gylį, palyginti su tradicine senaties analize, o .NET ir RPA prototipų našumo skirtumai yra kelių eilių dydžio. Analizuojant tik uždarytas sąskaitas (40 368 įrašai), tradicinis senaties rodiklis (DaysPastDue) leido jas suskirstyti į penkis krepšelius pagal pradelstumo trukmę, tačiau EPDA rizikos žyma („Normal“, „EarlyOutlier“, „LateOutlier“) parodė, kad net „saugiais“ laikomuose krepšeliuose (≤ 0 ir 1–30 dienų) apie 9–13 % sąskaitų elgsenos požiūriu išsiskiria – klientai moka anksčiau ar vėliau nei įprasta, todėl tokie atvejai yra verti papildomo audito dėmesio. Tuo pačiu labai pavėluotos sąskaitos (31+ dienos) beveik visos priskiriamos vėlyvų išskirčių kategorijai, kas patvirtina, kad EPDA ne silpnina, o patikslina tradicinį rizikos signalą, atskirdamas tipiškai lėtai mokančius klientus nuo tų, kurių elgsena staiga pablogėjo.
5. Vertinant technologinius prototipus, nustatyta, kad .NET sprendimo vykdymo trukmė ir resursų sąnaudos didėjant duomenų apimčiai auga labai nuosaikiai, tuo tarpu RPA prototipas naudoja gerokai daugiau CPU ir atminties, o vykdymo laikas didėja daug sparčiau. Normalizuotas palyginimas parodė, kad vieną EPDA skaičiavimo iteraciją 1 000 sąskaitų .NET aplikacija atlieka milisekundžių, o RPA – sekundžių masteliu, t. y. .NET yra tūkstančius kartų spartesnis, nors abu prototipai grąžina tas pačias EPDA reikšmes.
6. Remiantis tyrimo rezultatais suformuluotos rekomendacijos patvirtina EPDA metodo ir pasirinktų technologinių sprendimų praktinę naudą vidiniam auditui, kartu aiškiai įvardijant taikymo ribotumus. EPDA rekomenduojama naudoti kaip papildomą elgsenos rodiklį kartu su tradicine senaties ataskaita – ypač tais atvejais, kai auditoriams svarbu identifikuoti klientus, kurių mokėjimo elgsena keičiasi, arba atskirti „chroniškai vėluojančius“ nuo tų, kurie tik neseniai pradėjo vėluoti. .NET sprendimas labiau tinka scenarijams, kai reikia reguliariai apdoroti didelius duomenų kiekius, integruoti sprendimą į IT infrastruktūrą ir garantuoti našumą bei stabilumą, o RPA („Power Automate Desktop“) prototipas yra patogus ten, kur reikalingas greitas, konfigūruojamas sprendimas, glaudžiai sąveikaujantis su vartotojo aplinka ir standartiniais „Office“ įrankiais. Kartu pažymėtina, kad darbo išvados ribojamos tuo, jog naudotas vienas viešai prieinamas (sintetinis, bet realistiškas) duomenų rinkinys, EPDA modelis pagrįstas palyginti paprasta statistine logika, o technologinis palyginimas apsiriboja dviem platformomis; todėl tolesni tyrimai galėtų plėsti metodą įtraukiant kelis realių įmonių duomenų rinkinius, sudėtingesnius prognozavimo modelius ir platesnį technologijų spektrą.

LITERATŪROS SĄRAŠAS

1. Fernandez, D., ir Aman, A. (2021). The challenges of implementing robotic process automation in Global Business Services. *International Journal of Business and Society*, 22(3), 1269–1282. <https://doi.org/10.33736/ijbs.4301.2021>
2. Gotthardt, M., Koivulaakso, D., Paksoy, O., Saramo, C., Martikainen, M. ir Lehner, O. (2020). Current state and challenges in the implementation of smart robotic process automation in accounting and auditing. *ACRN Journal of Finance and Risk Perspectives*. <https://www.acrn-journals.eu/resources/jofrp09g.pdf>
3. Hradecká, M. (2019) Robotic Internal Audit – Control Methods in the Selected Company. *AGRIS on-line Papers in Economics and Informatics*, 10(3), 31-42. https://ageconsearch.umn.edu/record/294153/files/413_agris-on-line-2019-2-hradecka.pdf
4. Jhonson D., Smith J. ir Shehzadi T. (2023). Strategic Automation: The Role of Robotics Process Automation in Modern Organizations. https://www.researchgate.net/publication/376315043_Strategic_Automation_The_Role_of_Robotics_Process_Automation_in_Modern_Organizations
5. Kapur, N. ir Singh, A. K. (2022). Robotics process automation and internal audit: A transformation. *The Management Accountant Journal*, 57(7), 58-63. <https://nirdprojms.in/index.php/maj/article/view/170854/116212>
6. Kogan, G., Kokina, J., Stampone, A. ir Boyle, D. M. (2022). RPA in Accounting Risk and Internal Control: Insights from RPA Program Managers. *Accounting Horizons*, 1-12. <https://doi.org/10.2308/HORIZONS-2022-191>
7. Kokina, J. ir Blanchette, S. (2019). Early evidence of Digital Labor in Accounting: Innovation with Robotic Process Automation. *International Journal of Accounting Information Systems*, 35, 100431. <https://doi.org/10.1016/j.accinf.2019.100431>
8. Precedence Research (2024) *Robotic Process Automation Market (By Type: Software, Service; By Deployment: Cloud, On-Premise; By Industry: BFSI, Pharma & Healthcare, Manufacturing, Logistics, and Energy & Utilities, Others) - Global Industry Analysis, Size, Share, Growth, Trends, Regional Outlook, and Forecast 2024 – 2033*. Prieiga per <https://www.precedenceresearch.com/robotic-process-automation-market>
9. Rawashdeh, A., Shehadeh, E., Rababah, A. ir Al-Okdeh, S. K. (2022). Adoption of robotic process automation (RPA) and its effect on business value: an internal auditors perspective. *Journal of*

10. Xin, N. L., Asokan, K. A. ir Suresh, A. B. (2022). The Impact of Implementing Robotic Process Automation (RPA) on the Internal Audit Function: A Malaysian Study. *International Journal of Early Childhood Special Education*, 14(3). https://imcra-az.org/uploads/public_files/2022-05/issue-4-283-364.pdf
11. Upflow blog (2024) *Aging Report: Meaning & Role in Accounts Receivable (AR)*. Prieiga per <https://upflow.io/blog/ar-metrics/aging-reports>
12. Zone and Co (2025) *Accounts receivable aging: What it is and why it matters*. Prieiga per <https://www.zoneandco.com/articles/accounts-receivable-aging-what-it-is-and-why-it-matters>
13. Tabs platform (2025) *The Aging of Receivables Method: A Guide for SaaS*. Prieiga per <https://www.tabs.com/blog/aging-of-receivables-method>
14. Appel, A. P., Oliveira, V., Lima, B., Malfatti, G. L., De Santana, V. F., ir De Paula, R. (2019). *Optimize Cash Collection: Use Machine learning to Predicting Invoice Payment*. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1912.10828>
15. AuditBoard (2022) *Hyperautomation in Action: Revenue and Receivables*. Prieiga per <https://auditboard.com/blog/hyperautomation-in-action-revenue-and-receivables>
16. FloQast (2021) *How Robotic Process Automation Is Transforming Accounting*. Prieiga per <https://www.floqast.com/blog/robotic-process-automation-transforming-accounting>
17. Moffitt, K. C., Rozario, A. M., ir Vasarhelyi, M. A. (2018). Robotic process automation for auditing. *Journal of Emerging Technologies in Accounting*, 15(1), 1–10. <https://doi.org/10.2308/jeta-10589>
18. Schoonbee, L., Moore, W., ir Van Vuuren, J. (2022). A MACHINE-LEARNING APPROACH TOWARDS SOLVING THE INVOICE PAYMENT PREDICTION PROBLEM. *The South African Journal of Industrial Engineering*, 33(4). <https://doi.org/10.7166/33-4-2726>
19. Peiris, M., ir Premaratne, S. (2025). Optimizing invoice distribution through machine learning and natural language processing techniques. *Procedia Computer Science*, 258, 737–744. <https://doi.org/10.1016/j.procs.2025.04.306>
20. AIMultiple (2025) *RPA Accounting: 6 Processes & Real-Life Examples*. Prieiga per <https://research.aimultiple.com/rpa-accounting/>

21. Kassar, M., ir Jizi, M. (2025). Artificial intelligence and robotic process automation in auditing and accounting: a systematic literature review. *Journal of Applied Accounting Research*, 1–25. <https://doi.org/10.1108/jaar-05-2024-0175>
22. Kaggle (2023) Customer Invoices Dataset. Prieiga per <https://www.kaggle.com/datasets/pradumn203/payment-date-prediction-for-invoices-dataset>
23. Charabuddi, N. R. R. (2025). Secure document automation using blockchain anchors and AI-Validated semantic hashing for invoice integrity. *Journal of Information Systems Engineering & Management*, 10(55s), 648–658. <https://doi.org/10.52783/jisem.v10i55s.11629>
24. Net Suite (2022) Accounts Receivable Aging Defined. Prieiga per <https://www.netsuite.com/portal/resource/articles/accounting/accounts-receivable-aging.shtml>
25. HighRadius (2025) *Accounts Receivable Aging Report: Importance, How to Create & Use It?*. Prieiga per <https://www.highradius.com/resources/Blog/accounts-receivable-aging-report/>
26. Stripe (2024) What is an ageing report? What's in one and how to use it. Prieiga per <https://stripe.com/en-lt/resources/more/what-is-an-aging-report-what-is-in-one-and-how-to-use-it>
27. PwC (2017) *Robotic process automation: A primer for internal audit professionals*. Prieiga per <https://www.pwc.com/us/en/services/consulting/cybersecurity-risk-regulatory/library/robotic-process-automation-internal-audit.html>
28. Metricstream (2020) *Importance of Data Analytics and RPA in Internal Audit*. Prieiga per <https://assets.metricstream.com/pdf/insights/importance-of-data-analytics-and-rpa-in-internal-audit.pdf>
29. Asman, F. N., Fernandez, D., Jamaludin, N. A., Zaki, H. O., ir Ghazali, A. W. (2022). Cash conversion cycle and firm's performance. *International Journal of Academic Research in Business and Social Sciences*, 12(11). <https://doi.org/10.6007/ijarbss/v12-i11/15276>
30. Schmidt, J. (2024). *Accounts receivable aging*. Prieiga per <https://corporatefinanceinstitute.com/resources/accounting/accounts-receivable-aging>
31. Kenton, W. (2025). *Understanding Days Sales Outstanding (DSO): key calculation and applications*. Retrieved from <https://www.investopedia.com/terms/d/dso.asp>
32. Paul, S. Y., Devi, S. S., ir Teh, C. G. (2012). Impact of late payment on Firms' profitability: Empirical evidence from Malaysia. *Pacific-Basin Finance Journal*, 20(5), 777–792. <https://doi.org/10.1016/j.pacfin.2012.03.004>

33. Chen, H., Wang, S., Yang, D., ir Zhou, N. (2025). COSO-Based Internal Control and Comprehensive Enterprise Risk Management: Institutional Background and Research Evidence from China. *Encyclopedia*, 5(3), 106. <https://doi.org/10.3390/encyclopedia5030106>
34. EFRP Group (2024) *Auditing Accounts Receivable (3 Series) A Comprehensive Guide*. Prieiga per <https://efpradvisory.com/wp-content/uploads/2024/10/3-series-Accounts-Receivable-Audit-Guide-Final-2024.pdf>
35. Bahrami, M., Bozkaya, B., ir Balcisoy, S. (2020). Using behavioral analytics to predict customer invoice payment. *Big Data*, 8(1), 25–37. <https://doi.org/10.1089/big.2018.0116>
36. KPMG (2012) *Leveraging data analytics and continuous auditing processes for improved audit planning, effectiveness, and efficiency*. Prieiga per <https://assets.kpmg.com/content/dam/kpmg/pdf/2016/05/Leveraging-Data-Analytics.pdf>
37. Brown, K. E. (2024). *How data analytics is transforming internal audit*. Prieiga per <https://www.wipfli.com/insights/articles/ra-how-data-analytics-is-transforming-internal-audit>
38. Joshi, P. L., ir Marthandan, G. (2020). Continuous internal auditing: can big data analytics help. *International Journal of Accounting Auditing and Performance Evaluation*, 16(1), 25. <https://doi.org/10.1504/ijaape.2020.106766>
39. O. Didkovskyi, N. Jean, G. Le Pera ir C. Nordio (2024) *Cross-Domain Behavioral Credit Modeling: transferability from private to central data*. Prieiga per <https://arxiv.org/html/2401.09778v1>
40. S. Nanda (2018) *Proactive Collections Management: Using Artificial Intelligence to Predict Invoice Payment Dates*. Prieiga per <https://www.crfonline.org/wp-content/uploads/2018/04/HighRadius-Proactive-Collections-Management-Using-AI-1Q2018-Journal.pdf>
41. E. C. Malesiou (2021) *Analysis of monthly payment delays using machine learning*. Prieiga per <https://pergamos.lib.uoa.gr/uoa/dl/object/2965184/file.pdf>
42. M. Martikainen (2023) *Predicting late payment of sales invoices with statistical learning methods*. Prieiga per https://lutpub.lut.fi/bitstream/handle/10024/166112/Matti_Martikainen_Predicting_late_payment_of_sales_invoices.pdf?sequence=1
43. Kong, L., ir Brintrup, A. (2025). A causal artificial intelligence model for payment delay optimisation in supply chain financing. *Supply Chain Analytics*, 11, 100138. <https://doi.org/10.1016/j.sca.2025.100138>

44. Caseware (2023) Using Data Analytics to Improve Audit Quality. Prieiga per <https://www.caseware.com/ca/resources/blog/using-data-analytics-to-improve-audit-quality/>
45. Wewerka, J., ir Reichert, M. (2020). Robotic Process Automation -- A systematic literature review and assessment framework. *arXiv* (Cornell University). <https://doi.org/10.48550/arxiv.2012.11951>
46. C. Dilmegani (2025) *Top 9 RPA Use Cases & Examples in Finance*. Prieiga per <https://research.aimultiple.com/rpa-finance/>
47. Ramsey R. (2024). From Order to Cash: Automating Accounts Receivable with UiPath. Prieiga per <https://capitalizeconsulting.com/from-order-to-cash-automating-accounts-receivable-with-uipath/>
48. Fomin S. (2020) *9 Use Cases of RPA in Finance and Accounting*. Prieiga per <https://www.blueprintsys.com/blog/rpa/9-use-cases-of-rpa-in-finance-and-accounting>
49. Corcentric (2023) *Maximizing Efficiency in Order to Cash with Robotic Process Automation*. Prieiga per <https://www.corcentric.com/maximizing-efficiency-in-order-to-cash-with-robotic-process-automation>
50. BlackLine (2022) *BlackLine Unveils Industry's 1st Accounts Receivable Automation Capabilities to Deliver Real-Time Insights into Customer Risk and Growth Opportunities Based on Behavioral Analysis*. Prieiga per <https://investors.blackline.com/news-releases/news-release-details/blackline-unveils-industrys-1st-accounts-receivable-automation>
51. Almayyahi, A., Fotouh, S. A., ir Metawee, A. K. (2025). Advantages of implementing robotic process automation in internal auditing. In *Communications in computer and information science* (pp. 153–163). https://doi.org/10.1007/978-981-95-2011-4_15
52. Verma D. (2025) *Top 9 Credit Risk Management Software in 2025*. Prieiga per <https://www.highradius.com/resources/Blog/top-credit-risk-management-tools/>
53. Internal Audit Foundation (2019) *Moving Internal Audit Deeper Into the Digital Age: Part 1*. Prieiga per <https://www.theiia.org/globalassets/site/foundation/latest-research-and-products/moving-ia-deeper-download.pdf>
54. Deloitte (2025) Adopting automation capabilities for internal Audit. Prieiga per <https://www.deloitte.com/us/en/services/consulting/articles/internal-audit-robotic-process-automation-adoption.html>

55. Toor H. (2020) Robotic Process Automation for Internal Audit. Prieiga per <https://www.isaca.org/resources/isaca-journal/issues/2020/volume-6/robotic-process-automation-for-internal-audit>
56. Stoica, O. C., ir Ionescu-Feleagă, L. (2021). Digitalization in Accounting: A Structured Literature review. In *Sciendo eBooks* (pp. 453–464). <https://doi.org/10.2478/9788366675704-045>
57. Aldemîr, C., ir Uysal, T. U. (2024). AI competencies for internal auditors in the public sector. *Edpacs*, 69(1), 3–21. <https://doi.org/10.1080/07366981.2024.2312001>
58. Euleric M., Gruene J., ir Wood D. (2024) Achieving Effective Internal Control Over Robotic process automation. Prieiga per <https://www.scribd.com/document/809432381/COSO-RPA-Dec-2024>
59. Eulerich, M., Waddoups, N., Wagener, M., ir Wood, D. A. (2024). Development of a framework of key internal control and governance principles for Robotic Process Automation (RPA). *Journal of Information Systems*, 38(2), 29–49. <https://doi.org/10.2308/isys-2023-067>
60. Armijo-Perea, J. D., Sepúlveda, S., Velásquez, I., ir Villanueva, E. (2025). Robotic Process Automation (RPA) in the internal audit of large companies in Medellín, Colombia: Implementation and lessons learned. *Multidisciplinary Business Review*, 18(1), 1–15. <https://doi.org/10.35692/07183992.18.1.2>
61. Ukląńska, A. (2023). Robotic Process Automation (RPA) – bibliometric analysis and literature review. *Foundations of Management*, 15(1), 129–140. <https://doi.org/10.2478/fman-2023-0010>